Algorithmen für Implizite Delegation zur Vorhersage von Präferenzen

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Software Engineering & Internet Computing

eingereicht von

BSc Benjamin Krenn
Matrikelnummer 01327294

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ.Prof. Dipl.-Ing. Dr.techn. Stefan Woltran
Mitwirkung: Projektass.(FWF) Dipl.-Ing. Dr.techn. Martin Lackner

Wien, 10. Dezember 2019

Benjamin Krenn   Stefan Woltran

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.ac.at

FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

Algorithms for Implicit Delegation to Predict Preferences

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Software Engineering & Internet Computing

by

BSc Benjamin Krenn
Registration Number 01327294

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Prof. Dipl.-Ing. Dr.techn. Stefan Woltran
Assistance: Projektass.(FWF) Dipl.-Ing. Dr.techn. Martin Lackner

Vienna, 10th December, 2019

Benjamin Krenn   Stefan Woltran

Technische Universität Wien
A-1040 Wien ▪ Karlsplatz 13 ▪ Tel. +43-1-58801-0 ▪ www.tuwien.ac.at

Erklärung zur Verfassung der Arbeit

BSc Benjamin Krenn

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, 10. Dezember 2019

Benjamin Krenn

Acknowledgements

I want to thank my parents for always supporting me. My friend and fellow student David Jaffry helped me motivate myself to work five days a week. I am also thankful to my Advisor Univ.Prof. Dipl.-Ing. Dr.techn. Stefan Woltran to make this thesis possible for me. And last but not least the person who showed me the topic and advised me until the end with good feedback and help Projektass.(FWF) Dipl.-Ing. Dr.techn. Martin Lackner.

Kurzfassung

Das Leben der Menschen wird immer digitaler. Damit können ihre Vorlieben zu verschiedenen Themen wie Lieblingsmusik oder Websites gesammelt und analysiert werden. Ein mögliches Ziel einer solchen Analyse ist es, ein Gesamtranking mit den Präferenzen mehrerer Personen oder Länder zu erstellen. Hier spielen Computational Social Choice Algorithmen eine Rolle, die mehrere Rangfolgen als Eingabe nutzen und als Ausgabe eine zurück liefern. Dies resultierende Rangfolge soll die gesamte Gruppe so gut und fair wie möglich repräsentieren. Diese Algorithmen werden als Social Welfare Funktionen bezeichnet. Wenn eine Anwendung regelmäßige Ranglisten erstellen möchte, die die Präferenzen aller Benutzer enthalten, muss eine Möglichkeit vorhanden sein, die aktuellen Präferenzranglisten der Benutzer, die den Dienst derzeit nicht nutzen, vorherzusagen. Dafür könnte implizite Delegierung die Lösung sein, da sie bekannte vorherige Präferenzdaten verwendet und versucht, daraus neue Rangfolgen für den angegebenen Benutzer / die angegebene Person zu erstellen. Für diese Arbeit wurden einige Algorithmen entwickelt, die dies versuchen. Ein Ziel dieser Algorithmen ist es, Rankings zu erstellen, die den tatsächlichen Top-k-Präferenzen eines Benutzers oder auch als Wähler bezeichnet, entsprechen. Als zweites Ziel wird versucht, mehrere fehlende Rankings zu ersetzen und diese dann mithilfe von Social Welfare Funktionen zu aggregieren. Hier wird optimalerweise ein Ranking erstellt, das einem aggregierten Ranking mit den realen Daten so ähnlich wie möglich ist. Für die Ähnlichkeit werden Kendall Tau Algorithmen verwendet. Um zu sehen, wie die impliziten Delegierungsmethoden für reale Daten funktionieren, werden sie an realen Datensätzen aus Quellen wie Spotify getestet.

Abstract

The lives of people are becoming more and more digitalized. With this their preferences over different topics like their favorite music or websites can be collected and analyzed. One possible goal of such an analysis is to create an overall ranking with the preferences of multiple people or countries. For this computational social choice comes into play with algorithms that take multiple rankings as input and output one ranking that represents the whole group as good and fair as possible, these algorithms are called social welfare functions. If some application wants then to create regular rankings that include the preferences of all its users there needs to be a way to predict the current preference rankings of users that did pause using the service. For this implicit delegation could be the solution as it takes previous preference data that is known and tries to create new rankings from them for the given user/person. For this thesis some algorithms were developed that try to accomplish this. One goal of these algorithms is to produce rankings that match the actual top-k preferences of a user or also called voter. As second goal it is attempted to replace multiple missing rankings and then use social welfare functions to aggregate them. Here it optimally produces a ranking that is as similar to an aggregated ranking with the real data as possible. For the similarity Kendall tau algorithms are used. To see how the implicit delegation methods work on real world data they are tested on real data sets from sources like Spotify.

Contents

CHAPTER 1

Introduction

Since long ago humans have different preferences for almost anything. This can range from what a group eats for dinner, to who should rule the country and up to how humanity should act upon a worldwide crisis. With this need for group decision making social choice theory was born [EH89][SSA+02][ASS10].

Over time different approaches were developed. One of the more simple voting systems is Plurality, where every person's most preferred alternative is rewarded with one point and the alternative with the most points is chosen [You75]. Some methods were developed to decide on one single winner, those are called Social Choice Functions. But many methods are also Social Welfare Functions, these methods not only decide on a winner, but generate a new preference ranking which should represent the overall preferences of the voters. They are all preference aggregation methods [BCE+16].

