

Addressing metric challenges: Bias and Selection - Efficient Computation - Hubness Explanation and Estimation

DISSERTATION

zur Erlangung des akademischen Grades

Doktor der Technischen Wissenschaften

eingereicht von

Mag. Abdel Aziz Taha

Matrikelnummer 9527395

an der Fakultät für Informatik
der Technischen Universität Wien

Betreuung: A.o.Prof. Andreas Rauber

Diese Dissertation haben begutachtet:

Andreas Rauber

Jenny Benois-Pineau

Wien, 21. September 2015

Abdel Aziz Taha

Addressing metric challenges: Bias and Selection - Efficient Computation - Hubness Explanation and Estimation

DISSERTATION

submitted in partial fulfillment of the requirements for the degree of

Doktor der Technischen Wissenschaften

by

Mag. Abdel Aziz Taha
Registration Number 9527395

to the Faculty of Informatics
at the Vienna University of Technology

Advisor: A.o.Prof. Andreas Rauber

The dissertation has been reviewed by:

Andreas Rauber

Jenny Benois-Pineau

Vienna, 21st September, 2015

Abdel Aziz Taha

Erklärung zur Verfassung der Arbeit

Mag. Abdel Aziz Taha
Schleiergasse 9/25
1100 Vienna

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 21. September 2015

Abdel Aziz Taha

Acknowledgements

First and foremost, I would like to express my appreciation and thanks to my adviser A.o.Prof. Andreas Rauber, Institute of Software Technology and Interactive Systems, TU Wien, thank you for giving me the fortunate opportunity to work with you and in your team; thank you for creating the conditions that made this dissertation possible. I also would like to express special gratitude to you for being available to encourage challenging discussions that inspired me to take the next steps. Equally important, I would like to express my appreciation and thanks to my adviser Dr. Allan Hanbury, Institute of Software Technology and Interactive Systems, TU Wien, thank you for your valuable guidance that was essential for finishing this thesis, but also for the moral inducement and motivation, that have been priceless for me and important to reach my goal. Among others, I owe you my scientific writing skill; I still remember my first attempt to write, when you had commented "Don't be too poetic!". Allan Hanbury, Andreas Rauber, you have been both a tremendous mentor for me. I thank you for encouraging my research with excellent support, and patient guidance. I am truly very fortunate to have the opportunity to work under your guidance as a student. It was both an honor and privilege to work with you.

I am also thankful to all my colleagues and committee members, who gave me the opportunity to conduct research with them and all those who have contributed directly or indirectly to this work. Special thank for my committee member Dr. Mihai Lupu, Institute of Software Technology and Interactive Systems, TU Wien, who was often available for interesting and useful discussions. I also thank Prof. Dr. Bjoern Menze, Computer Aided Medical Procedures, TU Munich, for providing the medical images to be used as test data.

Special thanks to my family. Words cannot express how grateful I am to my family members who have morally supported me. A special thank for my mother for her great motivation that helped me to strive towards my goal and for her prayer for me. Unfortunately, my father, who raised me with a love of science, is no longer with us to celebrate the completion of my dissertation. May God have mercy on his soul!

Lastly and most importantly, I would like to express my sincere appreciation to my beloved wife Lina, and children Josef, Yasmin, and Tamara, who were always my support and motivation in the moments when there was no one to answer my queries. Who have patiently endured many, many long hours alone without complaining while I was working on my dissertation. Who always attempt to hide their suffering and intent to show me

that it's alright although they often had to endure my absence. Thank you with all my heart and soul; I could not have completed this work without you by my side.

Kurzfassung

Featurespace (Merkmalsraum) ist ein wichtiges Konstrukt in Information Retrieval (IR) und Machine Learning (ML), in dem die zugrunde liegenden Objekte als Featurevektoren (Merkmalvektoren) dargestellt werden. Die bilden die Basis-Infrastruktur für Datenmodelle, welche den Kern von IR und ML darstellen. Diese Modelle beruhen auf den Beziehungen zwischen den Featurevektoren, die von Metriken gemessen werden. Metriken, wie Distanzen und Ähnlichkeiten, werden definiert, um Beziehungen zwischen einzelnen Vektoren (z.B. Distanz zwischen zwei Punkten) oder zwischen Gruppen von Vektoren (z.B. Ähnlichkeit zwischen zwei Punktwolken oder zwei Images) widerzuspiegeln. Es gibt jedoch drei Hauptprobleme in dieser Hinsicht: Das Erste ist, dass hunderte von Metriken existieren. Jede von diesen misst nur bestimmte Merkmale und Aspekte dieser Beziehungen, was bedeutet, dass verschiedene Metriken verschiedene Sichtweisen der Realität repräsentieren. Das kann auch so gesehen werden, dass verschiedene Metriken unterschiedliche Empfindlichkeiten bzw. Neigungen oder Verzerrungen zu bestimmten Eigenschaften, Aspekten oder Situationen haben. Diesen Bias zu verstehen erfordert ein formales Messverfahren dieser Empfindlichkeiten und Neigungen. Ein solches Verfahren, das eine formale Auswahlmethodik für Metriken möglich macht, um die für einen bestimmten Zweck am besten geeignete Metrik zu selektieren ist der erste Beitrag dieser Dissertation.

Das zweite Problem ist die Effizienz der Berechnung rechenintensiver Metriken, z.B. solche, die laut ihrer Definition die Abstände zwischen allen möglichen Paaren von Punkten berücksichtigen. Die Berechnung kann extrem ineffizient sein, insbesondere wenn Objekte verglichen werden, die aus einer enormen Anzahl von Punkten bestehen. Ein Beispiel ist die Berechnung der Hausdorff-Distanz zwischen Magnetresonanztomographiebildern (MRI). Solche Bilder können aus bis zu 100 Mio Punkten (z.B. ganz Körper MRI Images) bestehen. Das dritte Problem stellen Eigenheiten hochdimensionale Featurespaces dar, welche als Fluch der Dimensionalität (curse of dimensionality) bekannt sind, z.B. Sparsity (Spärlichkeit), Distanz Konzentration und Hubness. Diese Schwierigkeiten können auch als Sonderfälle der Metrik Sensitivität (asymptotische Neigung) betrachtet werden, welche entstehen, wenn die Dimensionalität ausreichend hoch ist. Einige der State-of-the-Art Methoden versuchen diese Schwierigkeiten durch die Verwendung von Merkmalsauswahl (feature selection) zu bewältigen, die die Dimensionalität des Merkmalraums durch die Beschränkung des Modells auf einer Teilmenge dieser Merkmale verringern, was mit Informationsverlust verbunden ist.

Bezüglich der der Sensitivität und Verzerrung von Metriken, präsentieren wir in einem ersten Schritt 20 Metriken für Evakuierung von 3D Medical Image Segmentierung, stellen für sie binär und fuzzy Definitionen bereit, und präsentieren eine umfassende Diskussion und Analyse ihrer Eigenschaften und Sensitivitäten. Basierend auf dieser Analyse stellen wir Richtlinien für die Auswahl von Metriken entsprechend der Eigenschaften der Segmentierungen und des Segmentierungsziels vor. In einem zweiten Schritt schlagen wir eine neue formale Methode vor, die automatisch auf die Sensitivität und Verzerrung der Metriken, basierend auf den Eigenschaften der zugrunde liegenden Objekte mit Berücksichtigung Benutzereinstellungen, rückschließt. Basierend auf dieser Methode präsentieren wir ein formales Verfahren zur Auswahl von Metriken für beliebige Evaluierungsprozess.

Für die Effizienz der Berechnung rechenintensiver Metriken stellen wir einen neuen Algorithmus zur Berechnung der exakten Hausdorff-Distanz zwischen beliebigen Punktwolken in einer Berechnungszeit vor, die linear mit der Größe der Punktwolken zunimmt. Dieser Algorithmus ist allgemein ohne Einschränkung über die zugrunde liegenden Objekten, die verglichen werden.

Im Hinblick auf hochdimensionale Datenräume präsentieren wir eine neue Erklärung für die Ursache von Hubness (curse of dimensionality), die auf Sparsity und Distanzkonzentration basiert. Auf der Grundlage dieser Erklärung leiten wir einen neuen Schätzer für Hubness ab, der auf Statistiken der Distanz zum Schwerpunkt beruht und in linearer Zeit berechnet werden kann. Wir stellen auch ein Verfahren zur Verringerung des Hubness anhand dieser Erklärung vor.

Abstract

In machine learning, data mining, and information retrieval, a feature space is an important construct, in which the underlying objects are represented as feature vectors, providing the base infrastructure required to build data models, which are the core of information retrieval and machine learning. These models are based on the relations between feature vectors, which are measured by metrics. Metrics, such as distances and similarities, are defined to reflect the relations between the individual feature vectors (e.g. distance between two vectors) or between groups of these vectors (e.g. similarity between classifications or images). However, there are three main problems in this regard:

The first is that metrics measure particular aspects of these relations, which means that different metrics represent different views of reality. This means that different metrics have different sensitivities and biases to particular properties, aspects, and contexts, which imposes the demand for bias and sensitivity measurement that enables defining a formal way for selecting the most suitable metrics depending on the nature of the underlying objects and the subjective user goals.

Regarding the metric bias and sensitivity, we provide in a first step a comprehensive discussion and analysis of 20 metrics for evaluating 3D medical image segmentation. Based on this analysis, we provide guidelines for metric selection based on the properties of the individual metrics, the properties of the segmentations, and the segmentation goal. In a second step, we propose a novel formal method that automatically measures the bias of metrics, based on the properties of the underlying objects and constraints determined by the user preferences. Based on this method, we provide a formal method for selecting evaluation metrics.

The second problem is efficiency of calculating computationally intensive metrics, like the Hausdorff distance, especially when comparing two point sets with huge size. One example of such a case is calculating the Hausdorff distance between medical volumes (3D images). Such volumes could have up to 100 Mio 3D pixels, e.g. whole body medical volumes. Metrics that are defined to calculate distances between all pairs of points become extremely inefficient when they are applied to such volumes.

Concerning the calculation efficiency of computationally intensive metrics, we propose a novel algorithm for calculating the exact Hausdorff distance in linear time. This algorithm is general and does not put any assumption on the underlying objects being compared.

The third problem is related to the curse of dimensionality, caused by the difficulties in relation to high dimensional feature spaces, including sparsity, distance concentration, and

hubness. These difficulties can also be seen as a special case of metric bias (asymptotic bias) arising when dimensionality is sufficiently increased. Some of the state-of-the-art methods deal with these difficulties by using feature selection, which aims to reduce the dimensionality of the feature space by restricting the model to a subset of the features.

Regarding high dimensional spaces, we propose a novel explanation of the cause of hubness (a common aspect of the curse of dimensionality) that is based on sparsity and distance concentration. Based on this explanation, we derive a novel estimator of hubness in linear time using statistics of the distance distribution. We also provide a method for hubness reduction, based on this explanation.

Contents

Kurzfassung	ix
Abstract	xi
Contents	xiii
List of Figures	xv
List of Tables	xxii
List of Algorithms	xxv
1 Introduction	1
1.1 Metrics and Metric Spaces	2
1.2 Feature Space	4
1.3 Difficulties in Using Metrics in Feature Space	4
1.4 Research Questions	7
1.5 Contribution	8
1.6 Formal Definitions and Notations	8
1.7 Structure of the Thesis	13
2 Metrics and Metric Bias	15
2.1 Introduction	15
2.2 State-of-the-Art	17
2.3 Evaluation of Medical Image Segmentation	20
2.4 Evaluation Metric Overview	21
2.5 Metric Analysis	23
2.6 Metric Properties and Metric Selection Guidelines	34
2.7 Metric Bias Inference	40
2.8 Summary	51
3 Computationally Intensive Metrics	53
3.1 Introduction	53
3.2 State-of-the-Art	56

3.3	Calculating the Hausdorff Distance	58
3.4	Analysis	69
3.5	Average Distance between Image Segmentations	78
3.6	Summary	84
4	Formal Analysis of Hubness	87
4.1	Introduction	87
4.2	State-of-the-Art	90
4.3	Distance Structure in High Dimensional Space	93
4.4	Cause of Hubness	103
4.5	Hubness Indicator	115
4.6	Analysis	118
4.7	Hubness Reduction	124
4.8	Testing with Real Data and other norms	129
4.9	Summary	135
5	Conclusion and Future Work	137
5.1	Conclusion	137
5.2	Future Work	139
A	Metric Definitions and Algorithms	143
A.1	2D and 3D Images	143
A.2	Basic Cardinalities of the Confusion Matrix	144
A.3	Spatial Overlap Based Metrics	146
A.4	Volume Based Metrics	150
A.5	Pair Counting Based Metrics	150
A.6	Information Theoretic Based Metrics	153
A.7	Probabilistic Metrics	154
A.8	Spatial Distance Based Metrics	156
A.9	Multiple Definition of Metrics in the Literature	158
A.10	Implementation	160
	Bibliography	161

List of Figures

2.1	The correlation between the rankings produced by 16 different metrics The pairwise Pearson’s correlation coefficients between the rankings of 4833 medical volume segmentations produced by 16 metrics. The color intensity of each cell represents the strength of the correlation, where blue denotes direct correlation and red denotes inverse correlation.	25
2.2	The effect of decreasing the true negatives (background) on the ranking Each of the segmentations in A and B is compared with the same ground truth. All metrics assess that the segmentation in A is more similar to the ground truth than in B . In \hat{A} , the segmentation and ground truth are the same as in A , but after reducing the true negatives by selecting a smaller bounding cube. The metrics RI , GCE , and TNR change their rankings as a result of reducing the true negatives. Note that some of the metrics are similarities (marked with S) and others are distances (marked with D).	26
2.3	The effect of overlap on the correlation between rankings produced by different metrics. The positions and heights of the bars show how metrics correlate with $DICE$ and how this correlation depends on the overlap between the compared segmentations. Four different overlap ranges are considered.	27
2.4	Metrics that fail to discover boundary errors. In (A), the star is compared with a circle and in (B) the same star is compared with another star of the same dimensions, rotated so that the resulting overlap errors (FP and FN) are equal in magnitude in both cases. All metrics that are based on FP and FN (overlap-based metrics) are not able to discover that the two shapes in (B) are more similar to each other than those in (A). On the contrary, all spatial distance based metrics discover the similarity and give (B) a higher score than (A). However, the metric most invariant to boundary error is the volumetric similarity, since it gives a perfect match in both cases.	29
2.5	Boundary errors: rewarding/penalizing recall. Illustration in 2D of boundary errors that decrease/increase recall. The ground truth image GT is compared with the image A that is smaller than GT and with another image B that is larger than GT. Although the boundary error in both cases is equal (δ), the magnitude of the resulting false negative (FN) with A is smaller than the resulting false positive (FP) with B. This causes that metrics, considering the absolute magnitudes of FN and FP, penalize high recall.	30
2.6	The effect of segment density. Two segmentations (B) and (C) are compared with the corresponding ground truth (A). (B) has a solid structure while (C) has a lower density due to large number of tiny holes uniformly distributed inside it. Although (C) has a higher accuracy of the boundary than (B), all metrics, excepts MHD and HD , give (B) a higher score than (C).	31

2.7	Illustration of sensitivity of overlap based metrics to segment size. In (A), two large segments having a displacement Δ . This results in a relatively large overlap. In (B), two thin segments with the same displacement, but without any overlap. Overlap based metrics give zero. Also in (C), small segments with the same displacement have no overlap. Overlap based metrics cannot differentiate the quality between (C) and (D) with a smaller displacement. . .	33
3.1	$x_{(i-1)}$ is the point already minimized in the previous iteration where <i>cm</i> was found to be the current maximum (temporary HD). Point x_i is being currently minimized by calculating its distances to B . Points $y_1..y_8 \in B$ are numbered according to their distance to x_i . An iteration order beginning with y_1, y_2 or y_3 is good because it will cause an immediate break whereas an iteration order beginning with other points is worse because the scan will continue.	61
3.2	The probability density function of a geometrical distribution.	64
3.3	Distribution of pairwise distances assuming a normal distribution for illustration. (A) Position of the Hausdorff distance h relative to the distribution affects p because $cm \leq h$. (B) h is large and <i>cm</i> can reach large values thereby increasing p . (C) h is small and <i>cm</i> remains small thereby decreasing p	65
3.4	The progress of <i>cm</i> in the first 10 thousand iterations (outer loop) when comparing two real brain tumor segmentations. Note that only 10 thousand of 15.6 million iterations in total (this is the number of voxels in the first segmentation) are shown and thus the curve does not reach the HD.	67
3.5	The average number of iterations in the inner loop until the early break at each iteration of the outer loop. Values are recorded from measuring the HD between 1000 pairs of trajectories generated from the road network of Oldenburg. Each trajectory contains 2000 points. Iterations of the outer loop are on the x-axis and the number of iterations in the inner loop averaged over all pairs on the y-axis.	68
3.6	Convergence behavior of the temporary HD (<i>cm</i>) along the iterations of the outer loop. Iterations of the outer loop are on the x-axis and the frequencies of <i>cm</i> values exceeding 90% and 99% of the corresponding HD at each iteration are on the y-axis.	69
3.7	Comparison between the performance of the proposed algorithm and the ITK algorithm in validating 240 real brain tumor segmentations against the corresponding ground truth. The set sizes of the segmentation pair being compared in kilo voxels are on the horizontal axis and the run time in seconds is on the vertical axis. The grid size varies from 125x125x125 to 250x250x250 voxels. The entries are sorted according to the sum of the two sizes ascending.	71

3.8 Comparison between the performance of the proposed algorithm and the ITK algorithm in comparing 300 pairs of volumes selected randomly so that the overlap between volumes in each pair is zero. The set sizes of the segmentation pair being compared in kilo voxels are on the horizontal axis and the run time in seconds is on the vertical axis. All volumes have a unified grid size of 250x250x250 voxels. The entries are sorted according to the sum of the two sizes ascending. 72

3.9 Comparison between the performance of the proposed *HD* algorithm and ITK Library implementation in validating 840 whole body segmentations against the corresponding ground truth. The data points are sorted according to the grid size ($w \times l \times h$). The set sizes of the segmentation pair being compared in kilo voxels as well as the grid size are on the horizontal axis and the run time in seconds is on the vertical axis. The ITK implementation failed with memory allocation error with all volumes over a particular grid size. The entries are sorted according to the sum of the two sizes ascending. 73

3.10 Performance comparison between the proposed algorithm and the ITK algorithm in comparing enlarged volumes. The set sizes of the segmentation pair being compared in kilo voxels are on the horizontal axis and the run time in seconds is on the vertical axis. All volumes have a unified grid size of 250x250x250 voxels. The entries are sorted according to the sum of the two sizes ascending. 75

3.11 Contribution of random sampling: Comparison between the efficiency of the proposed algorithm when using random sampling and direct scanning. The same data as in the experiment in Section 3.4.1 is used. The size of the compared images in kilo voxel is on the x-axis and the execution time in seconds, scaled logarithmically, is on the y-axis. 76

3.12 Comparison between the performance of the proposed algorithm and the ITK algorithm in measuring the HD of 300 pairs of Gaussians generated by randomly selected means and standard deviations for each of the 3 dimensions. The sizes of the point sets being compared in kilo voxels are on the horizontal axis and the execution time in seconds is on the vertical axis. 77

3.13 Comparison of the execution time of calculating the Hausdorff distance $HD(X,Y)$ by the proposed algorithm and the incremental Hausdorff calculation (INC) [NJS11]. In A, the size of X is fixed and the size of Y varies and conversely in B. Each data point is the average of 200 pairs of trajectories. The size of the point set is on x-axis and the execution time in milliseconds on the y-axis. 78

3.14	Illustration of the optimizations used in calculating the average distance (<i>AVD</i>). In (1) and (2), the images A and B, defined on the same grid, are to be compared using the <i>AVD</i> . In (3), the intersection of the images is identified. In (4), the points in the intersection are removed from the domain $D(NN)$, since they have zero distances. In (5), only distances from $R1$ to B are considered. In (6), only the boundary voxels (surface) of B are considered as range $R(NN)$	80
3.15	Finding the surface of a segmentation: In (A), a 3D segmentation of the edema of a real brain tumor viewed as three orthogonal slices. In (B), the surface (boundary) of the same segmentation as a result of applying the Algorithm 3.4 on the segmentation in (A).	81
3.16	Illustration of the optimizations achieved by reducing the search space when searching the nearest neighbor, a search sphere with radius r is found by moving from the query q toward the mean m and considering the first point crossed on the boundary.	82
3.17	Comparison between the performance of the proposed <i>AVD</i> algorithm and the <i>AVD</i> algorithm of the ITK Library in validating 240 brain tumor segmentations against the corresponding ground truth. The grid size is on the horizontal axis and the run time in seconds is on the vertical axis. The data points are sorted according to the grid size ($w \times l \times h$).	83
3.18	Comparison between the performance of the proposed <i>AVD</i> algorithm and ITK Library implementation in validating 840 whole body segmentations against the corresponding ground truth. The data points are sorted according to the grid size ($w \times l \times h$). The set sizes of the segmentation pair being compared in kilo voxels as well as the grid size are on the horizontal axis and the run time in seconds is on the vertical axis. The ITK implementation failed with memory allocation error with all volumes over a particular grid size. The entries are sorted according to the sum of the two sizes ascending.	85
4.1	Distribution of the pairwise distance between vertices of unit hypercubes of different dimensionalities; (A) for a 10-hypercube, fully enumerated; (B), (C), and (D) for 50-hypercube, 100-hypercube, and 500-hypercube respectively, sampled using the Monte-Carlo method.	99
4.2	Convergence of the DTM and the PWD with increasing dimensionality. Four random point sets were drawn from a normal distribution with dimensionalities 2, 10, 100, and 10000 in (A), (B), (C), and (D) respectively. Each point set consists of 500 points. For each point set all distances to mean are calculated, and 500 pairwise distances were sampled randomly. The DTM converges to \sqrt{d} and the PWD converges to $\sqrt{2d}$, which is in conformance with Equations 4.10 and 4.17 respectively.	101
4.3	Convergence of the DTM and the PWD with L_∞ norm. The same i.i.d. random point sets as in Figure 4.2 have been used.	102

4.4 Convergence of the DTM and the PWD with the cosine norm (COS). The same i.i.d. random point sets as in Figure 4.2 have been used. Note that the mean in terms of the cosine distance is not the origin, but rather the vector (point) that minimizes the angles to all other points. 103

4.5 In (A) there are symmetrical NN relations. In (B), the point P1 is deviated toward the mean of the square, which causes changes in the NN relations of the neighboring points, so that P1 becomes the nearest neighbor of P2 and P4, which makes P1 a hub. In (C) there are symmetrical NN relations between the cube vertices, since vertices have equal distances between them. In (D), P1 is deviated toward the mean. This causes P1 to become the NN neighbor of P2, P4, and P5. The number of points for which the NN relations change increases with dimensionality. 104

4.6 Deviation types: (A) The three types of deviation from the hypercube vertex that a point can have. δ_{r-} deviation is when a point leaves a vertex towards the mean (hypercube center), which we call the radial inside direction. δ_{r+} deviation is when a point deviates away from the mean, which we call the radial outside direction. δ_p deviation is any direction that is perpendicular to the radial direction, which we call the perpendicular hyperplane. (B) An arbitrary deviation δ of a point x can be resolved into its components as one or more of the three deviation types. (C) and (D) Two different distributions with a large $\tilde{\gamma}$ value (C), small $\tilde{\gamma}$ value (D). The shaded area illustrates the space where the point exists with high probability relative to the corresponding vertex. 106

4.7 In (A), points are exactly at the vertices of a hypercube. In (B), P4 is deviated along an edge of the hypercube affecting only one point regarding the NN relation, namely P3. In (C), the point P4 is deviated along a face diagonal, thereby affecting two points, namely P3 and P8. 109

4.8 Distribution of δ_r and the similarity to the hubness-optimal distribution. Although both of the distributions have the same variance, (B) has a higher kurtosis, which makes it more similar to the hubness-optimal distribution, since the mass is concentrated at the mean and few points deviate from the mean (tails). 110

4.9 The convergence of the sampled PWD variance of a random point set consisting of 500 i.i.d. normally distributed points. Each data point represents the variance resulting from sampling a number of point pairs (the x-axis). The figure shows that the variance quickly converges to the exact value. 113

4.10 Illustration in a low dimensionality of how a subset of hypercube vertices can build another hypercube of lower dimensionality. The vertices P2, P6, P8, and P4 of the cube (3-hypercube) are occupied by points. These four points build a fully occupied square (2-hypercube, i.e. a hypercube of lower dimensionality) 114

4.11	Q-Q plots of the exact hubness calculated using the basic hubness measure Equation 4.1 versus the estimated hubness $H(x)$ according to the proposed hubness indicator Equation 4.41 using a scaling constant $\alpha = \frac{1}{3}$. In both cases, 950 point sets, each of them consisting of 1000 points with dimensionalities uniformly distributed between 50 and 1000. In (A) for i.i.d points drawn from a uniform distribution over a hypercube, and in (B) for i.i.d. normal distribution.	117
4.12	Reducing the hubness by removing the nearest point to the mean. 5 point sets in the 500-dimensional space, each consisting of 500 points with i.i.d components from uniform distribution (A) and normal distribution (B). The number of points removed is on the x-axis and the hubness is on the y-axis, calculated using the basic algorithm, Equation 4.1. The values of ϑ before point removal are encoded in the legend.	119
4.13	The density gradient in a distribution where the density decreases when moving farther from the mean. The dashed circles illustrate the density levels. $var(DTM)$ increases with increasing rate of density decay, because the difference between $min(\Delta r)$ and $max(\Delta r)$ increases, where $\Delta r = r_j - r_i$	120
4.14	The relation between hubness and the distributions of DTM and PWD. Five examples of point sets, each of them consists of 300 points with dimensionality 500, drawn from different i.i.d. distributions. (A) and (B) are tow different random point sets drawn from a uniform hypercube, (C) and (D) from a normal distribution, and (E) from a uniform distribution over a hyperball. (F) is a details view of the DTM distribution in (E). The factor γ (ratio between the DTM variance and the PWD variance) as well as the outliers left to the DTM distribution decide the strength of hubness. In (A) and (B), γ is mid-range and the hubness has also mid-range values, however in (B) hubness is higher because of the outliers at the left side. In (C) and (D), γ is high and hubness is relatively high, but however it is higher in (D) because of the outliers at the left side. In (E), γ is very low and hubness is also very low although there is a high rate of outliers at the left side of the DTM distribution, which is clear in the details plot (F). All hubness values are measured with the basic NN definition, Equation 4.1	122
4.15	Comparison of the decay rates of the DTM variance and PWD variance for 500 point sets with dimensionalities varying from 1 to 500 for three distributions, namely a normal distribution in (A), a uniform distribution over the hypercube in (B), and a uniform distribution in the hyperball in (C). In each of the three cases, the estimative variances of the DTM and PWD are visualized by showing the minimum and the maximum. In (D), the distribution of the DTM of one case from (C), namely a point set of the dimensionality 500 is shown. Note that the domain of (D) corresponds to a very small distance interval of about 0.001 unit, compared with the corresponding interval of the PWD, which is about 0.2, estimated as $max(PWD) - min(PWD)$ for the corresponding point set in (C)	123

4.16	The influence of removing hubs on the NN relations. Green arrows represent correct NN relations, and red arrows represent incorrect NN relations caused by hubness bias. The direction of the arrow means k -nearest neighbor of. In (A) the hub point h is measured by a metric to be the NN of six points, only one of them r is correct and the five others are incorrect. (B) illustrates the NN relations after removing the hub point h . A significant decrease in the incorrect relations, and increase in the correct relations are observed.	126
4.17	Hubness reduction by hub removal: 100 i.i.d point sets have been randomly generated, each containing 1000 points and having a dimensionality between 100 and 1000. For each point set, two hubness values were calculated using the basic hubness measure (Equation 4.1), one value of the complete point set and the other value after removing 1% of the points (10 points) that have the most hubness. In each case, a k -NN algorithm was considered with a k value randomly selected between 1 and 10. The point sets are sorted according to the original hubness (before removal). Sorting according to dimensionality does not make sense because the effect of dimensionality on hubness is considerably smaller than the effect of outliers.	127
4.18	Transformation for hubness reduction. In (A) the points are normally distributed which results in a density gradient. In (B), points far from the centroid are moved toward the centroid, so that the magnitude of movement corresponds to the distance from centroid, i.e. farther points far from the centroid are more moved than near ones. In (B), the points after transformation have distribution that is nearer to the uniform hyperball, and thus have less hubness.	128
4.19	Hubness reduction by hub transformation: 30 i.i.d point sets have been randomly generated, such that each has 1000 points and a dimensionality between 30 and 500. For each point set, two hubness values were calculated using the basic hubness measure (Equation 4.1), one value is the original hubness and the other value is the hubness after the transformation according to Equation 4.42. In each case, a k -NN algorithm was considered with a k value randomly selected between 1 and 10. The point sets are sorted according to the original hubness (before transformation).	130
4.20	Hubness reduction by hub removal on samples of the TREC-AP text collection, each consisting of 500 documents. A k -NN algorithm with $k = 1$ and the L_2 norm have been used. In (A), for each document x , the value of the hubness contribution $\vartheta(x)$ and the exact k -occurrence $n_k(x)$ are calculated. The documents are sorted according to ϑ . The top part of the list is shown. In (B), the documents are successively removed in the order of the list (highest ϑ first). The number of documents removed is on the horizontal axis and the resulting exact hubness of the sample is on the vertical axis.	132
4.21	The results of the previous experiment (Figure 4.20) repeated using k -NN algorithm with $k = 3$ instead of $k = 1$	133

4.22	The results of the same experiment repeated using the <i>COS</i> norm and a k -NN algorithm with $k = 1$. Results show that removing the documents with the highest ϑ results in a hubness reduction as expected. It is also observed that the hubness convergence is faster than with the L_2 norm.	134
4.23	The results of the experiment hubness reduction by removal repeated using the L_∞ norm and a k -NN algorithm with $k = 1$. There is no hubness reduction observed in (B) after removing the top documents of the list sorted according to $\vartheta(x)$ in (A).	135
4.24	Comparison between L_2 norm (A) and L_∞ norm (B) of how the $n_k(x)$ values are distributed in relation to the $\vartheta(x)$ values for all documents in Sample 1. The values are sorted according to $\vartheta(x)$. In order to make the plot readable, $\vartheta(x)$ has been scaled (x20) on the vertical axis. While $\vartheta(x)$ is distinguishable for all documents in (A), it does not seem so in (B) because $\vartheta(x)$ distribution has a step-like form. The distribution of the $n_k(x)$ values is however related to $\vartheta(x)$ because documents on the left side have in general higher n_k values than those on the right side.	136
5.1	Illustration of the suggested analogy between image and text. The analogies are represented by dashed lines. Grid cells correspond to terms. The image to the document. The distances between pixels correspond to the distance between terms (semantic). The color a grid cell has from the context of an image corresponds to the meaning added to a term from the context of a document.	141
A.1	Illustration of the <i>AUC</i> when only one measurement is available according to [Pow11]. In this case, the <i>AUC</i> is area of the trapezoid defined by the measurement point and the lines $TPR = 0$ and $FPR = 1$	156

List of Tables

1.1	Confusion matrix comparing two segmentations, S_g as the ground truth segmentation and S_t as the test segmentation	11
-----	---	----

2.1	Overview of evaluation metrics for 3D image segmentation. The symbols in the second column are used to denote the metrics. The column “reference of use” shows papers where the corresponding metric has been used in the evaluation of medical volume segmentation. The column “category” assigns each metric to one of the categories above. The column “definition” shows the equation numbers where the metric is defined.	22
2.2	Assignment between the properties defined in Section 2.6.1 and the metrics defined in Table 2.1. A particular metric has a particular property iff the corresponding cell is check marked.	37
2.3	Summary of metric selection guidelines. Each row corresponds to either a segmentation property or a requirement and each column corresponds to one of the metrics in Table 2.1. A checked cell (✓) denotes that the metric is recommended for the corresponding property/requirement, a crossed cell (X) denotes that the metric is not recommended, and empty cells denote neutrality.	40
2.4	Manual and automatic metric suitability rankings. In column “expert”, the average correlation between metric rankings and the manual rankings as well as corresponding suitability ranks according to descending correlation. In column “automatic”, the metric bias calculated automatically by the proposed method as well as the ranks according to ascending bias (detailed data and results available in [THJ14a])	52
3.1	Result summary for experiments on medical images of varying sizes and characteristics where $n_1..n_2$ is the size range of the compared point sets, L, B, H are the grid dimensions for medical volumes and the time values are the average execution time for calculating the HD	74
4.1	Notation used throughout this chapter	89
A.1	Confusion matrix comparing two segmentations, S_g as the ground truth segmentation and S_t as the test segmentation	145
A.2	Five examples show that the pair counting cardinalities (a , b , c , and d) cannot be used in place of the overlap cardinalities (TP , FP , FN , and TN) to calculate the Jaccard index.	159

List of Algorithms

3.1	NAIVEHDD straightforwardly computes the directed Hausdorff distance .	59
3.2	EARLYBREAK computes the directed HDD using the early break technique and random sampling	60
3.3	RANDOMIZE finds a random order of a given point set	62
3.4	HOLLOW returns the surface points of a 3D image	81

Introduction

The terms denoting similarity such as close, similar, near, etc. and those denoting distance such as different, far, dissimilar, discrepant, etc. are basic concepts. In some contexts, they are related to perceptual meanings, i.e. can be observed visually, like the similarity between the faces of two persons. In other contexts, there are no visual relations, such as the similarity between two problem solutions or two classifications. Here, the similarity is rather a measure of the qualities two things have in common. However, similarity is sometimes subjective. To illustrate this, consider the following question: which figure is more similar to a circle, a square with the same area or a circle with different radius. The answer could be the square if we are focusing on area and the circle if we are focusing on the form. This is a simple example. However in practice, there are numerous aspects that can be considered to decide on the similarity. Metrics are functions that attempt to measure similarity. Since it is impractical and also not desirable to consider all aspects at a time, there are numerous metrics. Each of them considers only a subset of these aspects depending on its definition, which is the reason behind metrics having different sensitivities, different biases, and different suitabilities to measure similarity in different applications. The remainder of this section is organized as follows: In Section 1.1, we discuss in general metrics and metric space. In Section 1.2, we discuss some aspects relating to feature space as a core framework for machine learning (ML) and information retrieval (IR). In particular, we link feature space to metric space and then we link feature space representing image data to feature space representing text data by introducing an analogy between them. In Section 1.3, we present some of the difficulties that arise when using metrics in feature spaces of particular properties. In particular, we present the problem of metric bias and the need for bias measurement; we present the efficiency problem of calculating complex metrics in large data collections; and we present some problems arising when the dimensionality of the underlying data is very high. Finally in Sections 1.4 to 1.7, we present the research questions and the contribution of this thesis, we provide general notation that is used throughout the thesis, and we describe the structure of the thesis.

1.1 Metrics and Metric Spaces

In this section, we discuss metrics and metric spaces. At first we define the term object to denote things that metrics are applied on. Then we define the term metric and clear the relation between distance metrics and similarity measures. Finally we define the concept of metric space.

Object: Since metrics measure the distance between things, i.e. compare two things, we define the notation object to denote everything that can be compared.

Definition 1. *Object:* We denote with object every thing that can be compared using metrics, i.e. a metric compares two objects.

Objects can be simple, e.g. single points or more complex, e.g. point sets. The definition of objects can differ depending on the nature of the underlying data and the complexity of these objects can vary from one domain to another. In some applications, objects are not simple points, but rather more complex, e.g. a set of points. Examples are objects represented by point clouds. We will denote such objects as point cloud objects. A special case of point cloud objects are objects defined on an imaginary grid such that each point is represented by a grid cell. 2D and 3D images are examples of this type. We will denote such objects as grid-based objects.

Metric: A metric is a function defined on a set of objects, such that for any pair of objects in the set, it provides a positive value indicating how far the two objects are from each other. The following provides a formal definition of a metric.

Definition 2. *Metric:* Let O be a set of objects with $A, B, C \in O$. Let the function ϕ be defined such that $\phi : O \times O \rightarrow \mathbb{R}$. The function ϕ is a metric iff it satisfies the following properties: (i) non-negativity, i.e. $\phi(A, B) \geq 0$, (ii) the coincidence axiom, i.e. $\phi(A, B) = 0$ if and only if $A = B$, (iii) symmetry, i.e. $\phi(A, B) = \phi(B, A)$, and (iv) the triangle inequality, i.e. $\phi(A, C) \leq \phi(A, B) + \phi(B, C)$.

Some distance measures do not satisfy all the metric properties above. Examples of such measures are the pseudo metric [Kel75] if they satisfy all the properties except the coincidence axiom, and the quasi metrics [Wil31] if they satisfy all the properties except the symmetry. More information about these and other extended metrics like the semi metrics and the quasi-pseudo metrics can be found in [Kel75].

There is a large variety of metrics. They differ in their nature, their sensitivities, their bias, and the aspects they measure. We categorize metrics according to general aspects to ease understanding them. There are different aspects according to which metrics can be categorized, e.g. their applications. However, we divide metrics into two types according to the type of objects they can compare, namely:

- Distance between single points: These metrics measure how close two points are. Examples of metrics in this category are the p-norms and the cosine similarity

(note that the cosine similarity in its standard form is not a metric as it does not satisfy the triangle inequality property and it also violates the coincidence axiom).

- Distance between point clouds: These metrics measure how close (similar) two point clouds are. These metrics are more complex; they attempt to summarize the similarity between two sets of points, i.e. to provide one value that summarizes the similarities among all points in the two sets.

Distance metrics vs. similarity measures: In contrast to distance metrics, similarity measures give information on how close/similar two objects are. There are different ways to convert a similarity to a distance and vice versa. The most straightforward method is the inversion, i.e. $s = 1/\phi$, where ϕ is a distance metric and s is the corresponding similarity. However, for some applications, e.g. those where the similarity has directly to do with human perception, there are other methods that could be more optimal than the inversion method and shown to be more suitable for perceived similarity, e.g. the exponential conversion proposed by Shepard [She87] given by

$$s = e^{-\phi} \tag{1.1}$$

Formally, similarity measures are not metrics because they do not satisfy the coincidence axiom, since two identical objects do not have a zero similarity. They also do not satisfy the triangle inequality, as given in Definition 2. However, given the similarity is normalized to have its range in $[0, 1]$, then the conversion to distance using the inverse of Shepard conversion (Equation 1.1) given by

$$\phi = -\log(s) \tag{1.2}$$

results in a distance that satisfies non-negativity, the coincidence axiom and the triangle inequality in Definition 2. Assuming the underlying similarity measure is also symmetric, then its conversion is a metric. Li et al. [LCL⁺04] provide more information about similarity normalization and conversion to distance.

Since there is at least one conversion that converts any symmetric similarity to a distance satisfying the metric properties, we will not differentiate between distance metrics and similarity measures in this thesis, i.e. we will denote both of them as metrics, unless otherwise explicitly stated.

Metric space: A set of objects X together with a metric ϕ build a metric space. One example is the 3D Euclidean metric space. In this case X is the set of all 3D points and ϕ is the Euclidean distance between each point pair (the length of the straight line between the two points).

Definition 3. *Metric space: The ordered pair (O, ϕ) consisting of a set of objects O and a metric ϕ defined on $O \times O$ is called a metric space.*

1.2 Feature Space

Feature space is one of the most important concepts of machine learning that provides a framework for representing objects, e.g. documents or images.

In this section, we discuss feature space to highlight some important related aspects. In the following, we define some of the core elements and concepts of feature space [WF05].

Features: A feature is a measurable property, quality, aspect, or characteristic of an object. Features can be symbolic (e.g. color) or numeric (like height). However, they can always be represented as numeric values by applying a data preparing step like encoding and normalization. Since data preparation is not in the scope of this thesis, we always assume numerical features, i.e. each feature is always represented by a single numeric value.

Feature space: In machine learning, data mining, and information retrieval applications, objects are represented using their features. Since the features can be numerous and/or some of them can be less relevant than others, a data preparation step called feature selection can be used to select a subset of the features to be considered. Once these features are known, each of the underlying objects is represented by a combination of these features. The union of all these object representations forms a d -dimensional vector (d is the number of features selected) called feature vector, which can be modeled as a hyperspace called feature space.

1.3 Difficulties in Using Metrics in Feature Space

In this section we describe some of the difficulties arising when metrics are used with feature spaces, to which we present solutions in this work. The solved difficulties are in three directions, namely selecting the most suitable metric for a feature space, efficiently calculating computationally intensive metrics when the size of the feature space (number of points and/or dimensionality) is huge, and addressing drawbacks related to hubness when the dimensionality of the feature space is high.

1.3.1 Metric Bias and Metric Selection

There are already dozens of metrics used in information retrieval, and more keep appearing [ACMS12]. Most researchers choose metrics (e.g. evaluation measures) arbitrarily or according to their popularity [ACMS12]. A poorly defined metric may lead to inaccurate results, e.g. selecting suboptimal models when comparing the performance of classifiers [FWM⁺09] [SC12] [CK97] [PRVtHR08]. Although our research on metric and metric bias is general, i.e. applicable to various domains, we concentrate in our analysis on effectiveness metrics, i.e. metrics used to compare objects, e.g. classifications, with reference objects, e.g. ground truth.

Radlinski et al. [RC10] show that the relative system improvement achieved is decreasing, which results in sensitivity and fidelity of evaluation metrics becoming increasingly critical. When improvements are small, metrics with high sensitivity are needed to measure small but real improvements and also with high fidelity to distinguish between improvements based on user preferences and improvements resulting from biased relevance judgments.

Blanco et al. [BZ11] show experiments demonstrating how random perturbation could lead to significant improvements measured by the standard IR evaluation methods; they warned researchers about misinterpretation and stressed the need for standard and reliable evaluation methodology.

The following are some examples of metric sensitivities from the literature when metrics are used for comparing images: Hausdorff distance is very sensitive to noise and least squares based evaluation methods are very sensitive to outliers [GJC01]. Mutual information doesn't utilize spatial information inherited in images because only voxel relationships are considered but not the neighborhoods [RTR⁺04]. The baseline level of a metric is a property that gives how capable is a metric to discover the amount of agreement caused by chance. The baseline level should ideally be zero, since a random classifier should ideally have a zero score. Information theoretical measures have a non-convergent baseline which depends on the ratio between the number of data points and the number of classes. Therefore this class of measure needs chance correction [VEB09]. Commonly used measures (precision, recall and F-measures) are biased and don't consider the level of chance [Pow11]. Choosing evaluation metrics is very important and application-dependent; when evaluating imbalanced datasets, the metric choice is not obvious [FWM⁺09]. Metrics have different properties with respect to their correlation with user satisfaction criteria and their ease of interpretation [BV00]. Benhabiles et al. [BLVD10] validated 250 automatic segmentations against their corresponding ground truth segmentations using four different evaluation metrics. The results were then compared with manual ratings from 40 human observers. They found that the correlations between the ranking based on the manual ratings and the rankings based on the evaluation metrics vary between 30% and 80% depending on the metric used.

There is still no real scientific approach to efficiently select the most suitable evaluation metric for a specific task and/or a specific data set. Investigating metrics would help researchers to better understand them and help companies and stakeholders to save effort and time reaching optimal systems [PRVtHR08].

1.3.2 Computationally Intensive Metrics

In Section 1.1, we have divided metrics into two categories depending on the nature of the objects they can compare, namely metrics that measure the distances between two points and metrics that measure the distance between point sets.

Metrics of the latter category could be computationally intensive. The runtime of calculation depends on two factors, namely (i) the point set size, and (ii) the definition of the metric, i.e. the way it performs the comparison. While some metrics are not com-

putationally expensive. e.g. because they are based on basic cardinalities (Section 1.6.4), i.e. the overlap between the point sets, some other metrics are computationally intensive, since they attempt to compare all point pairs in the two point sets being compared.

One example of computationally intensive metrics is the Hausdorff distance (HD), which is defined as the maximum of the minimums of the distances from the first point set to the second one. To straightforwardly compute the HD between the point sets $X1$ and $X2$, for each point $x_i \in X1$, the minimum distance to $X2$, i.e. $\min(d(x_i, X2))$, should be calculated. The HD is then the maximum of these minimums. The straightforward computation has thus a runtime complexity of $O(n^2)$, where n is the point set size. This runtime complexity becomes a problem when n is significantly large. Consider for example comparing two 3D medical images, e.g. magnetic resonance images (MRI). A whole body MRI can contain many millions of voxels (grid cells). In such cases, the direct computation of the HD is not practical.

Several approaches have been proposed that aim to overcome the computational complexity of the Hausdorff distance, either by finding an efficient approximation or by efficiently computing the exact HD for special types of objects like polygons, line segments, or special curves.

In this work, we propose a novel general algorithm for computing the exact Hausdorff distance between arbitrary point sets in linear time in terms of the point set size. Furthermore, we provide an efficient method for computing the average distance between two image segmentations that makes use of the nature of image segmentations being dense point sets.

1.3.3 High Dimensional Space

When the dimensionality of the feature space is significantly high, e.g. when the number of features is large, another category of difficulties arises when applying metrics. The curse of dimensionality is a common term denoting phenomena that are related to high dimensional feature space. Distance concentration [RNI10] is one of these phenomena. It denotes the phenomenon in high dimensional space, in which all pairwise distances tend to be equal. Distance concentration has been studied intensively, e.g. in [BGRS99] [HAK00] [AHK01] [Fra08] [FWVM07] [Koe00] [RS05] [ST83].

Another term, called hubness, is also related to the curse of dimensionality. Hubness denotes a phenomenon in relation to the neighbor neighbor (NN) algorithms when applied to high dimensional feature space (the formal definition of hubness is in Section 4.1). Hubness is a characteristic of the structure of the NN relations in a metric space. According to this characteristic, there are objects that are frequently found as NN of other objects; these objects are called hubs. On the other hand, there are objects that are rarely or never found as NN of other objects; such objects are called anti-hubs or orphans. Some researchers link hubness to distance concentration, while others consider hubness a matter of density gradient or boundary in finite datasets.

Hubness has a negative effect on the performance of information retrieval systems, for example music retrieval [FSS12], because objects represented by hubs are far more

frequently retrieved than other objects although they may have low similarity to the query object.

In contrast to distance concentration, hubness has not been deeply studied [RNI10]. In this work, we provide a novel explanation for the cause of hubness based on data sparsity and distance concentration in high dimensional spaces. In contrast to other approaches, our explanation does not make assumptions about the data distribution or distance metric. Furthermore, it generalizes other models, and explains observations and results documented in the literature. Based on this explanation, we propose a novel hubness indicator that predicts the hubness, given a data set. This indicator is calculated in linear time in terms of the dataset size, which is efficient compared with the basic hubness measure that calculates the kNN-lists of all points.

1.4 Research Questions

The general focus of this work are the difficulties that arise when measuring distance in feature space under particular conditions, namely when distance metrics are biased to some properties of the feature space, when the feature space is very large such that the distance computation is no longer efficient, or when the dimensionality of the feature space is high such that distance measures are subjects to the curse of dimensionality. In particular, the research questions are in three research areas:

(I) Metric bias and metric selection: This research area deals with the bias of evaluation metrics, when comparing objects represented by point sets in the feature space, i.e. the tendency of metrics to penalize or reward particular properties of the objects being compared and the degree of suitability of a particular metric to measure a particular quality. In this scope, the research aims to answer the questions, (1) how can metric bias be measured, and (2) how to formally select the most suitable metric(s) for evaluating a set of classifications, taking into consideration the classification task and the user preference?

(II) Computationally intensive metrics: this research area deals with the efficiency problem of calculating some distance metrics between two point sets, when the size of the point sets is huge. Distance metrics coming into consideration are computationally intensive metrics. In particular, these are metrics that attempt to calculate distances of all point pairs in the two point sets, e.g. the Hausdorff distance. This complexity becomes a problem when the set size is huge. Since the Hausdorff distance is based on finding the maximum of the minimum (i.e. for each point in the first set, the minimum distance in the second set should be found), it is of importance to find an efficient way to calculate such complex metrics.

(III) Hubness: this research area deals with the hubness problem (a common phenomenon of the curse of dimensionality), which arises when the dimensionality of the feature space is significantly increased. More specifically, this research area deals with

the questions: (1) what is the cause that leads to hubness emergence in high dimensional feature space, (2) how can we predict the probability of hubness emergence, i.e. how to measure the tendency of a particular data set to produce hubness, and (3) what are possible strategies to reduce the tendency to hubness?

1.5 Contribution

The contribution of this work is summarized as follows in relation to the research questions above:

- In the research area (I) metric bias and metric selection, we provide a comprehensive analysis of evaluation metrics of 3D medical image segmentation and conclude this analysis with guidelines for selecting evaluation metrics for 3D image segmentation based on the properties of the metrics and the properties of the segmentations.

Then, we generalize the results of this analysis to evaluation metrics for arbitrary domains, providing a novel formal method for measuring the bias of a given metric m to a particular property, based on the correlation between the rankings produced by the metric m and the rankings produced by the other metrics. Based on this method, we propose a formal general framework for metric selection. These contributions have been published in [THJ14b] and [TH15b].

- In the research area (II) computationally intensive metrics, we propose a novel efficient algorithm for calculating the exact Hausdorff distance in linear time in terms of the point set size. The algorithm is efficient in terms of speed and memory, and significantly outperforms state-of-the-art methods. Furthermore, it is general, i.e. it can be applied to arbitrary point sets. This algorithm has been published in [TH15a].

Furthermore, we provide an efficient algorithm for computing the average distance between two image segmentations that makes use of the nature of image segmentations being dense point sets.

- In the research area (III) hubness, we propose a formal explanation of the cause of hubness, based on the sparsity and distance concentration in high dimensional space. Based on this explanation, we propose a hubness indicator that predicts the tendency to hubness of a given data set in linear time without calculating the nearest neighbor lists of all points. Furthermore, we suggest two novel methods for hubness reduction, based on this formal explanation. These findings have been submitted in [THR15].

1.6 Formal Definitions and Notations

In this section we provide formal definitions, constructs, and notation that hold throughout the thesis to ensure consistency.

1.6.1 Metric and Metric Space

We provide definitions related to metrics and metric space. Some of these definitions, namely Definitions 1, 2, and 3, have already been defined in Section 1.1 and are restated here.

Definition 1. *Object:* We denote with object every thing that can be compared using metrics, i.e. a metric compares two objects.

Objects can be simple, e.g. single points or more complex, e.g. point sets. Images are also objects, since they are a special case of point sets. Objects can also be classifications or clusterings. We will use the term object when we talk about metrics in general.

A metric is a function defined on a set of object, such that for any pair of objects in the set, it provides a positive value indicating how far the two objects are from each other. The following provides a formal definition of a metric.

Definition 2. *Metric:* Let O be a set of objects with $A, B, C \in O$. Let the function ϕ be defined such that $\phi : O \times O \rightarrow \mathbb{R}$. The function ϕ is a metric iff it satisfies the following properties: (i) non-negativity, i.e. $\phi(A, B) \geq 0$, (ii) the coincidence axiom, i.e. $\phi(A, B) = 0$ if and only if $A = B$, (iii) symmetry, i.e. $\phi(A, B) = \phi(B, A)$, and (iv) the triangle inequality, i.e. $\phi(A, C) \leq \phi(A, B) + \phi(B, C)$.

Definition 3. *Metric space:* The ordered pair (O, ϕ) consisting of a set of objects O and a metric ϕ defined on $O \times O$ is called a metric space.

Depending on the nature of the objects in O , metric spaces can be of different types. In this work, we are mainly interested in metric spaces defined on top of feature spaces, In particular, we will deal with two types of metric spaces, namely when the objects are vectors (e.g. text documents) and point sets (e.g. images, graphs). In the following, we define these two types of metric space.

Definition 4. *Vector metric space* is a special case of the metric space according to Definition 2, in which the underlying objects are represented by vectors in the hyperspace. Let (X, ϕ) be a metric space defined on the point set $X = \{x_1, \dots, x_n\}$ with $x_i \in \mathbb{R}^d$ and the distance function $\phi : X \times X \rightarrow \mathbb{R}$. Without loss of generality we assume that X is normalized to have its mean at the origin. Unless it is explicitly specified, we will use $\|\cdot, \cdot\|$ to denote distance in general and $\|\cdot, \cdot\|_p$ to denote the p -norms, i.e.

$$\|x_i, x_j\|_p = \left(\sum_{m=1}^d |x_{im} - x_{jm}|^p \right)^{\frac{1}{p}}$$

is the distance between the two points x_i and x_j and

$$\|x\|_p = \left(\sum_{m=1}^d |x_m|^p \right)^{\frac{1}{p}}$$

is the distance of the point x to the mean.

Definition 5. *Point set metric space* is a special case of the metric space according to Definition 2, in which the underlying objects are represented by point sets in the (hyper)space. Let $X = \{x_1, \dots, x_n\}$ with $x_i \in \mathbb{R}^d$ be the set of all points in the d -hyperspace, and let $\mathcal{P}(X)$ be the powerset (the set of all subsets) of X . The point set metric space is defined as $(\mathcal{P}(X), \phi)$, where ϕ is a metric defined as $\phi : \mathcal{P}(X) \times \mathcal{P}(X) \rightarrow \mathbb{R}$.

1.6.2 Evaluation process

In this thesis, only one type of evaluation will be handled, namely the evaluation based on comparing objects with ground truth. We restate the definition of the evaluation process defined in Section 1.6, which will be considered in this chapter.

Definition 6. *Evaluation in the sense of this work: Let $O = \{o_1, \dots, o_n\}$ be a set of objects according to Definition 1 being evaluated. Let $\hat{O} = \{\hat{o}_1, \hat{o}_2, \dots\}$ be the set of ground truth objects. The evaluation is performed by comparing each object $o_i \in O$ against its corresponding ground truth object $\hat{o}_j \in \hat{O}$. Note that $|O|$ is not necessarily equal to $|\hat{O}|$ because a ground truth object normally corresponds to more than one object.*

1.6.3 2D and 3D Image Space

An image can be thought of as a set of points defined on a grid, i.e. the points are represented by grid cells, which we call pixels. Images can be 2-dimensional (2D) or 3-dimensional (3D). 3D images are also called volumes, and the 3D-pixels are called voxels. The metric space defined on a set of images is a special cases of the metric space according to Definition 3, in which the objects are images, and the metrics coming into consideration are only those according to Definition 5. Since 2D images are a special case of 3D images, we will only provide a definition for a 3D image, which implicitly holds for a 2D image as well.

Definition 7. *A 3D binary image (volume) is represented by the ordered pair (X, S) , where:*

- $X = \{x_1, \dots, x_n\}$ is a point set with $|X| = w \cdot h \cdot d$, where w , h and d are the width, height and depth of the grid on which the volume is defined, such that each point $x \in X$ corresponds to a grid cell (voxel). We will call w , h and d the grid dimensions and $w \cdot h \cdot d$ the grid size.
- S is a classification that assigns each grid cell (each point $x \in X$) to one of two classes, either the foreground or the background, such that S builds a partition $S = \{S^1, S^2\}$ on X represented by the assignment function $f^i(x)$ that provides the membership of the grid cell x in the subset S^i , where $f^i(x) = 1$ if $x \in S^i$ and $f^i(x) = 0$ if $x \notin S^i$.

The classification S can also be seen as a two class image segmentation. We denote S^1 by the foreground and S^2 by the background. We also define the term segment as the union of all voxels of the foreground.

Note that binary images are a special case of fuzzy images, in which the assignment function f has the range $\{0, 1\}$. This definition can be generalized to the fuzzy case by redefining the range of f to be $[0, 1]$ representing the degree of membership of a voxel to a particular class.

Definition 8. A fuzzy 3D image (volume) is an image according to Definition 7, in which the assignment function $f^i(x)$ is redefined to have its range in $[0, 1]$, where $f^i(x) \in [0, 1]$ represents the degree of membership of the grid cell x in the subset S^i .

1.6.4 Basic Cardinalities of the Confusion Matrix

Many of the metrics used for comparing 3D image segmentations can be derived from the four basic cardinalities of the so-called confusion matrix, namely the true positives (TP), the false positives (FP), the true negatives (TN), and the false negatives (FN). We define these cardinalities for the binary as well as the fuzzy case.

Basic cardinalities for binary segmentation: For two binary classifications that assign each element in a sets to one of two classes, in our case segmentations according to Definition 7, we define the four basic cardinalities (also called the confusion matrix), representing the overlap that results based on the agreement/disagreement of the assignments of the two classifications (segmentations). The four cardinalities are TP (true positive), FP (false positive), FN (false negative), and TN (true negative).

Definition 9. Let S_g and S_t be two segmentations according to Definition 7, with assignment functions f_g and f_t respectively. Let S_g denote the ground truth segmentation and S_t denote the segmentation being evaluated. The four cardinalities are given by the sum of agreement m_{ij} between each pair of subsets $i \in S_g$ and $j \in S_t$. That is

$$m_{ij} = \sum_{r=1}^{|X|} f_g^i(x_r) f_t^j(x_r) \quad (1.3)$$

where $TP = m_{11}$, $FP = m_{10}$, $FN = m_{01}$, and $TN = m_{00}$.

Table 1.1 shows the confusion matrix of the partitions S_g and S_t .

Table 1.1: Confusion matrix comparing two segmentations, S_g as the ground truth segmentation and S_t as the test segmentation

Subset	S_t^1	$S_t^2 (= \overline{S_t^1})$
S_g^1	$TP(m_{11})$	$FP(m_{12})$
$S_g^2 (= \overline{S_g^1})$	$FN(m_{21})$	$TN(m_{22})$

Generalization to fuzzy segmentation: Intuitively, one favorable way to generalize the metrics based on the basic cardinalities to the fuzzy is to generalize the cardinalities of the confusion matrix to the fuzzy case. To this end, the main task is to calculate the agreement between two segmentations, where the assignments of voxels to segments are probabilities (fuzzy). It is common for this purpose to use a suitable triangular norm

(t-norm) to calculate the agreement between two fuzzy assignments [KPM00][Cam07]. Given two probabilities $p1$ and $p2$ representing the memberships of a particular element (voxel) to a particular class (segment) according to two different classifiers (segmenters), we use the $\min(p1, p2)$ as a t-norm as the agreement between the two classifiers. That is, we define the agreement function $g : [0, 1] \times [0, 1] \rightarrow [0, 1]$ that models the agreement on a particular voxel being assigned to a particular segment as $g(p1, p2) = \min(p1, p2)$. This also means that the agreement on the same voxel being assigned to the background is given by $g(1 - p1, 1 - p2)$. Intuitively, the disagreement between the segmenters is the difference between the probabilities given by $|p1 - p2|$. However, since the comparison is asymmetrical (i.e. one of the segmentations is the ground truth and the other is the test segmentation), we consider the signed difference rather than the absolute difference as in Equations 1.5 and 1.7. The four cardinalities defined in Equation 1.3 can be now generalized to the fuzzy case as follows:

Definition 10. *Let S_g and S_t be two segmentations according to Definition 8, with assignment functions f_g and f_t respectively. Let S_g denote the ground truth segmentation and S_t denote the segmentation being evaluated. The four fuzzy cardinalities of the confusion matrix are given by*

$$TP = \sum_{r=1}^{|X|} \min(f_t^1(x_r), f_g^1(x_r)) \quad (1.4)$$

$$FP = \sum_{r=1}^{|X|} \max(f_t^1(x_r) - f_g^1(x_r), 0) \quad (1.5)$$

$$TN = \sum_{r=1}^{|X|} \min(f_t^2(x_r), f_g^2(x_r)) \quad (1.6)$$

$$FN = \sum_{r=1}^{|X|} \max(f_t^2(x_r) - f_g^2(x_r), 0) \quad (1.7)$$

Note that in Equations 1.4 to 1.7, $f_g^i(x_t)$ and $f_t^j(x_t)$ are used in place of $p1$ and $p2$ since each of the functions provides the probability of the membership of a given point in the corresponding segment, and in the special case of crisp segmentation, they provide 0 and 1.

Other norms have been used to measure the agreement between fuzzy memberships like the product t-norm, the L-norms, and the cosine similarity. We justify using the min t-norm by the fact that, in contrast to the other norms, the min t-norm ensures that the four cardinalities, calculated in Equations 1.4 to 1.7, sum to the total number of voxels, i.e. $TP + FP + TN + FN = |X|$ which is an important requirement for the definition of metrics. For example, applying other norms that do not satisfy this property to metrics based on the confusion matrix may lead to undesirable effects like negative metric values or perfect match values less/greater than one.

1.7 Structure of the Thesis

This thesis is structured according to the research areas of the research questions, that is each research area is addressed in a chapter. The remainder of the thesis is organized as follows. In Chapter 2, we cover metric bias and metric selection methods. In Chapter 3, we present a novel efficient algorithm for calculating the exact Hausdorff distance between two arbitrary point sets in linear time as well as an efficient method for calculating the average distance between image segmentations. In Chapter 4, problems related to high dimensionality of feature space are analyzed. Here, we introduce a novel explanation of hubness, a novel hubness indicator, as well as strategies for hubness reduction, based on the explanation proposed.

Metrics and Metric Bias

2.1 Introduction

Metric bias and sensitivity are a challenge in choosing evaluation metrics. In this chapter, we address the problem of measuring metric bias and selecting evaluation metrics. We do this in two related steps: In the first step, we address selection of evaluation metrics for a specific domain, namely medical 3D image segmentation by providing comprehensive analysis of the properties of 20 evaluation metrics, which is concluded by metric selection guidelines based on the properties segmentations and biases of the individual metrics. In the second steps, metric selection is generalized by providing a domain-independent framework for metric selection that generalizes the analysis provided in the first step by systematically inferring metric bias for a particular dataset based on the properties of the dataset.

Sensitivity to a particular property could prevent the discovery of particular errors or it could over/underestimate them. For example, when evaluating classifications, metrics can be sensitive to class imbalance, number of classes, etc. When evaluating image segmentation, metrics can be sensitive to outliers, number of segments, boundary complexity, etc. Another type of sensitivity is the inability of identifying classification caused by chance. This is related to the baseline value of the metric, which should ideally be zero when the classification is done at random, indicating a zero score [VEB10]. In Section 1.3.1, we introduced the need for a formal way for selecting evaluation metrics. Metric sensitivities are a challenge in choosing evaluation metrics.

Medical image segmentation, as an example of classification, suffers from a lack of standardization of evaluation methodology, and the absence of a formal way for selecting evaluation metrics. In medical image segmentation, an image (e.g. an MRI Volume) is automatically segmented, i.e. each of its pixels/voxels is either assigned or not assigned to a particular class, e.g. a tumor. There are different quality aspects in medical image segmentation according to which types of segmentation errors can be defined. Metrics

are expected to indicate some or all of these errors, depending on the data and the segmentation task.

Based on four basic types of errors (added regions, added background, inside holes and border holes), Shi et al. [SNL13] described four types of image segmentation errors, namely the quantity (number of segmented objects), the area of the segmented objects, the contour (degree of boundary match), and the content (existence of inside holes and boundary holes in the segmented region).

Fenster et al. [FC05] categorized the requirements of image segmentation evaluation into accuracy (the degree to which the segmentation results agree with the ground truth segmentation), the precision as a measure of repeatability, and the efficiency which is mostly related to time. Under the first category (accuracy), they mentioned two quality aspects, namely the delineation of the boundary (contour) and the size (volume of the segmented region). The alignment, which denotes the general position of the segmented object, is another quality aspect, which could be of more importance than the size and the contour when the segmented objects are very small.

Contributions: This part of the thesis investigates metric properties, metric sensitivities, and metric bias and provides a formal framework for selecting evaluation metrics for image segmentation. This is established in two main steps:

- A comprehensive investigation of a set of metrics for evaluating 3D medical image segmentation, namely 20 evaluation metrics that have been identified based on a literature review used for the VISCERAL Benchmarks [LMMH13]. This work has been published in [TH15b]. In particular, the work can be summarized by the following:
 - It provides an overview of 20 evaluation metrics for volume segmentation, selected based on a literature review. Cases where inconsistent definitions of the metrics have been used in the literature are identified, and unified definitions are suggested.
 - It provides fuzzy definitions for all selected metrics. This allows uncertainty in medical image segmentation to be taken into account in the evaluation.
 - It provides comprehensive analysis of the properties and biases of these metrics, based on the correlation among them under different conditions, and by means of empirical examples. Based on this analysis, it provides guidelines for selecting a subset of these metrics.
 - It provides an efficient open source implementation of all 20 metrics that outperforms state-of-the-art tools in terms of computation time and memory usage, especially when used for comparing huge 3D medical image segmentations.
- A general formal method for measuring metric bias and a framework for selecting evaluation metrics for arbitrary domains. In particular, we provide a generalization of the metric selection method, published in [THJ14b], which is restricted to image

segmentation. For this, we propose a novel method for inferring metric bias to the properties of the objects being evaluated, based on which we define a general framework for automatic selection of evaluation metrics for arbitrary evaluation task. However, this method has been demonstrated and tested using only 3D medical segmentations. Testing using other domains is recommended as future work.

Chapter organization: The remainder of this chapter is organized as follows. In Section 2.2, we present related work investigating metric properties or providing research on choosing evaluation metrics. In Section 2.3, we present research results done on evaluation of 3D medical image segmentation. In particular, we present in Section 2.4 a short literature review of 20 evaluation metrics used for evaluating 3D medical segmentation (definitions and algorithms for calculating these metrics are presented in Appendix A). In Section 2.5, we provide an in-depth analysis of the 20 metrics, and a discussion of their properties, bias, and utilities as well as guidelines for selecting a subset of these metrics. In Section 2.7, we present a formal method for measuring metric bias based on correlation among metrics as well as a framework for automatically selecting the most suitable evaluation metric(s), given a set of objects and an evaluation task. Finally, we conclude the chapter in Section 2.8.

2.2 State-of-the-Art

In this section, we present some related work, either studying metric properties to provide guidelines for choosing metrics or providing standardization of evaluation methodologies and evaluation tools.

2.2.1 Formal Foundation

Huang et al. [HL05] establish a formal framework for comparing two different measures and introduce two criteria for formal comparison of the goodness of evaluation metrics, namely the degree of consistency and the degree of discriminancy. To briefly describe this framework, consider the task of classification evaluation, where classifiers are evaluated by comparing automatic classification with ground truth classifications using metrics. The intuitive idea behind this framework is that if two measures f and g are used to evaluate two classifiers a and b , then it is desirable that at least f and g are consistent with each other, that is, when f stipulates that classifier a is (strictly) better than b , then g will not say b is better than a . Further, if f is more discriminating than g , we would expect to see cases where f can tell the difference between classifiers a and b but g cannot, but not vice versa. In particular, considering a sufficient number of classifications corresponding to the classifiers a and b , the degree of consistency (DoC) counts the cases where the metrics f and g agree on the goodness of the classifiers a and b , i.e. $f(a) > f(b) \iff g(a) > g(b)$. Once the DoC of two metrics reaches a satisfying level, then one of the metrics is selected depending on the degree of discriminancy (DoD),

which in contrast counts the cases where f can distinguish between the two classifiers, but g cannot do. If for example $Doc(f, g) > 0.5$ and $DoD(f, g) > 1$, then the metric f is better than g for evaluating these classifiers. Applying these criteria, Huang et al. showed theoretically and empirically that AUC (Area Under Curve) is a better measure than accuracy in evaluating the performance of classifiers. They also showed that AUC produces a better classifier ranking and gives better results when used in combination with a statistical analysis like ANOVA.

Busin et al. [BM13b] used axiomatics to define an abstract formal notation based on the concepts of measure, measurement, and similarity. They used these notations to define a set of axioms that should be satisfied by an effectiveness metric, i.e. these axioms are used as criteria to evaluate metrics. They claimed this axiom set to be extendable and the notation to fit any effectiveness metric. Other researchers [AGAV09] [VEB10] [Mei05] [WW07] also applied formal constraints based on axiometry to compare and judge evaluation metrics depending on the grade of satisfaction of these constraints. Since these axioms are of an abstraction level that is not in focus of this thesis, we forgo listing them in this place.

All these approaches deal with the problem only from a theoretical axiometrical point of view without taking into account the classification goal and the nature and properties of data being classified. To explain this, consider the sensitivity to outliers as an example. Using these approaches, the sensitivity of the metrics can be recognized, but this alone is not sufficient to select an effectiveness metric for a particular data set because of two reasons: (i) These approaches do not consider the level of outliers in the underlying particular data set, and (ii) they do not consider the evaluation goal, i.e. whether or not the outlier sensitivity is desirable for this particular evaluation goal. On the contrary, the novel approaches proposed in this chapter provide a way for metric selection that takes into consideration both of the underlying data set and the evaluation goal.

2.2.2 Bias and Sensitivity of Metrics

Metric bias is the tendency of a particular metric to reward/penalize objects because of particular characteristics (properties) they have. An example of bias is the sensitivity of some metrics to class imbalance, i.e. classification where one class vastly exceeds the other classes in size. Fatourehchi et al. [FWM⁺09] proposed a framework based on Desired Region of Operation (DROP) for selecting the best evaluation metric among a set of metrics for evaluating classification algorithms with an imbalanced dataset. They stated that when datasets are imbalanced, then care should be taken in evaluating classification algorithms.

Sakai [Sak06] proposed a method for evaluating evaluation metrics by measuring their sensitivity using Bootstrap Hypothesis Tests, and used this method in comparing seven evaluation metrics, namely *AveP* (average precision), nCG_{1000} (normalized cumulative gain at cut-off 1000), $nDCG_{1000}$ (normalized discounted cumulative gain at cut-off 1000), *Q-measure*, *G-AveP* (the geometric mean of *AveP*), *G-Q-measure* (the geometric mean of *Q-measure*), and $PDoc_{1000}$ (the precision at cut-off 1000). They concluded that

$Qmeasure$, $nDCG_{1000}$, and $AveP$ are very sensitive and $PDoc_{1000}$ is very insensitive while nCG_{1000} , $G - AveP$, and $G - Q - measure$ lie in the middle.

Buckley et al. [BV00] presented a way of estimating the stability of particular measures. They negated the belief that commonly used evaluation measures are equally reliable. Sakai [Sak07] provided comparisons between metrics depending on the sensitivity and stability using the Voorhees/Buckley swap method [VB02]. All these papers lack generality because they are methods designed either for specific metrics or for specific metric properties. Powers [Pow11] showed that commonly used measures (precision, recall and F-measures) are biased and don't consider the level of chance, that is they do not consider agreement caused by chance, e.g. they would measure an accuracy of 0.5 for a random classifier, whereas such a classifier should be given a zero score. Powers introduced the concepts of informedness and markedness as measures for the probability that a classification is caused by chance. This concept is based on betting as metaphor, that is pure guessing will leave a punter with nothing in the long run, while a punter with a certain knowledge will win every time. Based on this concept, he recommended using ROC (Receiver Operating Characteristic) curve as a less biased measure. Bradley [Bra97] also recommended using ROC and the area under the curve (AUC) as a measure of accuracy of machine learning algorithms because of its independence of decision threshold and its invariance to a prior class probabilities. Vinh et al. [VEB09] [VEB10] provided a study on using information theoretical measures for comparing clustering. They addressed the sensitivity of these metrics when the set size is small compared to the number of clusters and emphasize the need for chance adjusting in this case. Wallach [Wal06] recommends information theoretical measures like mutual information and variation of information for evaluating classifiers and shows that using classic precision, recall and accuracy could be misleading when comparing classifiers.

2.2.3 Metric Standardization and Tools

In the text retrieval domain, the TREC_EVAL tool¹ provides a standardization of evaluation that provides a standard reference to compare text retrieval algorithms.

Gerig et al. [GJC01] proposed a tool (Valmet) for evaluation of medical volume segmentation. In this tool only five metrics (volumetric overlap, probabilistic distance, Hausdorff distance, average distance, and interclass correlation) are implemented. There are important metrics, like information theoretical metrics as well as some statistical metrics like Mahalanobis distance, and metrics with chance correction like Kappa and adjusted Rand index, that are not implemented in the Valmet evaluation tool. Furthermore, this tool doesn't provide support for fuzzy segmentation. The ITK Library² provides a software layer that supports medical imaging tasks including segmentation. The ITK Library provides evaluation metrics that are mostly based on distance transform filters [MQR03]. However, this implementation has the following shortcomings: First, the ITK Library doesn't implement all metrics identified in a literature review (Chapter 2.4) to

¹More about TREC_EVAL under http://trec.nist.gov/trec_eval/

²National Library of Medicine Insight Segmentation and Registration Toolkit (ITK) www.itk.org

be relevant for evaluating medical segmentation. Second, since most of the metrics are based on distance transform filters, they are not scalable to increasing volume grid size, that is they are not efficient in terms of speed as well as memory used when the grid size is increased.

To illustrate the importance of a standard evaluation tool for medical image segmentation, we show in Section A.9 examples of metrics with more than one definition in the literature leading to different values, but each of them is used under the same name. There is a need for a standard evaluation tool for medical image segmentation which standardizes not only the metrics to be used, but also the definition of each metric.

2.3 Evaluation of Medical Image Segmentation

Medical image segmentation is an important image processing step in medical image analysis. Segmentation methods with high precision (including high reproducibility) and low bias are a main goal in surgical planning because they directly impact the results, e.g. the detection and monitoring of tumor progress [ZWB⁺04] [ZWKW04] [KMWV97]. Warfield et al. [WWG⁺99] denoted the clinical importance of better characterization of white matter changes in the brain tissue and showed that particular change patterns in the white matter are associated with some brain diseases. Accurately recognizing the change patterns is of great value for early diagnosis and efficient monitoring of diseases. Therefore, assessing the accuracy and the quality of segmentation algorithms is of great importance.

Medical 3D images are defined on a 3D grid that can have different sizes depending on the body parts imaged and the resolution. A formal definition of 3D image segmentation is provided in Section 1.6.3. The grid size is given as $(w \times l \times h)$ denoting the width, the length, and the height of the 3D image. Each 3D point on the grid is called a voxel. Given an anatomic feature, a binary segmentation can be seen as a partition that classifies the voxels of an image according to whether they are part or not of this anatomic feature. Examples of anatomic features are white matter, gray matter, lesions of the brain, body organs and tumors. Segmentation evaluation is the task of comparing two segmentations according to Definition 6, i.e. by comparing the segmentation being evaluated with its corresponding ground truth segmentation.

Medical segmentations are often fuzzy meaning that voxels have a grade of membership in $[0, 1]$. This is e.g. the case when the underlying segmentation is the result of averaging different segmentations of the same structure annotated by different annotators. Here, segmentations can be thought of as probabilities of voxels belonging to particular classes. One way of evaluating fuzzy segmentations is to threshold the probabilities at a particular value to get binary representations that can be evaluated as crisp segmentations. However, thresholding is just a workaround that provides a coarse estimation and is not always satisfactory. Furthermore, there is still the challenge of selecting the threshold because the evaluation results depend on the selection. This is the motivation for providing metrics that are capable of comparing fuzzy segmentations without loss of information. In this work, we provide fuzzy definition for each of the metrics analyzed in this section.

2.4 Evaluation Metric Overview

In this section, we present a set of 20 metrics for validating 3D medical image segmentation that were selected based on a literature review of papers in which 3D medical image segmentations are evaluated. Only metrics with at least two references of use are considered. An overview of these metrics is given in Table 2.1. Depending on the relations between the metrics, their nature and their definition, we group them into six categories, namely overlap based, volume based, pair-counting based, information theoretic based, probabilistic based, and spatial distance based. Column “category” in Table 2.1 assigns each metric to one of these categories. The aim of this grouping is to first ease discussing the metrics in this paper and second to enable a reasonable selection when a subset of metrics is to be used, i.e. selecting metrics from different groups to avoid biased results. In the following, we shortly describe each of these categories:

- **Spatial overlap based (Category 1):** These are metrics defined based on the spatial overlap between the two segmentations being compared, namely the four basic overlap cardinalities (TP, TN, FP, FN) described in Definition 9.
- **Volume based (Category 2):** Metrics from this category are based on comparing the volume of the segmented region, i.e. they aim to measure the number of voxels segmented compared with the number of voxels in the true segmentation (ground truth).
- **Pair counting based (Category 3):** Metrics from this category are based on $\binom{n}{2}$ tuples that represent all possible voxel pairs in the image. These tuples can be grouped into four categories depending on where the voxels of each pair are placed according to each of the segmentations being compared. These four groups are Group I: if both voxels are placed in the same segment in both segmentations. Group II: if both voxels are placed in the same segment in the first segmentation but in different segments in the second. Group III: if both voxels are placed in the same segment in the second segmentation but in different segments in the first. Group IV: if both voxels are placed in different segments in both segmentations.
- **Information theoretic based (Category 4):** Metrics of this category are based on basic values of the information theory like entropy and mutual information.
- **Probabilistic based (Category 5):** These metrics consider the segmentations being compared as two distribution. Under this consideration, the metrics are defined based on classic comparison method of statistics of these distributions.
- **Spatial distance based (Category 6):** These metrics aim to summarize distances between all pairs of voxels in the two segmentations being compared, i.e. they provide a one value measure that represents all pairwise distances.

An efficient implementation of all metrics in Table 2.1 is provided as an open source evaluation tool named EvaluateSegmentation. More details about EvaluateSegmentation is provided in Appendix A.10.

Complete definitions and calculation algorithms for each of these metrics as well as fuzzy definitions are presented in Appendix A. The column “definition” in Table 2.1 provides the equation numbers for the definition of each metric.

Table 2.1: Overview of evaluation metrics for 3D image segmentation. The symbols in the second column are used to denote the metrics. The column “reference of use” shows papers where the corresponding metric has been used in the evaluation of medical volume segmentation. The column “category” assigns each metric to one of the categories above. The column “definition” shows the equation numbers where the metric is defined.

Metric	Symb.	Reference of use in medical images	cat.	Definition
Dice (=F1-Measure)	<i>DICE</i>	[ZWB ⁺ 04], [ZWKW04], [KvdHR ⁺ 07], [CHL ⁺ 06], [GSP ⁺ 08], [KMJK ⁺ 10], [BPA ⁺ 08], [MJB ⁺ 12], [KCAB09], [CdLGBC09], [AFNIS13]	1	(A.6)
Jaccard index	<i>JAC</i>	[MJB ⁺ 12], [GSP ⁺ 08], [RPR13a], [VYPP11], [CdLGBC09], [AFNIS13], [KMJK ⁺ 10], [RPR13b]	1	(A.7)
True positive rate (Sensitivity, Recall)	<i>TPR</i>	[AFNIS13], [MJB ⁺ 12], [KMJK ⁺ 10], [KCAB09], [PLH ⁺ 12], [ULZ ⁺ 06]	1	(A.10)
True negative rate (Specificity)	<i>TNR</i>	[AFNIS13], [MJB ⁺ 12], [KMJK ⁺ 10], [ULZ ⁺ 06]	1	(A.11)
False positive rate (=1-Specificity, Fall-out)	<i>FPR</i>	→ Specificity	1	(A.12)
False negative rate (=1-Sensitivity)	<i>FNR</i>	→ Sensitivity	1	(A.13)
F-Measure (F1-Measure=Dice)	<i>FMS</i>	→ Dice	1	(A.15), (A.17)
Global Consistency Error	<i>GCE</i>	[RPR13a], [VYPP11], [YM13a], [YM13b], [RPR13b]	1	(A.18) to (A.20)
Volumetric Similarity	<i>VS</i>	[GSP ⁺ 08], [RPR13a], [VYPP11], [BPA ⁺ 08], [CdLGBC09], [GHS07], [RPR13b]	2	(A.22)
Rand Index	<i>RI</i>	[RPR13a], [VYPP11], [YM13a], [YM13b]	3	(A.31)
Adjusted Rand Index	<i>ARI</i>	[WBFR04], [MVvW05]	3	(A.34)

Mutual Information	MI	[ZWKW04], [RTR ⁺ 04], [KvdHR ⁺ 07]	4	(A.35) to (A.40)
Variation of Information	VOI	[RPR13a], [YM13a], [VYPP11], [YM13b]	4	(A.41), (A.37)
Interclass correlation	ICC	[GJC01], [DKC ⁺ 11]	5	(A.43)
Probabilistic Distance	PBD	[GJC01], [GSP ⁺ 08]	5	(A.45)
Cohens kappa	KAP	[MJB ⁺ 12], [ZWB ⁺ 04]	5	(A.46) to (A.48)
Area under ROC curve	AUC	[ZWKW04], [PLH ⁺ 12], [MVvW05]	5	(A.49)
Hausdorff distance	HD	[MJB ⁺ 12], [GJC01], [MNLBR07], [GSP ⁺ 08], [BPA ⁺ 08], [KCAB09], [CdLGBC09], [PN12]	6	(A.50), (A.51)
Average distance	AVD	[MJB ⁺ 12], [KCAB09]	6	(A.52), (A.53)
Mahalanobis Distance	MHD	[NVV99], [CdLGBC09]	6	(A.54) to (A.56)

2.5 Metric Analysis

In this section, we provide an analysis of the metrics in Table 2.1, namely a discussion about their properties, i.e. their strength, weakness, bias, and sensitivities in evaluating medical segmentation. For this, we use two strategies, the first is examining the correlation between rankings of segmentations produced by different metrics in different situations. The second method is analyzing the metric values for particular empirical examples, where the segmentations have particular properties. Based on this analysis, we provide guidelines for selecting a subset of these metrics for evaluating a set of medical image segmentations. This analysis should give a motivation for the need of a formal method for selecting evaluation metrics, which we will propose in Section 2.7.

2.5.1 Correlation among Metrics

In this section, we examine the correlation between rankings of segmentations produced by different metrics without putting any constraints on the segmentations being ranked. Figure 2.1 shows the result of a correlation analysis between the rankings produced by 16 of the metrics presented in Table 2.1 when applied to a data set of 4833 automatic MRI and CT segmentations. In this data set, all medical volumes provided by all the participants

in the VISCERAL project [LMMH13] Anatomy 1 and Anatomy 2 Benchmarks were included. Each medical image is a segmentation of only one of 20 anatomical structures varying from organs like lung, liver, and kidney to bone structures like vertebra, glands like thyroid, and arteries like aorta. More details on these structures are available in [JdTGM⁺14]. Note that the Jaccard (*JAC*) and F-Measure (*FMS*) were excluded because they provide the same ranking as the Dice coefficient (*DICE*), a fact that follows from the equivalence relations described in Section A.3.1. Also *FPR* and *FNR* were excluded because of their relations to *TNR* and *TPR* respectively, as given in Equations A.12 and A.13. In a first step, the volume segmentations were evaluated by comparing each of them with its corresponding ground truth using each of the 16 metrics (evaluation according to Definition 6), and then they were ranked using each of the metrics to get 16 rankings in total. Then, the pairwise Pearson’s correlation coefficients were calculated. Note that analyzing the correlation between rankings instead of metric values solves the problem that some of the metrics are similarities and others are distances and avoids the necessity to convert distances to similarities as well as to normalize metrics to a common range. Each cell in Figure 2.1 represents the Pearson’s correlation coefficients between the rankings produced by the corresponding metrics. The color intensity of the cells represent the strength of the correlation. Metrics in Figure 2.1 can be divided into three groups based on the correlation between the rankings produced by them, one group is at the top left (Group 1) including *ARI*, *KAP*, *ICC*, *DICE*, *AVD*, *MHD*, *PBD*, and *VS* and another group is at the right bottom (Group 2) including *TNR*, *RI*, *GCE*, and *VOI*. The metrics in each of these groups strongly correlate with each other, but have no correlation with metrics in the other group. The remaining metrics (Group 3) including *MI*, *AUC*, *TPR*, and *HD* have medium correlation between each other and the other groups. A deeper consideration of the metric definitions shows that Group 1 and Group 2 classify the metrics according to whether they consider or do not consider the true negatives (background voxels) in their definitions. While all metrics in Group 2 include the true negatives in their definitions, none of the metrics in Group 1 does this. Note that the adjusted Rand index and the kappa measures principally include the true negatives in their definitions, but both of them perform chance adjustment, which eliminates the impact of the true negatives, i.e. avoids that the influence of the background dominates the result [FWM⁺09]. Also note that the average distance (*AVD*) and the Mahalanobis distance (*MHD*) in Group 1 do not consider the true negatives, since they are based on the distances between the foreground voxels (non-zero voxels). Considering the true negatives in the evaluation has a large impact on the result, since the background (normally the largest part of the segmentation) contributes to the agreement. Figure 2.2 illustrates, by means of a real example, how metrics based on the true negatives change the resulting rankings when the true negatives are reduced by selecting a smaller bounding cube [ULZ⁺06]. Such metrics are biased against the ratio between the total number of foreground voxels and the number of the background voxels, which is denoted as the class imbalance. This leads to segmentations with large segments being penalized and those with small ones being rewarded, a case that is common in medical image segmentation e.g. when the quality of two segmentations is to be compared, where one

	ARI	KAP	ICC	DICE	AVD	MHD	PBD	VS	MI	AUC	TPR	HD	TNR	RI	GCE	VOI	
ARI	1.00	1.00	1.00	1.00	0.95	0.93	0.91	0.81	0.80	0.75	0.74	0.52	-0.07	-0.07	-0.15	-0.15	Group 1
KAP	1.00	1.00	1.00	1.00	0.95	0.93	0.91	0.81	0.80	0.75	0.74	0.52	-0.08	-0.08	-0.16	-0.16	
ICC	1.00	1.00	1.00	1.00	0.95	0.93	0.91	0.81	0.81	0.75	0.74	0.52	-0.08	-0.09	-0.17	-0.17	
DICE	1.00	1.00	1.00	1.00	0.95	0.93	0.91	0.81	0.81	0.75	0.74	0.52	-0.08	-0.09	-0.17	-0.17	
AVD	0.95	0.95	0.95	0.95	1.00	0.93	0.86	0.76	0.67	0.70	0.69	0.70	0.07	0.08	0.00	0.00	
MHD	0.93	0.93	0.93	0.93	0.93	1.00	0.83	0.71	0.73	0.74	0.74	0.53	-0.07	-0.06	-0.13	-0.13	
PBD	0.91	0.91	0.91	0.91	0.86	0.83	1.00	0.74	0.71	0.65	0.64	0.45	-0.07	-0.09	-0.16	-0.16	
VS	0.81	0.81	0.81	0.81	0.76	0.71	0.74	1.00	0.60	0.45	0.44	0.40	-0.03	0.00	-0.08	-0.07	
MI	0.80	0.80	0.81	0.81	0.67	0.73	0.71	0.60	1.00	0.65	0.65	0.22	-0.49	-0.58	-0.64	-0.64	Group 3
AUC	0.75	0.75	0.75	0.75	0.70	0.74	0.65	0.45	0.65	1.00	1.00	0.35	-0.35	-0.14	-0.19	-0.19	
TPR	0.74	0.74	0.74	0.74	0.69	0.74	0.64	0.44	0.65	1.00	1.00	0.34	-0.36	-0.15	-0.20	-0.20	
HD	0.52	0.52	0.52	0.52	0.70	0.53	0.45	0.40	0.22	0.35	0.34	1.00	0.32	0.35	0.30	0.30	
TNR	-0.07	-0.08	-0.08	-0.08	0.07	-0.07	-0.07	-0.03	-0.49	-0.35	-0.36	0.32	1.00	0.84	0.84	0.84	Group 2
RI	-0.07	-0.08	-0.09	-0.09	0.08	-0.06	-0.09	0.00	-0.58	-0.14	-0.15	0.35	0.84	1.00	0.99	1.00	
GCE	-0.15	-0.16	-0.17	-0.17	0.00	-0.13	-0.16	-0.08	-0.64	-0.19	-0.20	0.30	0.84	0.99	1.00	1.00	
VOI	-0.15	-0.16	-0.17	-0.17	0.00	-0.13	-0.16	-0.07	-0.64	-0.19	-0.20	0.30	0.84	1.00	1.00	1.00	
	Group 1							Group 3					Group 2				

Figure 2.1: The correlation between the rankings produced by 16 different metrics. The pairwise Pearson’s correlation coefficients between the rankings of 4833 medical volume segmentations produced by 16 metrics. The color intensity of each cell represents the strength of the correlation, where blue denotes direct correlation and red denotes inverse correlation.

of them is larger, and the other one is smaller than the ground truth segmentation. Vinh et. al [VEB10] stated that such metrics need chance adjustment, since they do not meet the constant baseline property.

The correlation between metric is determined by factors of three categories. The first category are factors concerning the definitions of the metrics, examples of this category are whether or not the false positives are included in the definition and whether or not the spatial positions of the points (e.g. voxels) are considered. The second category are factors concerning the objects being compared. Examples of this category are the level of outliers in the segmentations being compared and the size of the segments in each segmentation. The third category are factors concerning the relations between each pair







	Ground truth		Segmentation			
A		↔		DICE	0.939	s
				AVG	0.204	D
				VS	0.953	s
				RI	0.986	s
				GCE	0.013	D
				TNR	0.994	s
B		↔		DICE	0.839	s
				AVG	1.149	D
				VS	0.855	s
				RI	0.970	s
				GCE	0.026	D
				TNR	0.999	s
Á		↔		DICE	0.939	s
				AVG	0.204	D
				VS	0.953	s
				RI	0.878	s
				GCE	0.124	D
				TNR	0.897	s

Figure 2.2: The effect of decreasing the true negatives (background) on the ranking Each of the segmentations in A and B is compared with the same ground truth. All metrics assess that the segmentation in A is more similar to the ground truth than in B . In \acute{A} , the segmentation and ground truth are the same as in A , but after reducing the true negatives by selecting a smaller bounding cube. The metrics RI , GCE , and TNR change their rankings as a result of reducing the true negatives. Note that some of the metrics are similarities (marked with S) and others are distances (marked with D).

of objects being compared. One example of this category is the overlap between the objects in each pair, i.e. whether the two objects have small or large overlap between them. Note this factor is self a metric, e.g. the $DICE$.

Obviously, given a set of metrics, factors from the first category do not change, since they depend on the definition of these metrics. However, factors from the other two categories change with the data. We want to examine how the correlation changes with these factors, i.e. how consistent is the correlation presented in Figure 2.1.

In one experiment, we examined the correlation for random subset of the original data set. We performed the experiment (calculating the metric correlation), but for subsets, selected randomly from the original dataset with sizes varying from 10% to 100% of the original dataset. The result was almost identical correlation in all cases. This can be explained by the fact that a random selection of object keeps factors from the second and third categories unchanged in average, which results in an identical correlation. Now the question is how the correlation changes when selecting subsets not random, but rather according to a particular factor. This is what we analyze in the next section.

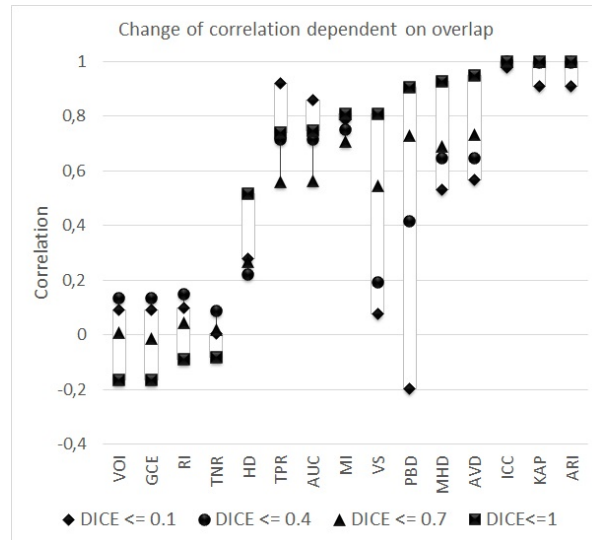


Figure 2.3: The effect of overlap on the correlation between rankings produced by different metrics. The positions and heights of the bars show how metrics correlate with *DICE* and how this correlation depends on the overlap between the compared segmentations. Four different overlap ranges are considered.

2.5.2 Effects of Overlap on the Correlation

In this section, we examine the correlation between metrics when the underlying objects have particular overlap between them. This experiment is motivated by the obvious fact that the strong correlation between overlap based metrics and distance based metrics (Figure 2.1) cannot hold in all cases. For example, consider the case where the overlap between segments is zero, here all overlap based metrics provide zero values regardless of the positions of the segments. On the contrary, distance based metrics still provide values dependent on the spatial distance between the segments. This motivated us to examine how the correlation described in Section 2.5.1 behaves when only segmentations with overlap values in particular ranges are considered.

Figure 2.3 shows the Pearson's correlation between the *DICE* and each of the other metrics when the measured *DICE* is in a particular range. One important observation is that the correlation between *DICE* and the distance based metrics (*AVD*, *HD*, and *MHD*) decreases with decreasing overlap, i.e. with increasing false positives and false negatives. This is intuitive because overlap based metrics, in contrast to distance based metrics, don't consider the positions of voxels that are not in the overlap region (false positives and false negatives), which means that they provide the same value independent of the distance between the voxels. It follows that increasing the false positives and/or false negatives (decreasing overlap) means increasing the probability of divergent correlation. Another observation is the strongly divergent correlation between volumetric similarity (*VS*) and *DICE*. This divergence is intuitive since the *VS* only

compares the volume (voxel count in case of binary images) of the segment(s) in the automatic segmentation with the volume in the ground truth, which implicitly assumes that the segments are optimally aligned. Obviously, this assumption only makes sense when the overlap is high. Actually, the VS can have its maximum value (one) even when the overlap is zero. However, the smaller the overlap, the higher is the probability that two segments that are similar in volume are not aligned, which explains the strong divergence in correlation when the overlap is low.

Finally, the highest divergence in the correlation is observed with the probabilistic distance (PBD). This is caused by the fact that PBD , in contrast to $DICE$, over-penalizes false positives and false negatives. This can be explained by means of the definition of the PBD in Equation A.44: differences in the voxel values in the compared segmentations have a double impact on the result because they increase the numerator and decrease the denominator at the same time, causing the distance to increase rapidly. Actually, the PBD even reaches infinity when the overlap reaches zero. PBD behaves the opposite of the VS regarding the sensitivity to the alignment, i.e. it strongly penalizes alignment errors (we mean with alignment errors that the segmented volume is correct, but the overlap is low). This makes PBD suitable for tasks where the alignment is of more interest than the volume and the contour.

2.5.3 Boundary Errors

Anatomy structures that are segmented can be of different grades of complexity in terms of boundary delimitation. They can vary from simple and smooth shapes, like a kidney, to irregular shapes, like tumors, but also branched and complex like the vessels of the eye retina. It depends on the goal of the segmentation, whether the exact delimitation of the boundary is important or not. For example, the boundary can be of importance when the goal is monitoring the progress of a tumor. In other cases, the goal is to estimate the location and the size or general shape of an anatomical structure, e.g. a lesion. Here the alignment and the extent are rather more important than the boundary. Another requirement could be maximizing the recall at the cost of the boundary delimitation, i.e. to ensure that the segmented regions contain (include) all of the true segment, e.g. when the goal is to remove a tumor. In this section, we analyze the metrics in terms of their capabilities of (i) penalizing boundary errors, (ii) rewarding recall, and (iii) discovering the general shape, thereby ignoring small details.

Penalizing boundary errors Figure 2.4 illustrates the fact that metrics differently consider boundary delimitation. In (A) a star is compared with a circle and in (B), the same star is compared with another star that has the same shape and dimensions, but is slightly rotated so that the resulting overlap errors FP and FN (obviously also the TP and TN) are the same as in (A). It follows that all metrics defined based on the overlap error cardinalities provide the same similarity between the two shapes in each case, which has been also confirmed empirically. This means that they do not discover that the shapes in (B) are more similar than those in (A), which also implies that such metrics

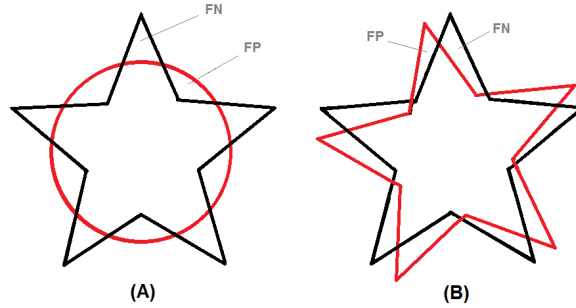


Figure 2.4: Metrics that fail to discover boundary errors. In (A), the star is compared with a circle and in (B) the same star is compared with another star of the same dimensions, rotated so that the resulting overlap errors (FP and FN) are equal in magnitude in both cases. All metrics that are based on FP and FN (overlap-based metrics) are not able to discover that the two shapes in (B) are more similar to each other than those in (A). On the contrary, all spatial distance based metrics discover the similarity and give (B) a higher score than (A). However, the metric most invariant to boundary error is the volumetric similarity, since it gives a perfect match in both cases.

are not recommended when segmentation algorithms are expected to provide accurate boundaries. However, the spatial based distance metrics, in particular the HD and the AVD , discover these boundary errors and provide higher similarity values for case (B). This makes these two metrics more suitable for cases where the boundary delimitation is of interest. Actually, as already mentioned in Section 2.5.2, this suitability follows from the fact that spatial based metrics consider the positions of the FP and FN in contrast to the overlap based metrics where FP voxels as well as FN voxels count the same regardless of their distances from the true positions. The volumetric similarity (VS) is also not recommended to discover boundary errors. Note that in (A) and (B), the VS provides a perfect match, given $|FP| = |FN|$ regardless of the boundary. VS is recommended for cases where the segmented volume is in the focus of interest regardless of the boundary and the alignment.

Rewarding recall Segmentation errors can be due to missing regions (parts in the ground truth that are missing in the automatic segmentation) or added regions (parts in the automatic segmentation without corresponding parts in the ground truth). Depending on the application, sometimes missing regions harm more than added regions, which means that algorithms are preferred that aim to maximize recall on cost of precision, i.e. avoid missing regions, even on cost of having added regions. In this case, metrics that reward recall could be a good choice. Figure 2.5 illustrates in 2D how metrics differ in evaluating segmentations in terms of missing and added regions. In one case, the ground truth segment GT is compared with a smaller segment A and in another case GT is compared with a larger segment B. The distance between the boundary of the ground truth and the boundary of the segment δ is equal in both cases. However, the

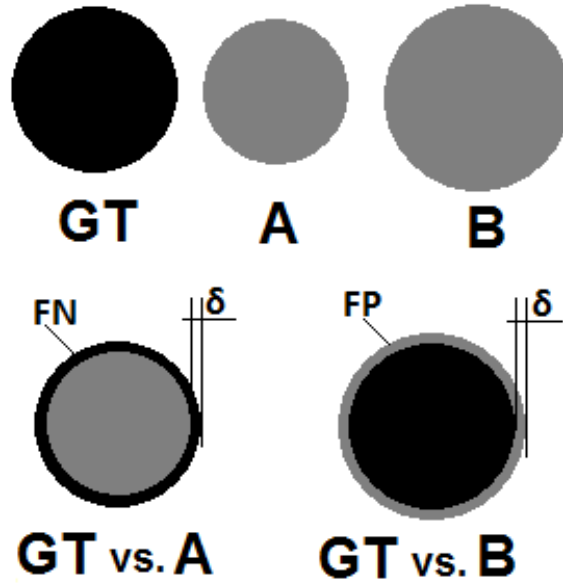


Figure 2.5: Boundary errors: rewarding/penalizing recall. Illustration in 2D of boundary errors that decrease/increase recall. The ground truth image GT is compared with the image A that is smaller than GT and with another image B that is larger than GT . Although the boundary error in both cases is equal (δ), the magnitude of the resulting false negative (FN) with A is smaller than the resulting false positive (FP) with B . This causes that metrics, considering the absolute magnitudes of FN and FP, penalize high recall.

volume differences (FN and FP) are not equal, which causes metrics based on the four cardinalities (TP, TN, FP, FN) to evaluate the two cases differently. The metrics MI (mutual information) and TPR (recall) reward recall and hence evaluate B as better than A . This is because MI measures how much information the segmentations have in common, which obviously increases with recall.

General shape and alignment The Mahalanobis distance MHD (Equations A.54 to A.56) measures the distance between two segmentations by comparing estimates of them, in particular it considers the two ellipsoids that best represent the segmentations [Mah36]. This way of comparison ignores the boundary details and considers only the general shape and the alignment of the segments. This could be a good choice when obtaining the exact shape of the segment is not a requirement.

2.5.4 Effect of Segmentation Density

The density of segments in automatic segmentations can vary depending on the strategies used by the segmentation algorithms. While some algorithms produce solid segments,

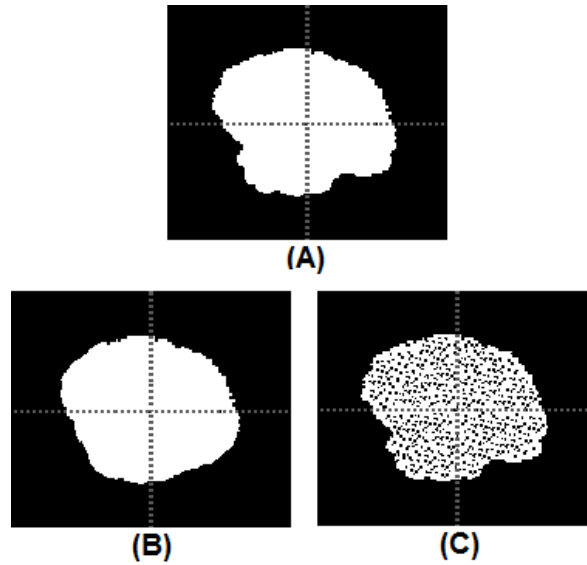


Figure 2.6: The effect of segment density. Two segmentations (B) and (C) are compared with the corresponding ground truth (A). (B) has a solid structure while (C) has a lower density due to large number of tiny holes uniformly distributed inside it. Although (C) has a higher accuracy of the boundary than (B), all metrics, excepts MHD and HD , give (B) a higher score than (C).

others produce segments with low density, e.g. due to a huge number of uniformly distributed tiny holes. It depends on the goal of the segmentation, whether the density of a segment is of importance or not. In some cases, the density has a meaning e.g. when it should measure the progress of a disease, and in other cases it is meaningless, e.g. when anatomical structures are to be localized, e.g. organs. There are cases where algorithms work very well in identifying the boundary of the structure being segmented, but produce segments with low density. Figure 2.6 shows a real example of brain tumor segmentation from the BRATS 2012 challenge, where a segmentation algorithm provides a solid segment (B) with low accuracy in identifying the boundary, and another algorithm (C) produces a segment with a boundary of higher accuracy, but the density is low due to numerous tiny holes. When comparing each of these cases with the corresponding ground truth (A), all the metrics, except the Mahalanobis distance (MHD) and the Hausdorff distance (HD), measure a higher similarity (or smaller distance) in (B) than in (C). The explanation is obvious, since all tiny holes are calculated as false negatives, which has impact on all metrics defined based on the four cardinalities (TP, TN, FP, FN). On the other hand, since the MHD estimates the general shape of the segment, thereby ignoring small details, it is not sensitive to segment density. Also the HD is not sensitive, since it is a max min operation, which means that errors caused by the tiny holes are ignored, when there exist larger errors. Given that the task is to identify the tumor core using a crisp segmentation, i.e. assigning each voxel either as tumor core

or background, the question is whether it is justified to penalize the low density of the segment. However, in cases where the segment density is to be ignored, metrics with such sensitivity should be avoided.

2.5.5 Effect of Segment Size

There is an inverse relation between segment size (relative to the grid size) and the expectation value of the alignment error, which directly follows from the degree of freedom for the segment location being higher when the segment is small. Furthermore, there is a direct relation between the expectation of alignment error and overlap between the segment in the ground truth and that in the segmentation under test. For small segments, the expectation value of the alignment error can be comparable in magnitude with the segment size, which results in the probability of small (or zero) overlap being high. In such a case, all metrics based on the four overlap cardinalities (TP, TN, FP, FN), e.g. the overlap based metrics, are not suitable, since they would provide the same value regardless of how far the segments are from each other, once the overlap is zero. To illustrate this effect, consider comparing two linear segments using *DICE*. Assume that these linear segments have almost exact match, but the overlap is zero. Here, the *DICE* provides the same value (zero) for these two lines and for another two lines that are far from each other. Figure 2.7 illustrates this effect. In (A), two large segments having a displacement Δ from each other. This results in a relatively large overlap, which makes overlap based metrics conceivable. On the contrary when the segments are thin and long as in (B) or very small as in (C) so that they have no overlap, in such a case although having the same displacement Δ , overlap based metrics give zero similarity, which is not always reasonable. For example, a segmentation algorithm that provides a segmentation of a blood vessel that does not exactly match the ground truth vessel, but tight beside it, such algorithm may be worth a higher score than zero. Another problem with overlap based metrics is that they are not sensitive to voxel positions, that is once a voxel is a false positive, it does not matter where it is. Figure 2.7 (C) and (D) illustrate this effect: Although the displacement δ in (D) is smaller than the displacement Δ in (C), overlap based metrics cannot discover that the segmentation (D) is better than (C). Obviously, metrics based on the volume, e.g. the volumetric similarity have also the same drawback.

Distance based metrics are recommended when the segments are small because they always provide values that are proportional to the spatial distance regarding of the overlap.

The question that arises here is, can we estimate a threshold that separates small from large segments? To answer this question, we first define small segment to be when the smallest dimension of this segment, i.e. $\min(\text{length}, \text{width}, \text{height})$, is significantly less than the corresponding dimension of the grid on which the image is defined, which means that at least one dimension should be small compared with the corresponding dimension of the grid. This results in three types of small segments, namely (i) segments that are small in only one dimension (planar shapes), (ii) small in two dimensions (linear shapes), (ii) or small in three dimensions (point similar shapes). The decision about whether segments are small or not can be done based on the distribution of the segment

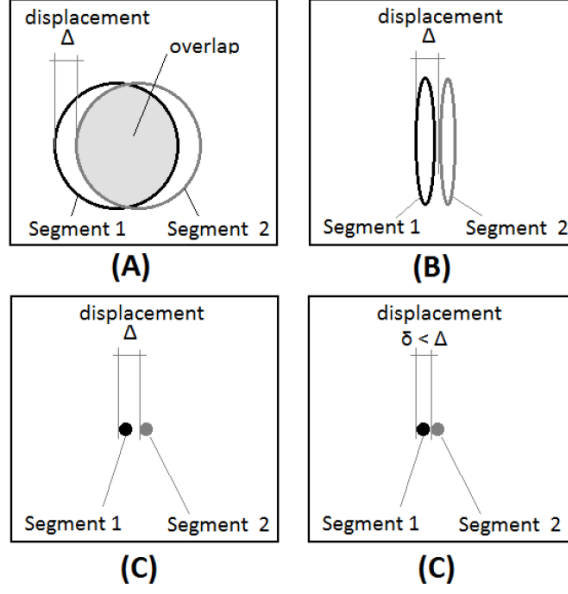


Figure 2.7: Illustration of sensitivity of overlap based metrics to segment size. In (A), two large segments having a displacement Δ . This results in a relatively large overlap. In (B), two thin segments with the same displacement, but without any overlap. Overlap based metrics give zero. Also in (C), small segments with the same displacement have no overlap. Overlap based metrics cannot differentiate the quality between (C) and (D) with a smaller displacement.

sizes, i.e. by considering how likely two segments have no overlap. However, it is helpful to define a threshold as a rule of thumb for judging segment size. We define this threshold as follows:

Let S_1 and S_2 be two segments being compared, and assume for simplicity that both of them have the same size. Let s be the smallest dimension of the segments and g be the corresponding dimension of grid. We will calculate the threshold $\epsilon = \frac{s}{g}$ based on the expectation value of the distance between the centers of the two segments S_1 and S_2 .

We calculate this expectation value by considering all possible locations that the two segments can take along the dimension g . Note that we consider the possible locations only in the critical direction (direction of the smallest dimension). The degree of freedom of the locations is governed by $g - s$ because we assume that the segments should be entirely within the grid, which means they cannot be located farther than $s/2$ from the borders. In the a first calculation alternative (which only has explanation purpose) we assume that the locations are uniformly distributed. In this case the expectation value of the distance between the segments $E[d(S_1, S_2)]$ is given by

$$E[d(S_1, S_2)] = \frac{1}{(g-s)^2} \sum_{i=1}^{g-s} \binom{g-s}{i} = \frac{g-s}{3} \quad (2.1)$$

Since a threshold is required that separates segments having overlap from those having no overlap, the segment size s should not exceed the expectation value of the distance between segments (note that the distance between the centers is meant), i.e.

$$s > \frac{g-s}{3} \implies s > \frac{g}{4} \quad (2.2)$$

Obviously, this threshold is not realistic, because it assumes that the locations are uniformly distributed, which only holds for a random segmentation algorithm. In practice, the locations are concentrated at a particular region of the grid which is dependent on the general quality of the segmentation algorithms being evaluated. We will use the Pareto method to estimate this region, i.e. we assume that 80% of the segment locations will be in 20% of the possible space. Applying this to Equations 2.1 and 2.2, we obtain:

$$E[d(S_1, S_2)] = \frac{20}{100} \frac{80(g-s)}{3 \cdot 100} + \frac{80}{100} \frac{20(g-s)}{3 \cdot 100} = \frac{32}{100} \frac{g-s}{3} \quad (2.3)$$

$$s > \frac{32}{100} \frac{g-s}{3} \implies s > \frac{32}{332}g \implies \epsilon \sim 0.10 \quad (2.4)$$

Note that the threshold depends on the general quality of the segmentations. For example if we assume instead a Pareto distribution of 10% : 90%, we get a threshold $\epsilon \sim 0.05$.

2.6 Metric Properties and Metric Selection Guidelines

In this section, based on the analysis presented in Section 2.5, we define some properties of evaluation metrics (Section 2.6.1), some properties of image segmentation (Section 2.6.2), and some general requirements that can be put on the segmentation task (Section 2.6.3). Based on these definitions, we provide metric selection guidelines that can be used to select evaluation metrics for 3D image segmentation.

2.6.1 Metric Properties

Based on the results of the discussion so far, we summarize the properties of the metrics that are relevant for segmentation. In particular, we define these properties and assign them to the metrics listed in Table 2.1.

- *Outlier sensitivity*: Sometimes automatic segmentations have outliers in form of few pixels outside the segment. The underlying property describes metrics that strongly penalize such outliers.
- *True negatives consideration*: In a two class segmentation, the voxels are assigned either to the single segment or to the background. The voxels that are assigned as background by both the automatic segmentation and the ground truth are called the true negatives. The underlying property describes metrics that calculate the true negatives as a part of the agreement between the automatic segmentation and the ground truth.

- *Chance adjustment:* The agreement between two segmentations could be caused by chance. The score measured for a segmentation performed randomly is called the baseline. The base line value of a metric should ideally be zero. The underlying property describes metrics that are defined to consider agreement caused by chance, i.e. to minimize the baseline value.
- *Sensitivity to point positions:* Some metrics, e.g. overlap-based metrics, do not consider the position of false positive voxels, i.e. they provide the same result wherever these voxels are. The underlying property describes metrics that do consider the position of the false positive, i.e. their values differ depending on where these voxels are.
- *Ignoring alignment errors:* alignment errors are when the segment in the automatic segmentation has similar shape and similar volume as the corresponding segment in ground truth, but it is not correctly aligned, e.g. translated or rotated. Some metrics are invariant to alignment error, i.e. they cannot discover them, like the volumetric similarity.
- *Recall rewarding:* Describes metrics that are not sensitive to errors increasing recall, in particular they penalize boundary errors that decrease the segmented volume more than errors that enlarge the segmented volume.
- *General shape and alignment:* Describes metrics that ignore small details and judge only the general shape and alignment of the segmented region.
- *Overlap-based:* This property describes metrics that are based on four types of overlap (TP, TN, FP, FN) between the automatic segmentation and the ground truth.
- *Distance-based:* This property describes metrics that are defined as functions of the Euclidean distances between the voxels of the segment in the automatic segmentation and the voxels of the segment in the ground truth.
- *Information theoretical-based:* Describes metrics based on information theoretical factors like the entropy.
- *Probabilistic-based:* Describes metrics defined as functions of statistics calculated from the voxels in the overlap regions of the segmentations.
- *Pair-counting-based:* Considering that a segmentation is a partitioning of an image, pair-counting-based metrics divides the tuples representing all possible object pairs into four groups depending on where the objects of each tuple are placed regarding the partitions, i.e. whether they are placed in the same partition or in different partitions. More details can be found in Appendix A.5
- *Volume-based:* Describes metrics that are defined based on the volume of the segmented region (e.g. the voxel count in case of binary segmentations). Note

that volume-based metrics are totally different from overlap-based metrics because the former consider the absolute volume of the segmented region regardless of the overlap.

Now, depending on whether each of these properties holds or does not hold for a particular metric, we present the property assignments in Table 2.2, in which a check marked cell denotes that the corresponding metric has the corresponding property. This assignment will be used later in Section 2.6.4 to define a protocol for selecting evaluation metrics.

2.6.2 Segmentation Properties

Metric selection should consider, among others, the properties of the segmentations being evaluated. In this section, we define some of the properties that segmentations can have, to which metrics can be sensitive. These properties will be used in combination with the metric properties to define a protocol for metric selection in Section 2.6.4.

- *Outliers*: In segmentation, outliers are relatively small wrongly segmented regions outside (normally far from) the segment. Metrics sensitive to outliers over-penalize them. When outliers do not harm, metrics with sensitivity to outliers, such as the *HD*, should be avoided.
- *Small segment*: When a segment size is significantly smaller than the background, so that it is comparable in magnitude with the expectation of the alignment error, then all metrics based on the four overlap cardinalities (TP, TN, FP, FN), e.g. the overlap based metrics, as well as volume based metrics (*VS*) are not suitable. Small segments are those with at least one dimension being significantly smaller than the corresponding dimension of the grid on which the image is defined (e.g. less than 5% of the corresponding grid dimension). In this case, distance based metrics are recommended.
- *Complex boundary*: While some segments have smooth boundaries, there are others that have a non-regular shaped complex boundary, which are denoted by this property. Metrics that are sensitive to point positions (e.g. *HD* and *AVD*) are more suitable to evaluate such segmentations than others. Volume based metrics are to be avoided in this case.
- *Low densities*: Some algorithms produce segmentations that have a good quality in terms of contour and alignment, but the segments are not solid, but rather have a lower density, e.g. because of numerous tiny holes. All metrics based on the four cardinalities are sensitive to segment density. They penalize low density and hence should be avoided in cases where the low density does not harm. In these cases, distance based metrics (*HD*, *AVD*, and *MHD*) are good choices.
- *Low segmentation quality*: This property describes segmentations that have in general a low quality, i.e. it can be assumed that the segments have in general low

Table 2.2: Assignment between the properties defined in Section 2.6.1 and the metrics defined in Table 2.1. A particular metric has a particular property iff the corresponding cell is check marked.

	Outlier sensitive	True negatives consideration	Chance adjustment	Sensitive to point positions	Ignoring alignment errors	Recall rewarding	General shape & alignment	Overlap-based	Distance-based	Information theoretical	Probabilistic-based	Pair-counting-based	volume-based
DICE								✓					
JAC								✓					
TPR						✓		✓					
TNR		✓						✓					
FPR								✓					
FNR								✓					
FMS								✓					
VS					✓								✓
GCE		✓											
RI		✓										✓	
ARI		✓	✓									✓	
MI		✓				✓				✓			
VOI		✓								✓			
ICC		✓	✓								✓		
PBD											✓		
KAP		✓	✓								✓		
AUC		✓									✓		
HD	✓			✓					✓				
AVD				✓					✓				
MHD				✓			✓		✓				

overlap with the corresponding segments in the ground truth segmentation. When the overlap is low, distance based metrics are more capable of differentiating between segmentation qualities than volume based metrics. The volumetric similarity VS should be avoided.

2.6.3 Requirements on the Segmentation Algorithms

Depending on the goal of the segmentation, there could be special requirements on the segmentation algorithms. Many different requirements could be defined, which can strongly differ from case to case. Some of the requirements that could be put on the segmentation algorithms are:

- *Contour is important:* Depending on the individual task, the contour can be of interest, that is the segmentation algorithms should provide segments with boundary delimitation as exact as possible. Metrics that are sensitive to point positions (e.g. HD and AVD) are more suitable to evaluate such segmentation than others. Volume based metrics are to be avoided in this case.
- *Alignment is important:* When the requirement is the location (general alignment) of the segment rather than the boundary delimitation. In this case, the volume based metrics are not a good choice.
- *Recall is important:* In some cases, it is an important requirement that the segmented region includes at least all the true segment, regardless of including parts of the false region. Obviously, the boundary delimitation in this case is of less interest, and the algorithms should rather maximize the recall. Metrics that reward recall are the mutual information MI and the true positive rate TPR .
- *Volume is important:* Sometimes the magnitude of the segmented region is of more importance than the boundary and the alignment. Here, algorithms should segment region to have a volume as near to that of the true segment as possible. The volumetric similarity VS is recommended.
- *Only general shape and alignment:* The exact boundary and high overlap are not always requirements. Depending on the goal, sometimes the general shape and the alignment (location) are sufficient, e.g. when the requirement is to identify lesions and give an estimation of the size. For this case, the Mahalanobis distance MHD is a good choice.

2.6.4 Guidelines for Selecting Evaluation Metrics

As has been stated in Section 2.1, different metrics have sensitivities to different properties of the segmentations, and thus they can discover different types of error.

Now, we provide guidelines for choosing a suitable metric based on the results so far. These guidelines are additionally summarized in Table 2.3 in form of matching between data properties/requirements and metric properties:

- i When the objective is to evaluate the general alignment of the segments, especially when the segments are small (the overlap is likely small or zero), it is recommended to use distance based metrics rather than overlap based metrics. The volumetric similarity (VS) is not suitable in this case.

- ii Distance based metrics are recommended when the contour of the segmentation, i.e. the accuracy at the boundary, is of importance [FC05]. This follows from being the only category of metrics that takes into consideration the spatial position of false negatives and false positives.
- iii The Hausdorff distance is sensitive to outliers and thus not recommended to be used when outliers are likely. However, methods for handling the outliers, such as the quantile method [HKR93], could solve the problem, otherwise the average distance (*AVG*) and the overlap based metrics as well as probabilistic based metrics are known to be stable against outliers.
- iv Probabilistic distance (*PBD*) and overlap based metrics are recommended when the alignment of the segments is of interest rather than the overall segmentation accuracy [ZWKW04].
- v Metrics considering the true negatives in their definitions have sensitivity to segment size. They reward segmentations with small segments and penalize those with large segments [ULZ⁺06]. Therefore, they tend to generally penalize algorithms that aim to maximize recall and reward algorithms that aim to maximize precision. Such metrics should be avoided when the objective is to reward recall.
- vi When the segmentations have a high class imbalance, e.g. segmentations with small segments, it is recommended to use metrics with chance adjustment, e.g. the Kappa measure (*KAP*) and the adjusted Rand index (*ARI*) [HA85] [FWM⁺09].
- vii When the segments are not solid, but rather have low densities, then all metrics that are based on volume or on the four cardinalities (TP, TN, FP, FN) are not recommended. In such cases distance-based metrics, especially *MHD* and *HD*, are recommended.
- viii Volumetric similarity is not recommended when the quality of the segmentations being evaluated is low in general, because the segments are likely to have low overlap with their corresponding segments in the ground truth. In this case, overlap-based and distance-based metrics are recommended.
- ix When the segmented volume is of importance, volumetric similarity and overlap based metrics are recommended rather than distance based-metrics.
- x When more than one objective is to be considered, which are in conflict, then it is recommended to combine more than one metric, so that each of the objectives is considered by one of the metrics. Thereby, it is recommended to avoid selecting metrics that are strongly correlated (Figure 2.1).

Table 2.3: Summary of metric selection guidelines. Each row corresponds to either a segmentation property or a requirement and each column corresponds to one of the metrics in Table 2.1. A checked cell (\checkmark) denotes that the metric is recommended for the corresponding property/requirement, a crossed cell (X) denotes that the metric is not recommended, and empty cells denote neutrality.

	DICE	JAC	TPR	TNR	FPR	FNR	FMS	VS	GCE	RI	ARI	MI	VOI	ICC	PBD	KAP	AUC	HD	AVD	MHD	
Outliers exist	\checkmark	\checkmark					\checkmark	\checkmark				\checkmark	\checkmark			\checkmark	\checkmark	X	\checkmark	\checkmark	
Small segment	X	X	X	X	X	X	X			X	X	X	X			X	X	\checkmark	\checkmark	\checkmark	
Complex boundary								X										\checkmark	\checkmark	X	
Low densities	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	\checkmark	\checkmark	\checkmark	
Low segmentation quality								X										\checkmark	\checkmark	\checkmark	
Contour is important								X										\checkmark	\checkmark	X	
Alignment is important								X											\checkmark		
Recall is important			\checkmark									\checkmark									
Volume is important								\checkmark													
General shape & alignment	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	\checkmark

2.7 Metric Bias Inference

In Section 2.3, we analyzed a set of metrics, identified the properties, sensitivities, and biases of each metric. Based on this analysis, we defined guidelines for selecting metrics from this set for segmentation evaluation settings.

In this section, we present a formal method for inferring the bias (sensitivity) of a particular evaluation metric to a particular property from a set of properties the underlying objects can have. Furthermore, based on bias inference, we provide a method for selecting the most suitable evaluation metric, given a data set and an evaluation task. In [THJ14b], we have proposed a formal method for selecting metrics for evaluating 3D medical image segmentation based on measuring metric bias. We now generalize this method to be used with any evaluation process, in which objects are compared with their corresponding ground truth objects.

Section 2.5 provides an analysis of 20 evaluation metrics for 3D medical image segmentation. It discusses the properties of segmentations and the biases of evaluation

metrics. It relates these properties and biases to requirements put on segmentation algorithms to provide a protocol (guidelines) for metric selection. The method proposed in this section also considers the properties of the objects being evaluated to infer metric biases and it uses these biases for metric selection. Here, we want to highlight the differences and the relations between the metric analysis in Section 2.5 and the bias inference method presented in this section. Section 2.5 aims to study particular metrics. It analyzes each individual metric from a theoretical point of view considering its definition and the nature how it works. It provides empirical experiments, examples and comments from the literature to define the biases of each individual metric. Once this is done, it relates these biases to segmentation properties and the requirements put on the segmentation algorithms to define selection guidelines. On the contrary, the method proposed in this section does not consider any theoretical knowledge of the metric or the metric biases, does not consider the definitions of the metrics, and even does not assume any particular metrics, because the bias is systematically inferred, given a set of properties defined on the objects being evaluated. Furthermore, the proposed method is not domain specific, but general and applicable to any evaluation task that is based on comparing objects with corresponding ground truth.

Evaluation process: We revisit (repeat) the definition of the evaluation process defined in Section 1.6.2, which will be considered in this chapter.

Definition 6. *Evaluation in the sense of this work: Let $O = \{o_1, \dots, o_n\}$ be a set of objects being evaluated. Let $\hat{O} = \{\hat{o}_1, \hat{o}_2, \dots\}$ be the set of ground truth objects. The evaluation is performed by comparing each object $o_i \in O$ against its corresponding ground truth object $\hat{o}_j \in \hat{O}$. Note that $|O|$ is not necessarily equal to $|\hat{O}|$ because a ground truth object may correspond to more or less than one object.*

Object Properties: Metrics can be biased to properties of the objects being evaluated. The methods proposed in this section are based on measuring the bias of metrics to the properties of the objects being evaluated. Examples of metric bias to object properties can be found in the analysis of 20 evaluation metrics for 3D medical images presented in Section 2.5. The following is the definition and discussion of the term object property (from now property).

Definition 11. *Object properties and property values: Let $O = \{o_1, \dots, o_n\}$ be a set of objects in an evaluation process according to Definition 6. We define $F = \{f_1, \dots, f_r\}$ to be a set of object properties (from now properties) that the objects O can have. These can be any properties thought to impact metrics e.g. size, class imbalance, number of classes, noise, deviation; for images, they can be shape signatures, sphericity, boundary smoothness, resolution, moments, etc. Furthermore, properties can also be metric-dependent e.g. precision and recall. The association between an object o and a property f is represented by a value that gives how strongly f exists in o . We will denote this by property values, i.e. property value f means the value of property f .*

In general, any properties of the underlying objects can be included in the process of metric selection. They can also be restricted to those properties known to be the only relevant ones for the evaluation task. Weighting is also another possibility to customize (personalize) the influence of particular properties on the metric selection. Weighting is described in Section 2.7.4.

There are in general two types of properties that can be involved:

- Properties exclusively related to the objects being evaluated, these vary strongly depending on the domain, for example in evaluating text retrieval systems, they can be document length, average term frequency, length of the corresponding query, etc. In evaluating classification, they could be class imbalance, class size, number of classes, noise, level of chance, and many others. In other domains, e.g. imaging, other properties can be included like shape signatures, descriptors, sphericity, smoothness, boundary complexity, resolution, moments, segment size, outliers, etc.
- Properties in relation to the ground truth, i.e. metric values. Note that metrics are used here as object properties that are in turn used to select metrics. The justification is that some metrics are sensitive to properties which are measured by metrics. We give two examples as illustration. The first example are metrics are sensitive to overlap, i.e. their average correlation with other metrics depends on the overlap between the object being evaluated and the ground truth. Here, the overlap value measured as the Dice can be included as a property to discover metric sensitivity to overlap (More details on sensitivity to overlap is in Section 2.5.2). Another example is the recall. Since there are metrics that penalize recall and others that reward recall, one can include the recall as an object property to infer sensitivities of other metrics to low/high recall (more details on rewarding recall is in Section 2.5.3).

Problem Definition: We define the problems to be solved:

Definition 12. *Bias inference:* Let O be a set of objects being evaluated according to Definition 6, and let F be a set of properties according to Definition 11. Furthermore, let M be a set of evaluation metrics. A metric $m \in M$ can be biased to a property $f \in F$, that is m tends to over/under estimate the quality of object $o \in O$, given o has the property f . This bias can differ in magnitude and in direction. In particular, the tasks are:

A Inferring the bias magnitude of a metric $m \in M$ to a property $f \in F$.

B Inferring the direction of this bias, i.e. whether metric m rewards or penalizes property f .

Definition 13. *Metric selection:* Given a set of objects O , ground truth objects \hat{O} according to Definition 6, a set of properties F according to Definition 11, and a set of metrics M , the task is to sort the metrics in M according to their suitabilities to evaluate the objects O .

The remainder of this section is organized as follows. Section 2.7.1 provides an example that illustrates the settings and the problems to be solved, which will also be used to illustrate further steps of the proposed methods in other sections. In Sections 2.7.2, we present a novel method for inferring the magnitude of metric bias to a particular property, i.e. a solution for the problem in Definition 13. In Section 2.7.3, we provide a method for measuring the direction of the bias, i.e. deciding, whether a metric penalizes or rewards a particular property. We present in Section 2.7.4 a framework for selecting evaluation metrics based on their bias, i.e. a solution for the problem in Definition 12. Finally in Section 2.7.5, we discuss and analyze the proposed methods.

2.7.1 Illustrative Example

To illustrate the problems to be solved, we give an illustrative example and link its elements to the corresponding Definitions. We will also use this example for the explanation in the next sections. In an evaluation process, the performance of information retrieval (IR) systems is evaluated by comparing their binary results with ground truth. Here binary results mean that upon test queries, the systems should classify a document collection into two classes, namely relevant and irrelevant. Each of these binary classifications is evaluated by comparing it with its corresponding ground truth classification. The set of all classifications being evaluated and their ground truth classifications correspond to the object sets O and \hat{O} in Definition 6 respectively. Five properties have been identified which are thought to have impact on the evaluation, and thus have been included as the property set corresponding to the set F in Definition 2.7. These properties are $F = \{\text{class size ratio (class imbalance), document average length, term average frequency, length of the corresponding query, level of chance (probability of classification being done by chance)}\}$. A set of seven metrics $M = \{\text{precision, recall, AUC, F-Measure, accuracy, Rand index, mutual information}\}$ contains the candidates from which metrics are to be selected. The problems to be solved corresponding to Definitions 12 and 13 are (i) to infer the bias of each metric to each property, e.g. the bias of precision to class imbalance, and (ii) to find the metric(s) most suitable to be used as evaluation metric(s) based on the average metric biases. This example will be used to illustrate further steps of the proposed methods in the next sections.

2.7.2 Measuring Bias Magnitude

Given a set of objects O , ground truth objects \hat{O} according to Definition 6, a set of properties F according to Definition 11, and a set of metrics M , the task is to measure the bias (sensitivity) of the metric $m \in M$ to the property $f \in F$. Using each of the metrics $m \in M$, we compare each object in O with its ground truth Object in \hat{O} to get a metric value. The resulting matrix of metric values is used in combination with the properties in F to infer metric bias.

The method proposed in this section mainly depends on the fact that if a metric m is sensitive to a property f , then m tends to generally over-reward or over-penalize objects having the property f . To illustrate this, consider the example in Section 2.7.1, and

the fact that mutual information is biased against high class imbalance, that is it tends to measure a lower similarity when the number of relevant documents is significantly smaller than the number of irrelevant documents.

Unfortunately, we cannot directly decide whether a metric is biased to a particular property or not, i.e. we do not know whether a particular metric value is related to the quality of the object or to an under/over-estimation due to a metric sensitivity to a particular property, provided that the object has this property. Therefore, our method measures the bias of a metric $m \in M$ to a property f indirectly by analyzing the average correlation between rankings produced by the metric m and rankings produced by the other metrics under the impact of the property f , which is described in more detail in the next paragraphs.

Recall the example again and consider that the mutual information (MI) is biased against higher class imbalance. If we group the classifications being evaluated into different subsets, such that classifications with equal class size ratio (CSR) are put in the same subset, then the average values of MI in each subset would be obviously dependent on the CSR in the subsets, i.e. the MI averaged over all classifications in a subset would be larger for subsets with higher CSR . The subsets can then be ranked based on this average MI to get the subset ranking of the metric MI under the impact of the property CSR . The same can be done for each metric $m \in M$ to get a subset ranking of metric m under the impact of the property CSR . If we now calculate the average correlation between the ranking of M and the rankings of all other metrics, we obtain the average correlation of MI under the impact of the property CSR . We call this the biased correlation of MI under the impact of the property CSR .

Now let us think about another case, namely when we group the classifications randomly. Here, the property CSR has no impact, because the subsets contain classifications with different $CSRs$. If we analogously average the MI in each subset and rank the subsets according these averages, we obtain the subset ranking of metric MI for random grouping. Doing the same for all other metrics and then calculating the average correlation between the ranking of MI and all other metrics, we obtain the average correlation of MI without an impact of any property, which we call the base correlation of the metric M .

To infer the bias of the MI to the property CSR , we consider the change in average correlation of MI between the two cases, i.e. the case of random grouping and the case of grouping according to CSR , that is the difference between biased correlation and base correlation.

We have illustrated the core idea of the proposed method by means of an example showing how to infer the bias of only one metric (mutual information) to only one property (class imbalance). The formal algorithm for inferring the bias of an arbitrary metric to an arbitrary property is presented below.

This algorithm is then applied to find the bias of each metric to each property, to obtain a bias matrix, which can be then used to (i) calculate the average bias of each metric, (ii) calculate the weighted bias of each metric, and (iii) apply metric selection based on average/weighted bias.

Algorithm - Bias Inference

1. For each metric $i \in M$, compare each of the objects $j \in O$ with its ground truth object to get the matrix s where $s(i, j)$ is the metric value of object j measured by metric i .
2. For each metric $i \in M$, rank the objects in O depending on their metric values s to get the rank matrix r where $r(i, j)$ is the rank of object j based on the metric i .
3. Construct a random partition P' on O by randomly assigning the objects to q equal subsets. Choose q such that there are statistically enough objects in each subset (a good selection is $q \approx \sqrt{n}$, where n is the number of objects). Each partition should have the same number of subsets q . The function $P'_k(j)$ that assigns an object j to the subset k is defined by

$$P'_k(j) = \begin{cases} 1 & j \in \text{subset } k \\ 0 & \text{otherwise} \end{cases} \quad (2.5)$$

Now for each metric $i \in M$, calculate for each subset k in P' a rank average $S'_i(k)$ of the individual ranks of objects in the subset.

$$S'_i(k) = \left(\sum_{P'_k(j)=1} r(i, j) \right) / n_k \quad (2.6)$$

Rank the subsets based on their rank averages S' to get $R' = R'(k, i)$ that gives the rank of the subset k according to metric i .

4. Construct a partition P^f of q equal subsets of O according to the property f , i.e. according to the property value f in the objects. The function $P^f_k(j)$ assigns the object j to the subset k .

$$P^f_k(j) = \begin{cases} 1 & j \in \text{subset } k \\ 0 & \text{otherwise} \end{cases} \quad (2.7)$$

Note that the q subsets are sorted according to the values of property f , i.e. the average value of property f in the subset $k + 1$ is larger than in the subset k .

Now analogously to the random partition, calculate for each subset k in P^f the rank averages $S^f_i(k)$ given by

$$S^f_i(k) = \left(\sum_{P^f_k(j)=1} r(i, j) \right) / n_k \quad (2.8)$$

where n_k is the number of segmentations in the subset k .

Rank the subsets based on their rank averages S^f to get $R^f = R^f(k, i)$ that gives the rank of the subset k according to metric i .

5. **Base correlation:** To calculate the base correlation of metric m , consider the random partition and calculate the average correlation between the ranking according to metric m and the rankings according to all other metrics. The base correlation $C'(m)$ is given by

$$C'(m) = \frac{1}{|M|} \sum_{i \in M} \text{corr}[R'(\cdot, m), R'(\cdot, i)] \quad (2.9)$$

where $\text{corr}(x1, x2)$ is the correlation (e.g. the Pearson's correlation coefficient) between the rankings $x1$ and $x2$, and $R'(\cdot, i)$ means all ranks (the ranking) according to metric i .

6. **Biased correlation:** To calculate the biased correlation of metric m to property f , consider the partition according to property f and calculate the average correlation between the ranking according to metric m and the rankings according to all other metrics. The biased correlation $C^f(m)$ is given by

$$C^f(m) = \frac{1}{|M|} \sum_{i \in M} \text{corr}[R^f(\cdot, m), R^f(\cdot, i)] \quad (2.10)$$

where $\text{corr}(x1, x2)$ is the correlation (e.g. the Pearson's correlation coefficient) between the rankings $x1$ and $x2$, and $R(\cdot, i)$ means all ranks (the ranking) according to metric i .

7. **Bias:** Finally, to calculate the bias of the metric m to the property f , calculate the difference between the base correlation and the biased correlation of the metric m , i.e. the bias $B(m)$ is given by

$$B^f(m) = C^f(m) - C'(m) \quad (2.11)$$

Note that ranking the subsets using the averages of the individual ranks in each subset is a ranking method inspired by the Mann-Whitney-Wilcoxon (MWW) test [MW47]. This is because straightforwardly computing the ranks directly from metric averages is sensitive to outliers and may produce incorrect rankings if the metric values are not normally distributed, because large metric values can compensate small ones [Dem06].

The algorithm above infers the magnitude of the bias of the metric m to a specific object property f . This inference is based on the change of the average correlation between the metric m and the other metrics as a result of grouping the objects according to the property f . However, the bias magnitude does not give any information on the type (direction) of bias, i.e. whether it is reward or penalization of the property f , which is addressed in the next section.

2.7.3 Measuring Bias Direction

In this section, we introduce a method for finding the sign (direction) of the bias calculated according to the algorithm in Section 2.7.2, i.e. for differentiating between rewarding bias and penalization bias.

The method is based on comparing the sum of subset ranks in the first $q/2$ subsets, i.e. $\sum_{k=1}^{q/2} R^f(k, m)$ with the sum of the subset ranks in the remaining subsets, i.e. $\sum_{k=q/2+1}^q R^f(k, m)$, that is the $D^f(m)$ is direction of bias of a metric m to a property f and given by

$$D^f(m) = \begin{cases} +1 & \text{for } \sum_{k=1}^{q/2} R^f(k, m) > \sum_{k=q/2+1}^q R^f(k, m) \\ -1 & \text{otherwise} \end{cases} \quad (2.12)$$

where $+1$ means that m rewards f and -1 means that m penalizes f .

Note that the subsets are sorted according to the property value f (see step 4 in the algorithm in Section 2.7.2). Equation 2.12 can be explained as follows: If the first half of the subsets (those having in general lower values of property f) have smaller rank sum than the remaining subsets (those having in general larger values of property f), this means that the metric m rewards the property f (plus sign), and vice versa, if the first half of the subsets has larger rank sum, this means that the metric m penalizes the property f (minus sign).

2.7.4 Metric Selection

In this section, we present a formal method for metric selection, i.e. measuring the suitability of a particular metric for evaluating a particular set of objects, based on the metric bias introduced in Section 2.7.2. Formally, given an evaluation process according to Definition 6, and a set of properties F according to Definition 11, the method proposed in this section provides a solution for the problem described in Definition 13, namely it sorts a set of metrics M according to their suitabilities to be used for the evaluation process.

The assumption behind this method is that the decision on the suitability of a metric depends on the biases of this metric regarding the properties of the underlying objects being evaluated. However, the relation between bias and suitability is not straightforward. We cannot generalize that metrics with high bias are not suitable or vice versa. In some cases, bias to a particular property is required; in other cases, bias to the same properties may be not preferred. The evaluation goal can have impact on how biases are to be considered in metric selection. In general, we differentiate between two cases:

- I There are no particular properties to be emphasized, preferred, or known to have impact related to the evaluation task. Here, it is assumed that metrics are to be selected that in general have less bias to the different properties of the objects, i.e. stable metrics that would not over-/under penalize particular properties.
- II There are properties known to be related to the evaluation goal or the nature of the object that the user intends to reward or penalize, because e.g. objects with these

properties are preferred or not preferred. For example, in evaluating a segmentation task, where it is important that the segmented region completely includes the true segment, i.e. recall is more important than precision, in this case bias that penalizes small segments or rewards recall would be preferred. In another case, where the exact boundary of an image is of importance, a metric bias penalizing complex boundaries is to be avoided. An example of properties related to the nature of the objects is outliers.

Average Bias

For Case I, the most suitable evaluation metrics are those with the least average bias over all properties in F . The idea behind this assumption is that since there are no properties known to be preferable or not preferable, the goal of the selection is to avoid metrics with sensitivities and biases as much as possible and to select metrics that are neutral and stable regarding the different properties, which can be achieved by selecting metrics with minimum average bias. The average bias of a metric m regarding the property set F is defined as follows:

Definition 14. *Average bias: given a set of objects O being evaluated according to Definition 6, a set of object properties F according to Definition 11, and a set of evaluation metrics M , then the overall bias $\bar{B}(m)$ of the metric $m \in M$ regarding the properties F is given by*

$$\bar{B}(m) = \frac{1}{|F|} \sum_{f \in F} B^f(m) = \frac{1}{|F|} \sum_{f \in F} \text{abs} \left(C^f(m) - C'(m) \right) \quad (2.13)$$

Weighted Average Bias

For Case II, it is required that the selection takes into consideration the evaluation goal, represented by some properties being more or less important than others. To this end, we introduce a weighted bias, which extends the average bias in Equation 2.13 by customizing it to the user preference using the weight vector W , which has the length $|F|$ and maintains a weight for each property $f \in F$. These weights are values in the interval $[-1, 1]$.

Definition 15. *Weighted average bias: Given a set of metrics M , a set of objects O being evaluated according to Definition 6, a set of properties F according to Definition 11, where some of the properties are known to be more relevant for the evaluation than others, then the weighted bias $B^w(m)$ of the metric $m \in M$ regarding the properties F is given by*

$$B^w(m) = \frac{1}{|F|} \sum_{f \in F} \left(1 - W^f D^f(m) \right) B^f(m) \quad (2.14)$$

where $B^f(m)$ is the magnitude of bias of the metric m to the property f , $D^f(m)$ is the direction of this bias, and W^f is the weight of the property f that is utilized as follows:

- $W^f = 0$ should be used in order that $B^f(m)$ is considered as is, i.e. as absolute value of the bias regardless of the bias direction. This is the case if there is special importance of property f . Setting zero weights for all properties results in Equation 2.14 reducing to Equation 2.13, i.e. average bias, which means that Equation 2.14 is the general form.
- $0 < W^f \leq 1$ should be used in order that metrics rewarding the property f are preferred. The value represents the preference, the higher the weight, the more preference is, i.e. one means strong preference.
- $-1 \leq W^f < 0$ should be used in order that metrics penalizing the property f are preferred. The value represents the preference, the higher the weight, the more preference is, i.e. one means strong preference.

To select metrics from M , Equation 2.14 is performed for each metric $m \in M$, then metrics with the lowest weighted average bias are selected. Note that average bias is a special case of weighted average bias, where all the weights have zero values.

2.7.5 Discussion

In this section, we present an experiment to validate the proposed method, which is described step by step to provide a clearer explanation of the formal definitions and algorithms.

Experiment

In this section, the proposed method for bias inference and metric selection is tested with a data set of 229 automatic brain tumor segmentations (MRI 3D volumes) from the BRATS2012 challenge³. These segmentations correspond to 47 medical cases, and were automatically generated by five different segmentation algorithms participating in the BRATS challenge. Note that the original dataset consists of 300 segmentations, but only those segmentations have been considered that correspond to medical cases where at least three segmentations per medical case exist. That is because the experiment requires building a ranking per medical case, for which only one or two segmentations are not sufficient.

We describe the experiment step by step and link each step to its corresponding formal definition. Beginning from the setting defined in Definitions 6, and 11, the 229 segmentations correspond to the set of objects O . For the set of metrics M , 18 metrics were selected from metrics listed in Table 2.1; note that Jaccard and F-Measure were excluded because they produce the same ranks as the Dice. For the set of properties F , 7 properties were defined, namely segment size, noise, class imbalance, connected component count, point variance, sphericity, and recall. A technical report of the complete data of this experiment in form of tabular results of each step is available in

³MICCAI 2012 Challenge on Multimodal Brain Tumor Segmentation, www2.imm.dtu.dk/projects/BRATS2012

[THJ14a]). The following steps (referenced to the algorithm described in Section 2.7.2) were performed:

- (Step 1): Using each metric m of the 18 metrics in M , each segmentation was evaluated against its corresponding ground truth segmentation to get the 18×229 dimensional metric matrix s .
- (Step 2): For each metric, the segmentations were ranked. The ranks were calculated based on the metric matrix s to get one ranking per metric, which results in a 18×229 dimensional segmentation rank matrix r . Note that here the rankings are not regarding the medical cases, but regarding the metrics globally over all segmentations.
- (Step 3): The segmentations were grouped randomly into 10 subsets ($q = 10$) to get the random partition P' , i.e. each subset consists of $\frac{229}{10} \approx 22$ segmentations. Now, the 10 subsets were ranked based on the rank sum of the individual segmentations in each subset obtained from Equation 2.6 to get a 18×10 dimensional subset rank matrix R' .
- (Step 4): For each of the properties $f \in F$, a partition on the segmentation set, consisting of 10 subsets, was constructed according to the property value f in the segmentations to get the partition P^f . Analogously to the random partition, the 10 subsets were ranked based on the rank sum of the individual segmentations in each subset (Equation 2.8) to get the 18×10 dimensional rank matrix R^f .
- (Steps 5, 6, and 7) Now having the random subset ranking for each metric $R'(m)$, and the subset ranking for each property and each metric $R^f(m)$, the biases of the individual metrics and properties were inferred using Equations 2.9, 2.10, and 2.11 and the average bias of each metric was calculated using Equation 2.13. Finally metrics were sorted according to increasing average bias. The results are displayed in Table 2.4.

The result of the metric suitability based on metric bias inference is displayed in Table 2.4. The column “automatic” shows the average bias of each metric, where the metrics with the least average bias are the most suitable. This suitability is represented by ranks, shown in column “rank”, where metrics with lower rank are more suitable. We call this the automatic metric suitability ranking.

In the remainder of this section, we validate the automatic metric suitability ranking using an expert segmentation ranking. In other words, to know how well this method can find suitable metrics, we find out which metrics a medical expert would select for evaluating the same segmentations.

To this end, a manual ranking done by a radiology expert was used: for each of the medical cases, the five corresponding segmentations were ranked by their quality from a medical point of view. I.e. the radiology expert visually evaluates the 5 segmentations by comparing them with their corresponding ground truth and gives each of them a

rank depending on its quality from a medical point of view. Let us call these the expert rankings. Analogously, each of the 5 segmentations was given a rank based on its distance (or similarity) to the ground truth segmentation, measured by each of the 18 metrics. Let us call the resulting rankings “the metric rankings”.

Now, the average correlation between the expert rankings and the metric rankings was computed for each metric and finally the metrics were sorted according to their average correlation. The resulting metric ranking (Table 2.4 column “expert”) was used as a ground truth suitability ranking to validate the automatic suitability ranking.

The Table 2.4 column “automatic” shows for each metric the average bias (Equation 2.13) and the corresponding suitability rank computed based on the average bias. The column “expert” shows for each metric the average correlation with the expert ranking, and the suitability rank based on it. A moderate to strong correlation between the two rankings can be observed. The six best metrics are the same in both rankings. This correlation shows that metrics with low bias produce rankings that are more correlated to expert rankings than others.

Note that the weighted average bias, described in Definition 15 have not been used in this experiment. Further investigation regarding weighted bias is suggested as future work.

2.8 Summary

We investigated metric properties, sensitivities, and metric bias. We related the findings to metric suitability for evaluation tasks and metric selection. This work was done in two parts.

In the first part, we investigated 20 evaluation metrics for 3D medical segmentations, which have been identified based on a literature review. For all these metrics, we provided definitions for binary and fuzzy segmentations. Furthermore provided a comprehensive analysis of these metrics, discussing their properties, sensitivities and biases, which we concluded by guidelines for selecting metrics for evaluating medical image segmentations. We have presented these findings in [TH15b].

In the second part, we generalized the concept of selecting metrics based on metric bias. Note that the first part also deals with metric selection based on metric bias, but with a difference, namely it is based on analysis, experiment and theoretical explanation of the properties of particular metrics for a specific domain, namely the 3D medical segmentation. On the contrary, the method proposed in this part aims to infer metric bias systematically using a formal general framework for any metrics and any evaluation task that is based on comparing objects with their corresponding ground truth. The method has been demonstrated only on medical segmentation. In particular, we generalize the metric selection framework that we had proposed in [THJ14b] to arbitrary evaluation tasks, given they are based on comparing objects with ground truth. The metric selection is based on a novel formal method for inferring metric bias to a particular property of the objects being evaluated. Given a set of properties defined to be relevant for an

Table 2.4: Manual and automatic metric suitability rankings. In column “expert”, the average correlation between metric rankings and the manual rankings as well as corresponding suitability ranks according to descending correlation. In column “automatic”, the metric bias calculated automatically by the proposed method as well as the ranks according to ascending bias (detailed data and results available in [THJ14a])

metric	expert		automatic	
	correl.	rank	bias	rank
Cohen’s Kappa	0.818	1	33.5	2
Adjusted Rand Index	0.818	1	33.1	1
Interclass Correlation	0.818	1	33.5	2
Probabilistic distance	0.802	2	34.7	5
Dice	0.800	3	33.6	3
Average Distance	0.798	4	33.9	4
Accuracy	0.791	5	64.0	14
Rand Index	0.791	5	64.0	14
Variation of Inform.	0.791	6	62.0	13
Mutual Information	0.753	7	46.5	12
Mahalanobis Distance	0.701	8	37.7	7
Global Consistency Err.	0.670	9	69.8	15
Hausdorff Distance	0.663	10	35.5	6
Area u. curve (AUC)	0.647	11	42.0	8
Sensitivity	0.615	12	44.4	10
Precision	0.608	13	44.5	11
Volumetric Similarity	0.590	14	43.6	9
Specificity	0.398	15	78.6	16
Correlation between expert & automatic ranking			0.607	

evaluation task, the propose method infers the overall bias of a given metric regarding these properties, which gives a basis for a formal metric selection framework.

Computationally Intensive Metrics

3.1 Introduction

Efficiency in speed as well as in memory usage is an important issue in metric calculation, especially those metrics that attempt to calculate the distance between two point sets, thereby considering distances of all point pairs in the two point sets, e.g. the Hausdorff distance (HD) and the average distance (AVD). This complexity becomes a problem when the set size is huge. An example of huge point sets are 3D medical image segmentations, which can have up to 100 Mio voxels.

There are several aspects that should be considered in order to achieve efficient evaluation tools. One of these aspects is avoiding unnecessary multiple computation when many metrics are to be calculated that are based on common basic elements or can be reduced to common elementary factors. This is an important consideration for evaluation tools that attempt to calculate a set of related metrics. We have successfully applied this method to optimize an implementation of an evaluation tool named EvaluateSegmentation¹, which we have implemented and described in [TH15b]. In EvaluateSegmentation, 20 evaluation metrics for 3D medical image segmentation have been implemented. A synergy between 15 of these metrics has been found by reducing their definitions to common basic factors, which results in these factors having to be computed only once, which avoids unnecessary computation and thus enables high efficiency in time and memory usage. Examples of such basic elements are the basic cardinalities of the confusion matrix (true/false positives and true/false negatives). Another advantage of reduction to common factors is enabling a generic extension of metrics, e.g. from binary to fuzzy, by only extending the elementary factors, based on which the metrics are defined.

¹EvaluateSegmentation is an open source project for evaluating medical volume segmentations available for download from <http://github/codalab/EvaluateSegmentation>.

Another aspect of efficiency is the calculation of computationally intensive metrics, especially those metrics that calculate distances between all possible pairs of points, e.g. the HD and the AVD. The naive computation of such metrics takes a time that is quadratically proportional to the point set size. When the point sets are huge, the computational time is critical. Memory usage is another challenge in metric computation in combination with huge point sets. Regarding these two aspects, we present (i) a linear general algorithm for calculating the exact Hausdorff distance, as one of the most computationally intensive metrics, and (ii) an optimization method for calculating the average distance between image segmentations, which makes use of them being rigid objects (dense point sets) to achieve an efficient computation.

The Hausdorff distance (HD) is a measure of dissimilarity between two point sets. The HD is an important metric that is commonly used in many domains like image processing and pattern matching as well as evaluating the quality of clustering. For example it is common to use the Hausdorff distance in the medical domain in applications like evaluation of medical segmentations and registration. In many cases medical images, such as magnetic resonance (MRI) and computed tomography (CT) volumes are compared e.g. to evaluate the performance of registration [BM92] [CR03] and segmentation algorithms [MNLBR07] [BPA⁺08] [KCAB09].

Formal definitions: The directed Hausdorff distance \check{H} between two arbitrary point sets A and B is the maximum of distances between each point $x \in A$ to its nearest neighbor $y \in B$. That is:

$$\check{H}(A, B) = \max_{x \in A} \{ \min_{y \in B} \{ \|x, y\| \} \} \quad (3.1)$$

where $\|\cdot, \cdot\|$ is any norm e.g. the Euclidean distance function. Note that $\check{H}(A, B) \neq \check{H}(B, A)$ and thus the directed Hausdorff distance is not symmetric. The Hausdorff distance H is the maximum of the directed Hausdorff distances in both directions and thus it is symmetric. H is given by:

$$H(A, B) = \max \{ \check{H}(A, B), \check{H}(B, A) \} \quad (3.2)$$

The Average Distance (AVD), is defined as the average of minimum distances from points in the first point set to the second one and vice versa. It is defined as

$$AVD(A, B) = \frac{d(A, B) + d(B, A)}{2} \quad (3.3)$$

where $d(A, B)$ is the directed Average Hausdorff distance that is given by

$$d(A, B) = \frac{1}{N} \sum_{x \in A} \min_{y \in B} \|x - y\| \quad (3.4)$$

According to the definitions of the HD and AVD , they are based on calculating all pairwise distances between points in the two point sets, which implies that a straightforward computation takes a time that is quadratically proportional to the point set size.

Requirements on distance algorithms: Many researchers have noted the computational complexity of the HD and AVD [NJS11] [GJC01] [HDAC12]. The most important characteristics to optimize are runtime and memory required. However, evaluating the quality of an algorithm should take into consideration how these two characteristics vary in relation to the following parameters, where the terms volume, grid size, and set size are according Definition 7:

- Point set size: For example, a brain MRI volume could reach a million voxels and that of a whole body could reach 100 million voxels. The runtime of the algorithm should remain reasonable when the set size increases extremely.
- Grid size: It is desirable that the complexity of the algorithm depends only on the point set size rather than the grid size. For example, in brain tumor segmentations, the volume of the tumor is normally a small fraction of the grid size and the rest is background. The background should not be included in the computation.
- Density and sparsity: an algorithm could perform better with sparse point sets like geographical locations and worse with dense point sets like MRI segmentations and vice versa.
- Generality: algorithms restricted to a special class of point sets cannot be applied in a general situation.

Contribution: In this chapter, we propose two algorithms:

The first is for calculating the HD . This algorithm is optimized to satisfy all the requirements stated above, i.e. it remains efficient when changing any of the four parameters. It has a nearly-linear complexity and an efficient performance for extreme point set sizes as well as for extreme grid size. It outperforms the standard HD algorithm of the ITK Library, the leading platform for image processing in the medical domain. Furthermore the proposed algorithm performs equally for sparse and dense point sets, and finally it is general without restrictions on the characteristics of the point set.

The second algorithm is for computing the AVD between 2D/3D image segmentations. This algorithm satisfies only the first two requirements, i.e. it operates efficiently with image segmentations having huge segments as well as huge grid sizes. It does not satisfy the generality and sparsity requirements because it is restricted to image segmentations since it makes use of their nature being dense objects.

Chapter organisation: The remainder of the chapter is organized as follows. Related work is discussed in Section 3.2. In Section 3.3 we propose the novel algorithm for computing the Hausdorff distance and provide a runtime analysis of the algorithm. Experiments and results are presented in Section 3.4. In Section 3.5, we present optimizations for calculating the average distance. Finally, the chapter is concluded in Section 3.6.

3.2 State-of-the-Art

Several approaches have been proposed that aim to overcome the computational complexity of the Hausdorff distance. These approaches can be generally divided into two categories, namely approximation and exact calculation of the Hausdorff distance. The first category contains those methods that try to efficiently find an approximation of the Hausdorff distance. This category is especially common with runtime-critical applications, for example pattern matching under transformation (e.g. moving object detection). Because the HD algorithm proposed in this chapter aims to calculate the exact Hausdorff distance, this category of research is actually not directly in the focus of this chapter; we therefore only give some representative references for this category. Alt et al. [ABJ91] used Voronoi diagrams to efficiently approximate the HD between simple polygons. Indyk et al. [IV03] proposed an algorithm for approximating the HD for matching patterns under transformation by using the Halls Theorem to reduce the necessary geometrical matching. Hossain et al. [HDAC12] proposed a linear time algorithm for finding an approximation of the HD with lower approximation error.

Most algorithms belonging to the second category try to efficiently compute the exact Hausdorff distance for specific classes of point sets or special types of objects like polygons, line segments, or special curves. The rest of these algorithms use complex structures that require a preprocessing phase causing long computation time and high memory need. In the next subsections we highlight some work related to this category in more detail.

3.2.1 Polygons

Atallah [Ata83] provided an algorithm for computing the Hausdorff distance for a special case of point sets, namely non-intersecting, convex polygons. The algorithm has the complexity of $O(n + m)$ where m and n are the vertex counts. The algorithm is mainly based on the fact that when minimizing/maximizing distances between two non-intersecting convex polygons, then points with extreme distances are always the vertices. This implies that only distances between vertices need to be computed to find the Hausdorff distance. Although this algorithm is simple and computationally efficient, it is restricted to a special class of point sets.

3.2.2 R-Trees

Papadias et al. [PTMH05] proposed an algorithm for finding aggregate nearest neighbors (ANN) in databases. Given two databases in form of point sets A and B , the algorithm finds for a given point $a \in A$ the nearest point $b \in B$. That is $ANN(a, B) = b \in B : dist(a, b) = mindist(a, B)$. The query data points are spatially indexed to produce an R-Tree which is then used to optimize searching for the ANN. In fact, $ANN(a, B)$ can be an elementary function for computing the directed Hausdorff distance because $\vec{H}(A, B) = max_{a \in A} ANN(a, B)$. But because the algorithm deals with B as a single object, it follows that the direct use of ANN to compute Hausdorff distance means

iterating all points $a \in A$ and performing *ANN* each time. Nutanong et al. [NJS11] extended the algorithm proposed in [PTMH05] by avoiding the iteration of all points in A : the algorithm achieves this by performing the aggregate nearest neighbor simultaneously in both directions, that is to use two R-Trees at the same time, one for each point set.

However, the drawbacks of both methods above are (i) they use complex structures with additional computational effort needed for building the index and (ii) the methods assume sparse point sets that are suitable for building an efficient R-Tree. If the underlying point sets are very dense or in the worst case rigid objects (e.g. medical segmentation), algorithms based on R-Trees may not be the best choice.

3.2.3 Distance Transform Based Algorithms

A distance transform (called also distance maps) is a representation of an image in which each pixel becomes a label that reflects its distance to the boundary or background. There are various transforms depending on the distance metric used [MQR03]. A common way to efficiently compute the HD in image processing is to use the distance transform. These methods compute the Hausdorff distance in linear time, given a distance transform, but the time required for computing the distance transform is proportional to the grid size, as it also takes background into account. Furthermore, these methods are based on labeling the pixels which makes them restricted to images, and thus they are not general. The ITK Library² uses distance transforms for computing the HD, described in [TSG06]. Ciesielski et al. [CCUG11] investigated the computational complexity of the algorithm described in [TSG06]. They concluded that this distance transform algorithm is a computationally expensive but ubiquitously needed operation in image processing. We use the ITK implementation of this algorithm as a reference to compare the performance of the proposed algorithm in Section 3.4.

3.2.4 HD for Mesh Surfaces

Guthe et al. [GBK05] proposed an algorithm for calculating the Hausdorff distance between mesh surfaces. This algorithm makes use of the specific characteristics of meshes to avoid sampling all points in the compared surfaces. To achieve this, two strategies are used. In the first strategy, the algorithm aims to recognize areas in the two compared surfaces where the pairwise triangles are expected to have maximum distance between them. Only these areas are intensively sampled thereby avoiding sampling all triangles of the surfaces. This is achieved by building a grid, in particular an octree, on each of the surfaces and then calculating the min/max distances between cells. The second strategy is to avoid sampling all points in a particular triangle when calculating the min/max distances of the cells. This is achieved by measuring the distances of the triangle vertices to the other mesh surface in a first step. Then, sampling further points inside the triangle is stopped if all distances of the vertices are less than the actual (yet unknown) HD. As mentioned above, this algorithm is based on the specific characteristics of meshes and

²National Library of Medicine Insight Segmentation and Registration Toolkit (ITK) www.itk.org

thereby lacks generality. In particular, the second strategy is only applicable on surfaces consisting of triangles (meshes) and because it is used in the first strategy, this implies that also the first strategy can be only applied on meshes efficiently.

3.2.5 Optimizing the k-Nearest Neighbors Algorithm

The k-nearest neighbor (k-NN) algorithm is a basic operation in many algorithms in information retrieval, machine learning, and data mining. It is also the core operation of many metrics that aim to compare two point sets, like the Hausdorff distance (HD) and the average distance (AVD). There is a lot of literature on k-NN optimization techniques. However, we are interested in an optimization presented by Zhao et al. [ZLX⁺14], on which we will build in Section 3.5 to optimize the calculation of the AVD . Zhao et al. proposed an optimized algorithm for finding the nearest neighbor (NN) based on a 3D uniform cell grid. In particular, this algorithm aims to find a convenient subspace of the grid that contains the NN by finding a hypersphere of a radius as small as possible containing the NN, such that unnecessary calculations are avoided. Using this technique, an efficient calculation of the NN is achieved. However, this optimization alone is not sufficient for calculating the AVD between huge 3D medical segmentations. Therefore, we propose two substantial modifications for this algorithm in Section 3.5 that result in an efficient AVD calculation algorithm that is sufficient for computing the AVD between huge 3D medical segmentations.

3.3 Calculating the Hausdorff Distance

In this section we propose a novel algorithm for calculating the exact Hausdorff distance. Before starting with the new algorithm, we will define some notations that will hold through the rest of the chapter. We also present the straightforward algorithm for calculating the Hausdorff distance in Algorithm 3.1 to ease explanation. Let $A = \{x_1, x_2, \dots, x_m\}$ and $B = \{y_1, y_2, \dots, y_n\}$ be two point sets in \mathbb{R}^d and let $\|x, y\|$ be any norm $\mathbb{R}^d \rightarrow \mathbb{R}$ where $x, y \in \mathbb{R}^d$. In the usual case this is the Euclidean distance function. Recall equations (3.1) and (3.2) and note that the Hausdorff distance is the maximum of the two directed Hausdorff distances in both directions. Thus, from now we will only concentrate on computing the directed Hausdorff distance $\check{H}(A, B)$

Note that we will only use two dimensional point sets in illustrations for simplicity, although the proposed algorithm is applicable for point sets in \mathbb{R}^d .

Obviously Algorithm 3.1 runs in $O(m * n)$ time where $m = |A|$ and $n = |B|$ because both loops in Algorithm 3.1, Lines 2 and 4, always run through all points. From now, we will call these loops the outer loop and the inner loop respectively.

Hereafter the three parts of the proposed algorithm are presented: in the first part (Section 3.3.1), we show that a complete scan in the inner loop is not always necessary (early breaking). The second part (Section 3.3.2) presents a sampling method that can replace the trivial scanning and considerably enhance the performance. The combination of early breaking and the sampling method provides a significant efficiency increase

Algorithm 3.1: NAIVEHDD straightforwardly computes the directed Hausdorff distance

Require: Two finite point sets A, B .

Ensure: Directed Hausdorff distance

```
1:  $cm_{ax} \leftarrow 0$ ;  
2: for  $x \in A$  do  
3:    $cm_{in} \leftarrow \infty$   
4:   for  $y \in B$  do  
5:      $d \leftarrow \|x, y\|$   
6:     if  $d < cm_{in}$  then  
7:        $cm_{in} \leftarrow d$   
8:     end if  
9:   end for  
10:  if  $cm_{in} > cm_{ax}$  then  
11:     $cm_{ax} \leftarrow cm_{in}$   
12:  end if  
13: end for  
14: return  $cm_{ax}$ 
```

compared to the application of these optimizations individually. In the third part (Section 3.3.3), a refinement technique is presented that excludes the intersection of the compared sets from computation in advance in the case where the intersection is defined, e.g. when the compared sets are images or volumes, which additionally provides a small increase in the speed of the algorithm. Finally, in Section 3.3.4 we present the runtime analysis of the proposed algorithm.

3.3.1 Early Breaking

It is not always necessary that the scan in the inner loop (Algorithm 3.1, Line 4) runs completely through. Since the Hausdorff distance aims to find the maximum of the minimums, the inner loop can actually break as soon as a distance is found that is below the temporary HD (cm_{ax}), because in this case cm_{ax} will definitely not change in the rest of the loop. This means the algorithm can break the inner loop and continue with the next point of the outer loop. Through the rest of the chapter, we will call stopping the inner loop because of finding some distance $d < cm_{ax}$ the early break. We modify Algorithm 3.1 to consider the early break as illustrated in Algorithm 3.2, Line 9.

Note that the run time of Algorithm 3.2 depends on at least the following factors:

- The order in which the outer loop iterates the points in A : detecting a point with a relatively large distance to B leads to a larger value of cm_{ax} and consequently to a higher probability of the occurrence of an early break. In fact detecting the point with maximum distance to B at the beginning leads to the best case.

Algorithm 3.2: EARLYBREAK computes the directed HDD using the early break technique and random sampling

Require: Two finite point sets A, B

Ensure: Directed Hausdorff distance

```

1:  $cmax \leftarrow 0$ ;
2:  $E \leftarrow A \setminus (A \cap B)$  {described in Sec. 3.3.3}
3:  $E_r \leftarrow \text{randomize}(E)$  {Randomization described in Sec. 3.3.2}
4:  $B_r \leftarrow \text{randomize}(B)$  {Randomization described in Sec. 3.3.2}
5: for all  $x \in E_r$  do
6:    $cmin \leftarrow \infty$ 
7:   for all  $y \in B_r$  do
8:      $d \leftarrow \|x, y\|$ 
9:     if  $d < cmax$  then {Early break described in Sec. 3.3.1}
10:       $cmin \leftarrow 0$ 
11:      break
12:     end if
13:     if  $d < cmin$  then
14:        $cmin \leftarrow d$ 
15:     end if
16:   end for
17:   if  $cmin > cmax$  then
18:      $cmax \leftarrow cmin$ ;
19:   end if
20: end for
21: return  $cmax$ 

```

- The order in which the inner loop performs the scan in B : Here it is advantageous to pick points with smaller distances, because a distance below $cmax$ leads to an early break. Figure 3.1 illustrates the relation between the iteration order and the occurrence of the early break.

3.3.2 Random Sampling in Place of Scanning

Now the question is how much is the improvement from using the early break alone? According to object coherence [GP93] based on the principle of spatial locality, in some classes of point sets, like images and volumes, the points are likely to be spatially distributed in a way that points iterated successively (e.g. line-wise or column-wise in an image) in the first set have similar distances to some reference point in the second set, which means that the early break could likely be delayed more than necessary. In other words, if no early break occurs, it is likely that it will not occur when a nearby point is tried. It is better in this case to continue the search in another region which is spatially far from the current point. In this section we describe how to use random

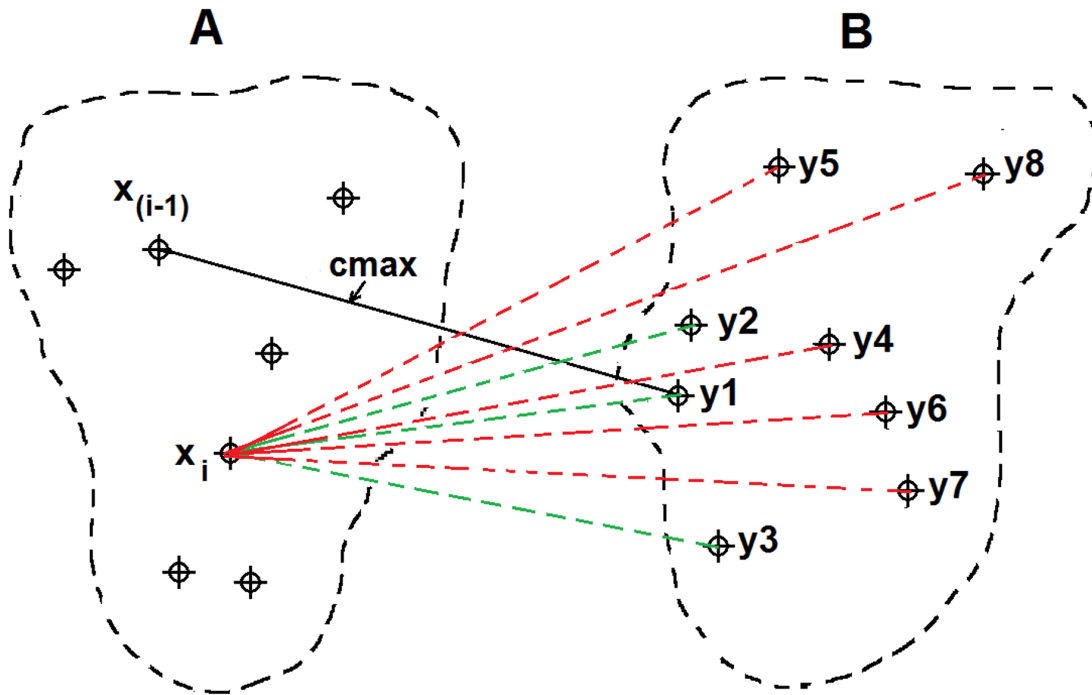


Figure 3.1: $x_{(i-1)}$ is the point already minimized in the previous iteration where $cmax$ was found to be the current maximum (temporary HD). Point x_i is being currently minimized by calculating its distances to B . Points $y_1..y_8 \in B$ are numbered according to their distance to x_i . An iteration order beginning with y_1 , y_2 or y_3 is good because it will cause an immediate break whereas an iteration order beginning with other points is worse because the scan will continue.

sampling instead of the trivial scanning to improve performance. This method leads to an algorithm with nearly-linear runtime as will be shown in Section 3.3.4

In random sampling, the aim is to avoid similar distances in successive iterations. This is achieved by randomly iterating the points in the inner loop. However, we randomize the sampling order also in the outer loop. We found that randomizing the sampling order additionally in the outer loop makes the runtime more efficient in some special cases. E.g. when the two point sets form generally linear shapes and one of them is nearly on the extension line of the other. The randomization additionally in the outer loop reduces the probability of worst cases with such point sets (this is e.g. frequent when the compared point sets are trajectories). For other cases, it is enough to only randomize the inner loop. But because the randomization doesn't need much computational effort and because there are no cases where it has a negative effect on the efficiency, we always randomize both of the loops. To achieve this, we prepare a list B_r with all points $y \in B$ randomly ordered and we use this set for iteration in the inner loop instead of set B . The same is done with iterating in the outer loop, more specifically the set E is also randomized

to get the set E_r . Algorithm 3.2, Line 3 and Line 4 show the additional randomization steps.

Note that preparing the random set in advance is necessary, because picking random candidates in the loop cannot ensure iterating through all the points. Generating the random order is possible in linear runtime by swapping each point in the set with a randomly selected point from the same set. Algorithm 3.3 illustrates this linear randomization.

The random scan eliminates the effect of the spatial locality in the point set and provides a significant improvement as shown in Sections 3.3.4 and 3.4.

Algorithm 3.3: RANDOMIZE finds a random order of a given point set

Require: A finite point set S

Ensure: Random order of S

```

 $S_r \leftarrow S;$ 
for all  $p_1 \in S_r$  do
   $p_2 \leftarrow \text{randompoint}(S_r)$ 
   $\text{swap}(p_1, p_2)$ 
end for
return  $S_r$ 

```

3.3.3 Excluding Intersection

In this section we describe a refinement that is applicable when the compared point sets are not disjoint but rather have a computable intersection. This is the case when the point sets are defined on a grid. For instance in the evaluation of medical volume segmentations, the compared images (test image and ground truth image) mostly have a large portion of voxels in common. We describe a technique that improves the performance of calculating the Hausdorff distance by excluding the intersection from the computation. This optimization generally provides a small increase in speed beyond the combination of early breaking and randomization. It is not a core part of the general HD algorithm, and can be used to achieve a small speed increase in cases where it is applicable. Let $S = A \cap B$ be the intersection between the compared point sets, then it is easy to conclude that $\check{H}(A, B) = \check{H}(A \setminus S, B)$. This follows from the fact that when iterating points $x \in A$ in the outer loop, $\forall x_i \in S \exists s_1 = x_i \in A, s_2 = x_i \in B : \|s_1, s_2\| = 0$, it follows that $\text{dist}(x_i, B) = 0$ which means that cmax doesn't change in the corresponding iteration. In other words, for each of the intersection points, a direct early break is guaranteed and therefore it is not necessary to include them in the outer loop and they can be excluded from A in advance. Note that the intersection points must be however included in the inner loop because they could be at minimum distance to some point $y \in B, y \notin S$. Algorithm 3.2, Line 2 shows the additional step needed to implement this improvement.

3.3.4 Runtime Analysis

Algorithm 3.2 has a runtime of $O(m)$ in its best case and a run time of $O(m * n)$ in its worst case (Note that we do not consider the complexity of measuring the distance between two points, which is dependent on the dimensionality of the points, because optimizing this measurement is not in the scope and thus assumed to be constant). The best case is when an early break occurs directly at the beginning of each iteration in the inner loop, that is when we always select a point with a distance below cm_{ax} . On the other hand, the worst case occurs when a full scan runs through completely in each iteration. The more important question is about the runtime of the average case.

Informally, it is expected that the average case runtime is biased towards the best case because the worst case generally requires conditions that are more difficult to satisfy. While a definite iteration order in the inner and the outer loops is required for the worst case so that the early break is prevented in each iteration, the best case requires only one condition, namely picking a point with a distance below cm_{ax} in each first iteration in the inner loop.

Now let us see the average case runtime in a more formal way. We consider the randomly picked point $y \in B$ in Algorithm 3.2 in the inner loop and define the random variable D to be the distance d measured between the point y and the current reference point $x \in A$. We also define the event e to be that distance d is larger than cm_{ax} that is $e \equiv d > cm_{ax}$. Note that event e means the non-appearance of an early break. Let us assume that event e always occurs with the probability q that is $P(e) = q$. Obviously the event \bar{e} occurs with probability $p = 1 - q$ and denotes picking a distance $d \leq cm_{ax}$.

We also define the random variable R to be the number of successive distances exceeding cm_{ax} followed by one distance below cm_{ax} i.e. the length of a sequence of successive events e followed by an event \bar{e} . For any iteration i , this is equivalent to $i - 1$ distances from the reference point x to the points y_1, y_2, \dots, y_{i-1} namely $d_1, d_2, \dots, d_{i-1} > cm_{ax}$ and one distance $d_i \leq cm_{ax}$. The probability density function of R is given by

$$\begin{aligned} f(x) &= P(d_1 > cm_{ax}, \dots, d_{x-1} > cm_{ax}, d_x \leq cm_{ax}) \\ &= q * \dots * q * p \\ &= q^{x-1}p \end{aligned} \tag{3.5}$$

which is a geometrical probability distribution. Figure 3.2 shows the probability distribution $f(x)$. Note the strong steepness of $f(x)$ that make longer runs of event e unlikely and intuitively explains the bias of the average case runtime towards the best case. To formally find the average case runtime, the expected value $E[R]$ of $f(x)$ should be found which is equivalent to the expected number of iterations until an early break.

$$E[R] = \sum_{x=1}^{\infty} x f(x) = \sum_{x=1}^{\infty} x q^{x-1} p \tag{3.6}$$

$E[R]$ is a geometrical series with $0 \leq p \leq 1$ that converges and has a sum. From Equation 3.6 it follows

$$E[R] = p + 2.q.p + 3.q^2.p + 4.q^3.p + \dots \tag{3.7}$$

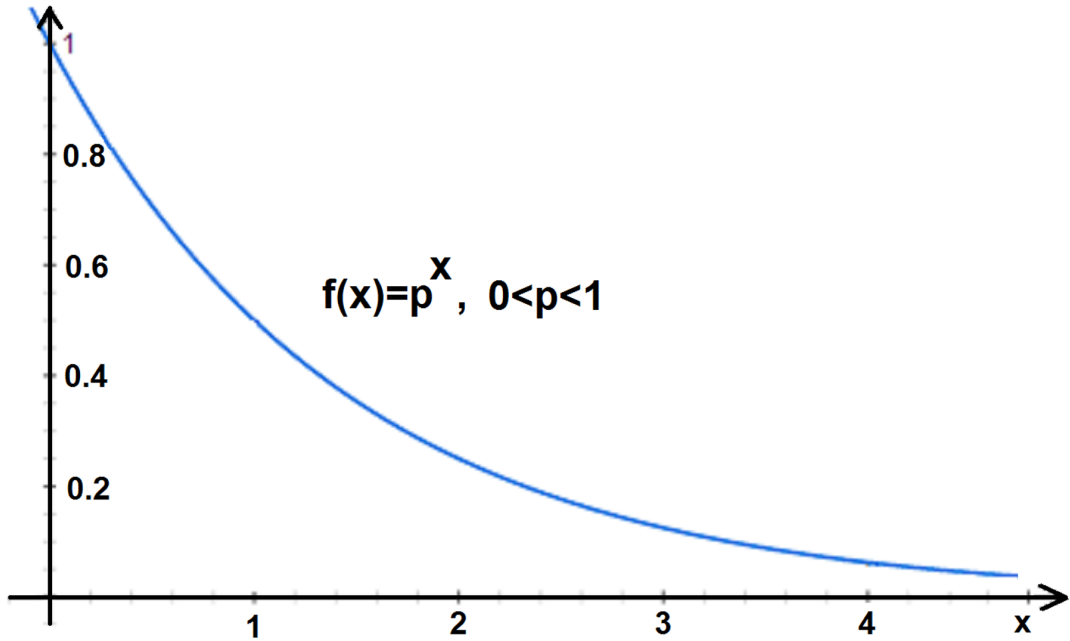


Figure 3.2: The probability density function of a geometrical distribution.

By multiplying both sides with q , subtracting the resulting equation from Equation 3.7, and then dividing by p

$$\frac{E[R](1 - q)}{p} = 1 + q + q^2 + q^3 + \dots \quad (3.8)$$

By multiplying both sides with q , subtracting the resulting equation from Equation 3.8, and then substituting $q = 1 - p$

$$E[R] = \frac{1}{p} \quad (3.9)$$

Equation 3.9 tells the important fact that the number of tries until an early break depends only on p which denotes the probability of picking a point with distance below cm_{ax} . The higher p is, the more likely that the inner loop terminates after a lower number of tries and vice versa.

But how high is p actually and what does it depend on? In fact, p depends mainly on how large cm_{ax} is and cm_{ax} is limited by the HD because ($cm_{ax} \leq h$) which means the HD determines how large cm_{ax} at most can be. If the Hausdorff distance is large, cm_{ax} can take larger values and thus it is more likely that a randomly selected point is below cm_{ax} which means a higher value of p and vice versa, that is $p \propto h$. Figure 3.3 illustrates the relation between cm_{ax} , the probability p , the Hausdorff distance, and the

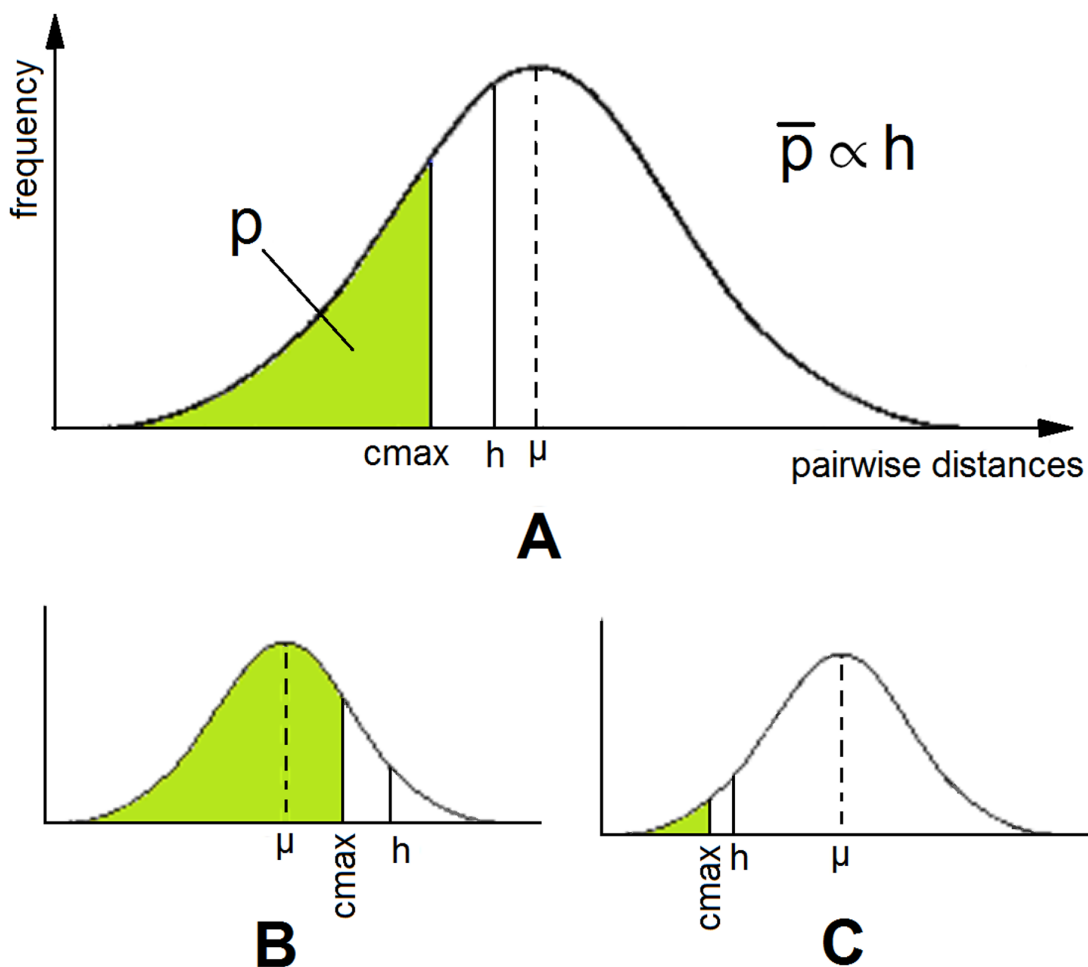


Figure 3.3: Distribution of pairwise distances assuming a normal distribution for illustration. (A) Position of the Hausdorff distance h relative to the distribution affects p because $cmax \leq h$. (B) h is large and $cmax$ can reach large values thereby increasing p . (C) h is small and $cmax$ remains small thereby decreasing p .

distribution of the pairwise distances d_{ij} where $d_{ij} = \|x_i, y_j\| : \forall x_i \in A, y_j \in B$. Here, a normal distribution is just for illustration and the relation holds for any distribution. The example is to show how p does not directly depend on the size of set B , but rather on the Hausdorff distance h and the distribution of the pairwise distances.

Note that we don't have to determine the distribution of the pairwise distances to conclude that the runtime depends only on h , this is because the distribution only determines the value of p , which is irrelevant for whether the runtime is dependent on B or not because the expected value $\frac{1}{p}$ is a constant value for any $p > 0$.

Formally, the average probability that the randomly picked distance $d \leq cmax$ is

given by

$$\bar{p} = \text{average} \left(\int_{x=0}^{cmax} f(x) dx \right) = c \int_{x=0}^h f(x) dx \quad (3.10)$$

where f is the probability density function that represents the distribution of the pairwise distances and c is a constant that results by estimating $cmax$ in terms of h ; the justification of this estimation is in the next subsection.

3.3.5 Convergence of the Temporary HD ($cmax$)

The value of $cmax$ geometrically increases during the progress of the outer loop (Algorithm 3.2) so that it already reaches values near h after a very small number of iterations compared with the total count of iterations, as demonstrated in Figure 3.4. We explain this geometrical increase as follows: At the beginning of the outer loop $cmax$ is zero, then it increases monotonically with the progress of the outer loop until it reaches the Hausdorff distance h . In each iteration, there are two possibilities, either the distance to the current point is smaller than $cmax$, here no $cmax$ update is performed or the distance is larger than $cmax$, in this case $cmax$ is updated to have the distance value. Let us observe only those iterations where $cmax$ is updated. For any such update iteration i we define $cmax(i)$ to be the $cmax$ value in that iteration (before update) and $d(i)$ to be the distance of the randomly selected point. The possible values that $d(i)$ can have are in the interval $[cmax(i), h]$. This means $d(i)$ has an expected value of $\frac{h-cmax(i)}{2}$ which is subsequently the expected value of $cmax(i+1)$. It follows that in the next iteration $i+1$, the expected interval in which $d(i+1)$ values can be is $[h - \frac{h-cmax(i)}{2}, h]$. Analogously, for iteration $i+2$, we likely get $cmax(i+2) = \frac{h-cmax(i)}{4}$ and an interval $[h - \frac{h-cmax(i)}{4}, h]$ and so on which implies a geometrical convergence of $cmax$ to h . To experimentally verify this geometrical convergence, we computed the Hausdorff distance between 1000 pairs of trajectories generated from the road network of Oldenburg (described in Section 3.4.7). Each of the trajectories consists of 2000 points. For each iteration in the outer loop, two values were recorded, namely the number of iterations in the inner loop until the early break n and the value of $cmax$ at the beginning of each iteration in the outer loop, hence getting 2000 values for each pair of trajectories, i.e. 2 million values in total. Two statistics were computed, the first by averaging the number of iterations until the early break (n) to get (\bar{n}) at each iteration of the outer loop. This is visualized with a logarithmic scale in Figure 3.5. At first, \bar{n} is very high because $cmax$ is zero and the inner loop is scanned completely. After that, \bar{n} decreases rapidly to converge finally at very low values. This statistic confirms the convergence behavior of the number of iterations until the early break predicted theoretically. The second statistic was made by converting the recorded $cmax$ values to percentage values of the HD between the corresponding pair of trajectories; this is because the HD is different in each pair. From these percentage values, we counted how many values exceed 90% and 99% at each iteration in the outer loop. Figure 3.6 shows the results. We show only the first 200 iterations to make the plot more readable. The results confirm the quick convergence of $cmax$ to the HD. In

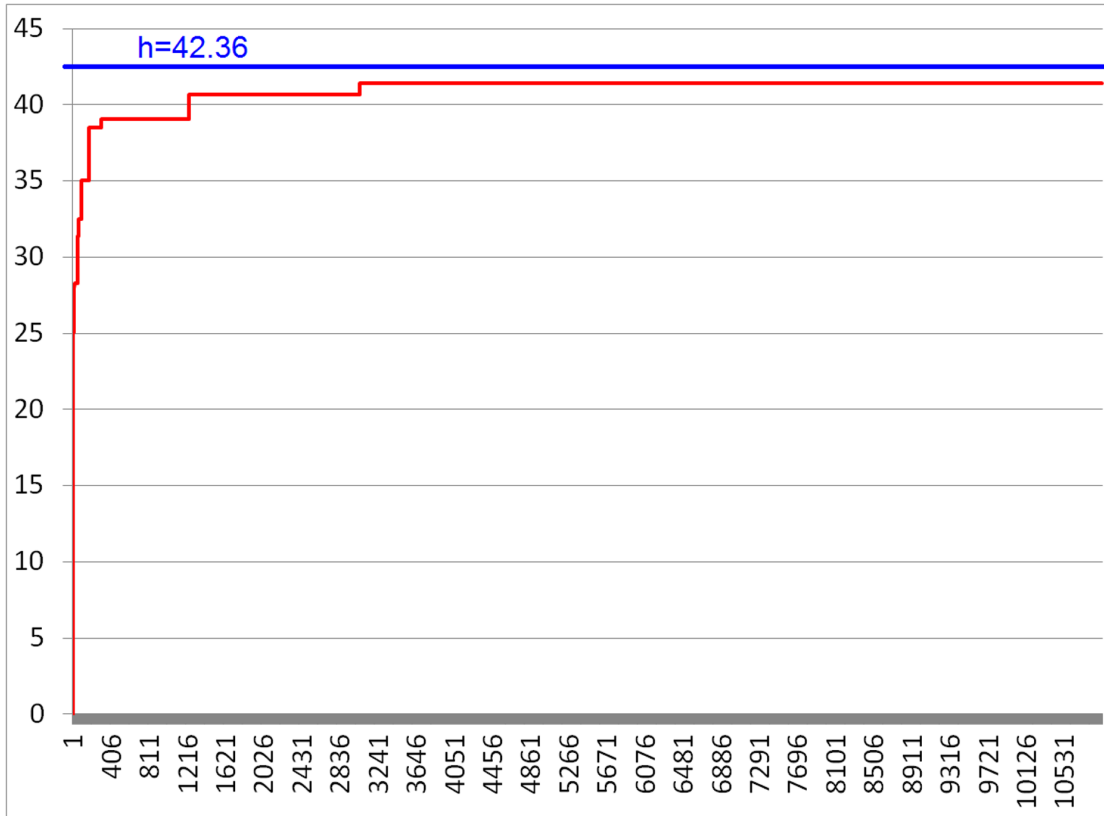


Figure 3.4: The progress of $cmax$ in the first 10 thousand iterations (outer loop) when comparing two real brain tumor segmentations. Note that only 10 thousand of 15.6 million iterations in total (this is the number of voxels in the first segmentation) are shown and thus the curve does not reach the HD.

about 80% of the cases, $cmax$ is already after 5 iterations above 90% of the HD, and already after 50 iterations above 99% of the HD.

From Equations 3.9 and 3.10, the expected number of iterations until the early break given a Hausdorff distance h is

$$E[R] = \frac{1}{\bar{p}} = \frac{1}{c \int_{x=0}^h f(x) dx} \quad (3.11)$$

Note that if h is very small, for example $h \approx 0$, the algorithm tends to get low performance. Nevertheless, low values of h mean high match between the point sets which means that it is likely that A and B have high intersection, which also means that the improvement introduced in Section 3.3.3 will compensate the loss of performance by excluding the intersection and this will keep a low overall runtime.

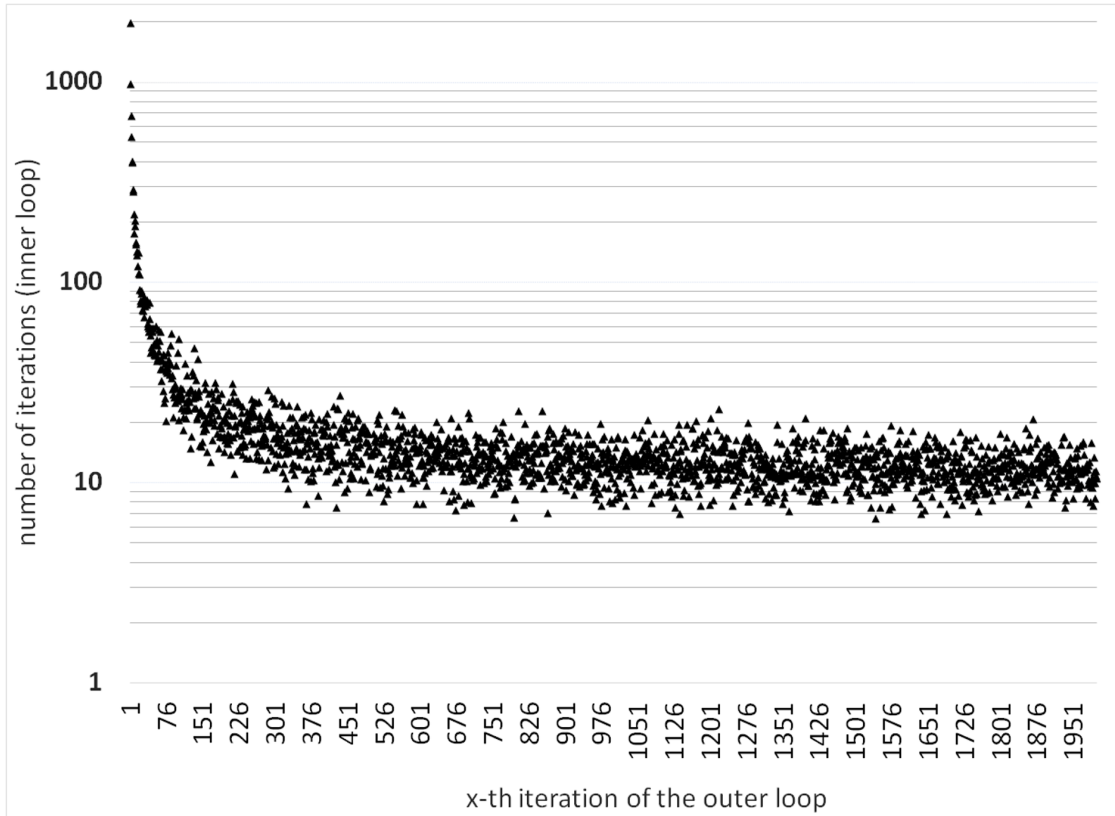


Figure 3.5: The average number of iterations in the inner loop until the early break at each iteration of the outer loop. Values are recorded from measuring the HD between 1000 pairs of trajectories generated from the road network of Oldenburg. Each trajectory contains 2000 points. Iterations of the outer loop are on the x-axis and the number of iterations in the inner loop averaged over all pairs on the y-axis.

3.3.6 Handling of Outliers

The Hausdorff distance is generally sensitive to outliers [EAN08] [HKR93]. The Hausdorff quantile is a method proposed in [HKR93] to solve the problem of outliers: according to the Hausdorff quantile method, the Hausdorff distance is defined to be the q^{th} quantile of distances instead of the maximum, so that possible outliers are excluded, where q is selected depending on the application and the nature of the measured point sets. The proposed algorithm can be easily extended to support the Hausdorff quantile by saving all distances measured and after the outer loop is finished, the distances are sorted and the q^{th} quantile is returned instead of *cmax*.

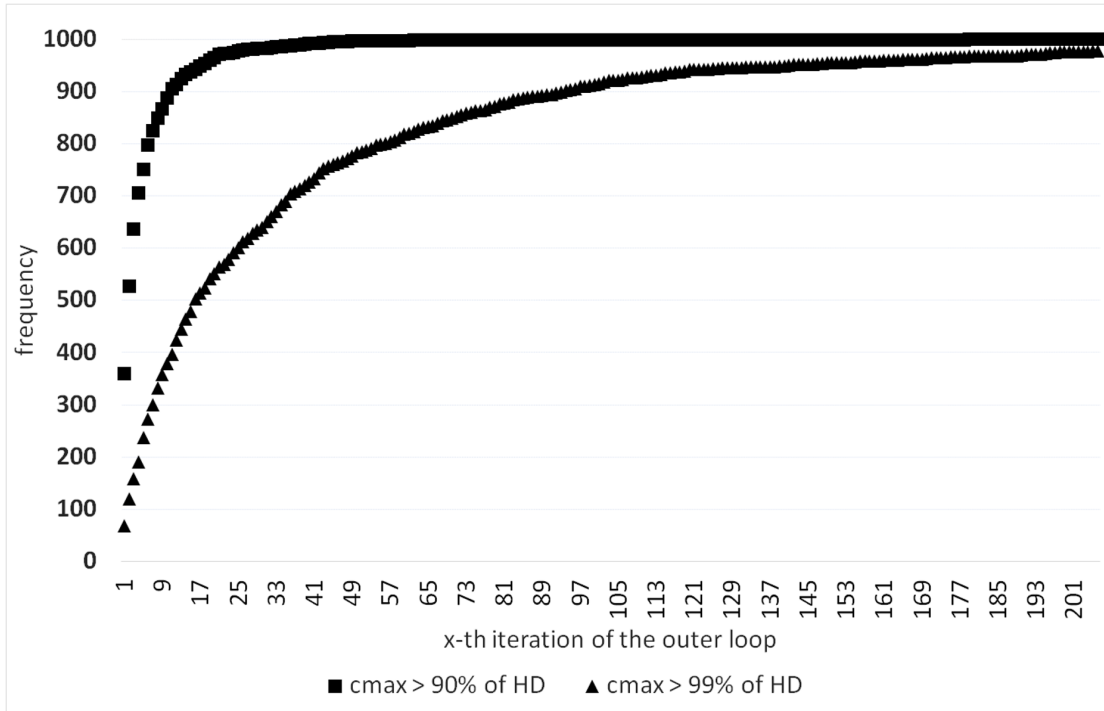


Figure 3.6: Convergence behavior of the temporary HD (c_{max}) along the iterations of the outer loop. Iterations of the outer loop are on the x-axis and the frequencies of c_{max} values exceeding 90% and 99% of the corresponding HD at each iteration are on the y-axis.

3.4 Analysis

The proposed algorithm was tested with three different types of data, namely 3D medical image segmentations, trajectories generated from a road network and random 3D Gaussians.

The terms volume, grid size, and set size used in combination with experiments using 3D medical image segmentations are defined in Definition 7.

Testing with real 3D medical image segmentations is done in four different variants against the ITK HD algorithm and in a fifth variant against a version of the proposed HD algorithm without the random sampling. In the first experiment (Section 3.4.1), the HD between the volumes and the corresponding ground truth segmentations was calculated. In the second experiment (Section 3.4.2), images were compared with randomly selected volumes from the same set, so that the volumes in each pair do not overlap to rule out that the general performance is dependent on the overlap between the compared images. In the third experiment (Section 3.4.4), new images were generated by merging up to 8 images in order to test the performance when the point set size increases. In the fourth experiment (Section 3.4.3), the sensitivity to increasing grid size is tested using whole

body MRI volumes. In the fifth experiment (Section 3.4.5), the same test as in the first experiment was performed, but against the proposed algorithm without the random sampling step to show the effect of combining the early break with the random sampling.

In the sixth experiment (Section 3.4.6), 3D point sets were generated based on random Gaussians and used to test the proposed algorithm to rule out that the efficiency of the proposed algorithm is dependent on the nature of medical images.

Finally, in the last experiment (Section 3.4.7), trajectories generated from a road network were used to test the proposed algorithm against the incremental Hausdorff distance calculation algorithm (INC), based on R-Trees.

The first six experiments were performed on a machine with 3GHz Intel core processor, 8GB Memory, and Windows 7 OS. The last experiment (Section 3.4.7) was done on a machine with the specification described in [NJS11], namely using a computer with a Core 2 Duo processor and 2GB of Main Memory, running Mac OS 10.5.

3.4.1 Comparing Volumes with Ground Truth

In this experiment, we used a test set of 300 automatic brain tumor segmentations (MRI 3D volumes) from the BRATS2012 challenge³. These volumes were produced by segmentation algorithms proposed by four participants of the BRATS challenge. The volumes vary widely in size and span the range from 2k to 600k voxels as point set size and from 125x125x125 to 250x250x250 voxels as grid size. Each of these volumes was validated against the corresponding ground truth segmentation made by human experts. The test set consists of 240 volumes and 60 ground truth segmentations. All volumes were validated using three algorithms: the first is an implementation of the straightforward algorithm (Algorithm 3.1) to ensure that the proposed algorithm computes the correct Hausdorff distance. The second one is the standard Hausdorff distance algorithm of the ITK library⁴, namely the `itk::HausdorffDistanceImageFilter`, assumed to represent the state-of-the-art. The ITK algorithm is based on the distance transform technique and is described in [TSG06] and [EAN08]. The third algorithm is the proposed algorithm, an implementation of Algorithm 3.2.

Figure 3.7 shows the performance of the proposed algorithm compared with the ITK algorithm: while the ITK algorithm took an average time of 2.09 seconds per volume, all runtimes of the proposed algorithm were below one second and have an average of 0.26 seconds per volume, which means that the proposed algorithm outperforms the ITK algorithm by about 7.6 times.

3.4.2 Testing with non-Overlapping Images

The aim of this experiment is to rule out that the performance depends on the overlap between the two compared volumes. To this end, we put all images (segmentations and

³MICCAI 2012 Challenge on Multimodal Brain Tumor Segmentation, <http://www2.imm.dtu.dk/projects/BRATS2012>

⁴National Library of Medicine Insight Segmentation and Registration Toolkit (ITK) <http://www.itk.org>

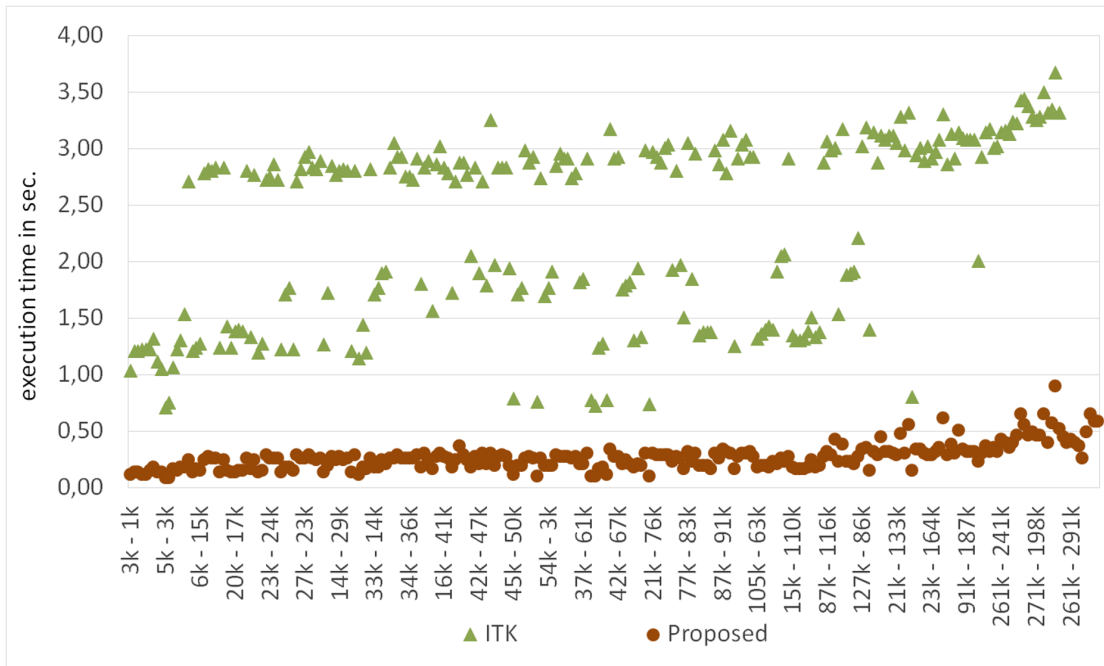


Figure 3.7: Comparison between the performance of the proposed algorithm and the ITK algorithm in validating 240 real brain tumor segmentations against the corresponding ground truth. The set sizes of the segmentation pair being compared in kilo voxels are on the horizontal axis and the run time in seconds is on the vertical axis. The grid size varies from 125x125x125 to 250x250x250 voxels. The entries are sorted according to the sum of the two sizes ascending.

ground truth volumes) in one pool of 300 images, then 300 pairs were selected from the pool so that the intersection (overlap) between the two images in each pair is zero. This was possible because brain tumors reside in different locations in the brain. The HD was calculated using the ITK algorithm and the proposed algorithm. Note that we had to unify all volumes to one grid size, namely 250x250x250 because the algorithms accept only pairs consisting of two volumes with the same grid size. Figure 3.8 shows the runtime plot of the proposed algorithm compared with that of ITK: again the proposed algorithm outperforms the ITK algorithm about by 7.8 times. While the runtimes of the proposed algorithm rarely exceed one second and have an average of 0.51 sec, the ITK algorithm took an average of 3.82 sec. The result shows that the efficiency of the method is not restricted to overlapped point sets and thus confirms the runtime analysis in Section 3.3.4, namely Equation 3.11 that shows that the algorithm tends to have a high efficiency when the HD is large. The increase in the efficiency compensates the efficiency lost when the intersection is not present. This is the case in this experiment because the HD is likely large, given that the images don't overlap.

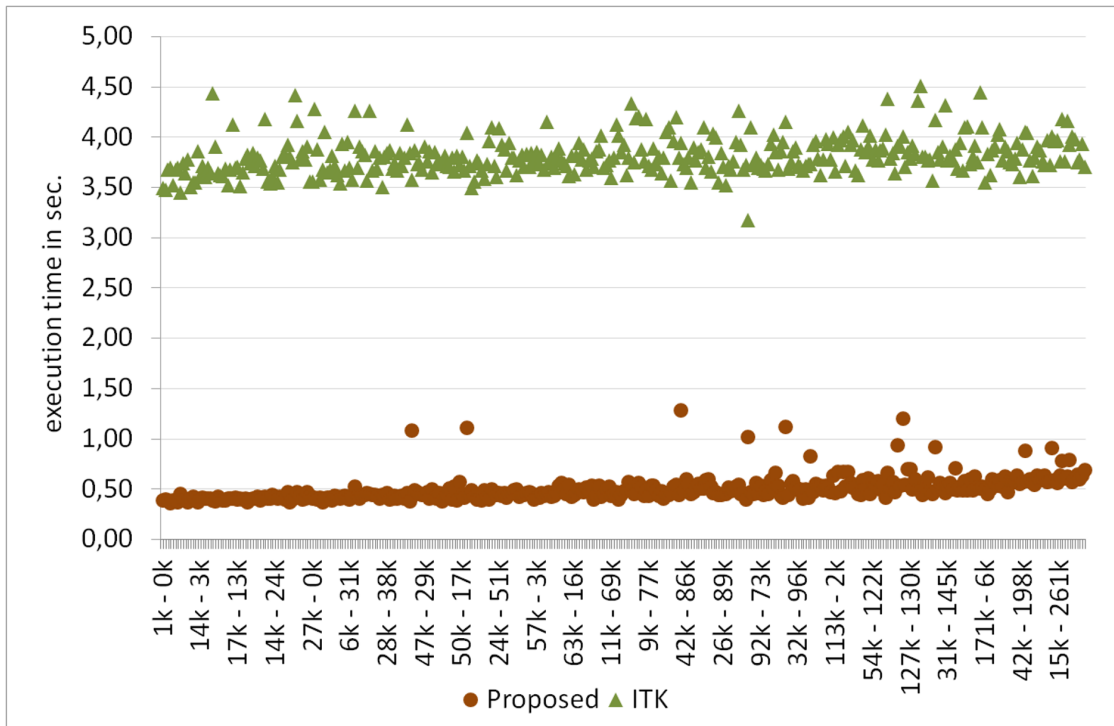


Figure 3.8: Comparison between the performance of the proposed algorithm and the ITK algorithm in comparing 300 pairs of volumes selected randomly so that the overlap between volumes in each pair is zero. The set sizes of the segmentation pair being compared in kilo voxels are on the horizontal axis and the run time in seconds is on the vertical axis. All volumes have a unified grid size of 250x250x250 voxels. The entries are sorted according to the sum of the two sizes ascending.

3.4.3 Efficiency Test with Whole Body Volumes

In this experiment we test the runtime behavior of the proposed algorithm of calculating the HD when the grid size of the volume is increased. For this, we tested it with very large 3D MR and CT volume segmentations from the VISCERAL project [LMMH13]. The set consists of 840 MRI and CT volume segmentations. These volumes were produced by segmentation algorithms proposed by five participants of the VISCERAL Anatomy 1 Benchmark. For each of the volumes there exists a ground truth segmentation. The volumes span the range from $387 \times 21 \times 1507$ to $511 \times 511 \times 899$ voxels as grid size and the range from 1000 to 5Mio voxels as set size. Each of these volumes was validated against the corresponding ground truth segmentation. In a first run, the proposed HD algorithm was executed and in a second run, the algorithm of the ITK Library was used. While the proposed HD algorithm ran through successfully with all volumes, the ITK algorithm broke down with a memory allocation error with all volumes over a particular grid size. The results in Figures 3.9 show that the proposed HD algorithm takes an

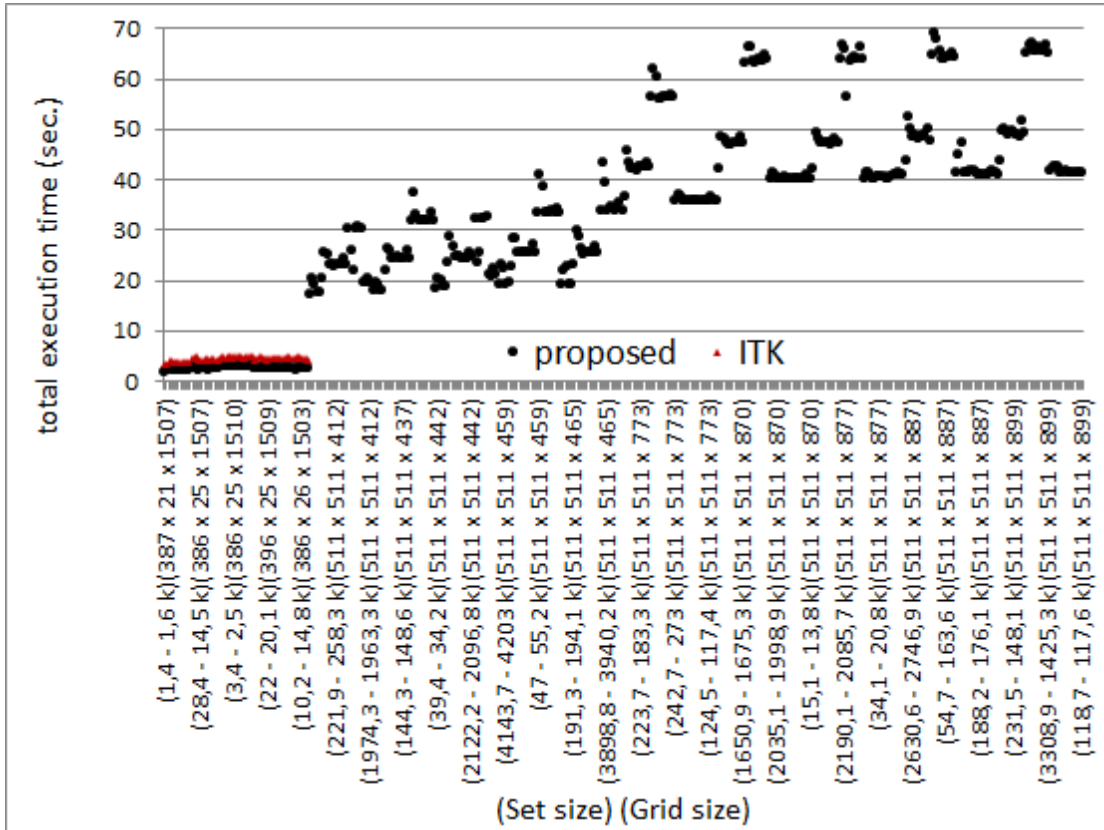


Figure 3.9: Comparison between the performance of the proposed *HD* algorithm and ITK Library implementation in validating 840 whole body segmentations against the corresponding ground truth. The data points are sorted according to the grid size ($w \times l \times h$). The set sizes of the segmentation pair being compared in kilo voxels as well as the grid size are on the horizontal axis and the run time in seconds is on the vertical axis. The ITK implementation failed with memory allocation error with all volumes over a particular grid size. The entries are sorted according to the sum of the two sizes ascending.

average execution time of 33.6 seconds for calculating the *HD*. The failing of the ITK implementation with images with large grid size can be explained by the fact that the distance transform based algorithms are sensitive to increasing grid size because all the background voxels should be labeled. On the contrary, the proposed algorithm is not sensitive to grid size increase because the background is not involved in the computation at all.

The result of this experiment can be explained by the fact that distance transform based algorithms are sensitive to increasing grid size because all the background voxels should be labeled by the algorithm. Ciesielski et al. [00252] investigated the computational complexity of the distance transform algorithm used in ITK and concluded that it is

computationally expensive but ubiquitously needed operation in image processing.

On the contrary, the proposed algorithm is not sensitive to grid size increase because the background is not involved in the computation at all.

The results of all experiments with MRI segmentations against the ITK HD algorithm are summarized in Table 3.1.

Table 3.1: Result summary for experiments on medical images of varying sizes and characteristics where n1..n2 is the size range of the compared point sets, L, B, H are the grid dimensions for medical volumes and the time values are the average execution time for calculating the HD

Testing with ..	LxBxH (grid size)	n1..n2 (set size)	proposed algorithm	ITK algorithm
ground truth	125 to 250	2k..350k	0.26 sec.	2.09 sec.
non-overlapping point sets	250x250x250	2k..350k	0.51 sec.	3.82 sec.
merged volumes	250x250x250	207k..1100K	0.93 sec.	3.45 sec.
Whole body volumes	387x21x1507 to 511x511x899	1.4k..3940k	33.6 sec.	allocation error

3.4.4 Testing with Large segments

The experiment in Section 3.4.3 ensures large grid sizes but does not ensure large segments since the organs can be small. In this experiment we test the runtime behavior when the set size increases. For this, we constructed a new pool of 300 volumes, where each of them is generated by merging up to 8 randomly selected volumes from the original test set without increasing the grid size, which is still 250x250x250 voxels, that is $V = V_1 \cup V_2 \cup V_3 \dots$ where V_1, V_2, \dots are the randomly selected volumes and V is the resulting test volume. The resulting volumes span a set size range from 150k to 850k voxels. Finally, 300 pairs were randomly selected and compared.

Figure 3.10 shows that the proposed algorithm outperforms the ITK algorithm and has no significant runtime increase with increasing the set size.

3.4.5 Testing the Effect of Random Sampling

This experiment is to show the contribution of the random sampling to the efficiency of the proposed algorithm. The same data and configuration of the experiment in Section 3.4.1 is used except that the random sampling is replaced by direct scanning. In particular, Lines 3 and 4 in Algorithm 3.2 are omitted and the sets E and B are used instead of the sets E_r and B_r respectively. The results in Figure 3.11 show that the random sampling is strongly related with the performance of the algorithm and has a significant contribution to the efficiency. Note that eight instances are removed to improve the visibility of the

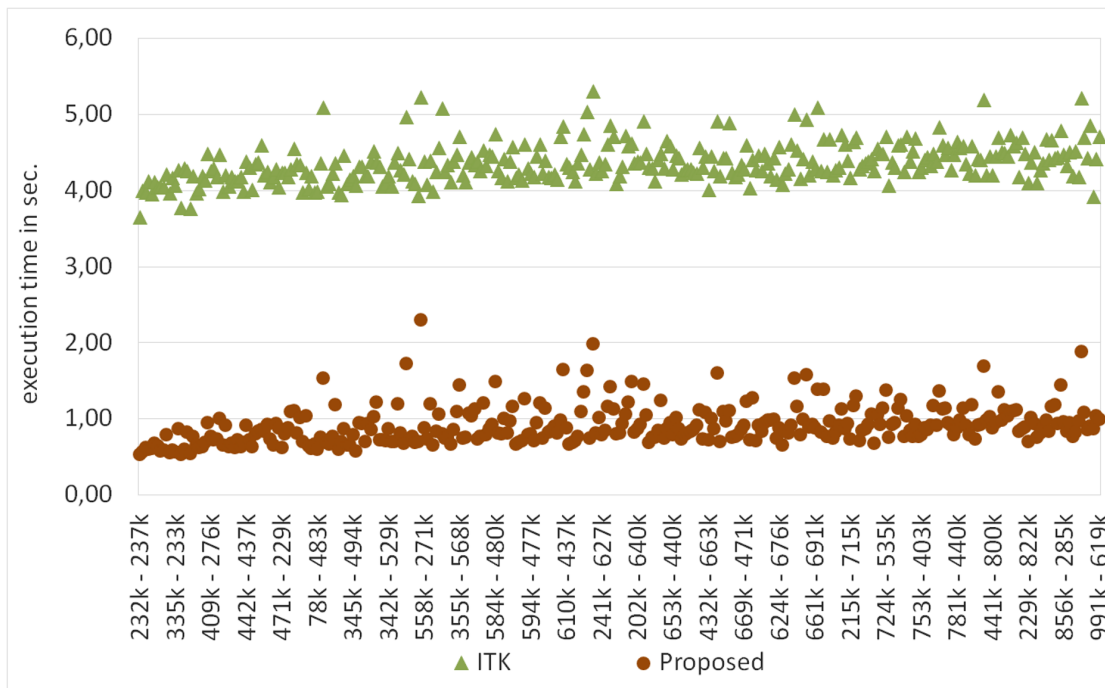


Figure 3.10: Performance comparison between the proposed algorithm and the ITK algorithm in comparing enlarged volumes. The set sizes of the segmentation pair being compared in kilo voxels are on the horizontal axis and the run time in seconds is on the vertical axis. All volumes have a unified grid size of 250x250x250 voxels. The entries are sorted according to the sum of the two sizes ascending.

plot because they have an execution time exceeding 100 seconds with direct scanning. The contribution of the random sampling is a factor of 36.8 measured as the ratio between the two execution times averaged over all pairs.

3.4.6 Testing with Random Gaussians

To rule out that the efficiency is dependent on the point distribution of medical volumes, the proposed algorithm was tested against random Gaussians. 300 point clouds were generated; each of them consisting of 50 thousand to 0.5 million points; the points in each cloud are normally distributed and satisfy a random Gaussian (i.e. the point coordinates x , y and z are generated according to three different Gaussians each with a random μ and a random σ) selected so that the points fit in a grid of 250x250x250 voxels. From these point clouds, 300 pairs were randomly selected. The HD distance between the point sets in each pair was measured by the proposed algorithm and the ITK algorithm. The results in Figure 3.12 show that the proposed algorithm still outperforms the ITK algorithm with a factor of about 4.35. The experiment shows that the proposed algorithm replicates its performance with normally distributed point sets.

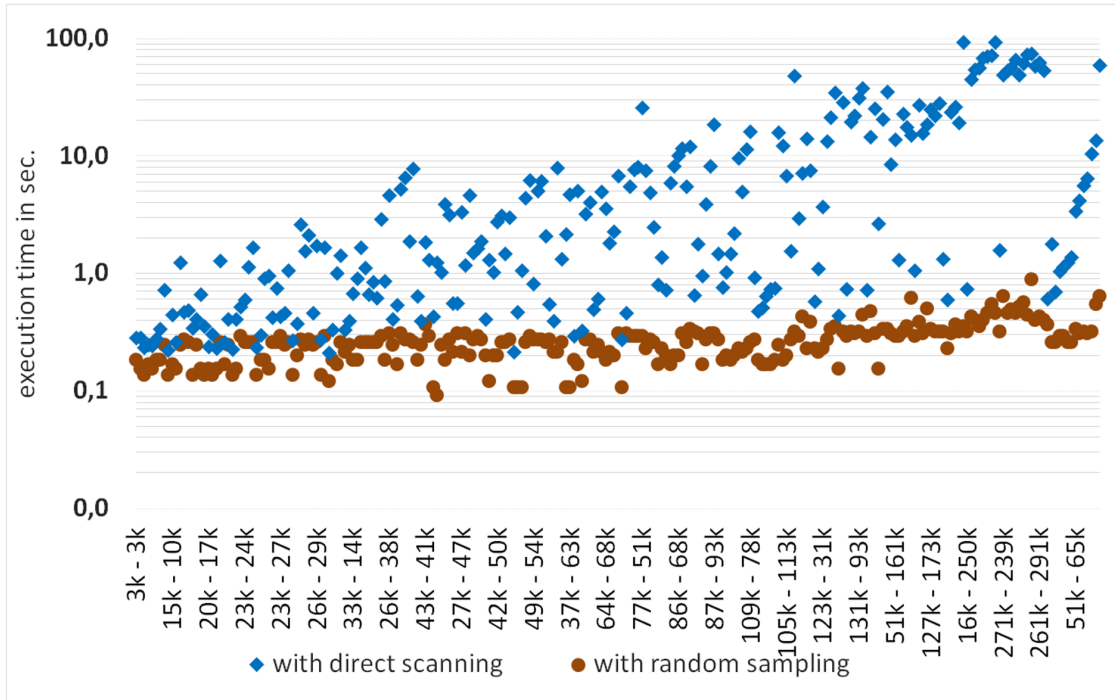


Figure 3.11: Contribution of random sampling: Comparison between the efficiency of the proposed algorithm when using random sampling and direct scanning. The same data as in the experiment in Section 3.4.1 is used. The size of the compared images in kilo voxel is on the x-axis and the execution time in seconds, scaled logarithmically, is on the y-axis.

We analyzed the few data points in Figure 3.12 where the proposed algorithm required a computation time of more than 4 seconds. We found that the relatively long runtime is not related to a particular point set, but rather to a combination between two point set configurations. In particular the runtime is relatively long when the pairwise distances are in average small compared with the HD, which causes that max grows slower and the early break consequently occurs less often. This observation is in conformance with the runtime analysis in Section 3.3.4, i.e. that the runtime is dependent on the value of the HD relative to the distribution of the pairwise distances between the compared point sets, as illustrated in Figure 3.3.

3.4.7 Testing against Incremental Hausdorff Distance

In this experiment, the proposed method was tested against the incremental Hausdorff distance calculation algorithm (INC) proposed by Nutanong et al. [NJS11]. To this end, we tested the proposed algorithm with the same data, the same setting, and on hardware of identical specification as described in [NJS11], Section 7.1 (Hausdorff Distance Calculation). As point sets, we used trajectories generated from the Oldenburg (OL)

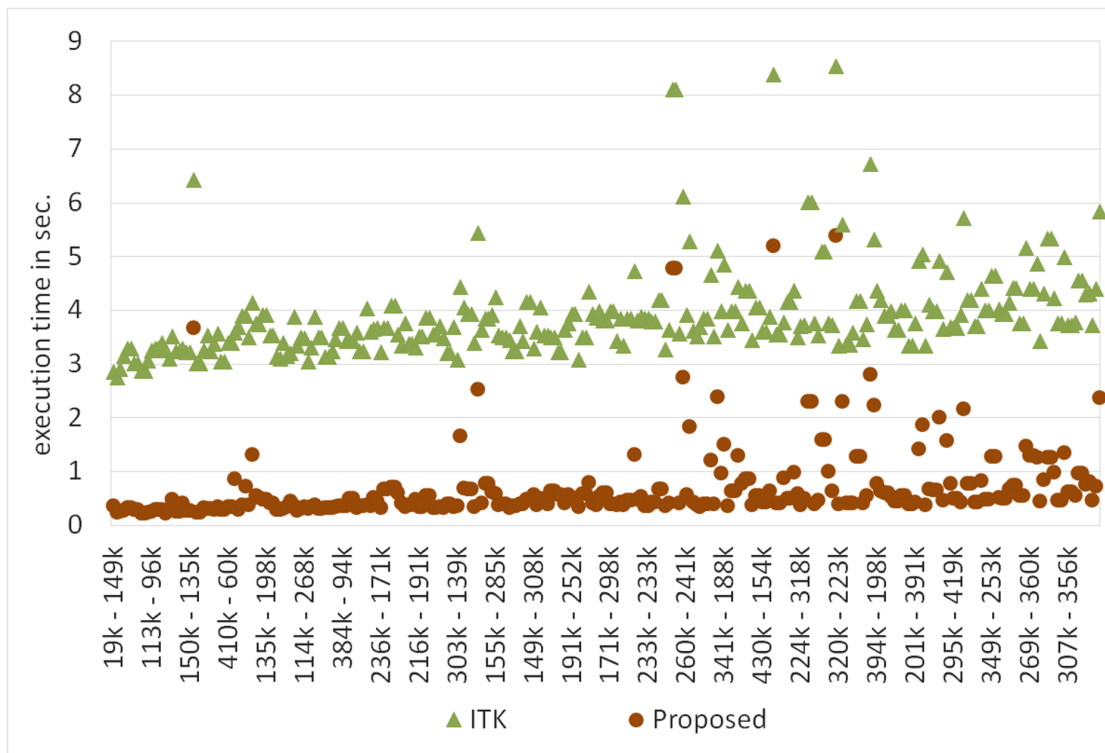


Figure 3.12: Comparison between the performance of the proposed algorithm and the ITK algorithm in measuring the HD of 300 pairs of Gaussians generated by randomly selected means and standard deviations for each of the 3 dimensions. The sizes of the point sets being compared in kilo voxels are on the horizontal axis and the execution time in seconds is on the vertical axis.

road network⁵ so that each trajectory is the shortest path between two randomly selected points in the network with a length of 2000 units. The points on each trajectory were sampled in different resolutions, i.e. the path was truncated into chunks with different lengths. Five groups of trajectories (G1 .. G5) were constructed so that each group contains trajectories sampled in a different resolution. G1, G2, G3, G4, G5 have 400, 800, 1200, 1600, 2000 sampled points respectively. The HD(X,Y) was calculated between trajectories by varying the point set size, i.e. selecting trajectories from different groups. In a first experiment set, the size of X was fixed and the size of Y was varied, and in a second experiment set the size of Y was fixed and the size of X was varied. The execution times of these experiments are compared with the execution times published in [NJS11], Section 7.1, Figure 8. Figure 3.13 shows the execution time where each data point is the average of 200 different pairs of trajectories. The results show that the proposed algorithm outperforms the INC algorithm by about 30 times.

⁵City of Oldenburg Road Network <http://www.cs.fsu.edu/~lifeifei/SpatialDataset.htm>

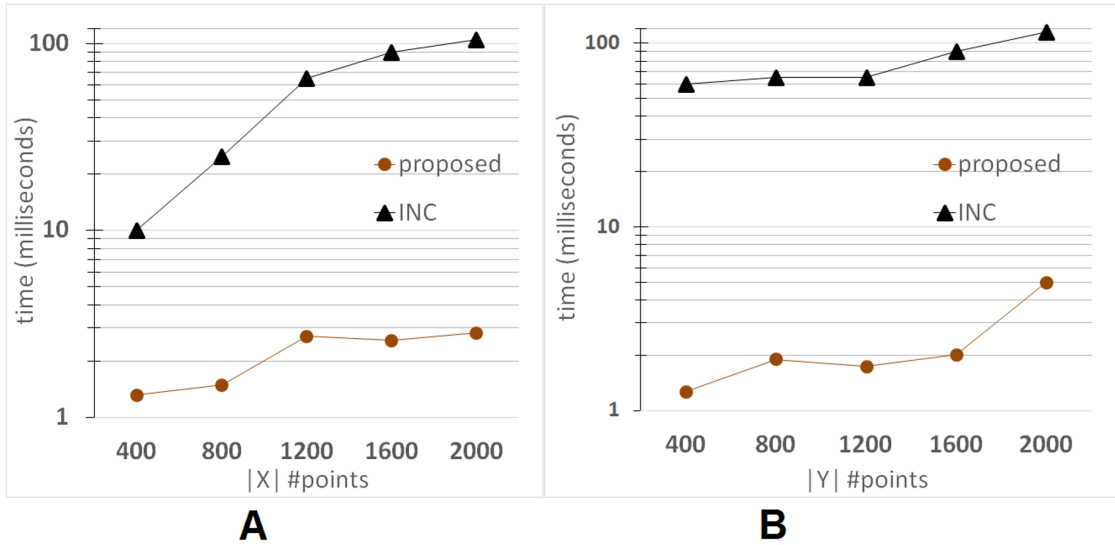


Figure 3.13: Comparison of the execution time of calculating the Hausdorff distance $HD(X,Y)$ by the proposed algorithm and the incremental Hausdorff calculation (INC) [NJS11]. In A, the size of X is fixed and the size of Y varies and conversely in B. Each data point is the average of 200 pairs of trajectories. The size of the point set is on x-axis and the execution time in milliseconds on the y-axis.

3.5 Average Distance between Image Segmentations

The Average Distance (AVD) is defined in Equations 3.3 and 3.4, which we restate here. The AVD is defined as the average of minimum distances from points in the first point set to the second one and vice versa. It is defined as

$$AVD(A, B) = \frac{d(A, B) + d(B, A)}{2} \quad (3.3)$$

where $d(A, B)$ is the directed Average Hausdorff distance that is given by

$$d(A, B) = \frac{1}{N} \sum_{a \in A} \min_{b \in B} \|a - b\| \quad (3.4)$$

Obviously, a straightforward computation of the AVD has a complexity that is quadratically proportional to the point set size.

The AVD is known to be stable and less sensitive to outliers than the HD . The AVD is commonly used to compare image segmentations, for example comparing medical image segmentations [SSS02], [MJB⁺12], [KCAB09]. Unfortunately, the two optimization techniques used for the Hausdorff distance (early break and randomization) cannot be applied for the AVD because the AVD attempts to calculate all the distances and finally considers their average, which makes early break optimization not applicable. In this section, we use two optimization techniques to achieve efficient calculation of the AVD

between two binary image segmentations. We formally define binary image segmentation as follows:

The nearest neighbor (NN) is the core operation used to calculate the average distance between two image segmentations. Given a point q and a segment A , the NN of q in A is defined as:

Definition 16. *Nearest neighbor (NN): Let A be a segment in a binary segmentation according to Definition 7. Let q be any query point. The nearest neighbor operation $NN(q, A)$ finds the point $p \in A$ such that $d(q, p) \leq d(q, x), \forall x \in A$.*

In this section, we provide optimizations for the NN operation used in combination with image segmentations. The first optimization is based on excluding the irrelevant voxels from the calculation, e.g. by considering representations of the segmentations as hollow objects (surfaces), thereby excluding the inside voxels from the calculation. The second optimization is based on reducing the search space of the NN by finding a convenient search radius.

3.5.1 Voxel Exclusion

Let A and B be segmentations according to Definition 7, illustrated in Figure 3.14 (1) and (2), which are to be compared using the average distance. Since both of the segmentations are defined on the same grid, the segments (foreground) in A and B result in three types of regions as illustrated in Figure 3.14 (3). Assuming that A is the ground truth segmentation and B is the segmentation being tested, then region $R1 = A \setminus B$ represents the false negative (FN), $R2 = A \cap B$ represents the true positive (TP), and $R3 = B \setminus A$ represents the false positive.

According to Equation 3.3, the average distance is the average of the directed average distances from A to B and from B to A . Since we are interested in optimizing the nearest neighbor function, we only describe one direction, namely from A to B . The same holds for the opposite direction.

A naive calculation of the directed AVD from A to B should calculate the nearest neighbors of all points $q \in A$, which we call the domain of the NN operation $D(NN)$. Note that $A = R1 \cup R2$. The nearest neighbors are found in B , which we call the range of NN operation $R(NN)$. Note that $B = R2 \cup R3$.

Obviously, The first voxel exclusion optimization is removing region $R2$ from the domain $D(NN)$, i.e. calculating the nearest neighbors of only those points of A that are in $R1$ and ignoring those lying in $R2$. This optimization is illustrated in Figure 3.14 (4) and (5). This is justified by the fact that all points $q \in A$ lying in $R2$ have a zero distance to the segment B . Note that the inverse does not hold, i.e. $R2$ is only removed from the domain $D(NN)$ and may not be removed from the range $R(NN)$, since a point $x \in B$ in $R2$ could be the NN of some point $q \in A$ that is not in $R2$.

The second voxel exclusion optimization makes use of the nature of segmentations being rigid objects (dense point sets). It suggests considering only the surface of $R(NN)$, instead of considering all points inside it, as illustrated in Figure 3.14 (6). Here $R(NN) =$

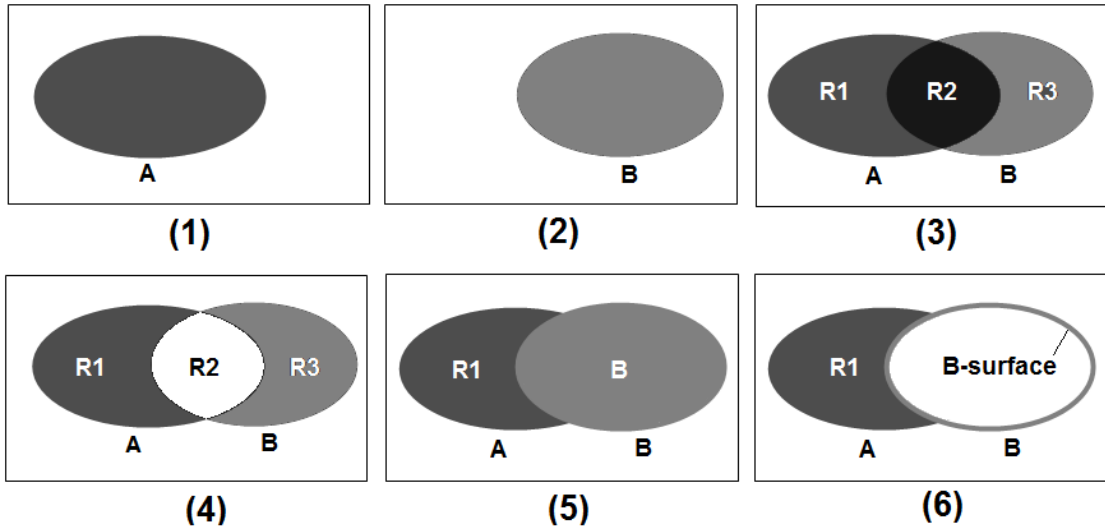


Figure 3.14: Illustration of the optimizations used in calculating the average distance (AVD). In (1) and (2), the images A and B , defined on the same grid, are to be compared using the AVD . In (3), the intersection of the images is identified. In (4), the points in the intersection are removed from the domain $D(NN)$, since they have zero distances. In (5), only distances from $R1$ to B are considered. In (6), only the boundary voxels (surface) of B are considered as range $R(NN)$.

surface(B) This is justified by the fact that when moving in a straight line from a point $q \in R1$ toward B , the first point crossed in B is on the surface, which means the nearest neighbor cannot be farther than this point. This implies that all points inside B and not on the surface are not relevant.

To extract the surface (boundary) of a 3D segmentation, any convenient contour tracing algorithm can be used, which has the following properties: (i) can be applied for 3D images, (ii) discovers the holes in an object, and (iii) visits all connected components in an image. More information about contour tracing algorithms is in [SHB07] [CKL14] [Zam82]. For illustration, we use Algorithm 3.4 that takes a segmentation as input and calculates its 3D boundary, i.e. a hollow segmentation having the same surface. Figure 3.15 illustrates the surface of a real brain tumor segmentation resulting from applying Algorithm 3.4.

3.5.2 Reducing the Search Sphere

Voxel exclusion optimization presented in Section 3.5.1 achieves a considerable efficiency improvement. However, more improvement is required to achieve satisfactory efficiency for huge image segmentations. In this section, we propose another optimization of the nearest neighbor (NN) operation for calculating the AVD . This optimization results in the search space required to find the NN of a query point being reduced, i.e. it avoids

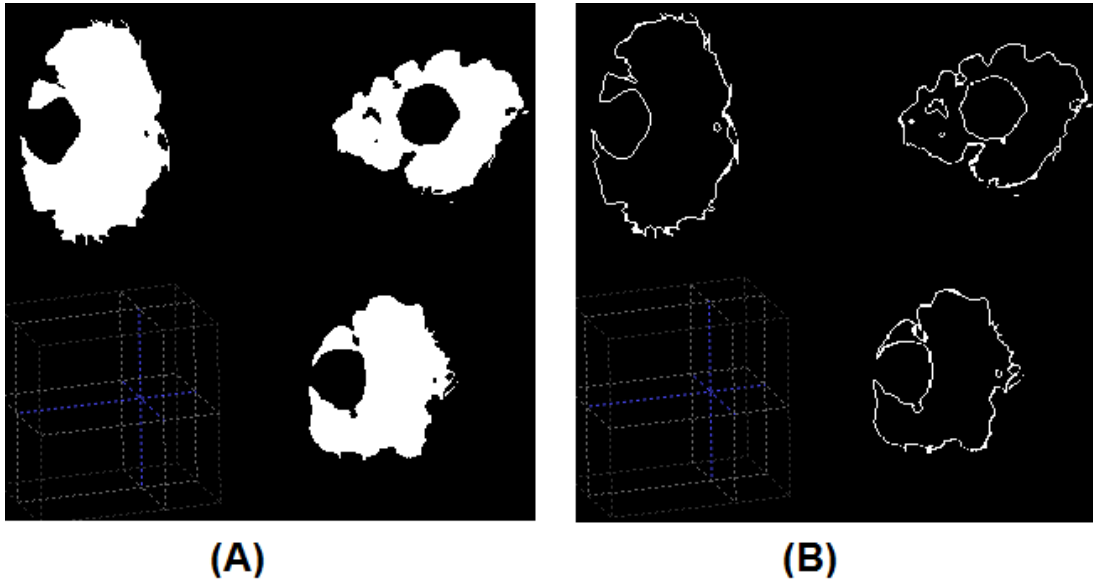


Figure 3.15: Finding the surface of a segmentation: In (A), a 3D segmentation of the edema of a real brain tumor viewed as three orthogonal slices. In (B), the surface (boundary) of the same segmentation as a result of applying the Algorithm 3.4 on the segmentation in (A).

Algorithm 3.4: HOLLOW returns the surface points of a 3D image

Input: Image as a set of pixels, I
Output: Hollow image $h(I)$

```

1  $H \leftarrow \emptyset$ ;
2 foreach  $p \in I$  do
3    $N \leftarrow$  the 6 neighbors of  $p$ ;
4   foreach  $n \in N$  do
5     if  $n$  is background then
6        $p$  add to  $H$ ;
7     end
8   end
9 end
10 return  $H$ 

```

scanning the whole segment surface (hollow segment) resulting from the first optimization. We use a modified version of the NN algorithm proposed by Zhao et al. [ZLX⁺14] in which a 3D cell grid is built on the point cloud and for each query point, a search subspace (a subset of the grid cells containing the nearest neighbor) is found to limit the search required to find the NN. The modification we added to this algorithm is to find a radius r of a convenient search sphere that for sure contains the NN in B of a given query point

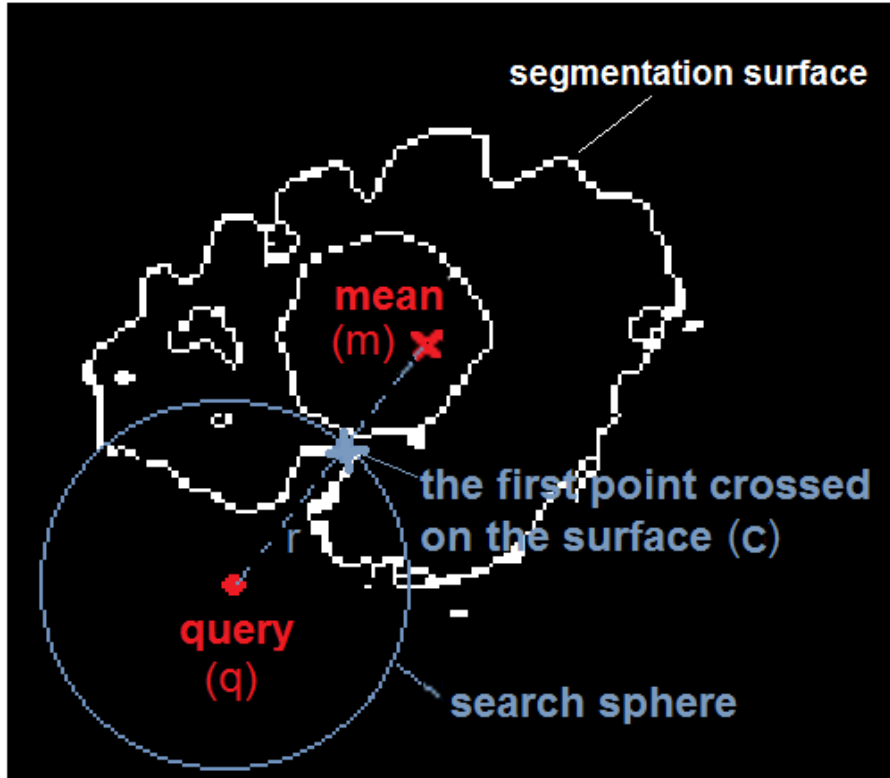


Figure 3.16: Illustration of the optimizations achieved by reducing the search space when searching the nearest neighbor, a search sphere with radius r is found by moving from the query q toward the mean m and considering the first point crossed on the boundary.

$q \in A$, as illustrated in Figure 3.16. Finding this convenient search radius r is done by moving in a straight line from the query point q toward the mean (centroid) m of the segment B and stopping as soon as a point $c \in B$ is crossed on the surface of segment B . The distance from the query q to the crossing point c is the search radius required, since it ensures finding the NN within the search sphere, i.e. $r = \overline{qc}$. After finding a convenient sphere, all grid cells contained by this sphere or crossed by its surface are searched. If no point c is found (which is unlikely to happen with segmentations), an exhaustive search is performed.

We present the experiments that validate the efficiency of the proposed algorithm for calculating the *AVD* with two different sets of real MR and CT volume segmentations, namely brain tumor segmentations (Section 3.5.3) and whole body image segmentations (Section 3.5.4). In both cases, the proposed *AVD* algorithm was tested against the implementation of the *AVD* algorithm of the ITK library version 4.4.1, assumed to represent the state-of-the-art. This ITK algorithm is based on the distance transform technique, described in [TSG06] and [EAN08]. All experiments were executed on a

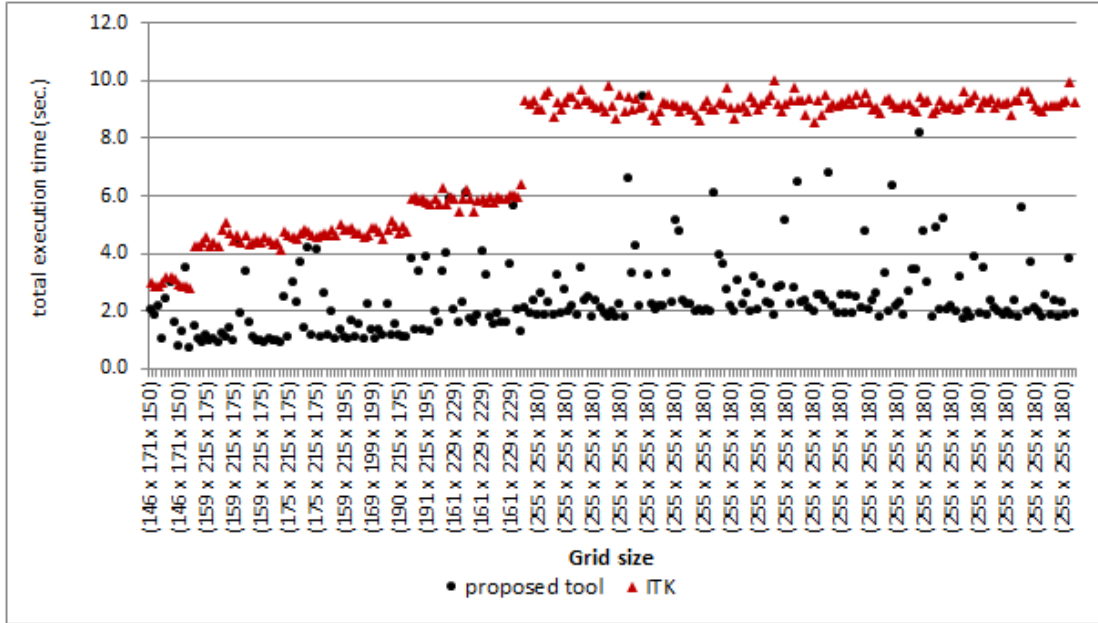


Figure 3.17: Comparison between the performance of the proposed *AVD* algorithm and the *AVD* algorithm of the ITK Library in validating 240 brain tumor segmentations against the corresponding ground truth. The grid size is on the horizontal axis and the run time in seconds is on the vertical axis. The data points are sorted according to the grid size ($w \times l \times h$).

machine with Intel Core (i5) CPU, 8 GB RAM and Windows 7 OS. Note that all execution times include the time for reading the images and calculating the metrics.

3.5.3 Efficiency Test with Brain Tumor Segmentation

In this experiment, the proposed algorithm of calculating the *AVD* was tested with real brain tumor segmentations (MR 3D volumes). We used the same dataset of the experiment in Section 3.4.1, namely 300 automatic brain tumor segmentations from the BRATS2012 challenge, consisting of 240 volumes and 60 ground truth segmentations (more details in Section 3.4.1). Each of these volumes was compared twice with the corresponding ground truth segmentation, one time using the proposed *AVD* algorithm, and one time using the *AVD* algorithm of the ITK Library.

Figure 3.17 shows that the proposed algorithm outperforms the ITK implementation in computing the *AVD* by a factor of 3.0 and takes an average of 2.5 seconds. It also shows that in contrast to the proposed algorithm, the performance of the ITK algorithm is strongly dependent on the grid size.

3.5.4 Efficiency Test with Whole Body Volumes

In this experiment we test the runtime behavior of the proposed algorithm of calculating the *AVD* when the grid sizes of the volumes are huge. For this, we tested it with whole body 3D MR and CT volume segmentations from the VISCERAL project [LMMH13]. We used the same dataset of the experiment in Section 3.4.3, consisting of 840 whole body MRI and CT volume segmentations with grid sizes varying from $387 \times 21 \times 1507$ to $511 \times 511 \times 899$ voxels and segment sizes varying from 1000 to 5Mio voxels. Each of these volumes was validated twice against the corresponding ground truth segmentation, one time using the proposed *AVD* algorithm and another time using the algorithms of the ITK Library.

While the proposed *AVD* algorithm ran through successfully with all volumes, the ITK algorithm broke down with a memory allocation error with all volumes over a particular grid size. The results in Figures 3.18 show that the proposed *AVD* algorithm takes an average execution time of 38.9 seconds for calculating the *AVD*. Similarly to the behavior of the *HD* distance in Section 3.4.3, the failing of the ITK implementation with images of large grid size can be explained by the fact that the distance transform based algorithms are sensitive to increasing grid size because all the background voxels should be labeled. The proposed algorithm is not sensitive to grid size increase because the background is not involved in the computation at all. Even if the segment is large (background is small), the proposed algorithm is still efficient since it considers only the surface of the segment.

3.6 Summary

We propose an efficient algorithm for computing the exact Hausdorff distance. We formally show that the proposed algorithm has a nearly-linear runtime in the average case. The proposed algorithm combines early breaking and randomization optimizations to achieve a significant increase in speed over other algorithms that do not use this combination. The proposed algorithm does not impose any restrictions on the input data, and is hence generalizable to all applications. Moreover, it does not require a complex setup phase needing high computational effort and extensive storage space.

We experimentally show a 36-fold increase in speed over an HD algorithm with only early breaking included i.e. without using the randomization. We also show experimentally that the proposed algorithm significantly outperforms in terms of speed the standard HD algorithm of the ITK Library in comparing medical volumes and the incremental HD algorithm in comparing trajectories generated from a road network. Moreover, the proposed algorithm is shown to work even when comparing volumes with extremely high dimensions (grid size).

Furthermore, we propose two optimizations that achieve an efficient calculation of the average distance (*AVD*). The first optimization is based on voxel exclusion by (i) removing the segment intersection and (ii) considering only the voxels on the segment surface, but not those inside the segment. The second optimization is based

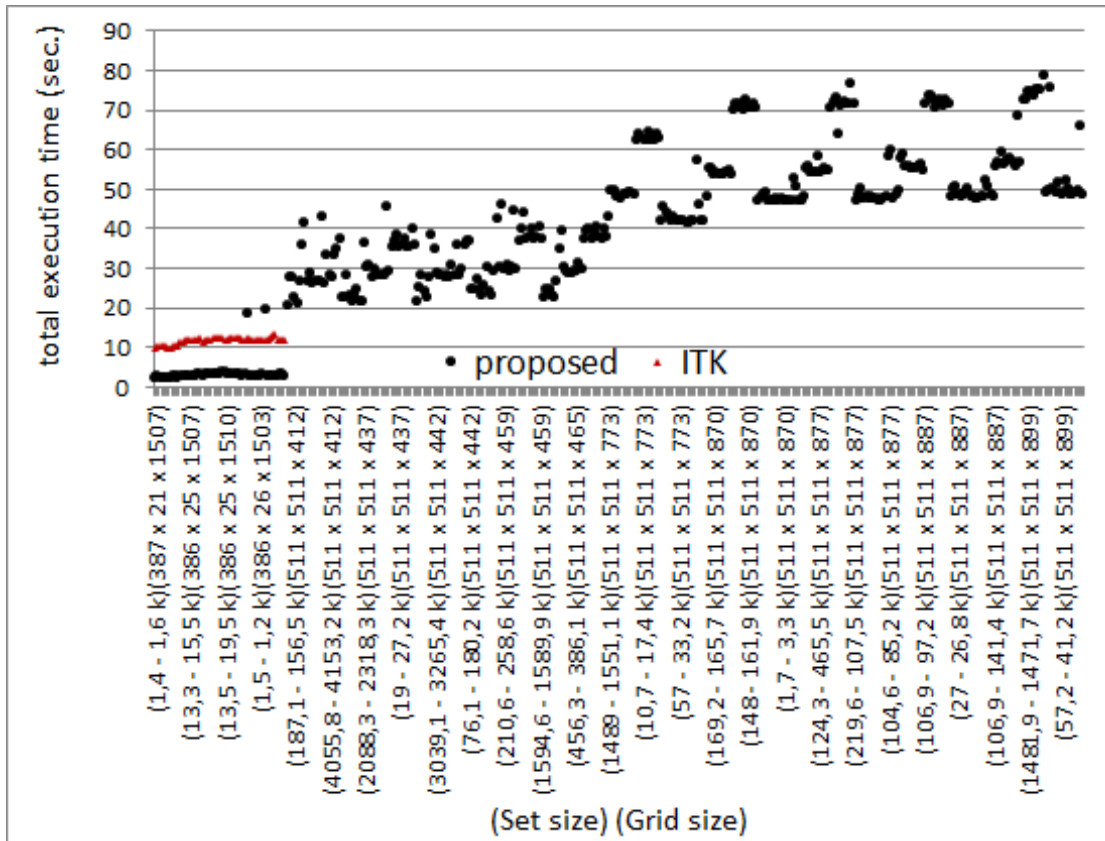


Figure 3.18: Comparison between the performance of the proposed *AVD* algorithm and ITK Library implementation in validating 840 whole body segmentations against the corresponding ground truth. The data points are sorted according to the grid size ($w \times l \times h$). The set sizes of the segmentation pair being compared in kilo voxels as well as the grid size are on the horizontal axis and the run time in seconds is on the vertical axis. The ITK implementation failed with memory allocation error with all volumes over a particular grid size. The entries are sorted according to the sum of the two sizes ascending.

on reducing the search subspace by finding a convenient search radius that contains the nearest neighbor. An implementation of these algorithms is available as part of the tool EvaluateSegmentation⁶.

⁶EvaluateSegmentation is an open source project for evaluating medical volume segmentations available for download from <http://github/codalab/EvaluateSegmentation>.

Formal Analysis of Hubness

4.1 Introduction

After having addressed a problem related to feature spaces with huge number of points, in this section, we address another problem related to feature spaces of high dimensionality, namely the hubness problem.

Hubness is a term used to denote a phenomenon related to the k -nearest neighbor algorithm (kNN) when applied to high dimensional data. Hubness is characterized by the emergence of hubs and anti-hubs in high-dimensional data. Hubs are data points for which the probability of appearing in the kNN -lists of other points is significantly higher than expected. On the contrary, anti-hubs are data points for which the probability to be nearest neighbor of other points is significantly lower.

In a recent analysis of hubness observed in a music retrieval system, Flexer et al. [FSS12] show by analyzing human gradings of songs retrieved by the system, that hub songs seem to exhibit less perceptual similarity to the songs they are close to. They also show that hubness is strongly related to the classification algorithms used and the features selected to build the model rather than being a property of particular songs. Furthermore, they show that hubness is also observed in very large databases, e.g. music retrieval with more than 1/4 Mio songs, and even gets worse with increasing collection size. Radovanovic et al. [RNI10] empirically show that the distance metric used in the kNN algorithm as well as the data distribution have a direct impact on the potential emergence of hubness and that least hubness is observed with the cosine distance combined with sets of points with normally distributed components.

The curse of dimensionality denotes phenomena that arise when dealing with data in high dimensional spaces. The distance concentration [RNI10] is one of these phenomena, known as the tendency of all pairwise distances to be equal when the dimensionality d is sufficiently high. Distance concentration has been studied intensively, e.g. in [BGRS99] [HAK00] [AHK01] [Fra08] [FWVM07] [Koe00] [RS05] [ST83]. Because our analysis of

hubness is closely related to the distance concentration, we present more details on relevant aspects of distance concentration in Section 4.3.

In contrast to distance concentration, hubness, as a further aspect of the curse of dimensionality, has not been deeply studied. While there is considerable research about the impact of hubness on machine learning and information retrieval, less research has been invested for understanding the origin of hubness [RNI10].

Formal definitions: One of the important algorithms performed to metric spaces is the nearest neighbor algorithm, which is a core computation in machine learning and information retrieval because many algorithms are based on it. Given an object $x \in X$, the task is to find $y \in X$ such that $d(x, y) \leq d(x, z), \forall z \in X, z \neq x$. In the following we provide basic definitions and notations related to the nearest neighbor algorithms.

Definition 17. *The k -nearest neighbor list of the point $x \in X$ is defined as the point subset $L_k(x) \subset X$ where $\forall x_i \in L_k(x), x_j \in X \setminus L_k(x) : \|x, x_i\| \leq \|x, x_j\|$*

Definition 18. *The k -occurrence $n_k(x)$ of the point $x \in X$ is defined to be the number of times the point x appears in the k -nearest neighbor lists of other points, i.e.*

$$n_k(x) = \sum_{i=1}^n \mathbb{1}_{L_k(x)}(x_i), \text{ where } \mathbb{1}_{L_k(x)} \text{ is the indicator function with respect to } L_k(x), \text{ i.e. } \mathbb{1}_{L_k(x)}(x_i) = 1 \text{ if } x_i \in L_k(x) \text{ and } 0 \text{ otherwise.}$$

Definition 19. *Hubness is defined as the asymmetry of the distribution of the k -occurrence in Definition 18, which leads to :*

- *the emergence of hubs, where a hub is a point x_h for which $n_k(x_h) \gg E[n_k]$, and*
- *the emergence of anti-hubs, where an anti-hub is a point x_u for which $n_k(x_u) \ll E[n_k]$.*

where $E[\cdot]$ denotes the expected value. A basic measure of hubness is the skewness of the k -occurrence, given by

$$S_{n_k} = \frac{\mu_3}{\sigma^3} = \frac{E[(n_k(x) - \mu)^3]}{(E[(n_k(x) - \mu)^2])^{3/2}} \quad (4.1)$$

where μ_3 is the third central moment, μ is the mean, and σ is the standard deviation of n_k

The basic measure of hubness (Equation 4.1) as the third standardized moment of the k -occurrence S_{n_k} has been used in [LBSN13] [RNI10]. We will use this measure as a reference for validating the proposed explanation of hubness and the proposed hubness indicator.

Further notations used throughout this chapter are defined in Table 4.1.

Notation	Definition
DTM	distance to mean
PWD	pairwise distance
NN	nearest neighbor
d	dimensionality of a hyperspace
d -space	a space of dimensionality d , likewise d -hypercube and d -hypersphere
HDS	high dimensional space
$E[.]$	the expected value
$\text{var}(.)$	the variance

Table 4.1: Notation used throughout this chapter

Contribution: This chapter results in the following contributions:

- An explanation of the cause of hubness, based on sparsity and distance concentration in high-dimensional space. This explanation is general and not restricted to any assumptions on the distribution and the distance norm used.
- A novel hubness indicator based on distance statistics that estimates the hubness in a data set in linear time in terms of the point set size.
- A novel hubness estimator of the amount of hubness caused by a particular point.
- Two novel strategies for hubness reduction.

The explanation of hubness proposed in this chapter is based, amongst others, on facts related to distance concentration and sparsity in the high dimensional space. We show that points in high dimensional space have a distance structure that is similar to the distance structure of the vertices of a hypercube. Based on this fact, we show that a hub is a point that deviates from its corresponding hypercube vertex towards the centroid, which makes it nearer to all points at vertices connected with its vertex by one edge. Furthermore, we suggest two novel strategies for hubness reduction based on the hubness explanation proposed.

Chapter organization: The remainder of this chapter is organized as follows: We provide in Section 4.1 a formal definition of the hubness problem and a basic function for measuring hubness as well as notation that holds throughout this chapter. Section 4.2 presents some relevant research related to hubness. In Section 4.3 we provide a model of distance structure in high dimensional space; thereby we provide a theoretical background that we use to explain the cause of hubness. In Section 4.4, we propose a novel explanation of the cause of hubness based on the model presented in Section 4.3. Novel hubness indicators are proposed in Section 4.5 that estimate the hubness probability in a given

point set and the amount of hubness caused by a particular point. Section 4.6 presents further analysis that confirms the theoretical findings by providing empirical results and explanations for phenomena described in the literature.

4.2 State-of-the-Art

Origin of hubness: Radovanovic et al. [RNI10] provide an analysis of the origin of hubness. They empirically show a correlation between the position of a data point relative to the data mean and the probability of it being a hub. While it is already well-known that a point nearer to the mean is on average nearer to all other points, the main contribution of Radovanovic et al. is that they formally show that this effect is amplified in a normal distribution when the dimensionality is increased. Since the Euclidean distances between the mean and the other points (*DTM*) are distributed according to the Chi square distribution (this follows directly from the definition of the Chi square), they use this as a model to show their claim. For this they assume an imaginary point set with i.i.d. normally distributed points, in which the dimensionality is successively increased and show using the Chi square distribution that the nearer the point is to the mean, the higher the probability of the point being a hub. For this, while increasing the dimensionality, they tracked and observed two points drawn from the data, but located at specific positions with respect to the origin. These positions are expressed in terms of the standard deviation σ and the expectation value of the *DTM*. For each of the two points, the average distance to all other points was observed. As an examples they considered one point p_1 located at the expected distance and another point p_2 located $2 \cdot \sigma$ closer to the mean than the first one. They showed (as a main contribution of their paper) that the average distance to all other points of the first point $\overline{\|p_1, \cdot\|_2}$ is larger than this of the second point $\overline{\|p_2, \cdot\|_2}$ and that the difference between the two averages increases with the dimensionality, i.e. $\overline{\|p_1, \cdot\|_2} > \overline{\|p_2, \cdot\|_2}$ and $(\overline{\|p_1, \cdot\|_2} - \overline{\|p_2, \cdot\|_2}) \sim d$. Note that this claim is proven only for the Euclidean distance and the normal distribution and thus lacks the generality.

Low et al. [LBSN13] challenge the claim of Radovanovic et al. [RNI10] and state that hubness is a matter of density gradient in data sets, e.g. at the boundary of finite data sets or in distributions with decreasing density gradient like the normal distribution, rather than a matter of high dimensionality (“... *that the emergence of hubs is an intrinsic effect of the dimensionality of the data - a view we dare to challenge here*” [LBSN13]). They provide different empirical examples by generating random point sets uniformly sampled over the d -hypercube and the d -hypersphere and demonstrated that hubness also occurs in low dimensional point sets, e.g. 2D and 3D images. In other experiments, they demonstrated that hubness can be increased by increasing the boundaries in the data set, i.e. when the point set consists of many chunks separated from each others, and each of them is uniformly distributed. Furthermore, they also demonstrate empirically that the strength of hubness depends on the distribution of the points; e.g. while hubness is strongly observed in normally distributed point clouds, it seems to be weak in points distributed in a hyperball. From these empirical observations, they concluded that

hubness is caused by density gradient. They claimed that the density gradient at the boundary of the point set causes hubness, which explains the increase of hubness when the point set consists of chunks, since spitted data has more boundaries than continuous data. They also claimed that the density gradient in normally distributed data, which results from decaying the density while moving farther from the mean, is the reason why hubness is in general higher in normally than in uniformly distributed data.

Retrievability: Azzopardi et al. [AV08] introduced the concept of retrievability as a measure of how a retrieval system affects the users' ability to access document, i.e. it gives knowledge about how retrievable the system makes individual documents. They provide analysis of TREC collections that demonstrate the importance of the utilities provided by the retrievability measure they proposed, especially with tasks that require the content being reliably accessible, e.g. higher order information access tasks, such as e-Government accessibility. They emphasizes that effectiveness is insufficient to evaluate biased retrieval systems due to the growing amount of content and the growing users' reliance on retrieval systems in finding contents. They justified this claim by showing that in highly biased retrieval systems, up to 80% of the document could be removed from the collection without significantly degrading performance, given the TRC-style evaluation is used, which sets maximizing effectiveness as a goal. Retrievability is related to hubness in the sense that retrievability assumes that documents are retrieved with different grades of ease due to retrieval system bias. From the hubness point of view, document with high retrievability are hubs and those with low retrievability are anti-hubs (orphans). However, the approaches proposed in this chapter and the approach proposed by Azzopardi et al. are totally different: While we provide explanation and measures of hubness at lower level that is based on the impact of metric bias on distance structure in high dimensional space, the methods by Azzopardi et al. are achieved at a higher level by tackling the retrieval system from outside, namely by observing the documents retrieved upon sending queries. That is, given a document collection D , the set of all possible queries Q , then the retrievability of a document d with respect to the collection D is a measure of how likely d will be retrieved upon sending the queries $q \in Q$. Since queries are not equal frequently used, the retrievability measure considers a weight O_q for each query q . If K_{dq} is the rank of the document d in the ranked list upon the query q , then the retrievability measure is defined as

$$r(d) = \sum_{q \in Q} O_q \cdot f(K_{dq}, c) \quad (4.2)$$

where $f(K_{dq}, c)$ is a generalized utility function and c denotes the maximum rank the user is willing to proceed down the ranked list.

Bashir et al. [AV08] investigated retrievability in patent retrieval and proposed a method for improving retrievability using novel approaches of pseudo relevance feedback and query expansion. Patent retrieval, as a prior-art retrieval, is a recall-oriented application domain, where not missing a relevant patent is considered more important than retrieving only the set of relevant patents at top rank results. In this domain,

queries are typically patent applications being examined for novelty. Patent documents have complex structures and diverse technical contents, which leads to extremely large dictionaries. Experiments using retrievability measurement proposed in [AV08] indicate a large bias toward a subset of patents in state of the art retrieval systems, which results in that a large subset of patent documents being not retrievable. State of the art methods suggest increasing the coverage of prior-art queries by improving queries using Query Expansion (QE) with Pseudo Relevance Feedback (PRF). Bashir et al. proposed a novel method for a better identifications of PRF patents. In particular, they suggested that PRF patents are identified based on their similarity with query patents via selected terms (instead of all terms as in state-of-the-art methods). They identify relevant terms from query patents based on their proximity distribution with prior-art queries. Using this approach, an increase in the retrievability of individual patents is obtained, which indicates that this prior-art retrieval approach provides better opportunity for retrieving individual patents in search space.

Hubness reduction: Much work has been done for hubness reduction. Schnitzer et al. [SFSW11] use mutual proximity based on the shared neighborhood between points as a distance metric in the NN algorithm. Mutual proximity between two points is defined as the number of common neighbors between these two points. Using mutual proximity as a distance metric, a new distance model is created in which the nearest neighbor relations are corrected which leads to a hubness reduction. However, the fact that hubness has negative impact on the distance metrics used for finding the shared neighborhood (mutual proximity) makes this approach sub-optimal. Tomasev et al. [TM12] show that using the distance space induced from the shared neighborhood does not eliminate hubs and anti-hubs, and thus does not entirely overcome the hubness problem. They propose a new hubness-aware method for calculating the shared neighbors. This methods defines similarity that increases the class separation. This is done by assigning a weight to each point that minimizes the intra-class distances while maximizing the inter-class distances, which decreases the impact of hubness on the creation of the shared neighbor distance model. Local scaling method [ZmP05] is an approach for building nearest neighbor relations model using distance scaling based on local neighborhood information. Local scaling approaches attempt to scale the distance between two points depending on the distances their k -nearest neighbors. In particular, the scaled distance $LS(x_i, x_j)$ between the points x_i and x_j is given by

$$LS(x_i, x_j) = \exp\left(\frac{\|x_i, x_j\|^2}{D_k(x_i)D_k(x_j)}\right) \quad (4.3)$$

where $D_k(x)$ is the distance between the point x and its k -nearest neighbor. Local scaling has been used as a method for hubness reduction, since the resulting nearest neighbor relations seem to be more symmetrical than those obtained without using the local scaling. Schnitzer et al. [SFSW12] propose a new hubness reduction method that uses a new version of scaling, which they called global level. In contrast to local scaling that uses the neighborhood information to scale distance, global scaling method attempt to use global

neighborhood information. In this method, the global point distribution is considered to transform the distance between x_i and x_j into a probability that x_i is the closest point to x_j . After that the scaled distance is calculated by combining the joint probabilities of the distances from x_i to x_j and from x_j to x_i . Suzuki et al. [SHS⁺13] proposed an approach for reducing hubness using the centering method. The centering method is a transformation in which the points are transformed such that their origin is shifted to the data centroid. They also extended this method by using weighted centering, i.e. shifting the origin explicitly to the hub points instead of the centroid. They empirically showed that the weighted centering method leads to considerable improvement in combination with natural language data.

Contribution: In this chapter, motivated by the insight of Radovanovic et al. [RNI10] that hubs tend to be closer to the distribution mean than other points, we propose an explanation of the cause of hubness based on data sparsity and distance concentration in high-dimensional spaces. In contrast to the analysis by Radovanovic et al., which is based on particular distributions and thus lacks generality, our proposed explanation does not make any assumptions about the distribution or distance norm used. For example, due to the assumptions by Radovanovic et al., the cause of hubness they provided does not explain why the dimensionality alone is not sufficient for the emergence of hubness, nevertheless there are some distributions where hubness does not occur regardless of the dimensionality (more in Section 4.6.3). It also does not provide an explanation for the impact of the distance norms on the hubness behavior. Our proposed cause of hubness provides answers to all of these questions. Furthermore, our explanation decreases the discord between the results of Radovanovic et al. [RNI10] and the results of Low et al. [LBSN13] by linking both of them to the same cause (more in Section 4.6.2). Based on the theoretical results presented in this chapter, we also propose a hubness indicator that predicts the hubness in a given data set as well as the hubness contribution by a particular data point.

4.3 Distance Structure in High Dimensional Space

In this section, we provide the theoretical background required to explain the cause of hubness. This will be based on (i) the sparse distribution of points in high dimensional space (HDS), and (ii) the distance convergence as a direct result of the distance concentration. As an outcome of this section, we show that when d is sufficiently large, points lie almost exclusively at the vertices of a hypercube, a fact that will be used to explain hubness, the main contribution of this chapter, in Section 4.4.

The remainder of this section is organized as follows: In Section 4.3.1, the sparse distribution of points in a high dimensional space (HDS) is discussed. In Section 4.3.2, we discuss the convergence of the distance to mean (DTM) and the pairwise distance (PWD) in HDS. The convergence of distance between hypercube vertices is discussed in Section 4.3.3. Finally in Section 4.3.4, we present a model of distance structure between high dimensional points that will be used to explain hubness in Section 4.4.

4.3.1 Sparsity in High Dimensional Space

In high dimensional space, data becomes sparse [BB61] [ST83] [AHK01] [RS05]. There are more than one concept referred to as sparsity, e.g. in information retrieval and data mining, sparsity denotes the property of data where data points have zeros as values for the majority of their components. However, for this research, we are interested in another aspect of sparsity, namely that high dimensional points lie in different orthants (an orthant is subset of the hyperspace analogous to the quadrant in 2D and 3D space), and that most of the orthants are empty.

Let's formally define the notation of the orthant.

Definition 20. *Notation of the orthant.* Let $\mathbb{R}_q^d, 1 < q < 2^d$, denote the d -dimensional orthants in \mathbb{R}^d , that is, the sets $I_1^\pm \times I_2^\pm \times \dots \times I_d^\pm$, where $I_j^+ = [0, \infty)$, $I_j^- = (-\infty, 0)$, and I_j^\pm means either I_j^- or I_j^+ .

There are exactly 2^d orthants in a d -dimensional space, which increases very rapidly as d increases. For example in a $100 - d$ space, there are $2^{100} \approx 10^{30}$ orthants. In such a space, any real world data set is sparse. E.g. in a dataset consisting of one trillion 100-dimensional points, only a tiny fraction (less than a trillionth) of the orthants have points.

In particular, we show in Lemma 1 that in a high dimensional space, points are with very high probability in different orthants, which implies that an orthant generally contains at most one point.

Lemma 1. *Given a data set X of $n \ll 2^d$ points with i.i.d components, then each orthant contains with high probability at most one point.*

Proof. Let the points X be represented by a random vector $R = \{R_1, \dots, R_d\}$, distributed according to the probability densities ψ_i with the means μ_j . Then $x_i \in X$ is given by the d -tuple (R_{1i}, \dots, R_{di}) , where R_{ji} is the i th value of the random variable R_j . Assume without loss of generality that R_i are normalized to have their means at the origin, i.e. $\mu_j = 0$.

Let the function vector $\omega = \{\omega_1, \dots, \omega_d\}$ be the probabilities of a point having its coordinates in I^+ , i.e. $\omega_j(x_i) = P(x_{ij} \in I^+)$ is the probability that x_i has a positive j^{th} coordinate and $1 - \omega_j(x_i) = P(x_{ij} \in I^-)$ is the probability that x_i has a negative j^{th} coordinate. Note that in case that ψ_i is a symmetrical distribution about its mean, then $\omega_i = \frac{1}{2}$. The orthant, in which each point x_i resides, is determined by the outcome of the random vector R represented by functions ω , which corresponds to an experiment with d independent Bernoulli trials each having the probability ω_j . Let v be the event of a point located in a particular orthant, we are primarily interested in the distribution of k occurrences of the event v within n points, which meets the Poisson distribution given by $f(k, \lambda) = \frac{\lambda^k e^{-\lambda}}{k!}$, where $\lambda = n \cdot \prod_{j=1}^d \omega_j$ is the average occurrence of the event v within n points. This is a very low probability, given high dimensional data of the real world. \square

To imagine the probability of two points being in the same orthant, consider 10^{20} points in a 100 dimensional space. Assume a symmetrical distribution, i.e. $w = \frac{1}{2}$, this

means $\lambda = 10^{20} \prod_{j=1}^{100} \frac{1}{2} \approx 10^{-6}$, which means the probability of two points in the same orthant is $f(2, 10^{-6}) = \frac{10^{-12} e^{10^{-6}}}{2} \approx 10^{-11}$. Lemma 1 shows that in a high dimensional space, an orthant contains with very high probability at most one point.

4.3.2 Distance Convergence

In this section, we will show that when the dimensionality d is sufficiently high, then the DTM and PWD converge and the following limits hold for any finite point set regardless of its distribution:

$$\lim_{d \rightarrow \infty} \|x\| = c_m \cdot \eta(d) \quad (4.4)$$

$$\lim_{d \rightarrow \infty} \|x_i, x_j\| = c_p \cdot \eta(d) \quad (4.5)$$

where c_m and c_p are constants determined by the distribution and η is some function of d .

This convergence of distance (Equations 4.4 and 4.5) directly follows from the definition of the distance concentration, which is known as the tendency of pairwise distances to be equal as d reaches infinity. Distance concentration has been deeply studied, e.g. in [Dem94] [Fra08] [HAK00] [Kha04] [Koe00] [ST83] [ZM14], and in relation to distance norms in [AHK01] [BM13a] [FWVM07].

Since the distance concentration is deeply studied, and well analyzed under various settings, e.g. for different distributions and different distance norms, it is not in the focus of this chapter to prove distance concentration. Rather, we present in the next paragraphs, as illustrative examples, proofs for the convergence of DTM and PWD in the Euclidean space, that do not make any assumptions about the distribution of the point components.

Let X be a point set according to Definition 3 with $\|\cdot, \cdot\|_2$ as a distance norm. To show that the DTM of the points X converges when $d \rightarrow \infty$ without assumptions on their distribution, we show that all the points X are at the surface of a hypersphere centered at the mean of X , which implies that they are at the same distance from the mean [ZM14]. To this end, consider the smallest hypersphere, S_d , centered at the mean of X and containing all the points in X , i.e. it has a radius $r = \max(\|X\|)$. Now let us calculate the volume of the outer thin shell of the hypersphere, E , with a thickness ϵ . The aim is to show that when d is large, then the vast majority of the hypersphere volume is in the thin shell E , i.e. as $d \rightarrow \infty$, $\text{vol}(E) \rightarrow \text{vol}(S_d)$.

The volume of the hypersphere s_d is given by [Ken04] as:

$$\text{vol}(S_d) = \frac{2\pi^{\frac{d}{2}}}{\Gamma(\frac{d}{2} + 1)} r^d \quad (4.6)$$

where Γ is the Gamma function. The volume of the shell E of thickness ϵ is the difference between the volume of the hypersphere S_d with radius r and a smaller hypersphere \check{S}_d

with radius $r - \epsilon$, i.e.

$$\text{vol}(\check{S}_d) = \frac{2\pi^{\frac{d}{2}}}{\Gamma(\frac{d}{2} + 1)}(r - \epsilon)^d \quad (4.7)$$

The ratio between the volume of the shell and the volume of the hypersphere is

$$\frac{\text{vol}(E)}{\text{vol}(S_d)} = \frac{\text{vol}(S_d) - \text{vol}(\check{S}_d)}{\text{vol}(S_d)} = 1 - \left(\frac{r - \epsilon}{r}\right)^d \quad (4.8)$$

Now, the volume of the thin shell E is almost equal to the volume of the hypersphere S_d when d is sufficiently large because

$$\lim_{d \rightarrow \infty} \frac{\text{vol}(E)}{\text{vol}(S_d)} = \lim_{d \rightarrow \infty} 1 - \left(\frac{r - \epsilon}{r}\right)^d = 1 \quad (4.9)$$

which implies that all the points X are with high probability in the shell E , given d is sufficiently high. This also implies that the points are at the same distance from the mean, i.e. their DTM converges to r .

The proof above shows that the DTM distance converges to some value, but it does not specify this value, since it depends on the distribution, i.e. the radius r of the hypersphere is determined by the distribution of the points. For example if the components of the points are normally distributed, then the DTM is distributed according to the Chi distribution \mathcal{F} with d degrees of freedom, which directly follows from the definition of the chi distribution. Here the DTM converges to the expected value of \mathcal{F} , given by

$$E[\|X\|_2] = E[x_{\mathcal{F}}] = \sqrt{d} \quad (4.10)$$

Theorem 1, from Demartines [Dem94], provides a formula for the convergence value of the DTM for any distribution.

Theorem 1. Demartines [Dem94] adapted. *Let $X \in \mathbb{R}^d$ be a random vector with i.i.d. components that meet the distribution \mathcal{R} then,*

$$E[\|x\|] = \sqrt{ad - b} + O(1/d) \quad (4.11)$$

$$\text{var}(\|x\|) = b + O(1/\sqrt{d}) \quad (4.12)$$

where a and b are constants that depend only on the distribution \mathcal{R} , and do not depend on the dimensionality d .

According to Theorem 1, the DTM converges to a constant value that depends on d which confirms Equation 4.4.

Now we will show the convergence of the pairwise distance (PWD) between high dimensional points. To this end, we start from the results reached above, i.e. the hyperpoints are in the outer thin shell E of the hyperball. Let $x = (x_1, \dots, x_d)$ and $y = (y_1, \dots, y_d)$ be any two points in E . Without loss of generality rotate the space so

that one of the points, say x has zero value for its second up to d^{th} component, i.e. $x = (r, 0, 0, \dots, 0)$. This is possible since x is at the surface of the hyperball. Now, for the Euclidean distance between the two points

$$\begin{aligned} (\|x, y\|_2)^2 &= \sum_{i=1}^d (x_i - y_i)^2 = (r - y_1)^2 + \sum_{i=2}^d y_i^2 \\ &= r^2 - 2ry_1 + \sum_{i=1}^d y_i^2 \end{aligned} \quad (4.13)$$

Since the distance of the point y to the mean is r because y is also in the shell, we have $\sum_{i=1}^d y_i^2 = r^2$. By substituting in Equation 4.13, we get

$$\|x, y\|_2 = \sqrt{2r^2 - 2ry_1} = \sqrt{2}r\sqrt{1 - \frac{y_1}{r}} \quad (4.14)$$

Again r , the convergence value of the DTM (the radius of the hypersphere), is distribution dependent. Considering the convergence according to Theorem 1, Equation 4.11, and assuming a sufficiently large d so that the term $O(1/d)$ vanishes, we get by substituting in Equation 4.14

$$\|x, y\|_2 = \sqrt{2(ad - b)}\sqrt{1 - \frac{y_1}{\sqrt{ad - b}}} \quad (4.15)$$

$$\begin{aligned} \lim_{d \rightarrow \infty} \|x, y\|_2 &= \lim_{d \rightarrow \infty} \sqrt{2(ad - b)}\sqrt{1 - \frac{y_1}{\sqrt{ad - b}}} \\ &= \sqrt{2(ad - b)} = \sqrt{2}E[\|x\|_2] \end{aligned} \quad (4.16)$$

This implies that in high dimensional space, the Euclidean DTM and PWD have always the following relation

$$PWD = \sqrt{2}DTM \quad (4.17)$$

which consequently implies that the vectors representing any two points are orthogonal, since the two vectors together with the PWD form a right-angled triangle. Thus the points on the hypersphere are actually only on the vertices of the inscribed hypercube, which is presented in more detail in the next section.

4.3.3 Hypercube Vertices

There are different classes of distance in a hypercube, e.g. edges, face diagonals, cell diagonals, etc. In a cube of edge length g , e.g. Figure 4.5 (C), a point pair sharing an edge (e.g. P1 and P2) has distance g , a point pair sharing a small diagonal (e.g. P1 and

P3) has distance $\sqrt{2}g$, and a point pair sharing a large diagonal (e.g. P1 and P7) has a distance $\sqrt{3}g$. That means there are three classes of distance in a cube. In general, there are d classes of distance in a d -hypercube [Aic97]. Assuming the Euclidean distance, these distance classes are $\{\sqrt{t}g, t = 1, \dots, d\}$ where g is the edge length of the hypercube. That means, with increasing dimensionality, new distance classes arise. However, it is rather of more importance, how the PWD between vertices is distributed among these distance classes. In the following Lemma, we show that the distance between hypercube vertices converges.

Lemma 2. *Let C_d be a hypercube in the d -space with an edge length g , and let $V_d = \{0, g\}^d$ be the set of its vertices, then the pairwise distance between the vertices converges when d is sufficiently increased, that is*

$$\lim_{d \rightarrow \infty} \|x, y\|_p = g \cdot \left(\frac{d}{2}\right)^{\frac{1}{p}} \quad (4.18)$$

where $x, y \in V_d$.

Proof. Without loss of generality, assume C_d has one of its vertices at the origin and its edges along the axes. Then the vertices x and y are the hyperpoints $x = (x_1, x_2, \dots, x_d)$, and $y = (y_1, y_2, \dots, y_d)$, where $x_i, y_i \in \{0, g\}$. The distance between the two vertices is

$$\|x, y\|_p = \left(\sum_{i=1}^d |x_i - y_i|^p\right)^{\frac{1}{p}} = \left(\sum_{i=1}^d h_i^p\right)^{\frac{1}{p}} = g \left(\sum_{i=1}^d b_i\right)^{\frac{1}{p}} \quad (4.19)$$

where $h_i \in \{0, g\}$ and $b_i \in \{0, 1\}$.

$\sum_{i=1}^d b_i$ is a random variable drawn from a Binomial distribution (bin), since it is the number of successes in a sequence of Bernoulli experiments (b_i), each of them with two outcomes $\{0, 1\}$ distributed uniformly with probability $\frac{1}{2}$, i.e. $\sum_{i=1}^d b_i \sim \text{bin}(d, \frac{1}{2})$, and has an expectation value $d/2$, a standard deviation of $\sqrt{d/4}$, a minimum of 1 and a maximum of d .

By substituting the expectation value $E\left[\sum_{i=1}^d b_i\right] = d/2$ in Equation 4.19, it follows that

$$E[\|x, y\|_p] = g \cdot (d/2)^{1/p} \quad (4.20)$$

Now, since the expectation value and the convergence value are in general not necessarily equal, we only need to show that they are in this case. This follows from the decay of the ratio between the standard deviation σ_{bin} and the domain of the Binomial distribution ($max_{bin} - min_{bin}$) when d is sufficiently increased, i.e.

$$\lim_{d \rightarrow \infty} \frac{\sigma_{bin}}{max_{bin} - min_{bin}} = \lim_{d \rightarrow \infty} \frac{\sqrt{d/4}}{d-1} = 0 \quad (4.21)$$

Since the ratio converges to zero as d increase, it follows that the distance converges to the expectation value (Figure 4.1), i.e.

$$\lim_{d \rightarrow \infty} \|x, y\|_p = g \cdot (d/2)^{1/p} \quad (4.22)$$

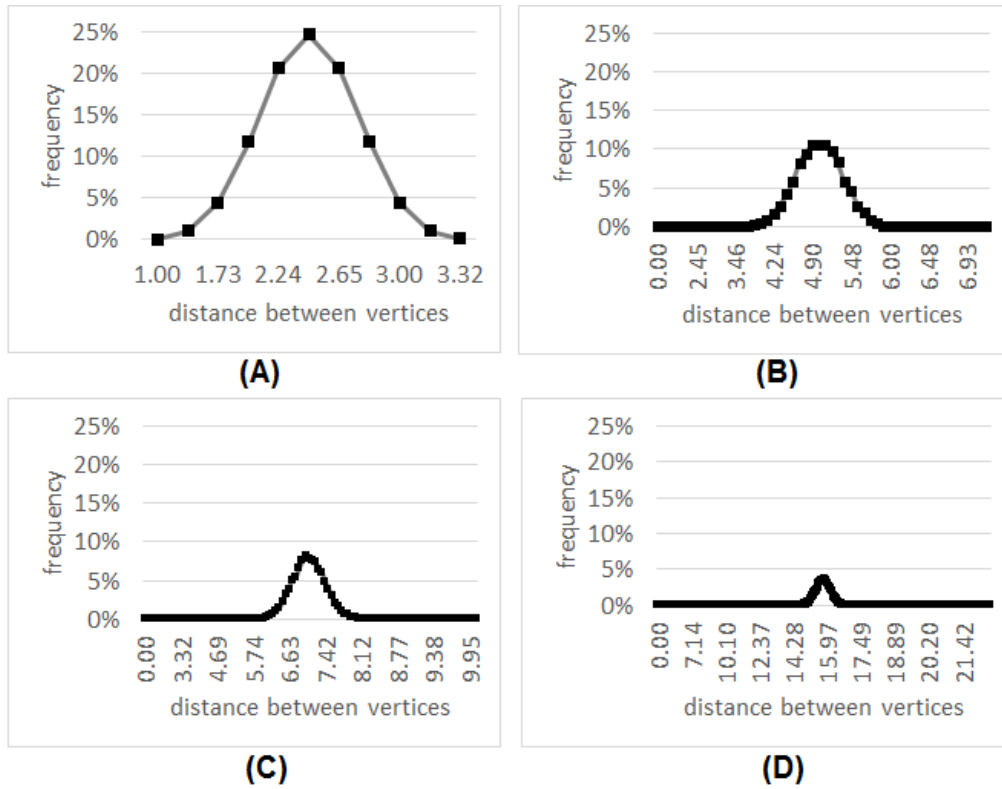


Figure 4.1: Distribution of the pairwise distance between vertices of unit hypercubes of different dimensionalities; (A) for a 10-hypercube, fully enumerated; (B), (C), and (D) for 50-hypercube, 100-hypercube, and 500-hypercube respectively, sampled using the Monte-Carlo method.

□

Lemma 2 implies that particular distances dominate the distance structure in HDS, which results in the fact that the distances between pairs of vertices are with high probability equal.

Figure 4.1 illustrates the distribution of distance between vertices of hypercubes of different dimensionalities (10, 50, 100, and 500), which empirically confirms the convergence according to Lemma 2, i.e. the interval containing most of the distances gets smaller as d increases.

4.3.4 Distance Structure of High Dimensional Points

This section shows that high dimensional points are at the vertices of a hypercube, or at least they can be modeled with hypercube vertices. In other words, we show that the distance structure of the hypercube vertices is equivalent to the distance structure of high dimensional points when d is sufficiently large.

Hypercube vertices have three relevant properties, each of which corresponds to a property of points in HDS: (i) Vertices have equal distance to the hypercube centroid; this property corresponds to the DTM of high dimensional points when it converges (Equation 4.4) which has been proven in Equations 4.6 to 4.9; (ii) vectors representing the vertices are orthogonal; this property corresponds to the relation between *DTM* and *PWD* (Equation 4.17), which implies that high dimensional points are orthogonal; and (iii) the vast majority of pairwise distances between hypercube vertices is equal according to Lemma 2; this property corresponds to the convergence of *PWD*, given by Theorem 1. Combined with the fact that there is at most one point in each orthant (Lemma 1), this leads to the fact that hypercube vertices provide a model for points in high dimensional space, which will be used in Section 4.4 to explain hubness.

Note that high dimensional points being on the vertices of a hypercube is not in conflict with the fact that they are on the surface of a hypersphere, as shown in Section 4.3.2, because they are actually at the vertices of the hypercube inscribed in the hypersphere of radius r , where r is the convergence value of the DTM.

Empirical demonstration of distance convergence using different norms: We illustrate the distance convergence and the distance structure described in this section using i.i.d. point sets for three different norms, namely the Euclidean distance (L_2), the sup norm (L_∞), and the cosine distance (*cos*).

For two points $x = \{x_1, \dots, x_d\}$ and $y = \{y_1, \dots, y_d\}$ The L_∞ norm is defined as

$$\|x, y\|_\infty = \max_{i=1}^d (|x_i - y_i|) \quad (4.23)$$

and the cosine distance is defined as

$$\|x, y\|_{\cos} = 1 - \frac{\vec{x} \cdot \vec{y}}{|\vec{x}| \cdot |\vec{y}|} = 1 - \frac{\sum_{i=1}^d x_i y_i}{\sqrt{\sum_{i=1}^d x_i^2} \sqrt{\sum_{i=1}^d y_i^2}} \quad (4.24)$$

To this end, we generated four i.i.d normally distributed point sets, each consisting of 500 points, with the dimensionalities 2, 10, 100, and 10000. Using each of the three norms, the DTM was calculated for each point. Only 500 pairwise distances (PWD) were sampled randomly and calculated using each of the norms.

Figure 4.2 shows the results for the Euclidean distance. While the distances in low dimensionalities (2 and 10) spread to fill most of the space, they tend to converge in higher dimensionalities (100 and 10000). Note that in each case the expectation value (the convergence value) of the *DTM* is \sqrt{d} and the expectation value of the *PWD* is $\sqrt{2d}$, which is in conformance with the theoretical results in Equations 4.10 and 4.17 respectively. This implies that any two points form with the mean a right angle triangle, which also implies that all point vectors are orthogonal to each other.

Figure 4.3 illustrates the convergence of L_∞ in high dimensional space. A distance convergence of *DTM* and *PWD* is observed as in the case of Euclidean distance, but

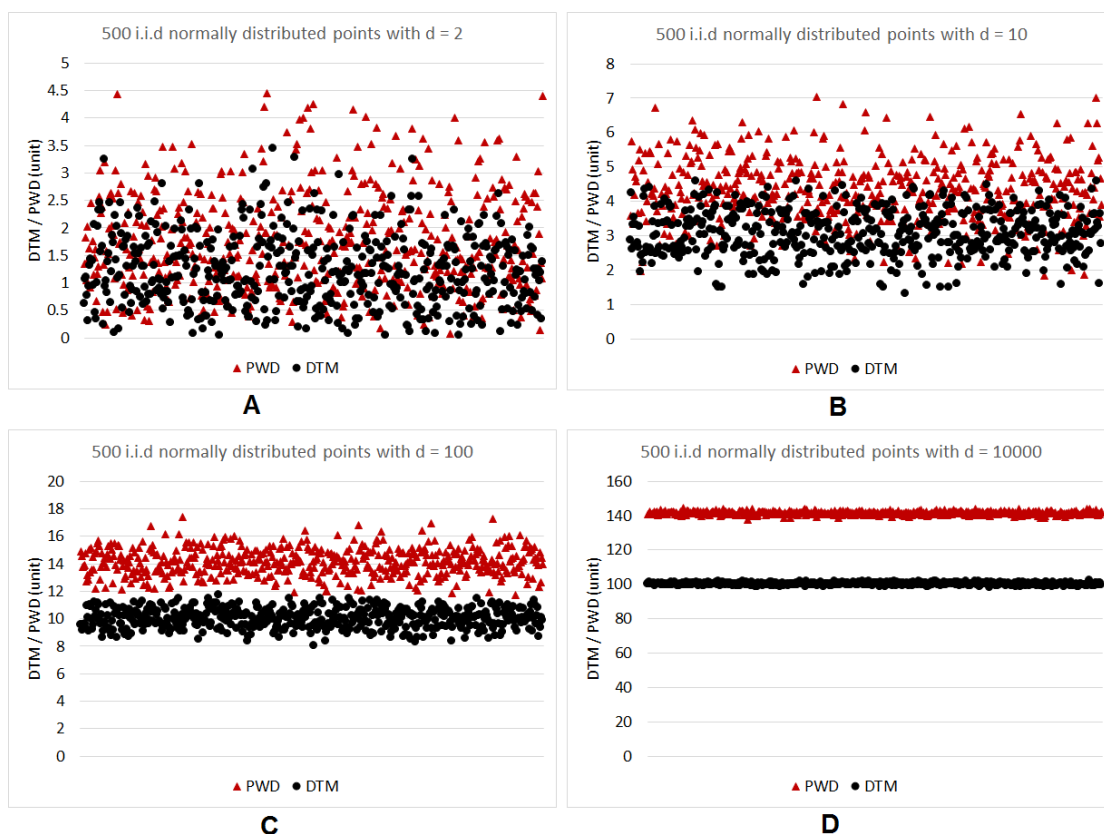


Figure 4.2: Convergence of the DTM and the PWD with increasing dimensionality. Four random point sets were drawn from a normal distribution with dimensionalities 2, 10, 100, and 10000 in (A), (B), (C), and (D) respectively. Each point set consists of 500 points. For each point set all distances to mean are calculated, and 500 pairwise distances were sampled randomly. The DTM converges to \sqrt{d} and the PWD converges to $\sqrt{2d}$, which is in conformance with Equations 4.10 and 4.17 respectively.

with two differences, namely (i) that the convergences values are different. This is in conformance with Equations 4.4 and 4.5 that state that the convergent values depend on the distribution and dimensionality. (ii) The convergence is slower than with the Euclidean distance, which is in agreement with the observation reported in literature [FWVM07] that L_∞ is less concentrated in high dimensional space than the Euclidean distance.

Figure 4.4 illustrates the convergence behavior of the cosine distance norm (COS) in high dimensional space. Note that the mean with respect to the COS is the vector (point) that minimizes the cosine distances (angles) to all other points. Such vector does not necessarily represent one of the points in the point set. However, for this experiment, we use as an approximation of the mean one of the points in the set, namely the one nearest to the mean, i.e. the point in the set that minimizes the cosine distances to all

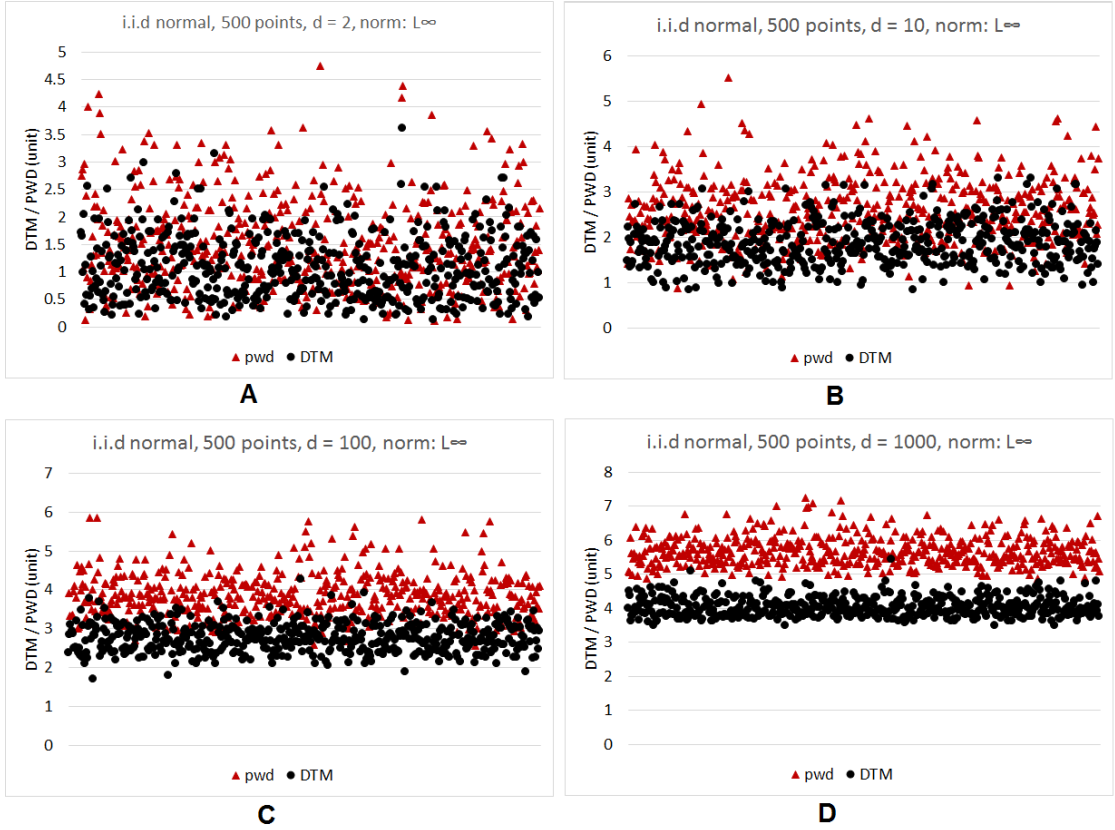


Figure 4.3: Convergence of the DTM and the PWD with L_∞ norm. The same i.i.d. random point sets as in Figure 4.2 have been used.

other points. This can be identified by calculating for each point the sum of pairwise distances to the other points and then selecting the one with the lowest sum.

Both DTM and PWD converge to the same value, namely one. This is an interesting observation that confirms the distance structure we have presented in this section as follows:

The convergence value of the COS PWD (which is observed to be one) gives information about the convergence value of the angle between vectors representing points in high dimensional space. In particular, the convergence of COS PWD to one implies that the angle converges to 90° , which confirms that all vectors are orthogonal, i.e. also with COS norm, points are located at the vertices of a hypercube. Formally, this follows from

$$\lim_{d \rightarrow \infty} \|x, y\|_{COS} = 1 \implies \lim_{d \rightarrow \infty} \cos(\widehat{xy}) = 0 \implies \lim_{d \rightarrow \infty} \widehat{xy} = 90^\circ \quad (4.25)$$

where \widehat{xy} denotes the angle between the vectors representing the point x and y . Note that cosine distance is $1 - \text{angle}$ according to Equation 4.24.

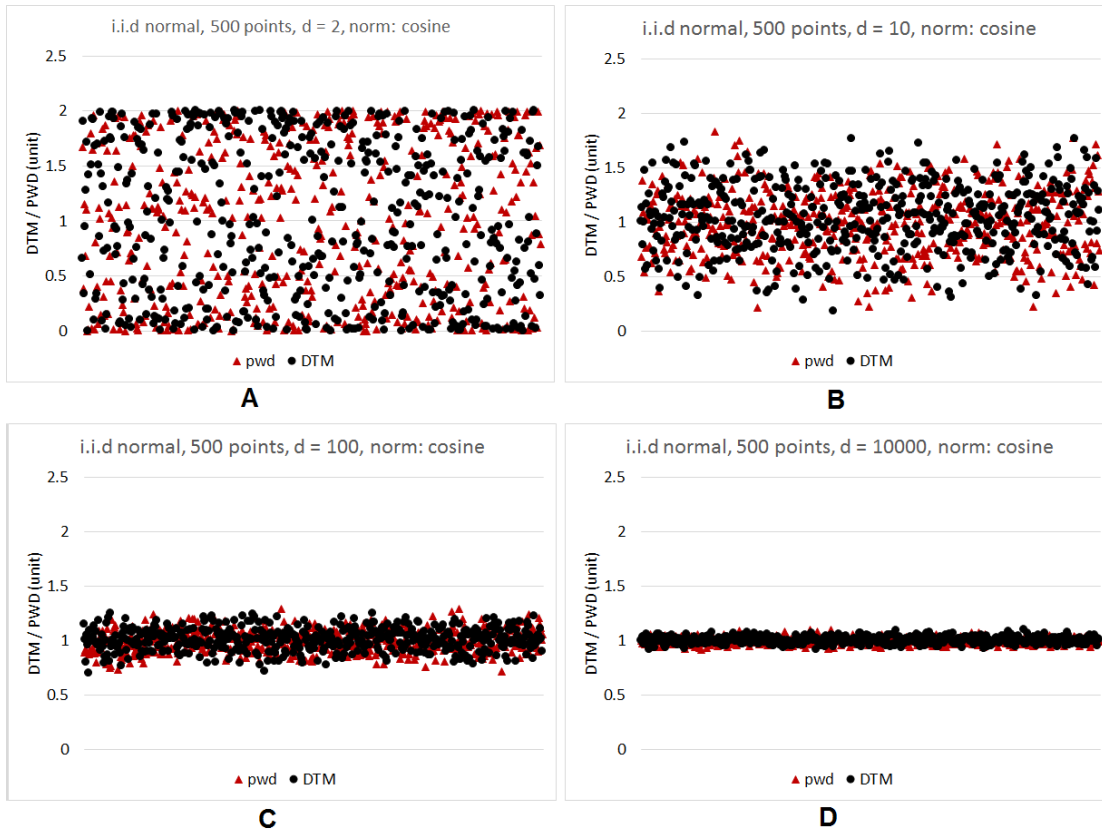


Figure 4.4: Convergence of the DTM and the PWD with the cosine norm (COS). The same i.i.d. random point sets as in Figure 4.2 have been used. Note that the mean in terms of the cosine distance is not the origin, but rather the vector (point) that minimizes the angles to all other points.

4.4 Cause of Hubness

In Section 4.3 we showed that high dimensional i.i.d points are concentrated at the vertices of a hypercube due to the sparsity combined with the distance concentration. Based on this, we provide in this section an explanation of the bias in the NN relations in a high dimensional space and thus the cause of hubness. Before we provide a general explanation of hubness, we give an example in lower dimensionality on the cause of hubness.

Assume that a 2-hypercube (square) has 4 points located exactly at its vertices as shown in Figure 4.5 (A). Because the distances between vertices are equal, points could be mutually nearest neighbors to each other so that the NN relations are symmetrical, and thus the hubness is low. Now suppose that the point P1 is deviated toward the mean as in Figure 4.5 (B). In this case, the NN relations change so that P1 becomes the nearest neighbor of P2 and P4. In other words P1 becomes a hub. This happens regardless of

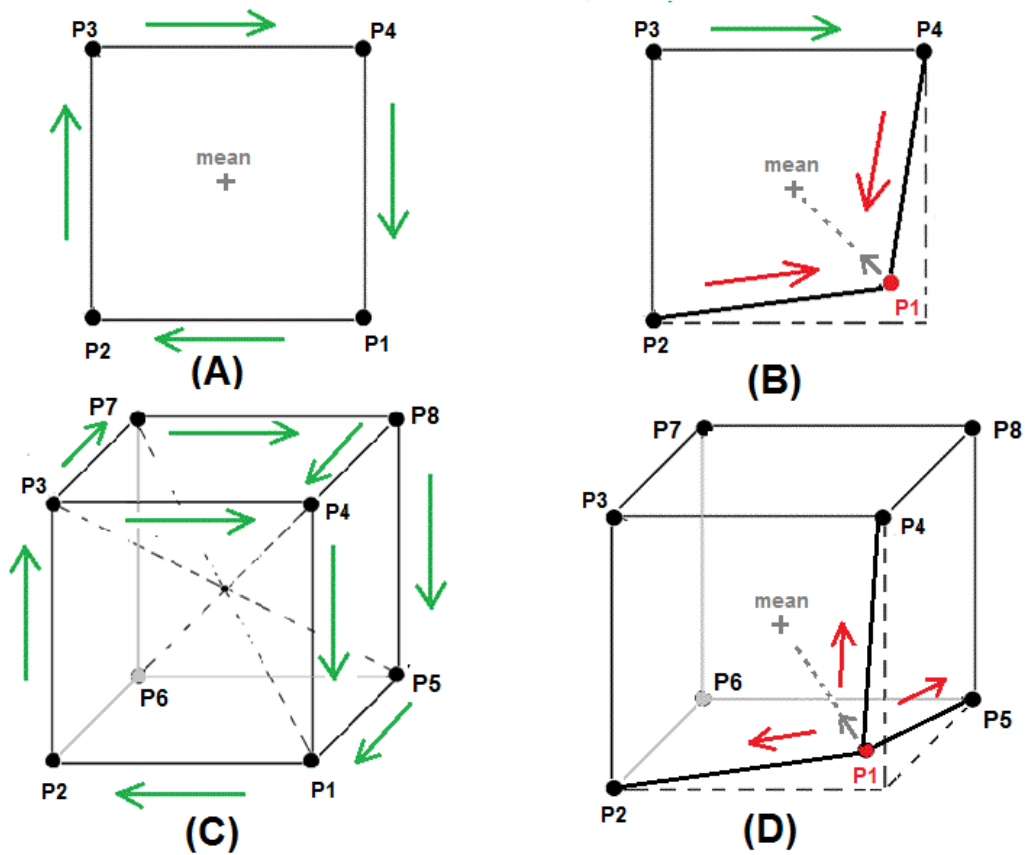


Figure 4.5: In (A) there are symmetrical NN relations. In (B), the point P1 is deviated toward the mean of the square, which causes changes in the NN relations of the neighboring points, so that P1 becomes the nearest neighbor of P2 and P4, which makes P1 a hub. In (C) there are symmetrical NN relations between the cube vertices, since vertices have equal distances between them. In (D), P1 is deviated toward the mean. This causes P1 to become the NN neighbor of P2, P4, and P5. The number of points for which the NN relations change increases with dimensionality.

how small this deviation is, given the points are exactly at the vertices. Analogously, in a 3-hypercube (cube) with 8 points located exactly at the vertices, as shown in Figure 4.5 (C), suppose that one point, say P1, has been deviated toward the mean, as shown in (D). The same will happen, i.e. the NN relations change, but with one difference, namely that more points are affected, in this case P1 becomes the nearest neighbor of P2, P4, and P5, which results in P1 becoming a hub. In higher dimensionalities, the deviated point becomes the nearest neighbor of more points. Actually, it becomes the nearest neighbor of all points at vertices connected with its vertex over one edge. Considering the fact that each vertex is connected with exactly d other vertices over one edge, this implies that the number of points affected by the deviation increases with the dimensionality.

The example described above is too simple because of the following:

- (I) It assumes that all points are exactly at the vertices of the hypercube and one point deviates from the vertex toward the mean, but in practice, all points have some deviation from the vertices depending on the distribution, the dimensionality, and the distance norm used. How does this fact affect hubness emergence?
- (II) In addition to the deviation of a point toward the mean, there are other types of deviation, e.g. away from the mean or perpendicular to this direction or a composition of multiple deviations. What is the impact of each type of deviation?
- (III) The example assumes that all vertices are occupied by points, but in practice only a very small fraction of them are (see Section 4.3.1). How can the explanation of hubness hold regardless of this fact?

Actually, the example above illustrates a point distribution that provides optimal conditions for the emergence of hubness. We will call this distribution the hubness-optimal distribution. The hubness in a data set is decided by how much the distribution of the data set is similar to hubness-optimal distribution. This similarity is defined formally in the next sections.

The remainder of this section is organized as follows. In Section 4.4.1, we define three types of deviation from the hypercube vertices that points can have. We discuss and formally define their impact on hubness. In Section 4.4.2, we discuss two categories of factors having impact on hubness, namely factors characterizing the general tendency to hubness and factors specifying the hubness caused by outliers. With these two sections we answer questions (I) and (II). In Section 4.4.3, we discuss the relation between the three types of deviation from the hypercube vertices and the statistics of the DTM and PWD. Finally in Section 4.4.4, we discuss the sparsity in high dimensional space and explain why sparsity does not violate the hubness-optimal distribution; thereby we answer question (III).

4.4.1 Deviation Types

In this section we discuss three types of deviation from the vertices of the hypercube that the points can have. We show that only one type of deviation promotes hubness while the other two types decrease hubness. The deviation types are illustrated in Figure 4.6 (A). These are: the δ_{r-} deviation, in which a point deviates in the radial direction toward the mean; the δ_{r+} deviation, in which a point deviates from a vertex in the radial direction away from the mean; and the δ_p deviation, which denotes deviations, in which a point deviates from a vertex in a direction perpendicular to the radial directions. Obviously, an arbitrary point deviation from a vertex is a composition of these three deviation types and can be resolved to them as components.

Let δ be any arbitrary deviation from a vertex of the hypercube in any direction. Obviously, δ is a d -dimensional vector, thus it has d components. If one of these components is along the radial direction (the direction from the vertex to the hypercube

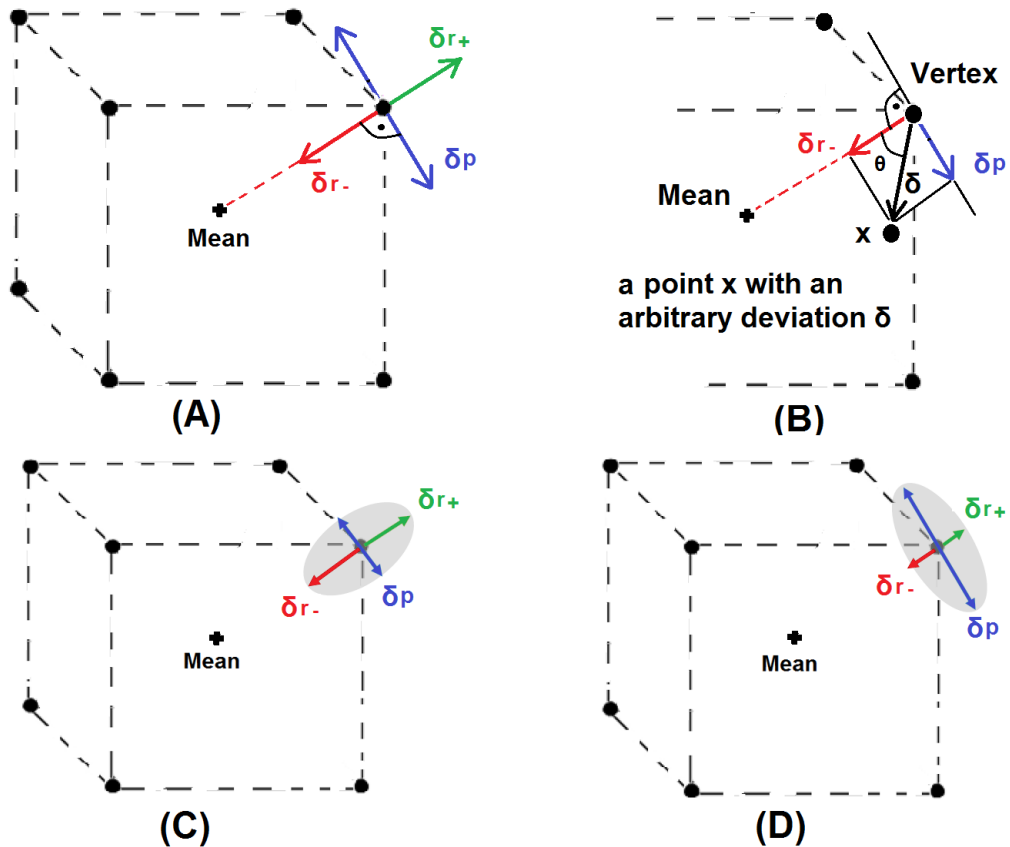


Figure 4.6: Deviation types: (A) The three types of deviation from the hypercube vertex that a point can have. δ_{r-} deviation is when a point leaves a vertex towards the mean (hypercube center), which we call the radial inside direction. δ_{r+} deviation is when a point deviates away from the mean, which we call the radial outside direction. δ_p deviation is any direction that is perpendicular to the radial direction, which we call the perpendicular hyperplane. (B) An arbitrary deviation δ of a point x can be resolved into its components as one or more of the three deviation types. (C) and (D) Two different distributions with a large $\tilde{\gamma}$ value (C), small $\tilde{\gamma}$ value (D). The shaded area illustrates the space where the point exists with high probability relative to the corresponding vertex.

centroid or vice versa), then, assuming a coordinate system for which the radial direction is an axis, there are other $d - 1$ components, forming a $(d - 1)$ -hyperplane orthogonal to the radial direction, which we will call the perpendicular hyperplane. Figure 4.6 (B) illustrates the decomposition of an arbitrary deviation δ of a point x in the 2-space into its components in the radial direction (δ_{r-}) and in the perpendicular plane (δ_p).

We formally define the three types of deviations as follows.

Definition 21. Let X be a data set according to Definition 3, and $x \in X$ be a d -dimensional data point with an expected position at V , the hypercube vertex of the

hypercube C_d with the centroid M . Let x be at position \acute{V} , having a deviation δ from its expected position in any arbitrary direction in the d -space, so that V is the nearest vertex to \acute{V} . Also let θ be the angle at the vertex V formed by M and \acute{V} , then

- The radial direction denotes the direction from M to V , or from V to M .
- The radial inside direction is the direction from V to M
- The radial outside direction is the direction from M to V
- The perpendicular hyperplane denotes the $(d - 1)$ -hyperplane orthogonal to the radial direction.
- The corresponding vertex V of a point $x = (x_1, \dots, x_d)$ is the nearest vertex to x , given by

$$V = (v_1, \dots, v_d) \text{ where } \begin{cases} v_i = 1, & x_i \geq 0 \\ v_i = 0, & \text{otherwise} \end{cases} \quad (4.26)$$

- The deviation in the radial direction δ_r is given by

$$\delta_r = \delta \cdot \cos(\theta) \quad (4.27)$$

Now the three deviation types are defined as follows:

1. δ_{r-} is any deviation component in the radial inside direction, given by

$$\delta_{r-} = \delta \cdot \cos(\theta), \text{ where } \theta < \pi/2 \quad (4.28)$$

2. δ_{r+} is any deviation component in the radial outside direction, given by

$$\delta_{r+} = \delta \cdot \cos(\theta), \text{ where } \theta \geq \pi/2 \quad (4.29)$$

3. δ_p is any deviation in the perpendicular hyperplane, given by

$$\delta_p = \delta \sqrt{1 - \cos(\theta)^2} \quad (4.30)$$

In this section, we discuss and formally define the impact of each of the three types of deviation in Definition 21 on the similarity of the point distribution to the hubness-optimal distributions, and consequently on hubness.

In general, a point distribution is similar to the hubness-optimal distribution if the minority of the points are deviated from their corresponding vertices toward the mean, while the majority of the points are concentrated at their corresponding vertices, i.e. may deviate from the corresponding vertices in any direction, but remain within a radius that is smaller than the deviation of the minority toward the mean. This means that the

degree of similarity can be roughly estimated as the ratio between the extent of deviation toward the mean and the extent of deviation otherwise. This rough estimation is $\tilde{\gamma}$ (a more accurate estimation is provided in Section 4.4.2), given by.

$$\tilde{\gamma} = \frac{\text{var}(\delta_r)}{\text{var}(\delta_p)} \quad (4.31)$$

To understand why the ratio between $\text{var}(\delta_r)$ and $\text{var}(\delta_p)$ gives a rough estimation of the similarity to the hubness-optimal distribution (where as we will see, only δ_{r-} has impact), consider the following two cases. Case 1, $\tilde{\gamma}$ is large, i.e. $\text{var}(\delta_p)$ is small compared with $\text{var}(\delta_r)$. Since δ_p is in the $(d-1)$ -hyperplane, this implies that the deviation of $d-1$ components is on average smaller than the deviation in the radial direction. Case 2, $\tilde{\gamma}$ is small, i.e. the opposite of Case 1, which implies that the deviation in $d-1$ components is on average larger than the deviation in the radial direction. It is clear that Case 1 is more similar to the hubness-optimal distribution than Case 2, since in Case 1, the points are more concentrated at the corresponding vertices than in Case 2. Figure 4.6 (C) and (D) illustrate the $\tilde{\gamma}$ estimation of the general tendency to hubness in a data set. The shaded area around the vertex illustrates the space where the point is likely to exist. In (C), $\text{var}(\delta_r)$ is large compared with $\text{var}(\delta_p)$, which results in a high tendency to hubness. In (D) is the opposite, and consequently the hubness emergence is unlikely. Note that this holds for all vertices, although illustrated for only one.

In the remainder of this section, we explain why δ_p and δ_{r+} deviations do not promote hubness like δ_{r-} , on the contrary, they even decrease hubness.

Let us consider the cases illustrated in Figure 4.7: In (A), the NN relations are symmetrical because points are exactly at the vertices of a cube. In (B), deviating the point P4 along an edge of the cube has a limited effect, since only P3 becomes nearer to P4 and all other points remain unaffected. In (C), deviating the point P4 on a face diagonal has a limited effect, because only P3 and P8 become nearer to P4. The effect is more than in (B), but however less than when deviating the point directly toward the mean. Note that θ (the angle between the deviation vector δ and the hypercube diameter from the corresponding vertex to the centroid) is inversely proportional to the number points affected by the deviation, i.e. θ is smaller in (C) than in (B), and in the direct deviation toward the mean is the smallest ($\theta = 0$). Imagine the case in a high dimensional space, where the hypercube vertex is connected to a large number of other vertices over one edge and there are many faces. The number of points becoming nearer to a deviated point depends on which face the point is deviated along, and consequently on the angle θ . This number is maximized when the deviation is directly in the radial inside direction, and minimized when the deviation is completely in the perpendicular hyperplane, otherwise, it depends on the angle θ , a fact that is conformant with the $\tilde{\gamma}$ estimation.

In addition to the fact that δ_p deviations do not promote hubness, as has been illustrated, they even compensate the hubness that may be caused when some point is deviated toward the mean. To understand this compensation effect, recall the illustrative example in Figure 4.5 (B) and (D), and imagine that the points are not at the vertices,

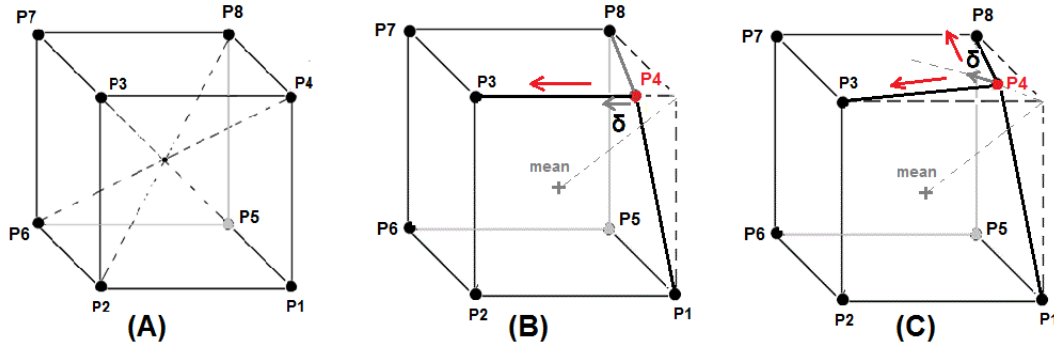


Figure 4.7: In (A), points are exactly at the vertices of a hypercube. In (B), P4 is deviated along an edge of the hypercube affecting only one point regarding the NN relation, namely P3. In (C), the point P4 is deviated along a face diagonal, thereby affecting two points, namely P3 and P8.

but instead have deviations in the perpendicular direction. In this case, the probability of the points changing their NN relations as a result of the deviation of P4, i.e. having P1 as NN, is reduced. This is what we call the compensation effect, which justifies including δ_p as a denominator of $\tilde{\gamma}$, the estimator of tendency to hubness, Equation 4.31.

Now let us consider the δ_{r+} deviation, i.e. away from the mean. Such deviations result in the deviated points becoming farther from the rest of the points, i.e. isolated, thereby becoming anti-hubs. Furthermore, such deviations have obviously a compensation effect due to the same reasons that δ_p deviations have, as mentioned above, i.e. they contribute in enlarging the radius around the vertices, where points exist with high probability, which reduces the similarity to the hubness-optimal distribution.

4.4.2 General Tendency vs. Outlier Specific Hubness

Equation 4.31 provides a rough estimation of the general tendency to hubness of a data set. In this section, we provide a more accurate estimation of the general tendency to hubness. Furthermore, we provide an estimation of the point specific hubness, which depends on (i) the general tendency to hubness in the data set and (ii) the position of the point.

General Tendency to Hubness

The statement of Equation 4.31 is that the less the deviation in the perpendicular hyperplane (δ_p), compared with the deviation in the radial direction (δ_r), the more similar the distribution is to the hubness-optimal distribution, and thus the tendency to hubness is higher. However, $\tilde{\gamma}$ is a rough estimate, because it considers only one statistic of the deviation toward the mean, namely $var(\delta_r)$, which measures the extent of the δ_r deviation on average, but does not take into consideration its distribution. To understand this, consider the distributions of δ_r as shown in Figure 4.8. $\tilde{\gamma}$ does not

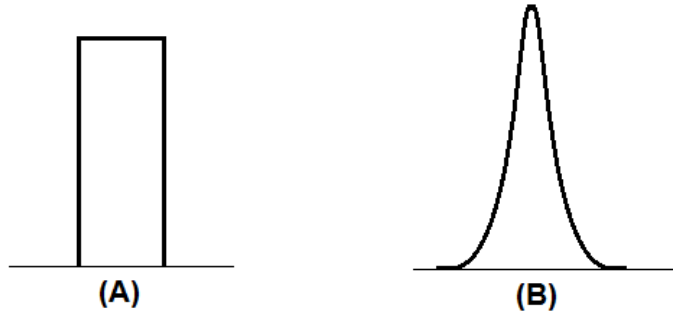


Figure 4.8: Distribution of δ_r and the similarity to the hubness-optimal distribution. Although both of the distributions have the same variance, (B) has a higher kurtosis, which makes it more similar to the hubness-optimal distribution, since the mass is concentrated at the mean and few points deviate from the mean (tails).

differentiate between these two distributions, given that both of them have the same variance. Actually, there is a difference between them regarding the tendency to hubness. The distribution in (B) has a higher kurtosis than in (A). A distribution with high kurtosis is characterized by a sharp peak and tails, which results in the mass concentrating in a small range around the mean, and the minority being in the tails. Such a distribution is more similar to the hubness-optimal distribution than in (A) because the majority of the points are concentrated at the vertices (the mass) and only few points are deviated toward the mean (the left tail). Kurtosis is a characteristic that has impact on the general tendency to hubness, which is a justification to include it in the definition of $\tilde{\gamma}$. The kurtosis of a random variable Y is defined by

$$kurt(Y) = \frac{E[(Y - E[Y])^4]}{(E[(Y - E[Y])^2])^2} \quad (4.32)$$

However, since hubs are points deviating toward the mean, i.e. points corresponding to the left tail of the distribution of δ_r , and kurtosis is in contrast defined for both sides, we define the function $kurt_L(\delta_r)$ that considers only the kurtosis of the left part of the distribution, i.e. the part having a direct impact on hubness, which enables a more accurate estimation of hubness tendency in the dataset X , that is

$$kurt_L(\delta_r) = \frac{E[f(\delta_r)^4]}{(E[f(\delta_r)^2])^2} \text{ where} \quad (4.33)$$

$$f(\delta_r) = \begin{cases} \delta_r - E[\delta_r] & \text{if } \delta_r < E[\delta_r] \\ 0 & \text{otherwise} \end{cases}$$

Now, we extend the hubness tendency estimation $\tilde{\gamma}$ to a more accurate form $\acute{\gamma}$, given by

$$\acute{\gamma} = kurt_L(\delta_r) \frac{var(\delta_r)}{var(\delta_p)} \quad (4.34)$$

Outlier Specific Hubness

The tendency to hubness γ (Equation 4.34) is a characteristic of a data set, since it is based on the distributions of δ_r and δ_p . However, most of the hubness in a data set is normally caused by few points, namely those outliers lying nearer to the mean than the other points, i.e. points corresponding to the left tail of the δ_r distribution. Such outliers may significantly increase hubness, which results in the hubness deviating from the typical value for the underlying distribution.

It is useful to have a way to identify those outliers in order to remove them, or to weight distance metrics to penalize them in order to reduce hubness. In the following, we propose an indicator for hubness caused by a particular point.

Let X be a point set according to Definition 3. Let $x \in X$ have a deviation $\delta_{r-}(x)$ from its expected position (the corresponding vertex). The hubness caused by x is proportional to the deviation of x toward the mean in relation to the average deviation of the other points, i.e. relative to the standard deviation of the radial deviations. We define $\hat{\nu}(x)$ as an estimate of the relative hubness caused by the point x , which is given by

$$\hat{\nu}(x) = \frac{E[\delta_r] - \delta_r(x)}{\sqrt{\text{var}(\delta_r)}} \quad (4.35)$$

Note that $\hat{\nu}(x)$ is a measure of the relative hubness contribution of a particular point and does not give information about the absolute amount of hubness. That is $\hat{\nu}(x)$ specifies the order of the points in terms of their hubness contribution, i.e. a point with a higher $\hat{\nu}$ value is expected to have a higher hubness contribution than another point in the same point set with a lower $\hat{\nu}$ value. This means that comparing points in different point sets using $\hat{\nu}$ does not make sense.

Also note that when δ_r in Equation 4.35 is a deviation toward the mean, i.e. δ_{r-} , it has a negative value, which results in a positive $\hat{\nu}$ value denoting a hub, and when δ_r is a deviation away from the mean (δ_{r+}), it has a positive value, which results in a negative $\hat{\nu}$ denoting an anti-hub.

In Section 4.6.1, we provide analysis that illustrates by means of i.i.d. random point sets the relation between tendency to hubness and the outlier specific hubness.

4.4.3 Relation to DTM and PWD

In this section, we show that $\text{var}(\delta)$, $\text{var}(\delta_r)$, and $\text{var}(\delta_p)$ can be estimated using statistics of two common distances, namely the distance to mean (DTM) and the pairwise distance (PWD).

Let X be a point set according to Definition 3. Let $\rho = \|x\|$, $x \in X$ be the DTM with a minimum $\min(\rho)$, an expected distance $E[\rho]$, and a variance $\text{var}(\rho)$. Furthermore, let $\eta = \|x, y\|$, $x, y \in X$ be the pairwise distances (PWD) in X . Now, δ_r is straightforwardly related to ρ (the DTM) because if a point deviates toward the mean by δ_r , then the DTM is reduced by the same amount, and vice versa. Since the expected value of the

DTM $E[\rho]$ corresponds to the distance of the hypercube vertex to the mean, it follows that

$$\delta_r = \rho - E[\rho] \quad (4.36)$$

Consequently, because $E[\rho]$ is constant for a given data set, this means that the distribution of ρ is the same as of δ_r , but shifted a distance $E[\rho]$ to the right. It follows that ρ has the same variance and the same kurtosis as δ_r , i.e. $var(\rho) = var(\delta_r)$, $kurt_L(\rho) = kurt_L(\delta_r)$.

The extent of $var(\delta_p)$ can be sufficiently estimated by the variance of η because $var(\delta_p)$ is the variance of the deviations from the vertices in $d - 1$ dimensions (most of the directions), and $var(\eta)$ is the variance of the PWD. As an illustration, imagine that the points are exactly at the vertices, then $var(\eta) = 0$ and also $var(\delta_p) = 0$. Now, starting from this setting, any deviation of a point from the corresponding vertex in the $(d - 1)$ -space causes an increase in $var(\delta_p)$ and consequently in $var(\eta)$.

By substituting in Equations 4.34 and 4.35, we define new variants of $\acute{\gamma}$ and $\acute{\vartheta}$, namely γ and ϑ , using only statistics of ρ and η , that is

$$\gamma = kurt_L(\rho) \frac{var(\rho)}{var(\eta)} \quad (4.37)$$

$$\vartheta(x) = \frac{E[\rho] - \|x\|}{\sqrt{var(\rho)}} \quad (4.38)$$

Stability: Estimating $var(\delta_r)$ and $var(\delta_p)$ using the DTM and PWD is a more stable method than the direct calculation using Equations 4.27 to 4.30, because the direct calculation assumes that the expected positions of the points are exactly at the vertices. This assumption is true as a convergence value only when the dimensionality is sufficiently high. On the contrary, using the DTM and PWD statistics to estimate the deviations does not make this assumption, since the actual expected positions are implicitly used, which results in a more stable estimation also with relatively lower dimensionalities.

Another advantage is saving the effort of implementing algorithms for the explicit calculation the point deviations when statistics about the DTM and PWD are already available, or when the distribution of the underlying data is already known so that these values can be easily estimated.

Sampling the PWD: It is important to note here that only an estimation of the PWD variance is sufficient, and that the exact PWD variance is not necessarily required, since we are interested only in the extent of the deviation. The estimation can be achieved by sampling n point pairs randomly, where n is the point set size. This avoids the complexity of calculating the exact $var(\eta)$, and thus enables calculation in linear time in terms of the point set size. Figure 4.9 empirically demonstrate the convergence of the sampled PWD variance of a random point set consisting of 500 i.i.d. normally distributed points. It shows that the variance quickly converges to the exact value. Already after $|X|$ out

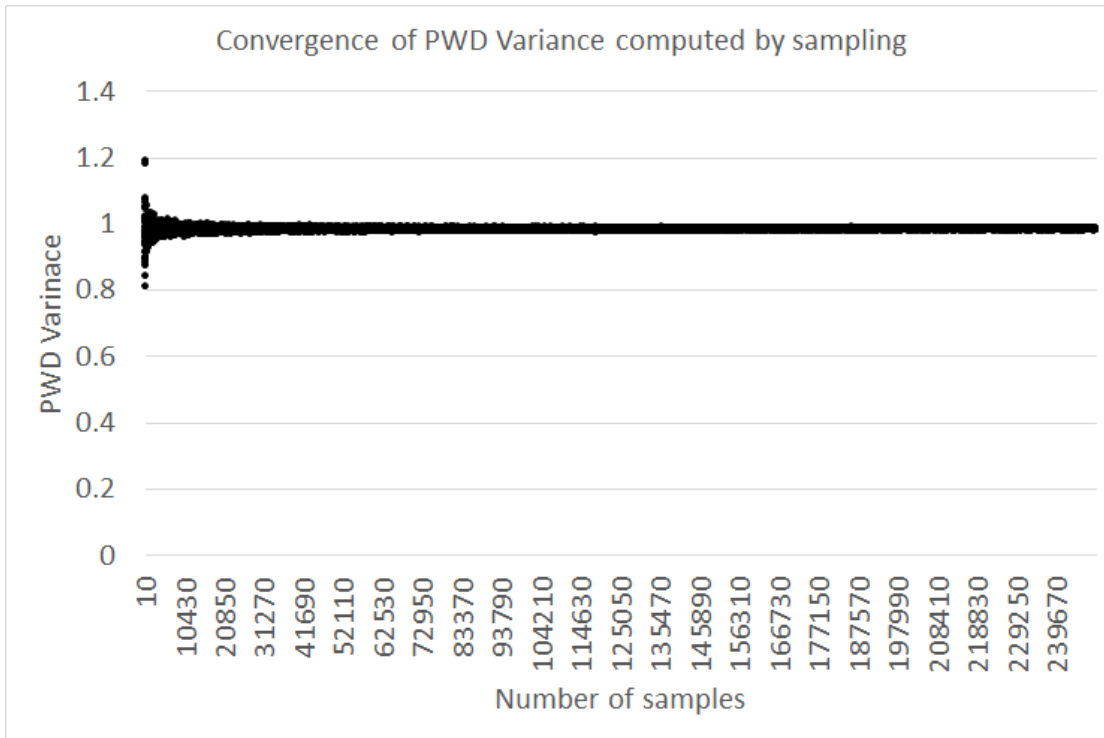


Figure 4.9: The convergence of the sampled PWD variance of a random point set consisting of 500 i.i.d. normally distributed points. Each data point represents the variance resulting from sampling a number of point pairs (the x-axis). The figure shows that the variance quickly converges to the exact value.

$|X|^2$ samples, it reaches a good estimation that is sufficient for the hubness indicator. In Section 4.11, we show in Figure 4.11 empirical results of the performance of a hubness indicator based on sampling the PWD as described above.

4.4.4 Distribution of Points among the Vertices

In Section 4.3.1, we showed that high dimensional data is sparse, and only a tiny fraction of the 2^d orthants have points, where the rest are empty. In Section 4.3.4, we suggested a distance structure in high dimensional space that states that points are concentrated at a hypercube vertices, which is the basis of the hubness explanation. Analogously, data sparsity implies also that only a tiny fraction of these 2^d hypercube vertices are occupied by points and the rest are empty. At first sight, this fact seems to violate the hubness-optimal distribution described in Section 4.4 (the core idea of the proposed hubness explanation), which states that when a point deviates from a vertex toward the mean, it becomes a hub as a result of becoming nearer to the points at the neighboring vertices. How can this hold, given that the neighboring vertices are most likely empty,

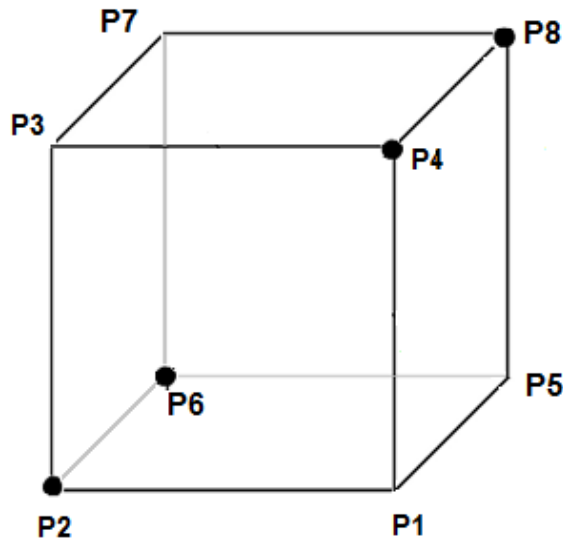


Figure 4.10: Illustration in a low dimensionality of how a subset of hypercube vertices can build another hypercube of lower dimensionality. The vertices P2, P6, P8, and P4 of the cube (3-hypercube) are occupied by points. These four points build a fully occupied square (2-hypercube, i.e. a hypercube of lower dimensionality)

since only a tiny fraction of vertices are occupied by points? This is the question that we answer in this section.

We showed in Section 4.3.3 that given the dimensionality of a hypercube is sufficiently high, the PWD between its vertices converges to a particular value, and that the vast majority of the pairwise-distances are of this value as illustrated in Figure 4.1. Now, imagine that we select a tiny subset of these pairwise-distances, it is most likely that all of them will be equal to the convergence value. In other words, a randomly selected subset of the 2^d hypercube vertices have the same PWD between them.

This fact leads to an interesting property of the high dimensional hypercube, namely that a randomly selected subset of vertices of a d -hypercube most likely forms another e -hypercube where $e < d$, i.e. another hypercube of lower dimensionality. In our case, the subset consisting of those vertices occupied by points build a another hypercube fully occupied by points and having a lower dimensionality. Figure 4.10 illustrates this property by means of an example in a low dimensionality, namely a cube with partially occupied vertices, i.e. only P2, P6, P8, and P4 have points. These vertices build another fully occupied hypercube of lower dimensionality, namely the square P2 P6 P8 P4.

4.5 Hubness Indicator

It is of advantage to have a way for predicting the hubness in a data set without calculating k-NN lists of all points, as required for Equation 4.1. In this section, we propose a hubness indicator that has a linear complexity in terms of the point set size. This indicator is given in two variants: using the direct calculation of δ deviation in Section 4.5.1, and using the DTM and PWD statistics in Section 4.5.2.

4.5.1 Indication Using δ Deviations

In Section 4.4.2, we described two factors that determine the strength of hubness in a point set, namely the general tendency to hubness $\hat{\gamma}$ (Equations 4.34) and the outlier specific hubness $\hat{\vartheta}(x)$ (Equation 4.35). In this section, we put $\hat{\gamma}$ and $\hat{\vartheta}$ together to define a hubness indicator that provides an estimate of the total Hubness in a point set. The total hubness consists of two following parts:

The base hubness: The first part, which we will call the base hubness, is caused by $\hat{\gamma}$, the general hubness tendency in the data set, which is characteristic, given a distribution, a dimensionality and a distance norm. For example, $\hat{\gamma}$ has a higher value for normally than for uniformly distributed data which results from the DTM variance of the normal distribution being larger due to the density gradient in the radial direction (detailed discussion in Section 4.6.2), and consequently normally distributed data has in general a stronger hubness than uniformly distributed data.

The outlier specific hubness The second part of hubness is the outlier specific hubness, which is caused by outliers and thus related to $\hat{\vartheta}$. Note that $\hat{\vartheta}(x)$ is an indicator of the relative hubness of a single point (outlier) x , while here, we define an indicator for the absolute total hubness of a point set, i.e. an indicator for the total set hubness, based on the relative point hubness $\hat{\vartheta}$. Here, there are two considerations:

The first consideration is that $\hat{\vartheta}(x)$ is the relative hubness of x , which only determines the hubness capability of x compared with the other points. Whether and how much this capability in deed causes hubness, depends also on the general tendency to hubness $\hat{\gamma}$, i.e. the same outlier would cause a higher hubness in a distribution with higher $\hat{\gamma}$ (e.g. normal distribution) than in a distribution with a lower $\hat{\gamma}$ (e.g. uniform distribution). Thus, the absolute hubness caused by an outlier is estimated by $\hat{\gamma} \cdot \hat{\vartheta}(x)$.

The second consideration is that the total outlier specific hubness in a data set is caused by more than one outlier. Normally, it is caused by a series of outliers at different distances from the mean and thus with different values of $\hat{\vartheta}(x)$. An intuitive way to calculate the total hubness is to sum the outlier specific hubness over all points. However, this estimation is not accurate because the absolute hubness of a point x depends also on the existence of another point y that is nearer to the mean than x . That is the hubness of x , given y , is significantly less than its hubness, given that y does not exist.

A more accurate estimator of the total outlier specific hubness is using only the point with the highest $\hat{\vartheta}$ (namely the nearest point to the mean), which we denote by x_{max} .

We estimate the total outlier specific hubness by the square of the hubness caused by this point, i.e. $\hat{\vartheta}^2(x_{max})$. This is justified as follows: Since there is normally a series of outliers at different distances from the mean, which have different hubness values starting from zero (for points far from the mean) and increasing until the maximum hubness value (for x_{max}), the total hubness can be estimated as the sum of an arithmetic series. This implies a total that is quadratically proportional to largest term in the series, i.e. $\hat{\vartheta}^2(x_{max})$. The following example illustrates how an arithmetic series sums to a value that is quadratically proportional to the largest term

$$\sum_{i=1}^n n_i = 1 + 2 + 3 + \dots + n = \frac{n(n+1)}{2} \approx \frac{n^2}{2} \quad (4.39)$$

We combine the base hubness and the outlier specific hubness in a hubness indicator of the total hubness in a point set X , which is given by

$$\begin{aligned} \acute{H}(X) &= \acute{\alpha} \cdot \acute{\gamma} + \acute{\beta} \cdot \acute{\gamma} \cdot \hat{\vartheta}^2(x_{max}) \\ &= \acute{\gamma}(\acute{\alpha} + \acute{\beta} \cdot \hat{\vartheta}^2(x_{max})) \end{aligned} \quad (4.40)$$

where x_{max} is the point with the highest δ_{r_-} deviation, i.e. the nearest to the mean, and $\acute{\alpha}$ as well as $\acute{\beta}$ are scaling constants that can be used as weights to balance the impact of global versus outlier-based hubness. They can also be used to scale $\acute{H}(X)$ to some hubness measure as a reference, e.g. the basic measure Equation 4.1; this is required because $\acute{H}(X)$ provides values that are proportional to the hubness, but not necessarily equal to those of the basic measure; thus the constants α and β can be used to obtain hubness estimation values that are comparable with those of the basic measure.

Section 4.6.1 provides an analysis by means of random i.i.d. point sets that illustrates that the total hubness in a data set consists of two components, the base hubness and the outlier specific hubness.

4.5.2 Indication Using DTM and PWD

In Section 4.5.1, we have proposed a hubness indicator $\acute{H}(X)$ that is based on the δ deviations. In this Section, we propose analogous to Equation 4.40 a new hubness indicator $H(X)$ based on Equations 4.37 and 4.38, using only statistics of the DTM and PWD (i.e. statistics of ρ and η instead of the direct calculation of δ deviations). This results in a more stable indicator since the calculating statistics of the DTM and PWD does not assume a complete distance concentration as the calculating statistics of the deviations. This has been discussed in more details in Section 4.4.3. The new hubness indicator $H(X)$ is given by

$$\begin{aligned} H(X) &= \gamma(\alpha + \beta \cdot \vartheta^2(x_{max})) \\ &= kurt_L(\rho) \frac{var(\rho)}{var(\eta)} \left(\alpha + \beta \frac{(E[\rho] - \|x_{max}\|)^2}{var(\rho)} \right) \end{aligned} \quad (4.41)$$

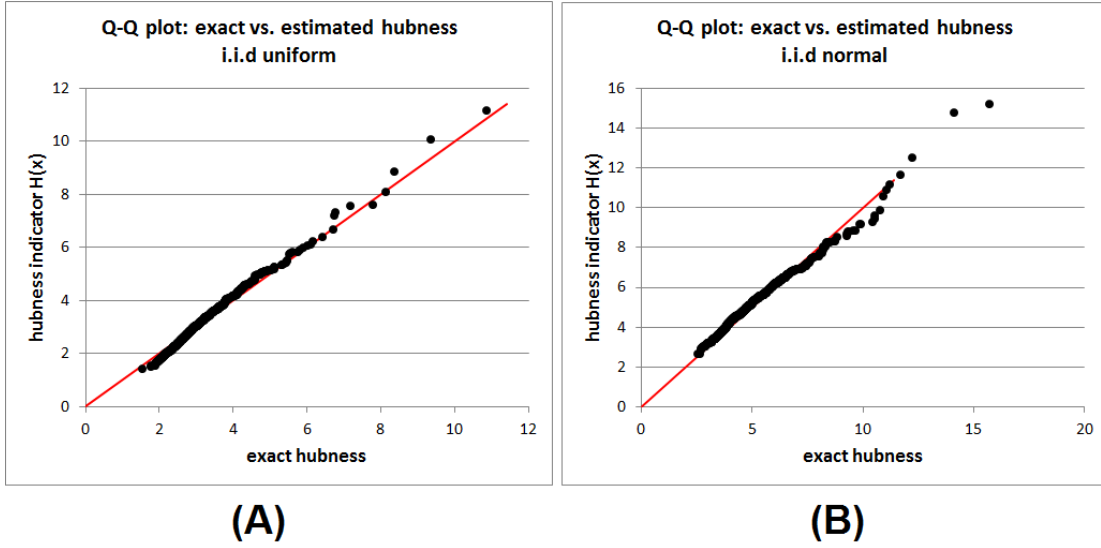


Figure 4.11: Q-Q plots of the exact hubness calculated using the basic hubness measure Equation 4.1 versus the estimated hubness $H(x)$ according to the proposed hubness indicator Equation 4.41 using a scaling constant $\alpha = \frac{1}{3}$. In both cases, 950 point sets, each of them consisting of 1000 points with dimensionalities uniformly distributed between 50 and 1000. In (A) for i.i.d. points drawn from a uniform distribution over a hypercube, and in (B) for i.i.d. normal distribution.

where x_{max} is the point with the highest ϑ value, i.e. the nearest point to the mean, and α as well as β are scaling constants that can be used as weights to balance the impact of global versus outlier-based hubness. They can also be used to scale $H(X)$ to some hubness measure as a reference. In order that $H(X)$ provides hubness values comparable with the basic measure in Equation 4.1, the constant α should be set to $\frac{1}{3}$ and β to 1.

To test the performance of the hubness indicator, Equation 4.41, we used 950 i.i.d. random point sets, each consisting of 1000 points and having a dimensionality, uniformly selected between 50 and 1000. For each of these 950 samples, two values have been calculated, namely the exact hubness using the basic function in Equation 4.1 and the estimated hubness value using the hubness indicator in Equation 4.41. The resulting 950 pairs of hubness values are then plotted on a quantile-quantile plot (Q-Q plot), which is a graph that visualizes the correlation between two distributions. Figure 4.11 (A) shows the Q-Q plot of one experiment using samples of uniformly distributed points, and (B) is the Q-Q of the same experiment repeated using samples of normally distributed points. Both plots show a strong correlation between the exact hubness and the estimated hubness (In a Q-Q plot, the strength of correlation is indicated by how approximately the points lie on the line $x = y$, i.e. on the red line).

Note that this hubness indicator has a complexity $O(d \cdot n)$, where n is the point set size, since a sampling of the PWD variance is sufficient, as mentioned in Section 4.4.3.

This is efficient compared with the classical way (Equation 4.1), that calculates the k-NN lists of all points, having at least $O(d \cdot n^2 \cdot \log n)$ complexity, assuming a sort complexity of $O(n \cdot \log n)$ and distance measurement complexity of $O(d)$, which is not efficient when n is large.

4.6 Analysis

In this section, we provide an analysis that supports the proposed cause of hubness, by explaining well-known observations or observations documented in the literature, as well as by presenting empirical results on point sets of different i.i.d. distributions and different distance norms.

4.6.1 Base Hubness vs. Outlier Hubness

Section 4.5 proposes a hubness indicator based on dividing the total hubness of a data set into two components, the base hubness, determined by γ (Equation 4.34) and the outlier specific hubness, determined by ϑ (Equation 4.35). While the base hubness is characteristic, i.e. stable, given a distribution, a dimensionality and a distance norm, the outlier specific hubness can lead to hubness of a data set strongly deviating from the hubness characteristic specified by γ . This is, for example, the reason why some point sets with uniform distribution can have higher hubness values than point sets with normal distributions if they have a high rate of outliers on the left side of the δ_r distribution. The two hubness components are illustrated in Figure 4.12 by showing how the hubness is reduced by removing the nearest n points to the mean, which are with high probability outliers. Figure 4.12 shows the hubness values (calculated using the basic algorithm in Equation 4.1) of i.i.d. point sets (uniform distribution in (A) and normal distribution in (B)) after successively removing the nearest points to the data mean. Furthermore, the values of ϑ of the point sets before removing the points is written in the legend. There are three interesting observations: (i) Removing only few points results in a significant reduction of hubness, namely the hubness was reduced by 50% by removing only 3 points in some point sets and removing 10 points on average, which means that hubness is reduced by 50% by removing 2% of the points on average. (ii) In all point sets from the same distribution, hubness converges to the same value after removing the outliers, which means that, after removing the outlier specific hubness, the remaining hubness is the base hubness. (iii) Considering the factor ϑ before the point removal, it is clear that ϑ correlates with the amount of hubness reduced until reaching the base hubness, i.e. with the outlier specific hubness.

4.6.2 Radovanovic et al. [RNI10] Versus Low et al. [LBSN13]

As has been mentioned in Section 4.2, Radovanovic et al. [RNI10] provide analysis on the origin of hubness. They emphasize the dimensionality and the data centrality as the main cause of hubness. However, Low et al. [LBSN13] challenge this claim and state that hubness is a matter of density gradient in data sets. To demonstrate the

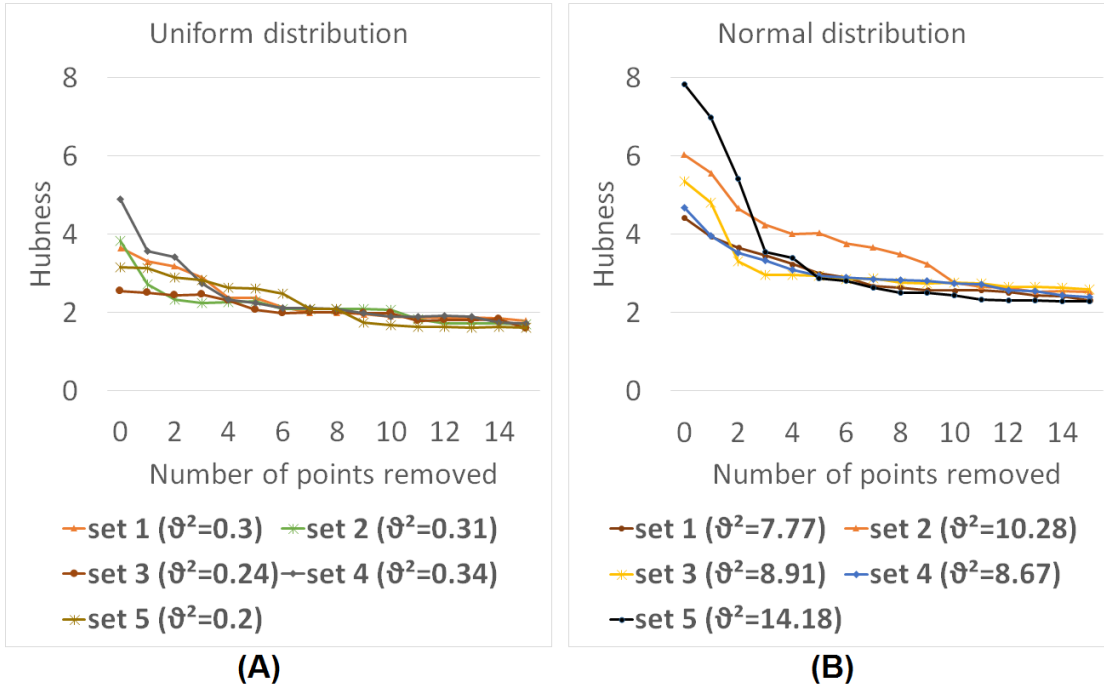


Figure 4.12: Reducing the hubness by removing the nearest point to the mean. 5 point sets in the 500-dimensional space, each consisting of 500 points with i.i.d components from uniform distribution (A) and normal distribution (B). The number of points removed is on the x-axis and the hubness is on the y-axis, calculated using the basic algorithm, Equation 4.1. The values of ϑ before point removal are encoded in the legend.

usefulness of our explanation as a theoretical background, we show that the discord between these two divergent research results can be reduced by linking both of them to the same explanation. Obviously, the observations of Radovanovic et al. that hubness is related to high dimensionality, and that hubs tend to be closer to the mean than other points, are straightforward outcomes of our explanation, since the number of vertices, connected to a given vertex over one edge, increases with the dimensionality, which consequently increases the impact of outliers on the NN relation of other points, as shown in Section 4.4.

However, the claim of Low et al. that hubness is a matter of density gradient can be also linked to the same explanation. An important result of our explanation is that the dimensionality alone is not sufficient for the emergence of hubness. In fact, even if the dimensionality is already high, the emergence of hubness depends on other conditions, namely the hubness tendency γ (Equation 4.34) and the existence of outliers on the left side of the distribution of the *DTM*. We agree with Low et al. that the density gradient promotes hubness insofar as it has an influence on the variance of δ_r and thus a direct impact on γ .

To explain how the density gradient has an impact on γ , we show that increasing the

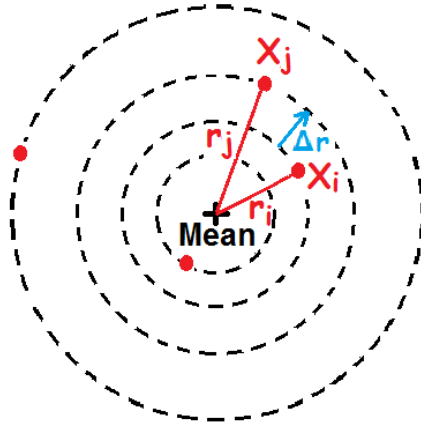


Figure 4.13: The density gradient in a distribution where the density decreases when moving farther from the mean. The dashed circles illustrate the density levels. $var(DTM)$ increases with increasing rate of density decay, because the difference between $min(\Delta r)$ and $max(\Delta r)$ increases, where $\Delta r = r_j - r_i$.

gradient results in an increase in the variance of the δ_r . Let us consider a distribution where the density decreases when moving away from the mean, e.g. the normal distribution, as shown in Figure 4.13. Here the density s_i at radius r_i is larger than the density s_j at radius r_j , given $r_i < r_j$. Since we are only interested in the $var(\delta_r)$ (the variance in the radial direction), we consider how the DTM changes as a result of changing the gradient. Since the density decreases when moving farther from the mean, the distance we should move until encountering the next point (Δr) increases, the farther we are from the mean. Consequently, the rate of change of DTM per unit radial distance increases, which means an increase in $var(DTM)$, which obviously means an increase in $var(\gamma)$.

Let us consider two examples presented by Low et al. as arguments that the density gradient is the cause of hubness. The first example is the observation of the high hubness in points drawn from a normal distribution compared to the lower hubness in uniformly distributed points. Here they link this observation to the high density gradient in the normally distributed points. Figure 4.14 (A) to (D) show how the strength of hubness is related to the variances of DTM and PWD, and consequently to the factor γ (ratio between variances of the DTM and the PWD). (A) and (B) are drawn from a uniform distribution. (C) and (D) are drawn from a normal distribution. The factor γ , which can be linked to the density gradient, is responsible for hubness being in general stronger in normal than in uniform distribution. Note that hubness can deviate from this general tendency specified by γ as a result of outliers at the left side of the DTM distribution, which have impact on the factor ϑ . For example in (B), the hubness is stronger than in (A) because of the outliers at the left side in (B). The outliers at the right side in (A) do not have impact on hubness. Analogously, the hubness in (D) is stronger than in (C) because of the outliers at the left side in (D).

The second example is the uniform distribution in the hyperball. Here Low et al.

argue the low hubness value to be a result of the low density gradient. As mentioned above, we agree with this claim insofar that the density gradient has an impact on γ . Figure 4.14 (E) and (F) shows that the uniform distribution in the hyperball has a very low γ because in contrast of the variance of PWD, the variance of the DTM is very low. The behavior of the hypersphere is discussed in detail in Section 4.6.3.

4.6.3 The Special Behavior of the Hyperball

The strange properties of the hyperball have been discussed by many researchers, e.g. [ZM14] [Ham50] [Fra08] [Koe00]. We will discuss some of these properties to explain the special property of the hyperball with respect to hubness. Low et al. [LBSN13] provide empirical results showing that no (or significantly low) hubness is observed in i.i.d points drawn from a uniform distribution in a hyperball. In this section, in a first step, we discuss the distance distribution and the convergence behavior of the uniform distributions in the hyperball, then we provide an explanation for the fact that such a distribution is almost free of hubness using the proposed cause of hubness.

Points uniformly distributed in a high dimensional hyperball with radius r are located almost exactly on the surface of the hyperball, i.e. at a distance r from the mean, a fact that has been confirmed frequently, e.g. [Ham50][ZM14]. We have provided a proof for distance convergence in Section 4.3.2. However, the unique property of the uniform hyperball is the quick convergence of the DTM compared to the PWD. When increasing the dimensionality d , the asymptotic Euclidean PWD in the hyperball converges to $r\sqrt{2}$ and the DTM converges to r . However, the PWD converges significantly more slowly than the DTM.

Figure 4.15 shows the convergence behavior of the DTM and the PWD for three different distributions, namely a normal distribution (A), a uniform distribution over the hypercube (B), and a uniform distribution in the hyperball (C). The variances of the DTM and PWD are estimated by plotting the minima and the maxima of the distances. Two unique properties of the uniform hyperball can be observed: (i) While the DTM and PWD in (A) and (B) continue growing asymptotically with d , the DTM and PWD are limited in (C) and do not depend on d . (ii) When increasing d , while the variance of the DTM in (A) and (B) continues to have relatively large values, the variance of the DTM in (C) decays very quickly and becomes almost zero already at $d = 50$. On the contrary, the variance of the PWD still keeps its relatively large values until the highest dimensionality considered in the experiment.

Now, the observation of the uniform hyperball being almost free of hubness can be straightforwardly explained by the proposed cause of hubness. The general hubness tendency of the uniform distribution over the hyperball is limited by $\gamma = \frac{\text{var}(DTM)}{\text{var}(PWD)}$, which decreases very quickly, since $\text{var}(DTM)$ decreases quickly whereas $\text{var}(PWD)$ keeps its value up to high dimensionalities. The low value of γ in the uniform hyperball explains the observation of being almost free of hubness. Figure 4.15 (D), shows the distribution of the DTM of one point set from (C) with $d = 500$. Note that the variance, estimated by $\max(DTM) - \min(DTM)$, is very low (~ 0.001) compared with the variance of the

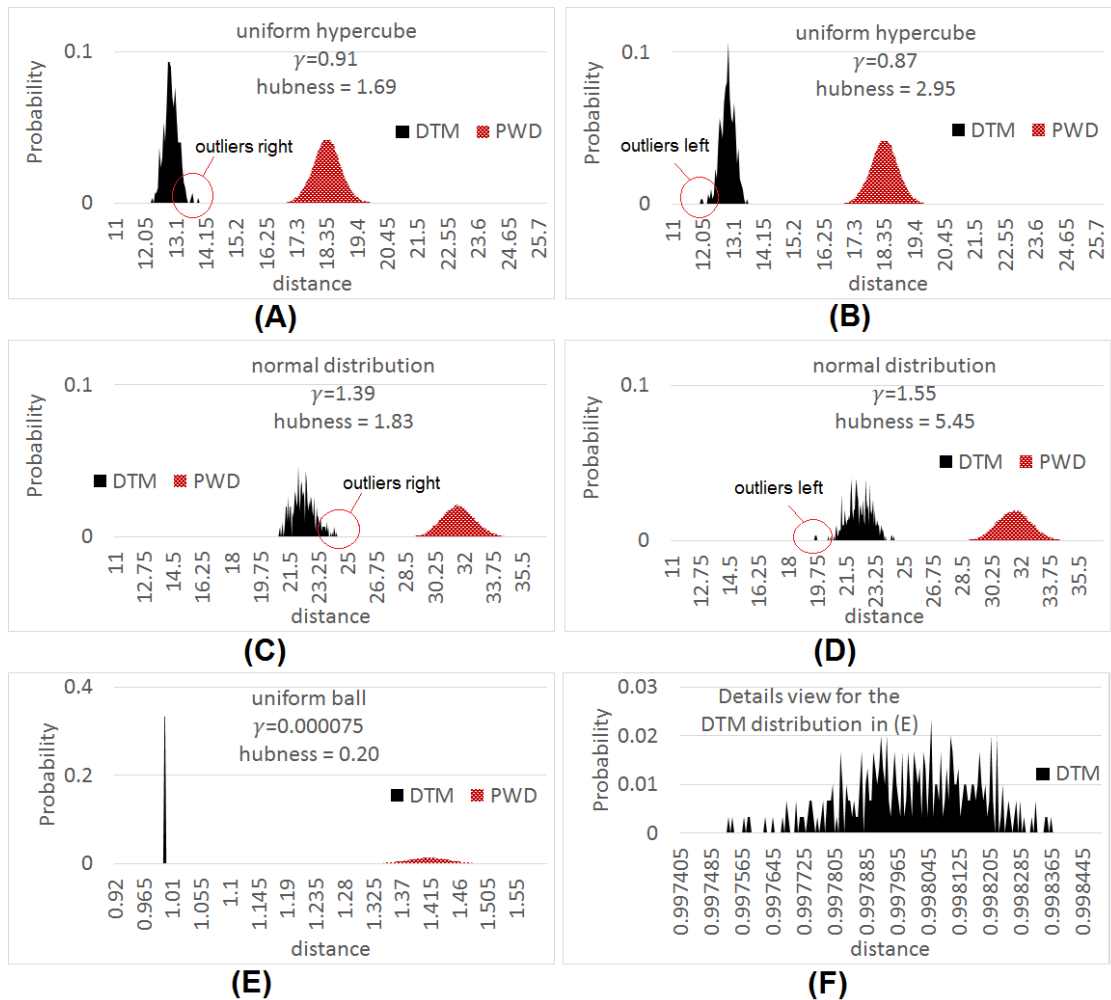


Figure 4.14: The relation between hubness and the distributions of DTM and PWD. Five examples of point sets, each of them consists of 300 points with dimensionality 500, drawn from different i.i.d. distributions. (A) and (B) are two different random point sets drawn from a uniform hypercube, (C) and (D) from a normal distribution, and (E) from a uniform distribution over a hyperball. (F) is a details view of the DTM distribution in (E). The factor γ (ratio between the DTM variance and the PWD variance) as well as the outliers left to the DTM distribution decide the strength of hubness. In (A) and (B), γ is mid-range and the hubness has also mid-range values, however in (B) hubness is higher because of the outliers at the left side. In (C) and (D), γ is high and hubness is relatively high, but however it is higher in (D) because of the outliers at the left side. In (E), γ is very low and hubness is also very low although there is a high rate of outliers at the left side of the DTM distribution, which is clear in the details plot (F). All hubness values are measured with the basic NN definition, Equation 4.1

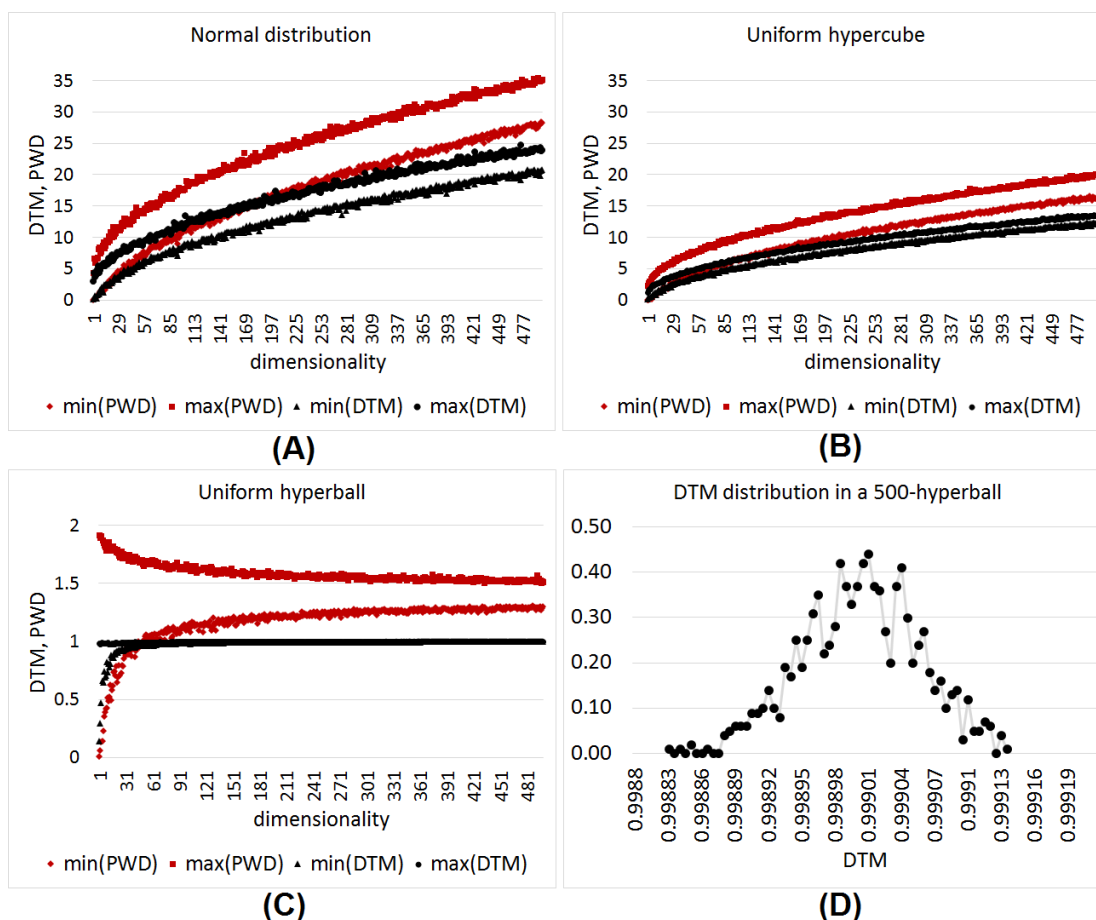


Figure 4.15: Comparison of the decay rates of the DTM variance and PWD variance for 500 point sets with dimensionalities varying from 1 to 500 for three distributions, namely a normal distribution in (A), a uniform distribution over the hypercube in (B), and a uniform distribution in the hyperball in (C). In each of the three cases, the estimative variances of the DTM and PWD are visualized by showing the minimum and the maximum. In (D), the distribution of the DTM of one case from (C), namely a point set of the dimensionality 500 is shown. Note that the domain of (D) corresponds to a very small distance interval of about 0.001 unit, compared with the corresponding interval of the PWD, which is about 0.2, estimated as $\max(PWD) - \min(PWD)$ for the corresponding point set in (C)

PWD (~ 0.2), estimated by $\max(PWD) - \min(PWD)$ from (C). Also note that this resulting low γ prevents the emergence of hubness, although there are outliers on the left side of the distribution, as shown in (D).

Also Figure 4.14 (E) and (F) shows the relation between hubness and the distribution of the DTM and PWD in the hyperball. It shows that such a distribution is characterized

by a very low γ compared with the uniform distribution (A) and (B), and the normal distribution (C) and (D).

4.7 Hubness Reduction

The theoretical background proposed in Section 4.4 and the factors affecting hubness defined in Section 4.5 provide a foundation for strategies of hubness reduction. As has been mentioned in Section 4.6.1, hubness consists of two parts, the base hubness determined by the distribution, the distance norm, and the dimensionality, which can be indicated by the factor γ (Equation 4.37), and outlier specific hubness, caused by points lying significantly nearer to the mean than the other points and indicated by the factor ϑ (Equation 4.38). In this section we suggest two strategies for hubness reduction based on these foundations. We test these methods only with i.i.d. random point sets and recommend testing them with real data sets in future work.

4.7.1 Hubness Reduction by Hub Removal

In this section, we suggest a strategy for hubness reduction based on the factor ϑ (Equation 4.38) determining the outlier specific hubness. In Section 4.6.1, it has been illustrated by means of i.i.d. random data sets that only $\sim 1\%$ of the points (outliers) cause most of the outlier hubness and also that removing these outlier points leads to a significant hubness reduction. This observation suggests the exclusion of outliers as a hubness reduction strategy.

This is motivated by two observations:

- i For the identification of such outlier points (hubs), the hubness indicator proposed in Section 4.5 can be used, which finds such points in linear time.
- ii Only very few points need to be removed in order to achieve a significant hubness reduction, as has been illustrated in Section 4.6.1.

Since there are usually few hubs, this means that the negative impact on performance caused by hub removal is small compared with the positive impact caused by hubness reduction. In the following, we present the justification for this claim. We start with an illustrative example and then generalize it.

Figure 4.16 illustrates a set of 13 points with their nearest neighbor (NN) relations for a k -NN algorithm with $k = 1$. We define correct relations to be NN relations measured by a distance metric and confirmed by a human as correct. Correct relations are illustrated as green arrows and the corresponding points are also colored in green. Furthermore, We define incorrect relations to be NN relations measured by a distance norm and judged by a human as incorrect because the measurement is affected by hubness bias. Incorrect relations are illustrated as red arrows and the corresponding points are also colored in red. In (A), the hub point h is recognized by a distance metric to be the NN of six points. Only one of these six relations, namely the point t , is correct. The other five relations

($w1$ to $w5$) are incorrect. In total, there are seven correct NN relations (green arrows), and five incorrect relations (red arrows). In (B), the hub point h has been removed, which leads to changes in the NN relations measured by the same metric. Now, the point t gets another NN, which is incorrect. But on the other hand, the points $w1$ to $w5$ get new NN, which results in their NN relations becoming correct. In total, there are 11 correct relations and only one incorrect relation.

To compare (evaluate) the two cases, let us define a score that results from incrementing for each correct relation and decrementing for each incorrect relation. In (A), the score is $7 - 5 = 2$. In (B), i.e. after removing the hub point h , the score is $11 - 1 = 10$. This means that removing one hub point h leads to a score increase of $10 - 2 = 8$.

Now, consider the general case with a k -NN algorithm and recall the k -occurrence n_k (Definition 18), where $n_k(x)$ denotes the number of points for which x is the k -NN. The expectation value of the k -occurrence in a k -NN algorithm is k , i.e. $E[n_k] = k$, which means that if the hubness would not exist, a point would likely be the NN of k points. Now assume that a hub point h is identified by a distance metric to be the NN of m other points and assume that m is sufficiently larger than k , i.e. $n_k(h) = m$ and $m \gg k$. Since the expectation value of the $E[n_k]$ is k , this means that there are likely $m - k$ incorrect NN relations corresponding to the hub point h . The score in this case is $k - (m - k) = 2k - m$ (the correct relations minus the incorrect relations).

Now, removing the hub point h from the data set has two impacts: (i) A negative impact caused by k cases, namely when the query point q is one of the k points having h as correct NN because these become incorrect after removing h . (ii) A positive impact related to $m - k$ cases, namely when the query point q is one of the $m - k$ points incorrectly measured to have h as NN, these become correct after removing h . This means that the number of correct relations changes from k to $m - k$ and the number of incorrect relations changes from $m - k$ to k . The score in this case is $(m - k) - k = m - 2k$.

The score increase caused by removing h is $(m - 2k) - (2k - m) = 2m - 4k$, which implies that removing the hub point h leads to a total improvement of $2m - 4k$ relations. Note that the higher m is compared with k , the more benefit is obtained by its removal.

We tested this method with randomly generated i.i.d. point sets, by generating 100 point sets, each containing 1000 points and having a dimensionality between 100 and 1000. For each point set, the hubness was calculated twice using the basic hubness measure (Equation 4.1), once with all the points included and a second time after removing 1% of the points (10 points) that have the most hubness, identified using the hubness indicator in Section 4.5. In each case, a k -NN algorithm was considered with a k value randomly selected between 1 and 10. The results are displayed in Table 4.17, which show that by removing 10 out of 1000 points, the hubness can be reduced to 51% of the original hubness in the average case and to 23% of the original hubness in the best case. Note that after removing 10 points, the remaining hubness is almost constant for all point sets, because the remaining hubness is the base hubness, which is characteristic for the distribution, an observation that is in conformance with the analysis in Section 4.6.1.

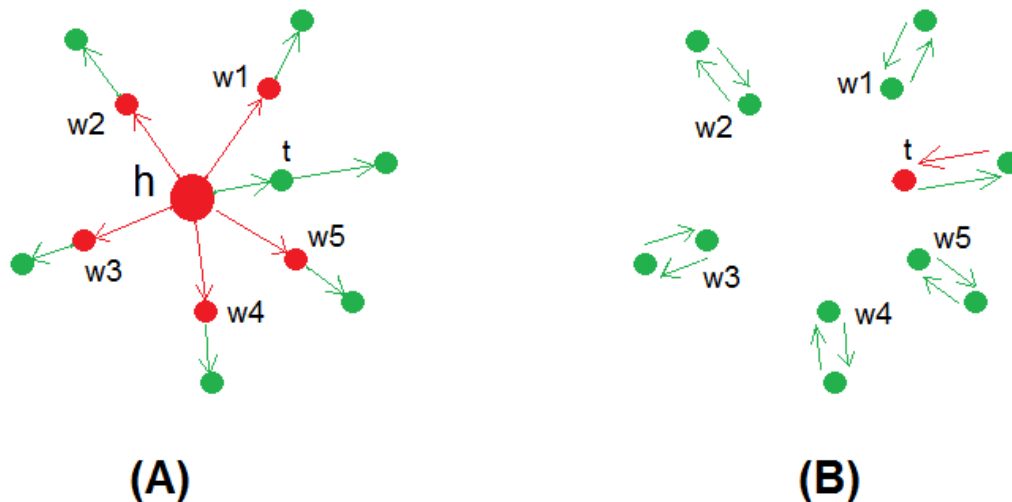


Figure 4.16: The influence of removing hubs on the NN relations. Green arrows represent correct NN relations, and red arrows represent incorrect NN relations caused by hubness bias. The direction of the arrow means k -nearest neighbor of. In (A) the hub point h is measured by a metric to be the NN of six points, only one of them t is correct and the five others are incorrect. (B) illustrates the NN relations after removing the hub point h . A significant decrease in the incorrect relations, and increase in the correct relations are observed.

4.7.2 Hubness Reduction by Transforming to the Hyperball

In this section, we suggest a method for hubness reduction based on the factor γ (Equation 4.37). In Section 4.6.3, we showed that the uniform distribution in a hyper ball is characterized by significantly low hubness due to a low γ . We propose a method for hubness reduction in normally distributed points by transforming the point components (coordinates) such that the points fit into a uniform distribution in a hyper ball without destroying their nearest neighbor relations.

As has been mentioned in Section 4.6.2, the density gradient in the normally distributed data results in high values of γ , which consequently cause high hubness values. The idea behind this reduction method is to reduce or eliminate the density gradient in the normal distribution.

To this end, points are moved toward the centroid so that the variance of the distribution of the DTM is reduced. Figure 4.18 shows the transformation that achieves a reduction of the DTM variance, while maintaining the relative spatial relations between points. Points far from the centroid are moved toward the centroid more than nearer points, but however, the DTM of points should still have the same order after the transformation. In Figure 4.18, (B), the point X2 is the farthest from the centroid, so it is moved the most, and likewise X3 is the nearest and thus it is moved the least.

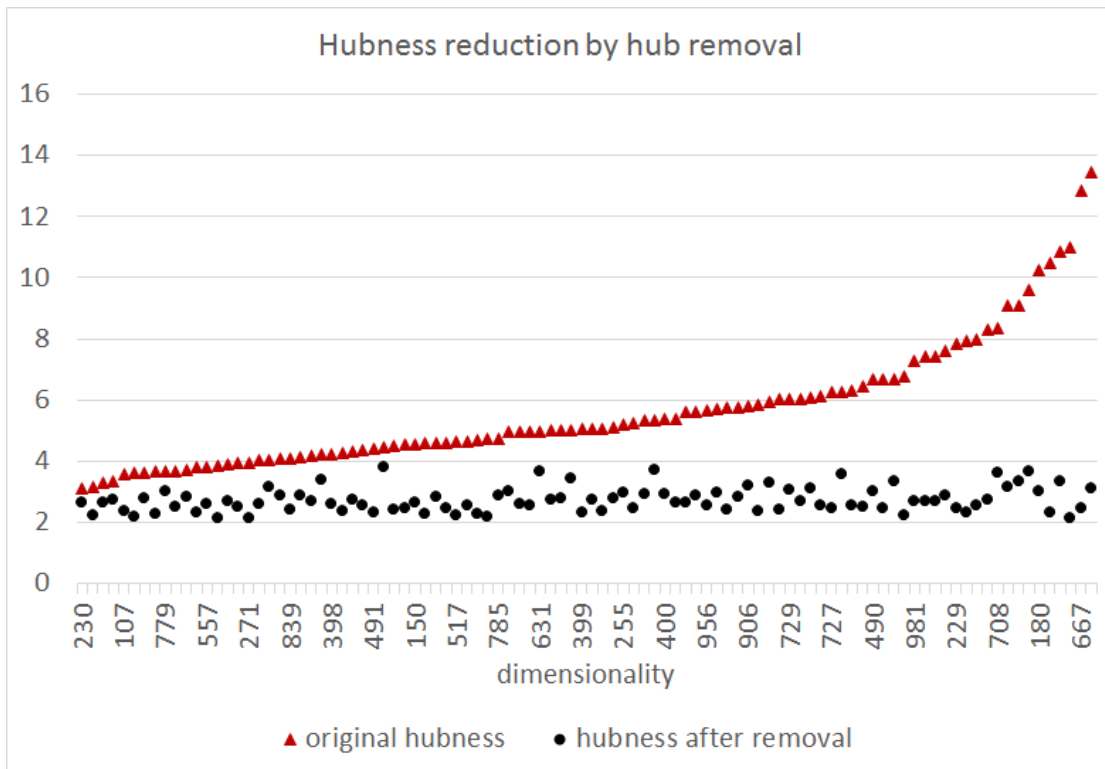


Figure 4.17: Hubness reduction by hub removal: 100 i.i.d point sets have been randomly generated, each containing 1000 points and having a dimensionality between 100 and 1000. For each point set, two hubness values were calculated using the basic hubness measure (Equation 4.1), one value of the complete point set and the other value after removing 1% of the points (10 points) that have the most hubness. In each case, a k -NN algorithm was considered with a k value randomly selected between 1 and 10. The point sets are sorted according to the original hubness (before removal). Sorting according to dimensionality does not make sense because the effect of dimensionality on hubness is considerably smaller than the effect of outliers.

The transformation required should meet three properties: (1) It should manipulate only distances in the direction of the density gradient, i.e. in the radial direction. Angles between vectors representing the points should not be changed to avoid unnecessary manipulation. (2) The order of the points with respect to their DTM should be maintained, which results in the NN relations remaining unchanged. (3) The transformation should be sensitive to the distance from the mean, i.e. farther points should be moved toward the centroid more than near ones.

One way to achieve this goal is to use the probability density function of the normal distribution for the transformation. Let X be a set of d -dimensional points with normally distributed components $X_1 \dots X_d$. Each component X_i is normally distributed according

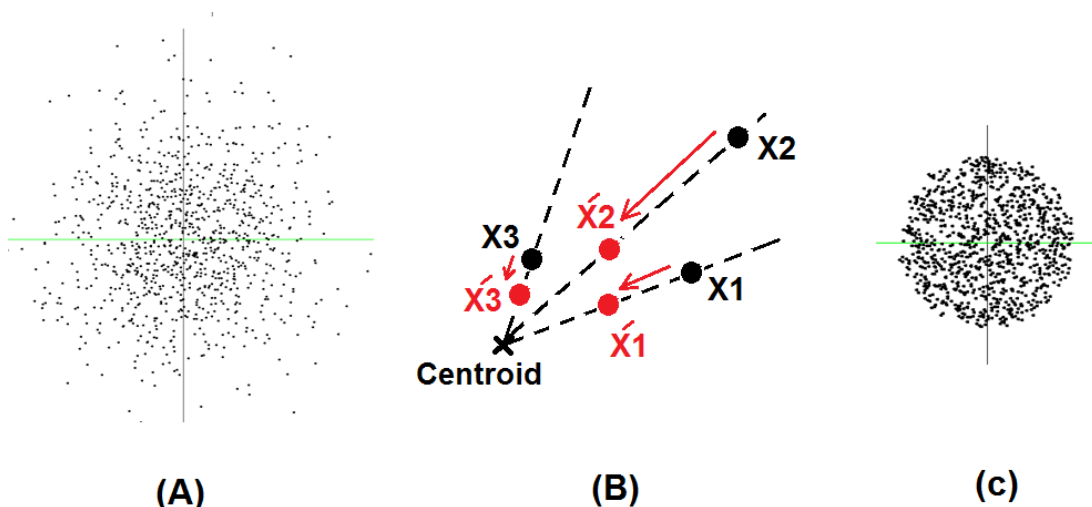


Figure 4.18: Transformation for hubness reduction. In (A) the points are normally distributed which results in a density gradient. In (B), points far from the centroid are moved toward the centroid, so that the magnitude of movement corresponds to the distance from centroid, i.e. farther points far from the centroid are more moved than near ones. In (B), the points after transformation have distribution that is nearer to the uniform hyperball, and thus have less hubness.

to the probability density f_i with σ_i and $\mu = 0$, i.e. the components are normalized to have their means at the origin. This implies that $f_i(x_i)$ is the probability of a point x having x_i as its value of i^{th} component.

In order that the points fit in a hypersphere, we should transform each individual component, such that the components become uniformly distributed, thereby keeping the angle of each vector unchanged, i.e. we want to transform only the lengths of the vectors. Transforming a vector goes in three steps, (i) calculating its length r and angle α , (ii) dividing all its components by r to get the unit vector representation of the point, and (iii) giving the unit vector a new length that is related to the probability density $f_i(r)$. The first two steps are the steps required for the common unit length normalization [SB88] [Buc93] [SBM96]. The third step aims to give each vector x a new length in $[0, 1]$, such that the original length hierarchy (order) is maintained, i.e. if two vectors x_1 and x_2 respectively have original lengths r_1 and r_2 , where $r_1 > r_2$, then the new lengths r'_1 and r'_2 should also meet the condition $r'_1 > r'_2$. This condition is satisfied when using the probability density f_i because of the fact that the farther the point is from the origin, the lower is the probability. Note that in order for this to hold, the point components should be normalized to have their means at the origin.

Formally, each point $x = (x_1, \dots, x_d)$ with a distance to the origin $r = \|x\|$ is trans-

formed to the point $\hat{x} = (\hat{x}_1, \dots, \hat{x}_d)$ where

$$\hat{x}_i = \frac{x_i - \mu_i}{r} \frac{1}{\sqrt{2\pi}} (1 - e^{-\frac{r^2}{2}}) \quad (4.42)$$

where $\mu = (\mu_1, \dots, \mu_d)$ is the mean. Note that the normalization of the points to have their mean at the origin is already included in the transformation. Figure 4.19 shows the result of applying the transformation in Equation 4.42 on i.i.d random point sets with normally distributed point components. In this experiment, 30 i.i.d. random point sets were generated, such that each has 1000 points and a dimensionality randomly selected between 30 and 500. The exact hubness values were calculated for the point sets using the basic algorithm (Equation 4.1) for k values selected between 1 to 10. For each point set, the points were transformed using Equation 4.42 and then the exact hubness values were calculated again. The results show a significant hubness reduction achieved by the transformation.

Note that in many state-of-the-art algorithms of IR, normalized vectors are used, where vectors representing data points are divided by their lengths which results in all vectors having the unit length [SB88] [Buc93] [SBM96]. Such normalization results in the variance of the DTM becoming zero, since all points are at the same distance to the mean, which is sufficient for hubness elimination. However, this normalization has the disadvantage of large loss of information compared with the transformation proposed in Equation 4.42, which does not require that the vectors representing the points have unit length.

The transformation according to Equation 4.42 can be used in two ways: As a transformation of the data space as a prior step, and then using a normal distance metric for measuring distance and building models; or as an integrated part of the distance metric, i.e. the distance metric considers the transformation as a part of its definition.

4.8 Testing with Real Data and other norms

In this section, we present a set of experiments that verify the hubness findings using real world data. As dataset, we used the TREC-AP ¹ text collection. The collection contains 209,783 documents and has a dimensionality of 237,368 terms. This collection was used in other research [AV08] to verify methods related to a similar topic, namely the retrievability.

These experiments require measuring the exact hubness to be used as a reference. Since the basic measure of hubness (Equation 4.1) has a complexity of an $O(d \cdot |X|^2 \cdot \log |X|)$, it is not realistic to test with the whole data set because this would take years of running time. To solve this problem, our strategy was to perform the experiments on random samples, each consisting of 500 documents selected uniformly from the collection. For significance, i.e. to ensure that the samples represent the collection, each experiment was repeated using 5 different samples.

¹The TREC-AP text categorization test collection is derived from proprietary AP news data. See the detailed description under <http://www.daviddlewis.com/resources/testcollections/trecap>

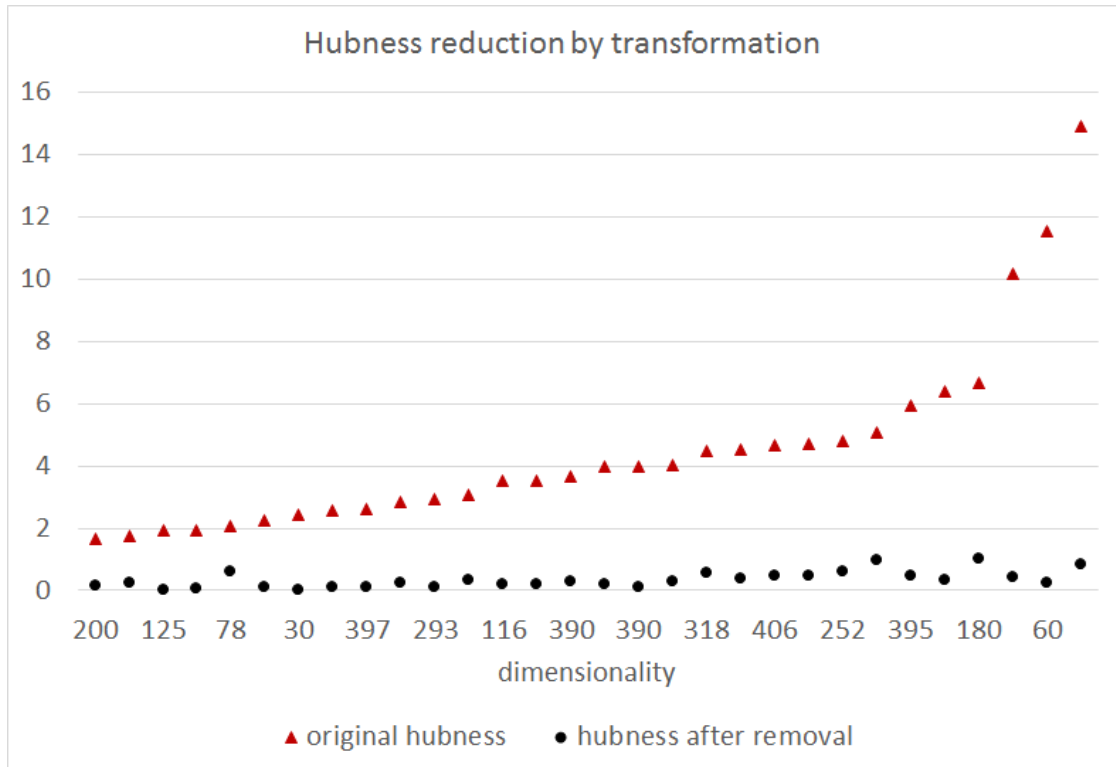


Figure 4.19: Hubness reduction by hub transformation: 30 i.i.d point sets have been randomly generated, such that each has 1000 points and a dimensionality between 30 and 500. For each point set, two hubness values were calculated using the basic hubness measure (Equation 4.1), one value is the original hubness and the other value is the hubness after the transformation according to Equation 4.42. In each case, a k -NN algorithm was considered with a k value randomly selected between 1 and 10. The point sets are sorted according to the original hubness (before transformation).

This set of experiments aims to (i) verify the proposed explanation of hubness using real world data, (ii) test whether this explanation is applicable to other norms, and (ii) evaluate the performance of the hubness indication method proposed in Section 4.5 using real data. In particular, we present experiments using three different norms. In two of them, namely the L_2 norm and the cosine distance COS , we aim to show that our findings are general and can be directly applied to norms of different categories. In the third one, namely the L_∞ , we aim to show that there are, however, norms, to which our findings hold in general, having special properties that cause side effects, which prevent a direct application of the hubness indicator and hubness reduction methods.

We describe here the experiment that was repeatedly performed on the samples each time under different settings, namely different norms and different values of k . The results of these experiments are presented and discussed in the next paragraphs. The experiment

was performed as follows: The exact hubness of each sample is calculated using the classic measure (Equation 4.1). Then for each document, the hubness contribution $\vartheta(x)$ is calculated using the proposed hubness indicator (Equation 4.38). Additionally, the k -occurrence $n_k(x)$ is calculated for each document. The documents are then sorted according to $\vartheta(x)$ descending. Now, the documents are successively removed from the sample in the sort order (the highest ϑ first). After each removal, the hubness of the sample is calculated using the basic measure. With the help of n_k values, it is possible to verify whether the documents identified as hubs using the hubness indicator, i.e. having high ϑ , are indeed hubs. And with the help of the exact hubness of the samples, it is possible to verify that removing hubs leads to overall hubness reduction.

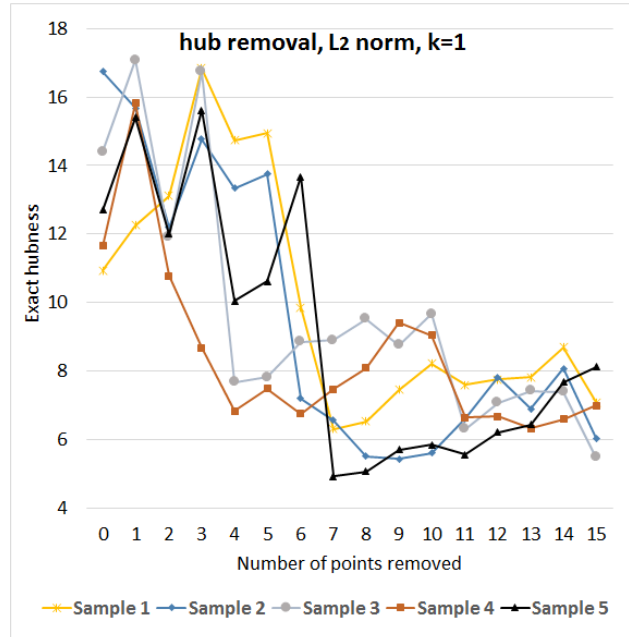
Experiments using L_2 norm: Figure 4.20 shows the results of the experiment performed using the L_2 norm and a k -NN algorithm with $k = 1$. The same experiment was repeated for each of the five samples. In (A), the top part of a listing of the documents sorted (separately for each sample) according to $\vartheta(x)$ (hubness contribution). For each document, the k -occurrence $n_k(x)$ (calculated according to Definition 18) is shown. The aim is to show that the documents removed based on $\vartheta(x)$ are indeed hubs by comparing $\vartheta(x)$ with $n_k(x)$. In (B), the vertical axis shows the exact hubness of each sample (measured using the basic function Equation 4.1) after successively removing the documents from the top of the listing in (A), i.e. in the order of decreasing $\vartheta(x)$. The number of documents removed are on the horizontal axis. Note that the hubness on the vertical axis in (B) is the overall hubness of the sample after the removal, while $\vartheta(x)$ in (A) is a hubness indicator corresponding to one single document, which is a value proportional to the hubness contribution of the document, which is not comparable with the sample hubness as absolute value. Also note that since $\vartheta(x)$ aims to predict for the document x the relative hubness contribution in the same point set, it makes no sense to compare ϑ values in different sets.

One important observation is that the decrease of hubness is not strictly monotonous, that is the removal of some points results in a hubness increase. This is a major difference between the results of this experiment and those of a similar experiment presented in Section 4.6.1, which was performed on i.i.d. random point sets with uniformly as well as normally distributed points. In contrast to the current experiment, the hubness in random data sets decreases strictly monotonous with hub removal. We explain this effect by text data being clustered and not uni-modal like the randomly generated point sets. Text data being clustered stems from sparsity, that is for a given document, the vast majority of the attributes are zeros. When the data is clustered, removing some hubs could lead to the emergence of other hubs, most likely in another cluster, but nearly at the same distance from the mean. This can be observed for documents with similar (or equal) ϑ values, e.g. the first two documents of Samples 1 in Figure 4.20 (A). The removal of the first one causes a hubness increase as shown in Figure 4.20 (B).

However, although the hubness decrease is not strictly monotonous, the hubness seems to converge after removing few hubs to a particular value. This confirms our findings regarding base- and outlier hubness presented in Section 4.6.1, i.e. after removing

removal order	Sample 1		Sample 2		Sample 3		Sample 4		Sample 5	
	$\vartheta(x)$	$n_k(x)$	$\vartheta(x)$	$n_k(x)$	$\vartheta(x)$	$n_k(x)$	$\vartheta(x)$	$n_k(x)$	$\vartheta(x)$	$n_k(x)$
1	1.93	57	2.30	180	2.28	130	1.89	60	1.80	110
2	1.93	22	2.15	19	2.14	96	1.84	91	1.71	58
3	1.92	30	2.06	8	2.02	4	1.72	23	1.61	16
4	1.90	89	2.04	4	2.02	3	1.71	12	1.58	18
5	1.64	12	1.91	11	1.74	6	1.67	3	1.51	2
6	1.61	3	1.75	3	1.73	2	1.67	3	1.47	2
7	1.51	0	1.65	0	1.73	0	1.64	2	1.44	12
8	1.48	0	1.61	2	1.72	0	1.62	3	1.38	2
9	1.47	0	1.61	1	1.70	8	1.61	3	1.38	2
10	1.44	6	1.60	0	1.64	0	1.58	2	1.38	2
11	1.43	1	1.59	1	1.62	0	1.58	9	1.37	0
12	1.42	4	1.58	2	1.55	0	1.55	1	1.36	1
13	1.38	0	1.57	0	1.52	3	1.54	6	1.36	0
14	1.38	1	1.57	0	1.50	2	1.53	1	1.35	1
15	1.36	1	1.54	1	1.50	1	1.52	4	1.35	1
16	1.35	5	1.52	2	1.49	0	1.51	1	1.33	0
17	1.35	0	1.51	0	1.47	0	1.50	3	1.31	0
18	1.34	0	1.48	1	1.46	0	1.48	1	1.31	0
19	1.33	0	1.47	1	1.46	0	1.47	1	1.29	0
20	1.32	0	1.47	3	1.44	0	1.45	0	1.26	1
21	1.32	1	1.46	1	1.44	0	1.44	2	1.25	2
22	1.32	4	1.43	1	1.44	0	1.41	2	1.25	2
23
...

(A)



(B)

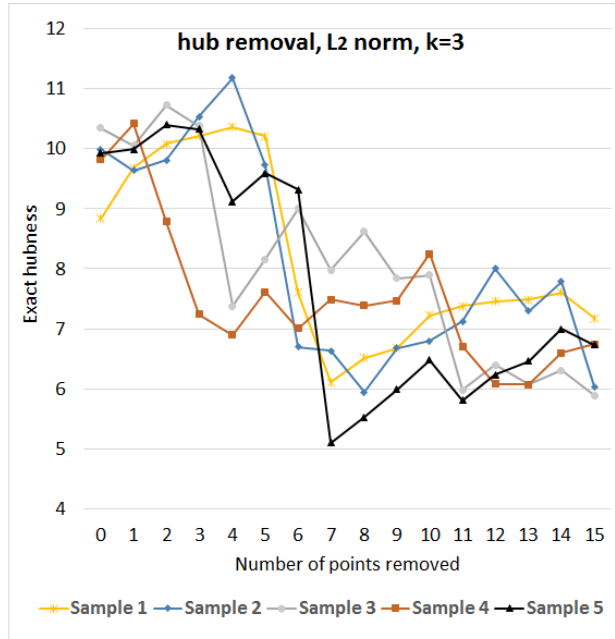
Figure 4.20: Hubness reduction by hub removal on samples of the TREC-AP text collection, each consisting of 500 documents. A k -NN algorithm with $k = 1$ and the L_2 norm have been used. In (A), for each document x , the value of the hubness contribution $\vartheta(x)$ and the exact k -occurrence $n_k(x)$ are calculated. The documents are sorted according to ϑ . The top part of the list is shown. In (B), the documents are successively removed in the order of the list (highest ϑ first). The number of documents removed is on the horizontal axis and the resulting exact hubness of the sample is on the vertical axis.

the outlier hubness, the hubness converges to the base hubness, which is distribution characteristic (note that all samples are drawn from the same distribution).

To evaluate the effect of k on the behavior of hubness, we repeated the previous experiment with $k = 3$ instead of 1. Figure 4.21 shows results that are similar to those of the previous experiment, but with one difference, namely that the hubness is in general lower for the same samples. This is in conformance with the observations documented in the literature [FSS12], namely that hubness is mostly stronger with smaller values of k . Note that the $n_k(x)$ values being larger does not necessarily mean higher hubness, because this is related to the higher value of k . The hubness has to do with the skewness of n_k distribution rather than with the absolute values. Also note that ϑ values are the same as in the previous example because ϑ depends only on the position of the document relative to the collection and does not depend on k , that is ϑ gives an indication of the relative hubness contribution of a particular document based on its position.

removal order	Sample 1		Sample 2		Sample 3		Sample 4		Sample 5	
	$\vartheta(x)$	$\text{nk}(x)$	$\vartheta(x)$	$\text{nk}(x)$	$\vartheta(x)$	$\text{nk}(x)$	$\vartheta(x)$	$\text{nk}(x)$	$\vartheta(x)$	$\text{nk}(x)$
1	1.93	213	2.30	260	2.28	256	1.89	179	1.80	229
2	1.93	143	2.15	218	2.14	267	1.84	208	1.71	228
3	1.92	130	2.06	134	2.02	111	1.72	104	1.61	116
4	1.90	190	2.04	73	2.02	88	1.71	50	1.58	81
5	1.64	19	1.91	39	1.74	18	1.67	20	1.51	18
6	1.61	19	1.75	13	1.73	6	1.67	28	1.47	9
7	1.51	2	1.65	0	1.73	6	1.64	16	1.44	27
8	1.48	0	1.61	5	1.72	3	1.62	13	1.38	2
9	1.47	0	1.61	3	1.70	14	1.61	13	1.38	2
10	1.44	10	1.60	2	1.64	2	1.58	5	1.38	4
11	1.43	4	1.59	1	1.62	2	1.58	29	1.37	6
12	1.42	9	1.58	12	1.55	0	1.55	10	1.36	2
13	1.38	1	1.57	5	1.52	7	1.54	18	1.36	1
14	1.38	3	1.57	2	1.50	3	1.53	2	1.35	5
15	1.36	2	1.54	4	1.50	5	1.52	7	1.35	1
16	1.35	12	1.52	4	1.49	1	1.51	4	1.33	0
17	1.35	0	1.51	0	1.47	1	1.50	12	1.31	2
18	1.34	1	1.48	1	1.46	1	1.48	2	1.31	2
19	1.33	0	1.47	2	1.46	1	1.47	2	1.29	0
20	1.32	1	1.47	5	1.44	0	1.45	1	1.26	4
21	1.32	2	1.46	3	1.44	0	1.44	5	1.25	6
22	1.32	8	1.43	1	1.44	0	1.41	4	1.25	4
23
...

(A)



(B)

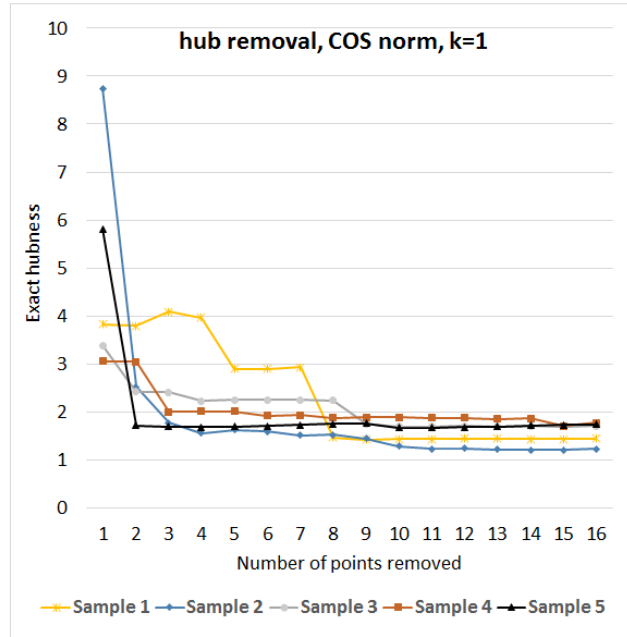
Figure 4.21: The results of the previous experiment (Figure 4.20) repeated using k -NN algorithm with $k = 3$ instead of $k = 1$.

Experiments using cosine distance norm (COS): The aim of this experiment is to show empirically that the proposed findings are applicable to a different category of norms than the p -norms, namely the cosine distance. In Section 4.3.4, we have demonstrated that the distance structure with respect to COS is in conformance with the hubness explanation model, i.e. that the points are located at the hypercube vertices. In this experiment, we empirically confirm that the hubness indicator and the hubness reduction by hub removal are applicable to the COS . We perform the same experiment as before using the COS norm and an k -NN algorithm with $k = 1$. Figure 4.22 presents the results, which show that the hubness has been reduced by removing the documents with the highest ϑ values. They also show that the hubness converges in all samples to almost the same value as expected. One can also observe that the convergence of hubness is faster than with L_2 and almost strict monotonous, which makes the hubness indicator and the hubness reduction method more effective. Note that the definition of the mean differ according to the distance norm. Recall that the mean with respect to COS is the vector that minimizes the angles to all other points. However, we use an approximation of the mean, namely the nearest point to the mean as has been described in Section 4.3.4.

Experiments using L_∞ distance norm: The results of repeating the experiment with L_∞ norm are presented in Figure 4.23 and show, in contrast to L_2 and COS , no hubness reduction.

removal order	Sample 1		Sample 2		Sample 3		Sample 4		Sample 5	
	$\vartheta(x)$	$n_k(x)$	$\vartheta(x)$	$n_k(x)$	$\vartheta(x)$	$n_k(x)$	$\vartheta(x)$	$n_k(x)$	$\vartheta(x)$	$n_k(x)$
1	26.03	13	30.52	25	27.27	13	25.96	6	26.28	20
2	6.86	3	11.30	5	5.29	1	4.74	12	5.05	4
3	6.41	4	9.25	4	4.17	4	3.77	4	4.06	1
4	6.14	12	7.09	0	4.16	3	3.45	0	4.04	3
5	6.05	7	5.25	5	3.86	1	3.14	0	3.30	4
6	5.66	4	4.81	0	3.57	2	3.08	1	3.17	3
7	5.51	9	4.74	2	3.56	3	2.73	0	2.83	2
8	4.81	1	4.53	1	3.40	8	2.55	2	2.74	0
9	4.64	1	4.17	3	3.30	6	2.52	4	2.72	6
10	4.51	2	3.95	5	3.29	2	2.36	2	2.64	4
11	4.22	4	3.73	0	3.24	2	2.24	3	2.62	1
12	4.16	1	3.67	3	3.06	1	2.24	1	2.62	0
13	3.97	2	3.55	1	2.98	0	2.19	3	2.54	4
14	3.88	2	3.52	0	2.96	1	2.17	0	2.35	0
15	3.69	1	3.49	3	2.90	0	2.04	2	2.35	1
16	3.51	1	3.43	0	2.84	2	2.03	0	2.33	1
17	3.50	0	3.36	1	2.75	1	1.97	1	2.32	1
18	3.39	1	3.32	3	2.73	2	1.95	1	2.25	0
19	3.19	4	3.23	4	2.58	0	1.81	1	2.17	3
20	3.18	4	3.17	0	2.57	0	1.81	0	2.11	1
21	3.10	2	3.15	1	2.43	1	1.78	8	2.05	1
22	2.92	0	3.05	0	2.41	3	1.74	0	2.04	4
23
...

(A)



(B)

Figure 4.22: The results of the same experiment repeated using the COS norm and a k -NN algorithm with $k = 1$. Results show that removing the documents with the highest ϑ results in a hubness reduction as expected. It is also observed that the hubness convergence is faster than with the L_2 norm.

This surprising result motivated us to look deeper at the distribution of the k -occurrences $n_k(x)$ and how they are related with the hubness indicator $\vartheta(x)$. Figure 4.24 shows a plot of all these values for Sample 1 in (A) for L_2 and in (B) and for L_∞ . The plot in (B) shows that there is a relation between $n_k(x)$ and $\vartheta(x)$, namely that the documents with high ϑ values (on the left side) have in general higher n_k values. Documents with negative $\vartheta(x)$ (those deviating away from the hypercube vertices) are mostly anti-hubs. This relation confirms that the proposed hubness explanation is in general valid also for L_∞ . However there is a special property of the L_∞ that prevents the hubness reduction by hub removal, which we explain as follows:

Since L_∞ is defined as the maximum of the dimension-wise displacements between two points, this definition results in that documents having equal largest term frequencies are at the same distance from the mean, although $d - 1$ of their components are totally different. This results in a step-like distribution of $\vartheta(x)$ as illustrated in Figure 4.24 (B). This step-like distribution makes distinguishing the hubness potential of the documents ineffective and thus prevents an exact identification of hubs because the hubness reduction by removal is based on removing only the top few documents in the list sorted according to $\vartheta(x)$. Such side effects caused by special properties related to particular norms require additional handling in order to apply the hubness indicators and reduction methods.

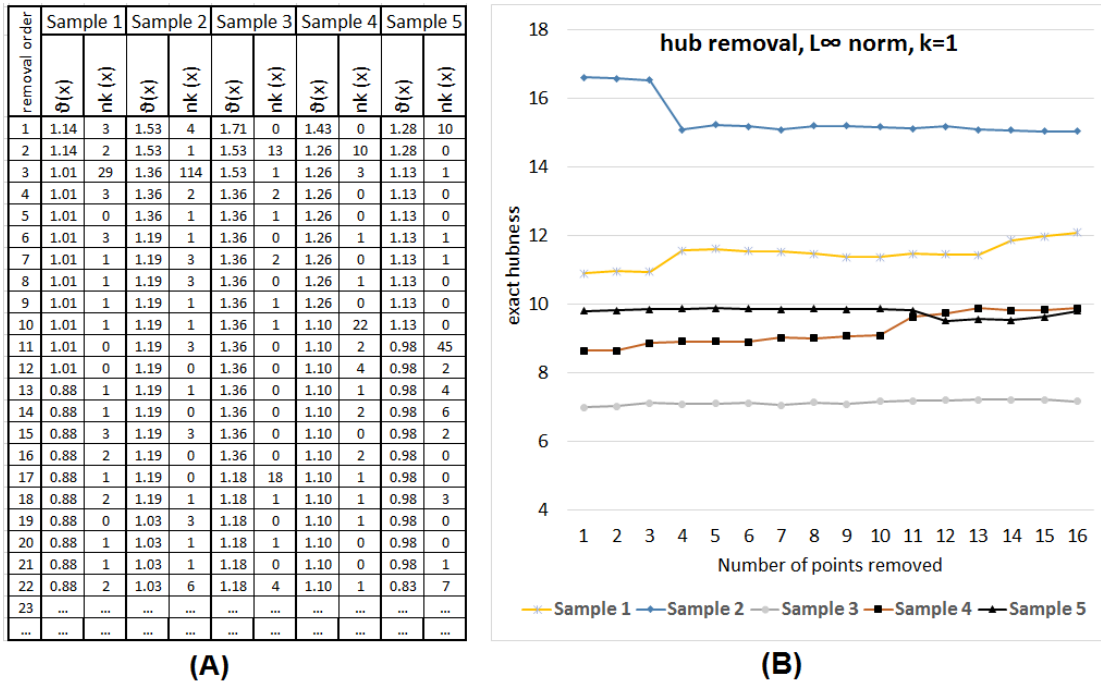
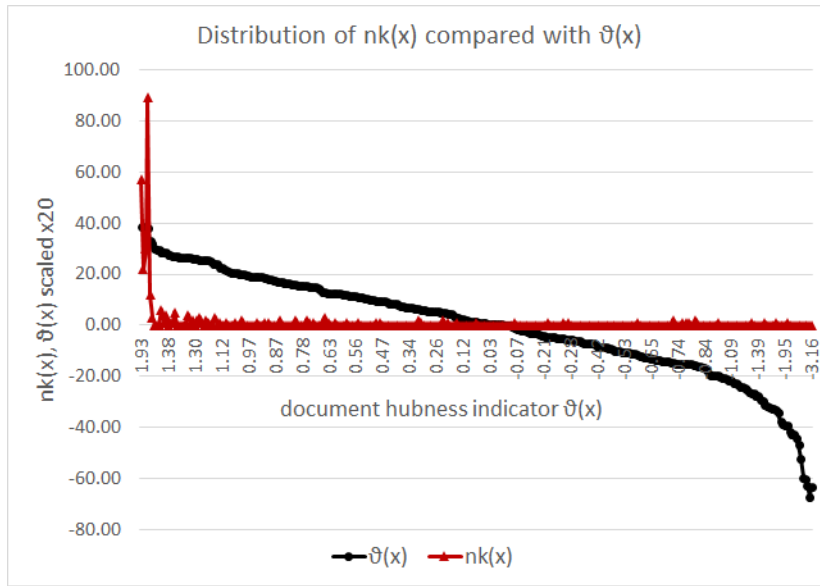


Figure 4.23: The results of the experiment hubness reduction by removal repeated using the L_∞ norm and a k -NN algorithm with $k = 1$. There is no hubness reduction observed in (B) after removing the top documents of the list sorted according to $\vartheta(x)$ in (A).

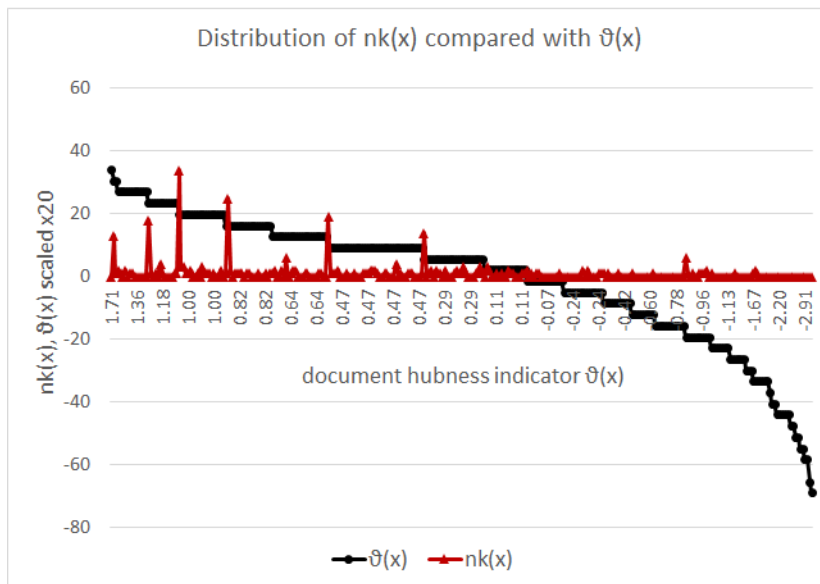
Finding generic guidelines for doing so is an essential challenge for future work.

4.9 Summary

We provided an analysis of hubness that results in a theoretical explanation of the origin of hubness based on the sparsity and distance concentration in high dimensional spaces. This explanation is general, since it does not assume particular distributions or particular distance norms. We demonstrated the strength of the proposed analysis by showing that three observations from the literature regarding hubness can be linked to the proposed origin of hubness and explained based on it. Based on this theoretical finding, we proposed hubness indicators with linear complexity in terms of the point set size. These indicators provide estimations of hubness in a given data set, or hubness caused by a particular data point. The indicators are based on the distribution of point deviations from their expected positions without calculating the nearest neighbor lists of all data points. Furthermore, we suggest a new strategy for hubness reduction based on transformation of the point components.



(A)



(B)

Figure 4.24: Comparison between L_2 norm (A) and L_∞ norm (B) of how the $n_k(x)$ values are distributed in relation to the $\vartheta(x)$ values for all documents in Sample 1. The values are sorted according to $\vartheta(x)$. In order to make the plot readable, $\vartheta(x)$ has been scaled ($\times 20$) on the vertical axis. While $\vartheta(x)$ is distinguishable for all documents in (A), it does not seem so in (B) because $\vartheta(x)$ distribution has a step-like form. The distribution of the $n_k(x)$ values is however related to $\vartheta(x)$ because documents on the left side have in general higher n_k values than those on the right side.

Conclusion and Future Work

5.1 Conclusion

In this work, we provide solutions for some difficulties arising when applying metrics to feature space of particular properties or under particular conditions. These solutions are in three different directions:

Metric bias and metric selection: We investigated biases and sensitivities of metrics to provide guidelines as well as a formal method for selecting evaluation metrics. This was achieved in two steps:

In the first step, we provided a comprehensive analysis of 20 evaluation metrics for 3D medical segmentations, which have been identified based on a literature review, such that only those metrics have been considered, for which there exist at least two papers confirming their usage in evaluating 3D medical segmentation. For each of these metrics, we provided binary and fuzzy definitions, as well as an efficient implementation in the form of an evaluation tool that has been provided as an open source project¹.

Furthermore we provided a comprehensive analysis of these metrics, discussing their properties, sensitivities and biases. To this end, (i) we analyzed the correlation between rankings produced by these metrics in general and under particular conditions, e.g. when the overlap between segments has particular levels. (ii) We analyzed particular examples of segmentation having special properties and investigated the behavior of each metric, given these properties. (iii) We designed synthetic al segmentations and illustrations to clear the strength and drawbacks of some metrics under particular conditions. (iv) We considered observations documented in the literature and related them to our analysis.

Finally, this analysis was summarized in form of metric properties (such as sensitivity to outliers, bias to segment size, ability to discover agreement caused by chance, rewarding

¹EvaluateSegmentation is an evaluation tool providing efficient implementation of 20 evaluation metrics for 3D image segmentation. It is available for download as open source under <http://github/codalab/EvaluateSegmentation>

high recall, etc.), and segmentation properties (such as complexity of boundary, segment density, outlier level, etc.). We then related between these metric and segmentations properties to provide guidelines for selecting suitable evaluation metric(s) for 3D medical images.

In the second step, a general metric selection framework has been proposed for selecting evaluation metrics for arbitrary evaluation task that is based on comparing objects with their corresponding ground truth objects. This framework is based on a novel method for inferring metric bias that provides a formal way to infer the metric sensitivity to a particular property of the objects being evaluated. Given a set of objects being evaluated and a set of properties that these objects can have, the proposed framework provides a method to automatically infer the bias of each metric to each property, based on which the overall bias to a particular data set is inferred.

Efficient metric calculation: We proposed two solutions to the efficiency problem of computing the distance between two huge point sets.

In the first solution, we proposed an efficient algorithm for calculating the exact Hausdorff distance (HD) between two arbitrary point sets. The HD is known to be complex in terms of computation time. A direct computation of the HD has a complexity that is quadratically proportional to the point set sizes, which makes a direct computation inefficient when the point sets are huge. The proposed algorithm calculates the exact HD in linear time in terms of the point set size. The proposed algorithm is general and can be used to measure the distance between two arbitrary point sets. The algorithm is based on two optimizations, namely the early breaking optimization, which makes use of the HD being a maximum of minimums, and the randomization optimization, which makes use of the principle of locality. In the early break optimization, unnecessary computations are avoided by breaking the loop of minimization when it is ensured that the completing the loop will not give additional information about the current maximum. The randomization optimization ensures that the order, in which minimum distances are computed, is random. This leads to a considerably more frequent occurrence of the early break and thus more avoidance of unnecessary computations. The idea behind the randomization optimization is that processing in the natural order of points (without randomization) leads, according to the principle of locality, to that the early break is not likely to occur, given that it has not occur in the previous iteration. This algorithm has been tested in various domains and found to significantly outperform the state-of-the-art algorithms.

In the second solution, we presented an efficient algorithm for calculating the average distance between image segmentations. This algorithm makes use of the segmentations being solid objects. The algorithm adds two optimizations to an existing algorithm that calculates the nearest neighbor using grid indexing. The first added optimization is avoiding the computation of distances to points inside the segments, i.e. measuring the distance between the surfaces of the segments. This is done by considering representations of the segments as hollow objects. The second optimization is reducing the search of nearest neighbors by finding a convenient search sphere (search radius) that ensures finding

the nearest neighbor, which considerably reduces the number of distance computations required. The algorithm has been tested on huge whole body magnet resonance volume segmentations and found to outperform the ITK algorithm as a state-of-the-art algorithm.

Hubness explanation, estimation, and reduction: We presented a formal explanation of the origin of hubness in high dimensional space. This explanation is based on the distance concentration of high dimensional points, which is well studied and has a solid basis. We provide a model of distance structure between points when the dimensionality is sufficiently high. According to this model, points are located (concentrated) at the vertices of a hypercube and have only small deviation from the vertices that decreases with dimensionality. When a point deviates more than the expectation value from a vertex toward the centroid of the hypercube, it becomes a hub because it becomes nearer to many other neighboring vertices.

Based on this hubness explanation, we proposed a hubness estimator that predicts the hubness of a particular point, i.e. the amount of hubness a particular point contributes with, which enables the identification of hubs and anti-hubs. Another hubness indicator has been proposed that predicts the extent of hubness in a given data set in a linear time in terms of the point set size.

Furthermore, based on the explanation, we suggested two hubness reduction methods, the first one is based on the identification of hubs using the hubness estimator and removing them from the data set, which leads to a considerable hubness reduction by removing only a tiny fraction (1%) of the points. The second hubness reduction method is based on transforming normally distributed points to a distribution known to have very low hubness while maintaining the nearest neighbor relations between points. In particular, points with normally distributed components are transformed to have a uniform distribution in a hyperball, by moving points toward the center, such that farther points are moved more than nearer ones in a manner that points fit in a uniform hyperball.

5.2 Future Work

The following suggestions are possibilities that are recommended to continue the research done in this thesis.

Hubness: In Chapter 4, we proposed a formal explanation of the origin of hubness and hubness indicators that predict the hubness contribution of a particular point as well the overall hubness in a data set. These findings are general and can be applied to any distance norm. However, although the basic theoretical principle is applicable for any norms, some special problems arise in combination with some norms, which seem as side effects related to the definition of these norms. One example is the L_∞ norm, which is defined as the maximum of the dimension-wise displacements between two points. This definition results in that all points are at the same distance from the mean, given that their largest component (coordinate) is equal, i.e. they are at the same distance from the

mean although $d - 1$ of their components may be totally different. To imagine the effect of this property, consider that in a text collection with term frequencies varying from 0 to r , there are only r levels of distance to mean, which means there is always a huge number of documents at the same distance from the mean. In this case, our expectation based on the proposed hubness explanation about the positions of hubs holds in general, i.e. that hubs are in general nearer to the mean than other points, but not the contrary, i.e. not every point near to the mean is a hub.

Another example of problems to be further investigated is the definition of the mean with respect to different norms. For example, this is the case with the BM25 function that is commonly used to measure the similarity between text documents. While the meaning of mean is clear with respect to the p-norms (e.g. the Euclidean distance), it does not seem to be straight forward with BM25. Such problems stemming from special properties of particular distance measures need to be addressed. We recommend therefore further investigation to enable a general application of the hubness findings in combination with these norms.

In Section 4.7.2, we suggested a hubness reduction method based on transforming normally distributed points to a distribution that fits in a uniform hyperball, which leads to a significant hubness reduction. In the proposed form of the transformation, the method works only with uni-modal normal distribution. However, data in practice is clustered (e.g. as a result of sparsity in text data). Such data can be often represented by a Gaussian mixture. We recommend further investigation to generalize the transformation to Gaussian mixture to enable the application of this hubness reduction method on a wider spectrum of real word data.

Analogy between text and images: Modeling the data in a feature space depends on the nature of the underlying data. Image and text are commonly used data in image retrieval and data mining. Metrics used to measure the similarity between images are usually different than metrics used to measure similarity between text documents. We suggest an analogy between text data and image data that enables representing both of them with the same framework, which allows the mutual application of the same techniques like the similarity metrics.

Images are defined on a grid. We link the union of all cells in this grid to the set of terms in a text feature space, such that image data is seen as a feature space of dimensionality $d = w.l.h$ (width · length · height), i.e. grid cells in image data correspond to terms in text data. Since an image is a collection of grid cells (pixels) and documents are collections of terms, images are linked to documents.

We recommend further investigating this analogy trying to answer questions like whether the techniques used for text processing can be transferred to images and vice versa. Can, for instance, BM25 be applied for image similarity, and the Hausdorff distance for document similarity. Another question is whether image feature space is subject to curse of dimensionality, e.g. hubness, since the image feature space is high dimensional according to this analogy.

A deeper insight in this analogy suggests that it is promising, since it can be extended

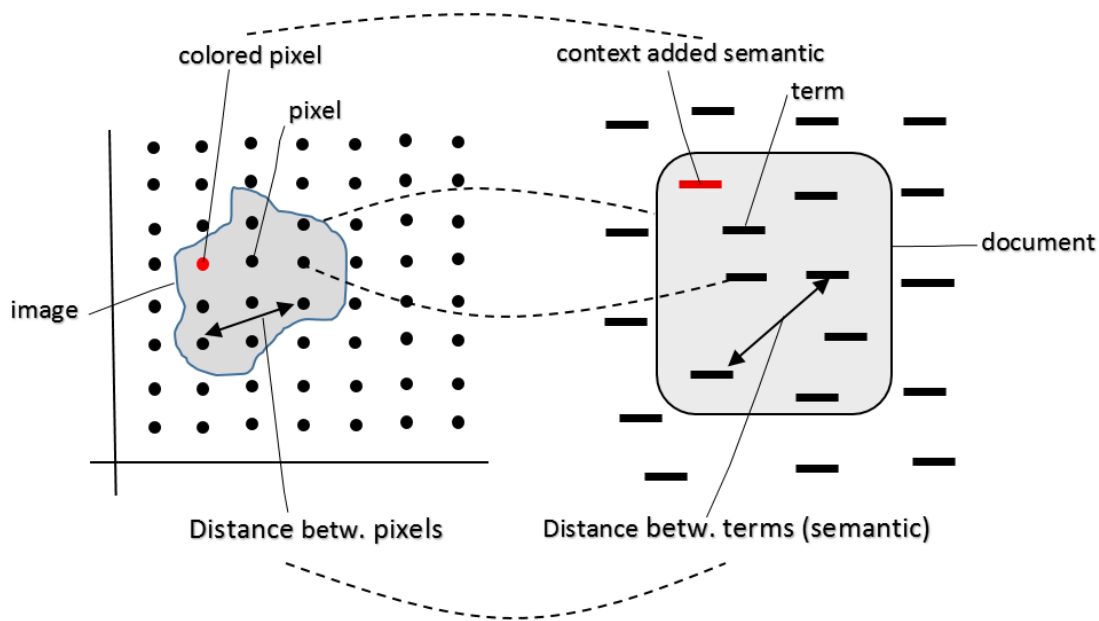


Figure 5.1: Illustration of the suggested analogy between image and text. The analogies are represented by dashed lines. Grid cells correspond to terms. The image to the document. The distances between pixels correspond to the distance between terms (semantic). The color a grid cell has from the context of an image corresponds to the meaning added to a term from the context of a document.

to more elements, namely semantic attributes. The 3-dimensional grid space of the image data gives information about the relations between grid cells, i.e. the distance between each pair of cells. This grid space is linked to the semantic in text that gives information about the similarity between terms. The analogy can thus be extended to context-added semantic. In a text document, a term can obtain a new semantic from the context of the document. Analogously the same grid cell can have two different colors, each one in the context of a different image. Figure 5.1 illustrates the suggested analogy between images and text.

It is worth investigation to figure out whether the accuracy of text retrieval systems can be increased by making use of measures designed for image similarity, that is to measure the similarity between text documents at syntax and semantic levels using these measures based on the analogy described above.

Metric Definitions and Algorithms

In this appendix, we provide the definitions of all metrics listed in Table 2.1, which have been selected, based on a literature review described in Section 2.4, as evaluation metrics for 3D medical image segmentation. Furthermore, we present in Section A.9 examples of inconsistency in the literature regarding the definition of the metrics to underline the need of a standardization of the definitions of evaluation metrics.

Formal general definitions for binary and fuzzy 3D images are provided in Section 1.6.3. Definitions of the basic cardinalities of the confusion matrix for the binary and fuzzy cases are provided in Section 1.6.4. Because these two Sections contain basic definitions that are important for this chapter, we repeat these two sections here to improve the readability.

A.1 2D and 3D Images

An image can be thought of as a set of points defined on a grid, i.e. the points are represented by grid cells, which we call pixels. Images can be 2-dimensional (2D) or 3-dimensional (3D). 3D images are also called volumes, and the 3D-pixels are called voxels. The metric space defined on a set of images is a special cases of the metric space according to Definition 3, in which the objects are images, and the metrics coming into consideration are only those according to Definition 5. Since 2D images are a special case of 3D images, we will only provide a definition for a 3D image, which implicitly holds for a 2D image as well.

Definition 7. *A 3D binary segmentation (binary segmented volume) is represented by the ordered pair (X, S) , where:*

- $X = \{x_1, \dots, x_n\}$ is a point set with $|X| = w \cdot h \cdot d$, where w , h and d are the width, height and depth of the grid on which the volume is defined, such that each point $x \in X$ corresponds to a grid cell (voxel). We will call w , h and d the grid dimensions and $w \cdot h \cdot d$ the grid size.
- S is a classification that assigns each grid cell (each point $x \in X$) to one of two classes, either the foreground or the background, such that S builds a partition $S = \{S^1, S^2\}$ on X represented by the assignment function $f^i(x)$ that provides the membership of the grid cell x in the subset S^i , where $f^i(x) = 1$ if $x \in S^i$ and $f^i(x) = 0$ if $x \notin S^i$. S can also be seen as a segmentation, i.e. the set of voxels that define a segment. We denote S^1 by the foreground voxels and S^2 by the background voxels.

Note that binary segmentations are a special case of fuzzy segmentations, in which the assignment function f has the range $\{0, 1\}$. This definition can be generalized to the fuzzy case by redefining the range of f to be $[0, 1]$ representing the degree of membership of a voxel to a particular class.

Definition 8. A fuzzy 3D segmentation (fuzzy segmented volume) is an image according to Definition 7, in which the assignment function $f^i(x)$ is redefined to have its range in $[0, 1]$, where $f^i(x) \in [0, 1]$ represents the degree of membership of the grid cell x in the subset S^i .

A.2 Basic Cardinalities of the Confusion Matrix

Many of the metrics used for comparing 3D image segmentations can be derived from the four basic cardinalities of the so-called confusion matrix, namely the true positives (TP), the false positives (FP), the true negatives (TN), and the false negatives (FN). We define these cardinalities for the binary as well as the fuzzy case.

Basic cardinalities for binary segmentation: For two binary classifications that assign each element in a sets to one of two classes, in our case segmentations according to Definition 7, we define the four basic cardinalities (also called the confusion matrix), representing the overlap that results based on the agreement/disagreement of the assignments of the two classifications (segmentations). The four cardinalities are TP (true positive), FP (false positive), FN (false negative), and TN (true negative).

Definition 9. Let S_g and S_t be two segmentations according to Definition 7, with assignment functions f_g and f_t respectively. Let S_g denote the ground truth segmentation and S_t denote the segmentation being evaluated. The four cardinalities are given by the sum of agreement m_{ij} between each pair of subsets $i \in S_g$ and $j \in S_t$. That is

$$m_{ij} = \sum_{r=1}^{|X|} f_g^i(x_r) f_t^j(x_r) \quad (\text{A.1})$$

where $TP = m_{11}$, $FP = m_{10}$, $FN = m_{01}$, and $TN = m_{00}$.

Table 1.1 shows the confusion matrix of the partitions S_g and S_t .

Table A.1: Confusion matrix comparing two segmentations, S_g as the ground truth segmentation and S_t as the test segmentation

Subset	S_t^1	$S_t^2(= \overline{S_t^1})$
S_g^1	$TP(m_{11})$	$FP(m_{12})$
$S_g^2(= \overline{S_g^1})$	$FN(m_{21})$	$TN(m_{22})$

Generalization to fuzzy segmentation: Intuitively, one favorable way to generalize the metrics based on the basic cardinalities to the fuzzy is to generalize the cardinalities of the confusion matrix to the fuzzy case. To this end, the main task is to calculate the agreement between two segmentations, where the assignments of voxels to segments are probabilities (fuzzy). It is common for this purpose to use a suitable triangular norm (t-norm) to calculate the agreement between two fuzzy assignments [KPM00][Cam07]. Given two probabilities p_1 and p_2 representing the memberships of a particular element (voxel) to a particular class (segment) according to two different classifiers (segmenters), we use the $\min(p_1, p_2)$ as a t-norm as the agreement between the two classifiers. That is, we define the agreement function $g : [0, 1] \times [0, 1] \rightarrow [0, 1]$ that models the agreement on a particular voxel being assigned to a particular segment as $g(p_1, p_2) = \min(p_1, p_2)$. This also means that the agreement on the same voxel being assigned to the background is given by $g(1 - p_1, 1 - p_2)$. Intuitively, the disagreement between the segmenters is the difference between the probabilities given by $|p_1 - p_2|$. However, since the comparison is asymmetrical (i.e. one of the segmentations is the ground truth and the other is the test segmentation), we consider the signed difference rather than the absolute difference as in Equations A.3 and A.5. The four cardinalities defined in Equation A.1 can be now generalized to the fuzzy case as follows:

Definition 10. Let S_g and S_t be two segmentations according to Definition 8, with assignment functions f_g and f_t respectively that satisfy the conditions $f_g^1(x) + f_g^2(x) = 1$ and $f_t^1(x) + f_t^2(x) = 1$ for all $x \in X$ (i.e. the memberships of a given point x always sum to one over all classes). Let S_g denote the ground truth segmentation and S_t denote the segmentation being evaluated. The four fuzzy cardinalities of the confusion matrix are given by

$$TP = \sum_{r=1}^{|X|} \min(f_t^1(x_r), f_g^1(x_r)) \quad (\text{A.2})$$

$$FP = \sum_{r=1}^{|X|} \max(f_t^1(x_r) - f_g^1(x_r), 0) \quad (\text{A.3})$$

$$TN = \sum_{r=1}^{|X|} \min(f_t^2(x_r), f_g^2(x_r)) \quad (\text{A.4})$$

$$FN = \sum_{r=1}^{|X|} \max(f_t^2(x_r) - f_g^2(x_r), 0) \quad (\text{A.5})$$

Note that in Equations A.2 to A.5, $f_g^i(x_t)$ and $f_t^j(x_t)$ are used in place of $p1$ and $p2$ since each of the functions provides the probability of the membership of a given point in the corresponding segment, and in the special case of crisp segmentation, they provide 0 and 1.

Other norms have been used to measure the agreement between fuzzy memberships like the product t-norm, the L-norms, and the cosine similarity. We justify using the min t-norm by the fact that, in contrast to the other norms, the min t-norm ensures that the four cardinalities, calculated in Equations A.2 to A.5, sum to the total number of voxels, i.e. $TP + FP + TN + FN = |X|$ which is an important requirement for the definition of metrics.

Basic setting: Based on the definitions in Sections A.1 and A.2, we provide the settings to be considered for the metric definitions in the following sections.

Let S_g and S_t be two segmentations according to Definition 7 or Definition 8, depending on whether the segmentations are binary or fuzzy, which is to be given in the context. Note that binary segmentation is just a special case of the fuzzy segmentation. Let S_g denote the ground truth segmentation and S_t denote the segmentation being evaluated. The segmentations S_g and S_t have the assignment functions f_g and f_t respectively, which provide the memberships of the voxels in the foreground. Note that in this chapter, unless it is explicitly stated, we only deal with partitions with two classes, namely the class of interest (e.g. anatomy or feature) and the background. We always assume that the first class (S_g^1, S_t^1) is the class of interest and the second class (S_g^2, S_t^2) is the background.

In the remainder of this appendix, we define the foundation of methods and algorithms used to compute all the metrics presented in Table 2.1. We structure the discussion in this appendix to follow the metric grouping given in the column “category”.

A.3 Spatial Overlap Based Metrics

Because all spatial overlap based metrics are based on four basic overlap cardinalities of the so-called confusion matrix, namely the true positives (TP), the false positives (FP), the true negatives (TN), and the false negatives (FN), we define these cardinalities for crisp as well as fuzzy segmentations, then we define the metrics based on them.

A.3.1 Calculation of Overlap Based Metrics

In this section, we define each of the overlap based metrics in Table 2.1 based on the basic cardinalities in Equation A.1 (crisp) or Equations A.2 to A.5 (fuzzy).

The Dice coefficient [Dic45] (*DICE*), also called the overlap index, is the most used metric in validating medical volume segmentations. In addition to the direct comparison between automatic and ground truth segmentations, it is common to use *DICE* to measure reproducibility (repeatability). Zou et al. [ZWB⁺04] used *DICE* as a measure of the reproducibility as a statistical validation of manual annotation where segmenters repeatedly annotated the same MRI image, then the pair-wise overlap of the repeated segmentations is calculated using *DICE*, which is defined by

$$DICE = \frac{2|S_g^1 \cap S_t^1|}{|S_g^1| + |S_t^1|} = \frac{2TP}{2TP + FP + FN} \quad (\text{A.6})$$

The Jaccard index (*JAC*) [Jac12] between two sets is defined as the intersection between them divided by their union, that is

$$JAC = \frac{|S_g^1 \cap S_t^1|}{|S_g^1 \cup S_t^1|} = \frac{TP}{TP + FP + FN} \quad (\text{A.7})$$

We note that *JAC* is always larger than *DICE* except at the extrema $\{0, 1\}$ where they are equal. Furthermore the two metrics are related according to

$$\begin{aligned} JAC &= \frac{|S_g^1 \cap S_t^1|}{|S_g^1 \cup S_t^1|} = \frac{2|S_g^1 \cap S_t^1|}{2(|S_g^1| + |S_t^1| - |S_g^1 \cap S_t^1|)} \\ &= \frac{DICE}{2 - DICE} \end{aligned} \quad (\text{A.8})$$

Similarly, one can show that

$$DICE = \frac{2JAC}{1 + JAC} \quad (\text{A.9})$$

That means that both of the metrics measure the same aspects and provide the same system ranking. Therefore, it does not provide additional information to select both of them together as validation metrics as done in [CdLGBC09][AFNIS13][CCH06].

True Positive Rate (*TPR*), also called Sensitivity and Recall, measures the portion of positive voxels in the ground truth that are also identified as positive by the segmentation being evaluated. Analogously, True Negative Rate (*TNR*), also called Specificity, measures the portion of negative voxels (background) in the ground truth segmentation that are also identified as negative by the segmentation being evaluated. However these two measures are not common as evaluation measures of medical image segmentation because of their sensitivity to segment size, i.e. they penalize errors in small segments more than in large segments [GJC01] [FC05] [ULZ⁺06]. Note that the terms positive and negative are rather for crisp segmentation. However, the generalization in Equations A.2 to A.5 extends the meaning of the terms to grade agreement. These two measures are defined as follows:

$$Recall = Sensitivity = TPR = \frac{TP}{TP + FN} \quad (\text{A.10})$$

$$Specificity = TNR = \frac{TN}{TN + FP} \quad (A.11)$$

There are two other measures that are related to these metrics, namely the false positive rate (FPR), also called Fallout, and the false negative rate (FNR). They are defined by

$$Fallout = FPR = \frac{FP}{FP + TN} = 1 - TNR \quad (A.12)$$

$$FNR = \frac{FN}{FN + TP} = 1 - TPR \quad (A.13)$$

The equivalence in Equations A.12 and A.13 implies that only one of each two equivalent measures should be selected for validation and not both of them together [ULZ⁺06], i.e. either FPR or TNR and analogously, either FNR or TPR . Another related measure is the precision, also called the positive predictive value (PPV) which is not commonly used in validation of medical images, but it is used to calculate the F-Measure. It is defined by

$$Precision = PPV = \frac{TP}{TP + FP} \quad (A.14)$$

F_β -Measure (FMS_β) was first introduced in [Chi92] as an evaluation measure for information retrieval. The F_β -Measure is a trade-off between PPV (precision, defined in Equation A.14) and TPR (recall, defined in Equation A.10). the F_β -Measure is given by

$$FMS_\beta = \frac{(\beta^2 + 1) \cdot PPV \cdot TPR}{\beta^2 \cdot PPV + TPR} \quad (A.15)$$

However, FMS is a special case of the Van Rijsbergen's effectiveness measure introduced in [Rij79]. FMS_β can be derived by setting $\alpha = \frac{1}{\beta^2 + 1}$ in Rijsbergen's effectiveness measure given by

$$E = 1 - \frac{1}{\alpha \frac{1}{PPV} + (1 - \alpha) \frac{1}{TPR}} \quad (A.16)$$

With $\beta = 1.0$ (precision and recall are equally important), we get the special case F_1 -Measure (FMS_1); we call it FMS for simplicity. It is also called the harmonic mean and given by

$$FMS = \frac{2 \cdot PPV \cdot TPR}{PPV + TPR} \quad (A.17)$$

Here, we note that the FMS is mathematically equivalent to $DICE$. This follows from a trivial substitution for TPR and PPV in Equation A.17 by their values in Equations A.10 and A.14, i.e. $TPR = \frac{TP}{TP + FN}$ and $PPV = \frac{TP}{TP + FP}$. The simplification directly results in the definition of $DICE$ in Equation A.6, i.e. $DICE = \frac{2TP}{2TP + FP + FN}$.

The global consistency error (GCE) [MFTM01] is an error measure between two segmentations. Let $R(S, x)$ be defined as the set of all voxels that reside in the same

region of segmentation S where the voxel x resides. For the two segmentations S_1 and S_2 , the error at voxel x , $E(S_1, S_2, x)$ is defined as

$$E(S_t, S_g, x) = \frac{|R(S_t, x) \setminus R(S_g, x)|}{|R(S_t, x)|} \quad (\text{A.18})$$

Note that E is not symmetric. The global consistency error (GCE) is defined as the error averaged over all voxels and is given by

$$GCE(S_t, S_g) = \frac{1}{|X|} \min \left\{ \sum_i^{|X|} E(S_t, S_g, x_i), \sum_i^{|X|} E(S_g, S_t, x_i) \right\} \quad (\text{A.19})$$

Equation A.19 can be expressed in terms of the four cardinalities defined in Equations A.2 to A.4 to get the GCE between the (fuzzy) segmentations S_g and S_t as follows

$$GCE = \frac{1}{|X|} \min \left\{ \frac{FN(FN + 2TP)}{TP + FN} + \frac{FP(FP + 2TN)}{TN + FP}, \frac{FP(FP + 2TP)}{TP + FP} + \frac{FN(FN + 2TN)}{TN + FN} \right\} \quad (\text{A.20})$$

A.3.2 Overlap Measures for Multiple Labels

All the overlap measures presented previously assume segmentations with only one label. However, it is common to compare segmentations with multiple labels, e.g. two-label tumor segmentation (core and edema). Obviously, one way is to compare each label separately using the overlap measures presented previously, but this would lead to the problem of how to average the individual similarities to get a single score. For evaluating segmentations with multiple classes, we use the overlap measures proposed by Crum et. al [CCH06], namely $DICE_{ml}$ and JAC_{ml} which are generalized to segmentations with multiple labels. For the segmentations A and B

$$JAC_{ml} = \frac{\sum_{labels, l} \alpha l \sum_{voxels, i} MIN(A_{li}, B_{li})}{\sum_{labels, l} \alpha l \sum_{voxels, i} MAX(A_{li}, B_{li})} \quad (\text{A.21})$$

where A_{li} is the value of voxel i for label l in segmentation A (analogously for B_{li}) and αl is a label-specific weighting factor that affects how much each label contributes to the overlap accumulated over all labels. Here, the $MIN(\cdot)$ and $MAX(\cdot)$ are the norms used to represent the intersection and union in the fuzzy case. $DICE_{ml}$ can be then calculated from JAC according to Equation A.9, i.e. $DICE_{ml} = 2JAC_{ml}/(1 + JAC_{ml})$. Note that the equations above assume the general case of multiple label and fuzzy segmentation. However, in multiple label segmentations, voxel values mostly represent the labels (classes) rather than probabilities, which means that in most available image formats, there are either multiple label or fuzzy segmentations [CCH06].

A.4 Volume Based Metrics

As the name implies, volumetric similarity (VS) is a measure that considers the volumes of the segments to indicate similarity. There is more than one definition for the volumetric similarity in the literature, however we consider the definition in [RPR13a], [VYPP11], [RPR13b] and [CdLGBC09], namely the absolute volume difference divided by the sum of the compared volumes. We define the Volumetric Similarity (VS) as $1 - VD$ where VD is the volumetric distance. That is

$$VS = 1 - \frac{||S_t^1| - |S_g^1||}{|S_t^1| + |S_g^1|} = 1 - \frac{|FN - FP|}{2TP + FP + FN} \quad (\text{A.22})$$

Note that although the volumetric similarity is defined using the four cardinalities, it is not considered an overlap-based metric, since here the absolute volume of the segmented region in one segmentation is compared with the corresponding volume in the other segmentation. This means that the overlap between the segments is absolutely not considered. Actually, the volumetric similarity can have its maximum value even when the overlap is zero. More details in Section 2.5.

A.5 Pair Counting Based Metrics

In this section, pair-counting based metrics, namely the Rand index and its extensions, are defined. At first we define the four basic pair-counting cardinalities, namely a , b , c , and d for crisp and fuzzy segmentations and then we define the metrics based on these cardinalities.

A.5.1 Basic Cardinalities

Given two partitions of the point set X being compared, let P be the set of $\binom{n}{2}$ tuples that represent all possible object pairs in $X \times X$. These tuples can be grouped into four categories depending on where the objects of each pair are placed according to each of the partitions. That is, each tuple $(x_i, x_j) \in P$ is assigned to one of four groups whose cardinalities are a , b , c , and d .

- Group I: if x_i and x_j are placed in the same subset in both partitions S_g and S_t . We define a as the cardinality of Group I.
- Group II: if x_i and x_j are placed in the same subset in S_g but in different subsets in S_t . We define b as the cardinality of Group II.
- Group III: if x_i and x_j are placed in the same subset in S_t but in different subsets in S_g . We define c as the cardinality of Group III.
- Group IV: if x_i and x_j are placed in different subsets in both partitions S_g and S_t . We define d as the cardinality of Group IV.

Note that the count of tuples in Groups I and IV represents the agreement ($a + d$) whereas the count of tuples in Groups II and III ($b + c$) represents the disagreement between the two partitions.

Obviously, because there are $\binom{n}{2} = n(n - 1)/2$ tuples, the direct calculation of these parameters needs $O(n^2)$ runtime. However, Brennan and Light [BL74] showed that these cardinalities can be calculated using the values of the confusion matrix without trying all pairs and thus avoiding the $O(n^2)$ complexity, that is

$$a = \frac{1}{2} \sum_{i=1}^r \sum_{j=1}^s m_{ij}(m_{ij} - 1) \quad (\text{A.23})$$

$$b = \frac{1}{2} \left(\sum_{j=1}^s m_{.j}^2 - \sum_{i=1}^r \sum_{j=1}^s m_{ij}^2 \right) \quad (\text{A.24})$$

$$c = \frac{1}{2} \left(\sum_{i=1}^r m_{i.}^2 - \sum_{i=1}^r \sum_{j=1}^s m_{ij}^2 \right) \quad (\text{A.25})$$

$$d = n(n - 1)/2 - (a + b + c) \quad (\text{A.26})$$

where r and s are the numbers of classes in the segmentations being compared (e.g. 2 for a 2-class segmentation), m_{ij} is the confusion matrix (Table A.1), $m_{i.}$ denotes the sum over the i th row, and $m_{.j}$ denotes the sum over the j th column. Note that here, in contrast to the overlap based metrics, there is no restriction on the number of classes in the compared partitions. However, for the evaluation of 3D medical segmentation, we are interested in segmentations with only two classes, namely the anatomy and the background; i.e. $r = s = 2$. We define the four cardinalities for this special case, more specifically for the segmentations S_g and S_t defined in Appendix A.1 based on the four overlap parameters defined in Appendix A.2

$$a = \frac{1}{2} \left[TP(TP - 1) + FP(FP - 1) \right. \\ \left. + TN(TN - 1) + FN(FN - 1) \right] \quad (\text{A.27})$$

$$b = \frac{1}{2} \left[(TP + FN)^2 + (TN + FP)^2 \right. \\ \left. - (TP^2 + TN^2 + FP^2 + FN^2) \right] \quad (\text{A.28})$$

$$c = \frac{1}{2} \left[(TP + FP)^2 + (TN + FN)^2 \right. \\ \left. - (TP^2 + TN^2 + FP^2 + FN^2) \right] \quad (\text{A.29})$$

$$d = n(n - 1)/2 - (a + b + c) \\ = n(n - 1)/2 - \quad (\text{A.30})$$

A.5.2 Generalization to Fuzzy Segmentations

As mentioned above, since the cardinalities a , b , c , and d are by definition based on grouping all the pairwise tuples defined on S_g and S_t , this requires processing $n(n-1)/2$ tuples which means a direct computation of these cardinalities for fuzzy segmentations takes $O(n^2)$ runtime. For medical segmentation, this complexity could be a problem since the number of voxels in a medical volume could reach 8-digit numbers. Methods (Huellermeier et al [HRHS12], Brouwer [Bro09], Campello [Cam07]) have been proposed that calculate the Rand index and its extension for fuzzy segmentations using different approaches. None of these approaches is efficiently applicable in the 3D medical imaging domain because they all have a run time complexity of $O(n^2)$. However, Anderson et al. [ABPK10] proposed a method that calculates the four cardinalities for fuzzy sets in $O(n)$ runtime. This is achieved by combining two already known strategies: (i) calculating the confusion matrix for fuzzy sets using some agreement function e.g. Equations A.2 to A.5 and (ii) calculating the four cardinalities by applying Equations A.23 to A.26 on the values of the fuzzy confusion matrix calculated in (i). We use this approach which means that Equations A.27 to A.30 already provide the fuzzy cardinalities according to [ABPK10], given the parameters TP , FP , TN and FN are calculated for fuzzy sets. In the next subsection, the Rand index and the adjusted rand index are calculated based on these cardinalities.

A.5.3 Calculation of Pair-counting Based Metrics

The Rand Index (RI), proposed by W. Rand [Ran71] is a measure of similarity between clusterings. One of its important properties is that it is not based on labels and thus can be used to evaluate clusterings as well as classifications. The RI between two segmentations S_g and S_t is defined as

$$RI(S_g, S_t) = \frac{a + b}{a + b + c + d} \quad (\text{A.31})$$

where a , b , c , d are the cardinalities defined in Equations A.27 to A.30.

The Adjusted Rand Index (ARI), proposed by Hubert and Arabie [HA85], is a modification of the Rand Index that considers a correction for chance. It is given by

$$ARI = \frac{\sum_{ij} \binom{m_{ij}}{2} - \sum_i \binom{m_{i.}}{2} \sum_j \binom{m_{.j}}{2} / \binom{n}{2}}{\frac{1}{2} [\sum_i \binom{m_{i.}}{2} + \sum_j \binom{m_{.j}}{2}] - \sum_i \binom{m_{i.}}{2} \sum_j \binom{m_{.j}}{2} / \binom{n}{2}} \quad (\text{A.32})$$

where n is the object count, m_{ij} is the confusion matrix (Table A.1), $m_{i.}$ denotes the sum over the i th row, and $m_{.j}$ denotes the sum over the j th column.

The idea behind this correction for chance is to abstract the agreement caused by chance from the enumerator, which is estimated here as the expectation value of the number of tuples from Group I, i.e. pairs in which the objects are placed in the same class in the first segmentation and in the same class in the second segmentation. This

expectation value is given by

$$E \left[\sum_{ij} \binom{m_{ij}}{2} \right] = \sum_i \binom{m_{i.}}{2} \sum_j \binom{m_{.j}}{2} / \binom{n}{2} \quad (\text{A.33})$$

The *ARI* can be expressed by the four cardinalities as

$$ARI = \frac{2(ad - bc)}{c^2 + b^2 + 2ad + (a + d)(c + b)} \quad (\text{A.34})$$

A.6 Information Theoretic Based Metrics

The Mutual Information (*MI*) of two variables is a measure of the amount of information one variable has about the other, or simply the amount of information they share. Or in other words, the reduction in uncertainty of one variable, given that the other is known [CT91]. It was firstly used as a measure of similarity between images by Viola and Wells [VW97]. Later, Russakoff et al. [RTR⁺04] used the *MI* as a similarity measure between image segmentations; in particular, they calculate the *MI* based on regions (segments) instead of individual pixels. The *MI* is related to the marginal entropy $H(S)$ and the joint entropy $H(S_1, S_2)$ between images defined as

$$H(S) = - \sum_i p(S^i) \log p(S^i) \quad (\text{A.35})$$

$$H(S_1, S_2) = - \sum_{ij} p(S_1^i, S_2^j) \log p(S_1^i, S_2^j) \quad (\text{A.36})$$

where $p(x, y)$ is joint probability, S^i are the regions (segments) in the image segmentations and $p(S^i)$ are the probabilities of these regions that can be expressed in terms of the four cardinalities *TP*, *FP*, *TN* and *FN*, which are calculated for the fuzzy segmentations (S_g and S_t) in Equations A.2 to A.5 as follows

$$\begin{aligned} p(S_g^1) &= (TP + FN)/n \\ p(S_g^2) &= (TN + FN)/n \\ p(S_t^1) &= (TP + FP)/n \\ p(S_t^2) &= (TN + FP)/n \end{aligned} \quad (\text{A.37})$$

where $n = TP + FP + TN + FN$ is the total number of voxels. Because *TP*, *TN*, *FP* and *FN* are by definition cardinalities of disjoint sets that partition the volume, the joint probabilities are given by

$$p(S_1^i, S_2^j) = \frac{|S_1^i \cap S_2^j|}{n} \quad (\text{A.38})$$

which implies

$$\begin{aligned}
p(S_1^1, S_2^1) &= \frac{TP}{n} \\
p(S_1^1, S_2^2) &= \frac{FN}{n} \\
p(S_1^2, S_2^1) &= \frac{FP}{n} \\
p(S_1^2, S_2^2) &= \frac{TN}{n}
\end{aligned} \tag{A.39}$$

The MI is then defined as

$$MI(S_g, S_t) = H(S_g) + H(S_t) - H(S_g, S_t) \tag{A.40}$$

The Variation of Information (*VOI*) between two variables is a measure of the amount of information lost (or gained) when one variable is changed to the other. Marin [Mei03] first introduced the *VOI* measure for comparing clustering partitions. The *VOI* is defined using the entropy and mutual information as

$$VOI(S_g, S_t) = H(S_g) + H(S_t) - 2 MI(S_g, S_t) \tag{A.41}$$

A.7 Probabilistic Metrics

The Interclass Correlation (*ICC*) [SF79] is a measure of correlations between pairs of observations that don't necessarily have an order, or are not obviously labeled. It is common to use the *ICC* as a measure of conformity among observers; in our case it is used as a measure of consistency between two segmentations. *ICC* is given by

$$ICC = \frac{\sigma_S^2}{\sigma_S^2 + \sigma_\epsilon^2} \tag{A.42}$$

where σ_S denotes variance caused by differences between the segmentations and σ_ϵ denotes variance caused by differences between the points in the segmentations [SF79]. Applied to the segmentations S_g and S_t , *ICC* is defined as

$$\begin{aligned}
ICC &= \frac{MS_b - MS_w}{MS_b + (k-1)MS_w} \quad \text{with} \\
MS_b &= \frac{2}{n-1} \sum_x (m(x) - \mu)^2 \\
MS_w &= \frac{1}{n} \sum_x (f_g(x) - m(x))^2 + (f_t(x) - m(x))^2
\end{aligned} \tag{A.43}$$

where MS_b denotes the mean squares between the segmentations (called between group MS), MS_w denotes the mean squares within the segmentations (called within group MS), k

is the number of observers which is 2 in case of comparing two segmentations, μ is the grand mean, i.e. the mean of the means of the two segmentations, and $m(x) = (f_g(x) + f_t(x))/2$ is the mean at voxel x .

The Probabilistic Distance (PBD) was developed by Gerig et al. [GJC01] as a measure of distance between fuzzy segmentations. Given two fuzzy segmentations, A and B , then the PBD is defined by

$$PBD(A, B) = \frac{\int |P_A - P_B|}{2 \int P_{AB}} \quad (\text{A.44})$$

where $P_A(x)$ and $P_B(x)$ are the probability distributions representing the segmentations and P_{AB} is their pooled joint probability distribution. Applied on S_g and S_t , defined in Appendix A.1, the PBD is defined as

$$PBD(S_g, S_t) = \frac{\sum_x |f_g(x) - f_t(x)|}{2 \sum_x f_g(x) f_t(x)} \quad (\text{A.45})$$

The Cohen Kappa Coefficient (KAP), proposed in [Coh60], is a measure of agreement between two samples. As an advantage over other measures, KAP takes into account the agreement caused by chance, which makes it more robust. KAP is given by

$$KAP = \frac{P_a - P_c}{1 - P_c} \quad (\text{A.46})$$

where P_a is the agreement between the samples and P_c is the hypothetical probability of chance agreement. The same can be expressed in the form of frequencies to facilitate the computation as follows

$$KAP = \frac{f_a - f_c}{n - f_c} \quad (\text{A.47})$$

where $n = |X|$ is the total number of observations, in our case the voxels. The terms in Equation A.47 can be expressed in terms of the four overlap cardinalities, calculated for fuzzy segmentations (Equations A.2 to A.5), to get

$$\begin{aligned} f_a &= TP + TN \\ f_c &= \frac{(TN + FN)(TN + FP) + (FP + TP)(FN + TP)}{n} \end{aligned} \quad (\text{A.48})$$

The ROC curve (Receiver Operating Characteristic) is the plot of the true positive rate (TPR) against the false positive rate (FPR). The area under the ROC curve (AUC) was first presented by Hanley and McNeil [HM82] as a measure of accuracy in diagnostic radiology. Later, Bradley [Bra97] investigated its use in validating machine learning algorithms. The ROC curve, as a plot of TPR against FPR , normally assumes more than one measurement. For the case where a test segmentation is compared to a ground truth segmentation (one measurement), we consider a definition of the AUC

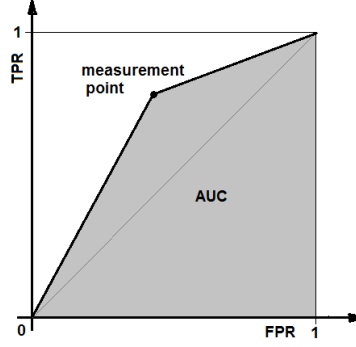


Figure A.1: Illustration of the AUC when only one measurement is available according to [Pow11]. In this case, the AUC is area of the trapezoid defined by the measurement point and the lines $TPR = 0$ and $FPR = 1$.

according to [Pow11], namely the area of the trapezoid defined by the measurement point and the lines $TPR = 0$ and $FPR = 1$ as illustrated in Figure A.1, which is given by

$$\begin{aligned}
 AUC &= 1 - \frac{FPR + FNR}{2} \\
 &= 1 - \frac{1}{2} \left(\frac{FP}{FP + TN} + \frac{FN}{FN + TP} \right)
 \end{aligned} \tag{A.49}$$

A.8 Spatial Distance Based Metrics

Spatial distance based metrics are widely used in the evaluation of image segmentation as dissimilarity measures. They are recommended when the segmentation overall accuracy, e.g. the boundary delineation (contour), of the segmentation is of importance [FC05]. Distance-based metrics is the only category of metrics to take into consideration the spatial position of voxels. More about the properties of distance metrics is in Section 4.6. In this section, we present three distance metrics, namely the Hausdorff distance, the Average distance and the Mahalanobis distance. All distances calculated in this section are in voxels, which means the voxel size is not taken into account.

A.8.1 Distance Between Crisp Volumes

The Hausdorff Distance (HD) between two finite point sets A and B is defined by

$$HD(A, B) = \frac{h(A, B) + h(B, A)}{2} \tag{A.50}$$

where $h(A, B)$ is called the directed Hausdorff distance and given by

$$h(A, B) = \max_{a \in A} \min_{b \in B} \|a - b\| \tag{A.51}$$

where $\|a - b\|$ is some norm, e.g. Euclidean distance. That is the directed Hausdorff distance $h(A, B)$ is the maximum of distances between each point $x \in A$ to its nearest neighbor $y \in B$. An algorithm that directly calculates the HD according to Equation A.51 takes an execution time of $O(|A||B|)$. There are many algorithms that calculate the HD with lower complexity. We use the algorithm proposed in Chapter 3 which calculates the HD in a nearly-linear time complexity.

The HD is generally sensitive to outliers. Because noise and outliers are common in medical segmentations, it is not recommended to use the HD directly [GJC01] [ZL04]. However, the quantile method proposed by Huttenlocher et al. [HKR93] is one way to handle outliers. According to the Hausdorff quantile method, the HD_q is defined to be the q^{th} quantile of distances instead of the maximum, so that possible outliers are excluded, where q is selected depending on the application and the nature of the measured point sets.

The Average Distance, or the Average Hausdorff Distance (AVD), is the HD averaged over all points. The AVD is known to be stable and less sensitive to outliers than the HD . It is defined by

$$AVD(A, B) = \max(d(A, B), d(B, A)) \quad (\text{A.52})$$

where $d(A, B)$ is the directed Average Hausdorff distance that is given by

$$d(A, B) = \frac{1}{n} \sum_{a \in A} \min_{b \in B} \|a - b\| \quad (\text{A.53})$$

where $n = |X|$ is the number of voxels. We use the algorithm proposed in Chapter 3 which efficiently calculates the AVD between image segmentations.

The Mahalanobis Distance (MHD) [Mah36] between two points in a point cloud, in contrast to the Euclidean distance, takes into account the correlation of all points in the point cloud containing the two points. The MHD between the points x and y in the point cloud A is given by

$$MHD(x, y) = \sqrt{(x - y)^T S^{-1} (x - y)} \quad (\text{A.54})$$

where S^{-1} is the inverse of the covariance matrix S of the point cloud and the superscript T denotes the matrix transpose. Note that x and y are two points in the same point cloud, but in the validation of image segmentation, two point clouds are compared. For this task, we use the variant of MHD according to G. J. McLachlan [McL99], where the MHD is calculated between the means of the compared point clouds and the common covariance matrix of them is considered as S . Hence the Mahalanobis distance $MHD(X, Y)$ between the point sets X and Y is

$$MHD(X, Y) = \sqrt{(\mu_x - \mu_y)^T S^{-1} (\mu_x - \mu_y)} \quad (\text{A.55})$$

where μ_x and μ_y are the means of the point sets and the common covariance matrix of the two sets is given by

$$S = \frac{n_1 S_1 + n_2 S_2}{n_1 + n_2} \quad (\text{A.56})$$

where S_1, S_2 are the covariance matrices of the voxel sets and n_1, n_2 are the numbers of voxels in each set.

A.8.2 Extending the Distances to Fuzzy Volumes

Different approaches have been proposed to measure the spatial distance between fuzzy images. The approaches described in [SLN11] are based on defuzzification (finding a crisp representation) either by minimizing the feature distance, which leads to the problem of selecting the features, or by finding crisp representations with a higher resolution, which leads to multiplication of the grid dimensions and therefore negatively impacts the efficiency of time consuming algorithms, like HD and AVD . For evaluating 3D medical segmentation, we use a discrete form of the approach proposed in [ZKB87] i.e. the average of distances at different α -cuttings depending on a given number of cutting levels k . The HD distance between the fuzzy segmentations A and B is thus given by

$$\overline{HD}_k(A, B) = \frac{1}{k} \sum_{i=1}^k HD_{\frac{i}{k}}(A, B) \quad (\text{A.57})$$

$$HD_\alpha(A, B) = HD(A_\alpha, B_\alpha) \quad (\text{A.58})$$

where A_α and B_α are the crisp representations resulting from thresholding the fuzzy volumes A and B at cutting level α , HD_α is the HD at cutting level α , and $k > 0$ is an integer that gives the number of cutting levels considered.

Analogously, the AVD and MHD between the fuzzy volumes A and B are given by

$$\overline{AVD}_k(A, B) = \frac{1}{k} \sum_{i=1}^k AVD(A_{\frac{i}{k}}, B_{\frac{i}{k}}) \quad (\text{A.59})$$

$$\overline{MHD}_k(A, B) = \frac{1}{k} \sum_{i=1}^k MHD(A_{\frac{i}{k}}, B_{\frac{i}{k}}) \quad (\text{A.60})$$

If the parameters k and α are omitted, i.e. HD , AVD and MHD , we assume distances at the cutting level $\alpha = 0.5$.

A.9 Multiple Definition of Metrics in the Literature

We present three examples representing three categories of inconsistency in the literature regarding the definition of the metrics to underline the need of a standardization of evaluation metrics and motivate a standard evaluation tool for medical segmentations. The first category is caused by misinterpretation resulting in misleading definitions, for example the confusion of the pair counting cardinalities (a , b , c and d) with the overlap cardinalities (TP , FP , TN and FN). In some papers [SY02] [ABPK10] [HRHS12] [Cam07], the pair-counting cardinalities are used in place of the overlap cardinalities although they are mathematically and semantically different. According to the definition,

the pair-counting cardinalities result from grouping $n(n - 1)/2$ tuples defined on $X \times X$ (Section A.5.1) whereas the overlap-based cardinalities (Section A.2) result from the class overlap i.e. pairwise comparison of n voxel assignments. In the papers mentioned above, several overlap-based metrics including the Jaccard index are defined using the pair-counting cardinalities in place of the overlap cardinalities. To illustrate how strongly the results differ in the two cases, we show examples in Table A.2. In each example, the partitions $P1$ and $P2$ are compared using the Jaccard index which is calculated in two ways: the first (JAC_1) using the overlap cardinalities according to [Jac12] and [JD88], the second (JAC_2) using the pair counting cardinalities according to [SY02], [ABPK10], [HRHS12] and [Cam07]. The values are different except in the first example.

Table A.2: Five examples show that the pair counting cardinalities (a , b , c , and d) cannot be used in place of the overlap cardinalities (TP , FP , FN , and TN) to calculate the Jaccard index.

P1	P2	TP	FP	FN	TN	JAC_1	a	b	c	d	JAC_2
1,0,1,1	1,1,0,0	1	2	1	0	0.25	1	2	1	2	0.25
1,1,1,1	0,0,0,1	1	3	0	0	0.25	3	3	0	0	0.5
0,1,0,1	1,1,0,0	1	1	1	1	0.33	0	2	2	2	0.0
0,0,0,0	0,0,0,1	0	0	1	3	0.0	3	0	3	0	0.5
1,0,0,1	1,1,0,1	2	0	1	1	0.67	1	2	1	2	0.25

The second category is naming inconsistency, where the same name is used to denote two different metrics. One example is the volumetric similarity (VS). While VS is defined in [RPR13a], [VYPP11], [RPR13b] and [CdLGBC09] as the absolute volume difference divided by the sum of the compared volumes (Equation A.22), there is another metric definition under the same name in [ISHV⁺12] defined as twice the volume of the intersection divided by the volume sum in percent, i.e.

$$VS = 2 \frac{|S_t \cap S_g|}{|S_t + S_g|} \cdot 100\% \quad (\text{A.61})$$

The last category is the multiple definition that stems from different theoretical approaches for estimating the same value. For example, the Interclass Correlation (ICC) has an early definition proposed by Fisher [Fis54]. Later, several versions of the ICC have been proposed. Some of these versions was designed to meet the needs of particular purposes, and others were proposed as alternative estimators. Shrout et al. [SF79] has discussed six versions of the ICC , one of them is the definition in Equation A.42. Note that this category is different from the second category in that here the versions of the same metrics aim to estimate the same statistic, while in the second category, the same name has been used to denote another metric that is totally different.

A.10 Implementation

The 20 metrics, identified in the literature review (Table 2.1) have been implemented in a tool named EvaluateSegmentation and provided as an open source project available under <http://github.com/codalab/EvaluateSegmentation>.

EvaluateSegmentation is an efficient command line tool that compares two 2D or 3D medical segmentations using the 20 evaluation metrics presented in Table 2.1. Being a pure command line tool without a GUI interface makes it suitable to be called using automation scripts when many segmentations are to be evaluated. The implementation has been generally designed to take advantage of the relations between the 20 implemented metrics represented in their definition in order to make use of the synergy between them to avoid repeating operations and hence to save execution time and memory. By default the evaluation result is displayed in a readable format on the System out, but it can be optionally saved as an XML file in a given path, e.g. to be parsed and processed by other tools.

The proposed tool uses the ITK Library, in particular the input/output layer, to read medical images, which gives it two important properties:

- The tool is fully compatible with a wide spectrum of medical image formats, namely all formats supported by the ITK framework.
- The tool is invariant to changes in file formats, e.g. it is also compatible with formats that are changed, or even introduced after its implementation. That is because the job reading the images is done by the ITK library, which is permanently maintained to support new standards.

EvaluateSegmentation is implemented in C++ using the CMake framework, which makes it operating system and compiler independent. CMake (www.cmake.org) is an open source platform that enables programs implemented in native languages like C++ to be operating system and compiler independent; it was originally created and funded by the National Library of Medicine (NLM) to provide a sufficient way for distributing the ITK application. The source of the project as well as builds for some operating systems are available under <http://github.com/codalab/EvaluateSegmentation>. To build the EvaluateSegmentation for any operating system, using any compiler, two resource components are required (i) the source code of the project and (ii) the ITK Library available as open source under <http://www.itk.org>.

Efficiency in speed as well as in memory usage is a critical point in metric calculation. EvaluateSegmentation uses various optimization techniques to achieve this purpose. More information about the efficiency optimization used in EvaluateSegmentation is available in Chapter 3.

Bibliography

- [ABJ91] Helmut Alt, Bernd Behrends, and Bloemer Johannes. Approximate matching of polygonal shapes (extended abstract). In *Proceedings of the Seventh Annual Symposium on Computational Geometry*, pages 186–193, 1991.
- [ABPK10] Derek T. Anderson, James C. Bezdek, Mihail Popescu, and James M. Keller. Comparing fuzzy, probabilistic, and possibilistic partitions. *IEEE Transactions on Fuzzy Systems*, 18(5):906–918, 2010.
- [ACMS12] James Allan, Bruce Croft, Alistair Moffat, and Mark Sanderson. Frontiers, challenges, and opportunities for information retrieval. *SIGIR Forum*, 46(1), May 2012.
- [AFNIS13] Ali Qusay Al-Faris, Umi Kalthum Ngah, Nor Ashidi Mat Isa, and Ibrahim Lutfi Shuaib. MRI breast skin-line segmentation and removal using integration method of level set active contour and morphological thinning algorithms. *Journal of Medical Sciences*, May 2013.
- [AGAV09] Enrique Amigo, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Information Retrieval*, 12(4):461–486, August 2009.
- [AHK01] Charu C. Aggarwal, Alexander Hinneburg, and Daniel A. Keim. On the surprising behavior of distance metrics in high dimensional space. In *Proceedings of the 8th International Conference on Database Theory*, pages 420–434, 2001.
- [Aic97] Oswin Aichholzer. *Combinatorial & Computational Properties of the Hypercube*. PhD thesis, IGI-TU Graz, Austria, 1997.
- [Ata83] Mikhail J. Atallah. A linear time algorithm for the Hausdorff distance between convex polygons. *Information Processing Letters*, 17(4):207–209, 1983.
- [AV08] Leif Azzopardi and Vishwa Vinay. Retrievability: An evaluation measure for higher order information access tasks. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, pages 561–570. ACM, 2008.

- [BB61] Richard Bellman and Richard Ernest Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1961.
- [BGRS99] Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. When is nearest neighbor meaningful? In *International Conference on Database Theory*, pages 217–235. Springer Berlin Heidelberg, 1999.
- [BL74] Robert L. Brennan and Richard J. Light. Measuring agreement when two observers classify people into categories not defined in advance. *British Journal of Mathematical and Statistical Psychology*, 27(2), 1974.
- [BLVD10] Halim Benhabiles, Guillaume Lavoue, Jean Phillippe Vandeborre, and Mohamed Daoudi. A subjective experiment for 3d-mesh segmentation evaluation. In *IEEE International Workshop on Multimedia Signal Processing (MMSP)*, 2010.
- [BM92] Paul J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.
- [BM13a] Gerard Biau and D.M. Mason, David. High-dimensional p-norms. arXiv:1311.0587 [math.ST], 2013.
- [BM13b] Luca Busin and Stefano Mizzaro. Axiometrics: An axiomatic approach to information retrieval effectiveness metrics. In *Proceedings of the 2013 Conference on the Theory of Information Retrieval*, pages 8:22–8:29, New York, NY, USA, 2013.
- [BPA⁺08] KO Babalola, B. Patenaude, P. Aljabar, J. Schnabel, D. Kennedy, W. Crum, S. Smith, T. F. Cootes, M. Jenkinson, and D. Rueckert. Comparison and evaluation of segmentation techniques for subcortical structures in brain MRI. *Medical image computing and computer-assisted intervention*, 2008.
- [Bra97] Andrew P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 1997.
- [Bro09] Roelof K. Brouwer. Extending the Rand, adjusted Rand and Jaccard indices to fuzzy partitions. *Journal of Intelligent Information Systems*, 32(3):213–235, 2009.
- [Buc93] Chris Buckley. The importance of proper weighting methods. In *Proceedings of the Workshop on Human Language Technology, HLT 93*, pages 349–352, 1993.
- [BV00] Chris Buckley and Ellen M. Voorhees. Evaluating evaluation measure stability. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 33–40. ACM, 2000.

- [BZ11] Roi Blanco and Hugo Zaragoza. Beware of relatively large but meaningless improvements. Technical report, Yahoo! Research 2011-001, 2011.
- [Cam07] Ricardo J. G. B. Campello. A fuzzy extension of the Rand index and other related indexes for clustering and classification assessment. *Pattern Recognition Letters*, 28(7):833–841, 2007.
- [CCH06] William R. Crum, Oscar Camara, and Derek L. G. Hill. Generalized overlap measures for evaluation and validation in medical image analysis. *IEEE Transactions on Medical Imaging*, 25(11):1451–1461, 2006.
- [CCUG11] Krzysztof Chris Ciesielski, Xinjian Chen, Jayaram K. Udupa, and George J. Grevera. Linear time algorithms for exact distance transform. *Journal of Mathematical Imaging and Vision*, 39(3):193–209, 2011.
- [CdLGBC09] Ruben Cardenas, Rodrigo de Luis-Garcia, and Meritxell Bach-Cuadra. A multidimensional segmentation evaluation for medical image data. *Computer Methods and Programs in Biomedicine*, 96(2):108–124, 2009.
- [Chi92] Nancy Chinchor. MUC-4 evaluation metrics. In *Proceedings of the 4th Conference on Message Understanding*, pages 22–29, 1992.
- [CHL⁺06] X. Cai, Y. Hou, C. Li, J. Lee, and W.G. Wee. Evaluation of two segmentation methods on MRI brain tissue structures. *Conference of the IEEE Engineering in Medicine and Biology Society*, 2006.
- [CK97] Vikram Chalana and Yongmin Kim. A methodology for evaluation of boundary detection algorithms on medical images. *IEEE Transactions on medical imaging*, 16(5), oct 1997.
- [CKL14] David Coeurjolly, Bertrand Kerautret, and Jacques-Olivier Lachaud. Extraction of connected region boundary in multidimensional images. *Image Processing On Line (IPOL)*, 4:30–43, 2014.
- [Coh60] Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20:37–46, 1960.
- [CR03] Haili Chui and Anand Rangarajan. A new point matching algorithm for non-rigid registration. *Computer Vision and Image Understanding*, 89(2):114–141, 2003.
- [CT91] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley-Interscience, New York, NY, USA, 1991.
- [Dem94] Pierre Demartines. *Analyse de donnees par reseaux de neurones auto-organises*. PhD thesis, Institut National Polytechnique de Grenoble, 1994.

- [Dem06] Janez Demsar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 17:30, 2006.
- [Dic45] Lee Raymond Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945.
- [DKC⁺11] Thomas M. Doring, Tadeu T.A. Kubo, L. Celso H. Cruz, Mario F. Juruena, Jiosef Fainberg, Romeu C. Domingues, and Emerson L. Gasparetto. Evaluation of hippocampal volume based on mr imaging in patients with bipolar affective disorder applying manual and automatic segmentation techniques. *Journal of Magnetic Resonance Imaging*, 33(3):565–572, 2011.
- [EAN08] Billet Eric, Fedorov Andriy, and Chrisochoides Nikos. The use of robust local Hausdorff distances in accuracy assessment for image alignment of brain MRI. *The Insight Journal*, 2008.
- [FC05] Aaron Fenster and Bernard Chiu. Evaluation of segmentation algorithms for medical imaging. In *Conference proceedings of the IEEE Engineering in Medicine and Biology Society*, volume 7, pages 7186–7189, 2005.
- [Fis54] Ronald Aylmer Fisher. *Statistical methods for research workers*. Oliver and Boyd, 1954.
- [Fra08] Damien François. *High-dimensional data analysis: optimal metrics and feature selection*. VDM Verlag, Germany, 2008.
- [FSS12] Arthur Flexer, Dominik Schnitzer, and Jan Schlueter. A mirex meta-analysis of hubness in audio music similarity. In *Proceedings of the 13th International Society for Music Information Retrieval Conference*, Porto, Portugal, 2012.
- [FWM⁺09] Mehrdad Fatourehchi, Rabab K. Ward, Steven G. Mason, Jane Huggins, Alois Schloegl, and Gary E. Birch. Comparison of evaluation metrics in classification applications with imbalanced datasets. In *IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 777–782, 2009.
- [FWVM07] Damien Francois, Vincent Wertz, Michel Verleysen, and Senior Member. The concentration of fractional distances. *IEEE Transactions on Knowledge and Data Engineering*, 19:873–886, 2007.
- [GBK05] Michael Guthe, Pavel Borodin, and Reinhard Klein. Fast and accurate Hausdorff distance calculation between meshes. *Journal of Winter School of Computer Graphics (WSCG)*, 13(2), 2005.
- [GHS07] Bram Van Ginneken, Tobias Heimann, and Martin Styner. 3d segmentation in the clinic: A grand challenge. In *MICCAI Workshop on 3D Segmentation in the Clinic*, pages 7–15, 2007.

- [GJC01] Guido Gerig, Matthieu Jomier, and Miranda Chakos. Valmet: A new validation tool for assessing and improving 3D object segmentation. In *Proceedings of the 4th International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 516–523, 2001.
- [GP93] E. Groeller and W. Purgathofer. Coherence in computer graphics. In *Visualization and Intelligent Design in Engineering and Architecture*. Elsevier Science Publishers, 1993.
- [GSP⁺08] Sylvain Gouttard, Martin Styner, Marcel Prastawa, Joseph Piven, and Guido Gerig. Assessment of reliability of multi-site neuroimaging via traveling phantom study. In *Proceedings of the 11th International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 263–270, 2008.
- [HA85] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2:193–218, 1985.
- [HAK00] Alexander Hinneburg, Charu C. Aggarwal, and Daniel A. Keim. What is the nearest neighbor in high dimensional spaces? In *Proceedings of the 26th International Conference on Very Large Data Bases*, pages 506–515. Morgan Kaufmann Publishers Inc., 2000.
- [Ham50] J. M. Hammersley. The distribution of distance in a hypersphere. *The Annals of Mathematical Statistics*, 21(3), 1950.
- [HDAC12] Julius Hossain, M. Dewan, Kiok Ahn, and Oksam Chae. A linear time algorithm of computing Hausdorff distance for content-based image analysis. *SOURCE Circuits, Systems and Signal Processing*, 31, 2012.
- [HKR93] Daniel P. Huttenlocher, Gregory A. Klanderman, and William A. Rucklidge. Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:850–863, 1993.
- [HL05] Jin Huang and Charles X. Ling. Using AUC and accuracy in evaluating learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 17:299–310, 2005.
- [HM82] James A. Hanley and Barbara J. Mcneil. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143:29–36, 1982.
- [HRHS12] Eyke Hallermeier, Maria Rifqi, Sascha Henzgen, and Robin Senge. Comparing fuzzy partitions: A generalization of the rand index and related measures. *IEEE Transactions on Fuzzy Systems*, 20:546–556, 2012.

- [ISHV⁺12] Laura Igual, Joan Carles Soliva, Antonio Hernandez-Vela, Sergio Escalera, Oscar Vilarroya, and Petia Radeva. Supervised brain segmentation and classification in diagnostic of attention-deficit/hyperactivity disorder. In *HPCS*, pages 182–187, 2012.
- [IV03] Piotr Indyk and Suresh Venkatasubramanian. Approximate congruence in nearly linear time. *Computational Geometry: Theory and Applications*, 24(2), 2003.
- [Jac12] Paul Jaccard. The distribution of the flora in the alpine zone. *New Phytologist*, 11(2):37–50, 1912.
- [JD88] Anil K. Jain and Richard C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, Inc., 1988.
- [JdTGM⁺14] Oscar Alfonso Jimenez del Toro, Orcun Goksel, Bjoern Menze, Henning Mueller, Georg Langs, Marc Andre Weber, Ivan Eggel, Katharina Grunenberg, Markus Holzer, Andras Jakab, Georgios Kontokotsios, Markus Krenn, Tomas Salas Fernandez, Roger Schaer, Abdel Aziz Taha, Marianne Winterstein, and Allan Hanbury. Visceral - visual concept extraction challenge in radiology: Isbi 2014 challenge organization. In *Proceedings of the VISCERAL Challenge at ISBI*, number 1194 in CEUR Workshop Proceedings, pages 6–15, 2014.
- [KCAB09] Hassan Khotanlou, Olivier Colliot, Jamal Atif, and Isabelle Bloch. 3D brain tumor segmentation in MRI using fuzzy classification, symmetry analysis and spatially constrained deformable models. *Fuzzy Sets and Systems*, 160(10):1457–1473, may 2009.
- [Kel75] John L. Kelley. *General Topology*. Springer, 1975.
- [Ken04] Maurice G. Kendall. *A course in the geometry of n dimensions*. Dover Publications, 2004.
- [Kha04] Rasul A. Khan. Approximation for the expectation of a function of the sample mean. *Statistics*, 38(2):117–122, 2004.
- [KMJK⁺10] Kasiri Keyvan, Dehghani Mohammad Javad, Kazemi Kamran, Helfroush Mohammad Sadegh, and Shaghayegh Kafshgari. Comparison evaluation of three brain mri segmentation methods in software tools. In *Biomedical Engineering (ICBME)*, pages 1–4, 2010.
- [KMVW97] David N. Kennedy, Nikos Makris, S. Caviness Verne, and Andrew J. Worth. Neuroanatomical segmentation in MRI: Technological objectives. *International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI)*, 11(8):1161–1187, 1997.

- [Koe00] Mario Koeppen. The curse of dimensionality. In *5th Online World Conference on Soft Computing in Industrial Applications (WSC5)*, pages 4–8, 2000.
- [KPM00] Erich Peter Klement, Endre Pap, and Radko Mesiar. *Triangular norms*. Trends in logic. Springer Netherlands, Netherlands, 2000.
- [KvdHR⁺07] Stefan Klein, Uulke A. van der Heide, Bas W. Raaymakers, Alexis N. T. J. Kotte, Marius Staring, and Josien P. W. Pluim. Segmentation of the prostate in mr images by atlas matching. In *IEEE International Symposium on Biomedical Imaging (ISBI)*, pages 1300–1303, 2007.
- [LBSN13] Thomas Low, Christian Borgelt, Sebastian Stober, and Andreas Nuernberger. The hubness phenomenon: Fact or artifact? In *Towards Advanced Data Analysis by Combining Soft Computing and Statistics*, volume 285, pages 267–278. Springer Berlin Heidelberg, 2013.
- [LCL⁺04] Ming Li, Xin Chen, Xin Li, Bin Ma, and P. M. B. Vitanyi. The similarity metric. *IEEE Transactions on Information Theory*, 50:3250 – 3264, 2004.
- [LMMH13] Georg Langs, Henning Mueller, Bjoern H. Menze, and Allan Hanbury. Visceral: Towards large data in medical imaging - challenges and directions. In *MICCAI Medical Content-based Retrieval for Clinical Decision Support workshop (MCBR-CDS)*, volume 7723, pages 92–98, Nice, France, 2013.
- [Mah36] Prasanta Chandra Mahalanobis. On the generalised distance in statistics. In *Proceedings National Institute of Science, India*, volume 2, pages 49–55, Apr 1936.
- [McL99] Geoff McLachlan. Mahalanobis distance. *Resonance*, 4:20–26, June 1999.
- [Mei03] Marina Meila. Comparing clusterings by the variation of information. In *Learning Theory and Kernel Machines*, volume 2777, pages 173–187. Springer, Berlin Heidelberg, 2003.
- [Mei05] Marina Meila. Comparing clusterings: an axiomatic view. In *Proceedings of the 22nd international conference on Machine learning*, pages 577–584. ACM, 2005.
- [MFTM01] David R. Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceeding of the 8th IEEE International Conference on Computer Vision*, volume 2, pages 416–423, July 2001.
- [MJB⁺12] Bjoern Menze, Andras Jakab, Stefan Bauer, Mauricio Reyes, Marcel Prastawa, and Koen Van Leemput, editors. *MICCAI 2012 Challenge on Multimodal Brain Tumor Segmentation BRATS2012*. MICCAI, Okt 2012.

- [MNLBR07] Fredric Morain-Nicolier, Stephane Lebonvallet, Etienne Baudrier, and Su Ruan. Hausdorff distance based 3D quantification of brain tumor evolution from MRI images. In *29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 5597–600, 2007.
- [MQR03] Calvin R. Maurer, Jr., Rensheng Qi, and Vijay Raghavan. A linear time algorithm for computing exact Euclidean distance transforms of binary images in arbitrary dimensions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(2):265–270, 2003.
- [MVvW05] Bart Moberts, Anna Vilanova, and Jarke J. van Wijk. Evaluation of fiber clustering methods for diffusion tensor imaging. In *IEEE Conference on Visualization*, pages 65–72, 2005.
- [MW47] H.B. Mann and D. R. Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The Annals of Mathematical Statistics*, 18(1):50–60, 1947.
- [NJS11] Sarana Nutanong, Edwin H. Jacox, and Hanan Samet. An incremental Hausdorff distance calculation algorithm. *Proceedings of the Very Large Database (VLDB) Endowment*, 4(8):506–517, May 2011.
- [NVV99] W.J. Niessen, K.L. Vincken, and M.A. Viergever. Evaluation of MR segmentation algorithms. In *International Society Magnetic Resonance in Medicine*, 1999.
- [PLH⁺12] Yachun Pang, Li Li, Wenyong Hu, Yanxia Peng, Lizhi Liu, and Yuanzhi Shao. Computerized segmentation and characterization of breast lesions in dynamic contrast-enhanced mr images using fuzzy c-means clustering and snake algorithm. *Computational and Mathematical Methods in Medicine*, 2012.
- [PN12] K. Somasundram P. Narendran, V.K. Narendira Kumar. 3D brain tumors and internal brain structures segmentation in mr images. *International Journal of Image, Graphics and Signal Processing*, 1:35–43, 2012.
- [Pow11] David M. W. Powers. Evaluation: From precision, recall and F-factor to ROC, informedness, markedness correlation. *Journal of Machine Learning Technologies*, 2:37–63, 2011.
- [PRVtHR08] T.H.J.M. Peeters, P.R. Rodrigues, A. Vilanova, and B.M ter Haar Romeny. Analysis of distance/similarity measures for diffusion tensor imaging. In *Visualization and Processing of Tensor Fields: Advances and Perspectives*. Springer, Berlin, 2008.

- [PTMH05] Dimitris Papadias, Yufei Tao, Kyriakos Mouratidis, and Chun Kit Hui. Aggregate nearest neighbor queries in spatial databases. *ACM Transactions on Database Systems (TODS)*, 30(2):529–576, 2005.
- [Ran71] William M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 1971.
- [RC10] Filip Radlinski and Nick Craswell. Comparing the sensitivity of information retrieval metrics. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 667–674. Association for Computing Machinery, 2010.
- [Rij79] C. J. Van Rijsbergen. *Information Retrieval*. Butterworth-Heinemann, Newton, MA, USA, 2nd edition, 1979.
- [RNI10] Milovs Radovanovic, Alexandros Nanopoulos, and Mirjana Ivanovic. Hubs in space: Popular nearest neighbors in high-dimensional data. *Journal of Machine Learning Research*, 11:2487–2531, 2010.
- [RPR13a] A. Ramaswamy Reddy, E. V. Prasad, and L. S. S. Reddy. Abnormality detection of brain mr image segmentation using iterative conditional mode algorithm. *International Journal of Applied Information Systems*, 5(2):56–66, 2013.
- [RPR13b] A. Ramaswamy Reddy, E. V. Prasad, and L. S. S. Reddy. Comparative analysis of brain tumor detection using different segmentation techniques. *International Journal of Computer Applications*, 82(14):14–28, 2013.
- [RS05] James Ramsay and Bernard Silverman. *Functional Data Analysis*. Springer Series in Statistics, 2nd edition, 2005.
- [RTR⁺04] Daniel B. Russakoff, Carlo Tomasi, Torsten Rohlfing, Calvin R. Maurer, and Jr. Image similarity using mutual information of regions. In *8th European Conference on Computer Vision, ECCV*, pages 596–607, 2004.
- [Sak06] Tetsuya Sakai. Evaluating evaluation metrics based on the bootstrap. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 525–532. ACM, 2006.
- [Sak07] Tetsuya Sakai. On the reliability of information retrieval metrics based on graded relevance. *Information Processing Management*, 43:531–548, 2007.
- [SB88] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

- [SBM96] Amit Singhal, Chris Buckley, and Mandar Mitra. Pivoted document length normalization. In *ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 21–29, 1996.
- [SC12] Mark D. Smucker and Charles L. A. Clarke. The fault, dear researchers, is not in Cranfield, but in our metrics, that they are unrealistic. In *The 2nd European Workshop on Human-Computer Interaction and Information Retrieval*, pages 11–12, The Netherlands, August 2012.
- [SF79] P. Shrout and J. Fleiss. Intraclass correlations: Uses in assessing rater reliability. *Psychol Bull*, 86:420–428, 1979.
- [SFSW11] Dominik Schnitzer, Arthur Flexer, Markus Schedl, and Gerhard Widmer. Using mutual proximity to improve content-based audio similarity. In *Proceedings of the 12th International Society for Music Information Retrieval Conference*, pages 79–84, 2011.
- [SFSW12] Dominik Schnitzer, Arthur Flexer, Markus Schedl, and Gerhard Widmer. Local and global scaling reduce hubs in space. *Journal of Machine Learning Research*, 13:2871–2902, Oct 2012.
- [SHB07] Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image Processing, Analysis, and Machine Vision*. CENGAGE-Engineering, Favoritenstrasse 9/4th Floor/1863, 2007.
- [She87] Roger N Shepard. Toward a universal law of generalization for psychological science. *Science*, 237:1317–1323, 1987.
- [SHS⁺13] Ikumi Suzuki, Kazuo Hara, Masashi Shimbo, Marco Saerens, and Kenji Fukumizu. Centering similarity measures to reduce hubs. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 613–623. ACL, 2013.
- [SLN11] Natasa Sladoje, Joakim Lindblad, and Ingela Nystrom. Defuzzification of spatial fuzzy sets by feature distance minimization. *Image and Vision Computing*, 29:127–141, 2011.
- [SNL13] Ran Shi, King Ngi Ngan, and Songnan Li. The objective evaluation of image object segmentation quality. In *Advanced Concepts for Intelligent Vision Systems (ACIVS)*, volume 8192, pages 470–479, 2013.
- [SSS02] J. S. Suri, S. K. Setarehdan, and S. Singh. *Advanced Algorithmic Approaches to Medical Image Segmentation*. Springer, 2002.
- [ST83] David W. Scott and James R. Thompson. Probability density estimation in higher dimensions. In *Computer Science and Statistics. Proceedings of the 15th Symposium on the Interface*, pages 173–179, 1983.

- [SY02] Gilbert Saporta and Genane Youness. Comparing two partitions: Some proposals and experiments. In *Proceedings in Computational Statistics*, pages 243–248, 2002.
- [TH15a] Abdel Aziz Taha and Allan Hanbury. An efficient algorithm for calculating the exact Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37:2153–2163, Mar 2015.
- [TH15b] Abdel Aziz Taha and Allan Hanbury. Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool. *BMC Medical Imaging*, 15:29, August 2015.
- [THJ14a] Abdel Aziz Taha, Allan Hanbury, and Oscar Jimenez. Test data and results of the automatic metric selection method. Technical report, Vienna University of Technology, http://publik.tuwien.ac.at/files/PubDat_229008.pdf, 2014.
- [THJ14b] Abdel Aziz Taha, Allan Hanbury, and Oscar Jimenez del Toro. A formal method for selecting evaluation metrics for image segmentation. In *2014 IEEE International Conference on Image Processing (ICIP) (ICIP 2014)*, pages 932–936, Paris, France, okt 2014.
- [THR15] Abdel Aziz Taha, Allan Hanbury, and Andreas Rauber. Hubness: Formal analysis and estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (SUBMITTED)*, 2015.
- [TM12] Nenad Tomasev and Dunja Mladenic. Hubness-aware shared neighbor distances for high-dimensional k-nearest neighbor classification. In *Hybrid Artificial Intelligent Systems - 7th International Conference*, volume 7209, pages 116–127, March 2012.
- [TSG06] Nicholas J Tustison, Marcelo Siqueira, and James C Gee. N-D linear time exact signed Euclidean distance transform. *The Insight Journal*, 2006.
- [ULZ⁺06] Jayaram K. Udupa, Vicki R. LeBlanc, Ying Zhuge, Celina Imielinska, Hilary Schmidt, Leanne M. Currie, Bruce Elliot Hirsch, and James Woodburn. A framework for evaluating image segmentation algorithms. *Computerized Medical Imaging and Graphics : the Official Journal of the Computerized Medical Imaging Society*, 30:75–87, 2006.
- [VB02] Ellen M. Voorhees and Chris Buckley. The effect of topic set size on retrieval experiment error. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 316–323, 2002.
- [VEB09] Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: is a correction for chance necessary?

In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1073–1080. ACM, 2009.

- [VEB10] Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 9999:2837–2854, December 2010.
- [VW97] Paul Viola and William M. Wells, III. Alignment by maximization of mutual information. *International Journal of Computer Vision*, 24(2):137–154, 1997.
- [VYPP11] Nagesh Vadaparthi, Srinivas Yarramalle, Suresh Varma Penumatsa, and P.S.R.Murthy. Segmentation of brain mr images based on finite skew gaussian mixture model with fuzzy c-means clustering and em algorithm. *International Journal of Computer Applications*, 28(10):18–26, 2011.
- [Wal06] Hanna M. Wallach. Evaluation metrics for hard classifiers. Technical report, Cambridge University, Cavendish Lab, 9 2006.
- [WBFR04] Ron Wehrens, Lutgarde M. C. Buydens, Chris Fraley, and Adrian E. Raftery. Model-based clustering for image segmentation and large datasets via sampling. *Journal of Classification*, 21(2):231–253, 2004.
- [WF05] Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers, 2005.
- [Wil31] W. A. Wilson. On quasi-metric spaces. *American Journal of Mathematics*, 53(3):675–684, Jul 1931.
- [WW07] Silke Wagner and Dorothea Wagner. Comparing clusterings - an overview. Technical report, Universitaet Karlsruhe (TH), 2007.
- [WWG⁺99] Simon K. Warfield, Carl-Fredrik Westin, Charles R. G. Guttmann, Marilyn S. Albert, Ferenc A. Jolesz, and Ron Kikinis. Fractional segmentation of white matter. In *Proceedings of Second International Conference on Medical Imaging Computing and Computer Assisted Interventions*, volume 1679, pages 62–71, 1999.
- [YM13a] Suchita Yadav and Sachin Meshram. Brain tumor detection using clustering method. *International Journal of Computational Engineering Research(IJCER)*, pages 11–14, 2013.
- [YM13b] Suchita Yadav and Sachin Meshram. Performance evaluation of basic segmented algorithms for brain tumor detection. *Journal of Electronics and Communication Engineering IOSR*, 5:08–13, 2013.

- [Zam82] Piero Zamperoni. Contour tracing of grey-scale images based on 2-d histograms. *Pattern Recognition*, 15(3):161–165, 1982.
- [ZKB87] Rami Zwick, Edward Karlstein, and David V. Budescu. Measures of similarity among fuzzy concepts: a comparative analysis. *International Journal of Approximate Reasoning*, 1(2):221–242, 1987.
- [ZL04] D. Zhang and G Lu. Review of shape representation and discription techniques. *PR Journal*, 37:1–19, 2004.
- [ZLX⁺14] Jianhui Zhao, Chengjiang Long, Shuping Xiong, Cheng Liu, and Zhiyong Yua. A new k nearest neighbors search algorithm using cell grids for 3d scattered point cloud. *Electronics and Electrical Engineering*, 20:81, 2014.
- [ZM14] Mohammed J. Zaki and Wagner Meira. *Data Mining and Analysis: Fundamental Concepts and Algorithms*. Cambridge University Press, 2014.
- [ZmP05] Lihi Zelnik-manor and Pietro Perona. Self-tuning spectral clustering. In *Advances in Neural Information Processing Systems 17*, pages 1601–1608. MIT Press, 2005.
- [ZWB⁺04] Kelly H. Zou, Simon K. Warfield, Aditya Baharatha, Clare Tempany, Michael R. Kaus, Steven J. Haker, William M. Wells, Ferenc A. Jolesz, and Ron Kikinis. Statistical validation of image segmentation quality based on a spatial overlap index. *Academic Radiology*, 11:178–189, 2004.
- [ZWKW04] Kelly H. Zou, William M. Wells, Ron Kikinis, and Simon K. Warfield. Three validation metrics for automated probabilistic image segmentation of brain tumours. *Statistics in Medicine*, 23:1259–1282, 2004.