Over time computer science and social choice theory began to work together and computational social choice was born. It considers aspects of both disciplines and combines them. For example social choice theory only considered if something such as manipulation in a voting system was possible and computer science added the computational complexity of such manipulation attempts that make certain methods useful because of high complexity to manipulate the outcome. And for example computer science benefits from the social choice theory with mechanisms for preference aggregation in other fields than political voting such as determining the best search results for a search engine. This thesis is placed in the field of computational social choice. It gives an overview of the complexity of different preference aggregation methods and it provides mechanisms to substitute missing voter data for aggregation purposes [CELM07][BCE+16].

At the time of writing this, many aspects of society are handled online. For example listening to music is being shifted in the direction of streaming instead of buying and playing the music offline. With this it is possible to track the music preferences of people or regions. If it would be the goal to decide on the overall best songs, but streaming numbers seems unfair, as some people or regions just stream more often or have more access to the internet, then it is a possibility to just take the rankings per person or region into account and see them all as equally important. On these rankings social welfare functions can decide on the overall ranking. As a side note, this problem of dealing with different expressions of preferences or utility between different people is called interpersonal comparison [BDW84]. It would be great if more information than just the rankings were available, like how much more does a person likes a song than another one, but this would lead to the problem of defining values for this and these values would not be equally used by every person. As no such information is available or could be easily and fair compared it was decided to weigh every persons (voters) vote as equally important linear order.

Now to continue the music example every person listening to the music is interpreted as one voter and votes through listening to the music ordered by how often they play each song. If then only the rankings of the people who had enough time to vote through listening to the music are taken into account this would exclude the opinion of other people. For this implicit delegation comes into play. In the case of this thesis it means approximating what the ranking of a voter would look like with the help of the rankings of the other voters together with the data from previous days. For example two people could have listened to nearly the same music for a few weeks and from this it would be likely to assume that if one of them did not listen to music on one day, they would still have nearly the same preference ranking over the music they currently like.

In general the meaning of implicit delegation can be explained with looking at the meaning of both words separately. It is attempted to predict a vote of someone who did not directly participate in a given election, however some data of previous elections is available that can be used to indirectly guess what the person/entity would have voted. Therefore it is called "implicit". The "delegation" part means that a representative is chosen like in politics where people vote for someone to represent their beliefs with a delegate, in the case of this thesis the representative can either be the ranking of another voter or some aggregated ranking that used the ballots of other voters as input. These two words are combined to "implicit delegation" which can be interpreted as indirectly and automatically elected representative.

To understand and interpret preference data some basic knowledge will be needed. Therefore Chapter 2 will give an introduction to some relevant topics. This includes definitions of ranked ballots, an overview of desirable properties for preference aggregation methods and some common mechanisms to compare different preference rankings.

Then in Chapter 3 multiple different aggregation methods or also called social welfare function (SWF) will be introduced. A selection of them will later be used in the experiments of this thesis. This is important as different functions have different properties and if one function is chosen because of its properties, it can not be simply assumed that an implicit delegation method tested on one function will have the same benefits on another function. Therefore a selection of some important ones will be used in the tests.

Implicit delegation in the context of Social Choice is a new concept. However there exists research using implicit delegation of responsibility in close relationships to determine self-control of people in a task based on previous tasks executed with the same people [vB11]. The goal of this work is to develop different approaches to predict preferences in the context of social choice theory.

Computer Experiments based on real word data have been designed and executed to test these approaches to see how good they can approximate missing preference data. This will include experiments where social welfare functions aggregate rankings with complete data compared to the aggregation without some voters. Then implicit delegation methods will analyze past data to fill those missing voters. The resulting aggregated ranking needs to be compared to the one with the full data. If this comparison has a higher similarity to the full data result than simply ignoring the missing voters before aggregating, this would mean the method works good on the corresponding data set. In contrast a lower similarity means it would be probably better to not use the method on the data set. Other approaches could be to simply test how good the top-k alternatives of a voter can be approximated.

The outcome will always depend on the data set, therefore different data sets will be collected and presented in Chapter 4. As focus real world data was chosen over artificially created data, as it is important to see how the methods work on real data. It will be attempted to find explanations of why a method works or does not work for each data set, to make it possible to decide which method should be used on different data sets based on some statistical analysis.

A description of the implicit delegation methods will follow in Chapter 5. For this different approaches will be used. First some search and analyze methods that find possible rankings from different voters in the present or in past rankings. Secondly some methods that combine the found rankings in an attempt to produce more accurate predictions.

The experiments themselves will be described in Chapter 6. This will be followed by a detailed representation of the results. These results are also used to see how good the results of the different methods can be predicted from some basic analysis over the full data sets. The results are also used to show which of the developed implicit delegation methods can be useful.

To sum up, the contribution of this thesis is as follows:

- This thesis introduces the concept of implicit delegation methods and proposes 8 concrete delegation methods.
- An overview of suitable preference aggregation methods for this setting is given, together with explanations of their properties and run-time estimates.

- It also presents a way to test such methods. For this a few data sets were collected and tested on. Those data sets were obtained from Spotify, I-Phone app store charts and Eurovision Song Contest.
- An analysis of the used data sets is given. This includes similarities between different voters and between different points in time.
- Experiments were executed that show which methods work best on which data sets. This lead to a general recommendation for certain implicit delegation methods.
- It shows the connections between similarity analysis of a whole data set to how good a given implicit delegation method performs.

CHAPTER 2

Prerequisites

To start off this chapter will give some basic knowledge of terminology and methods that are of importance in later chapters. As first an explanation of ranked ballot is given. This is followed by properties of preference aggregation methods. And last but not least some important ranking comparison metrics are introduced.

2.1 Definition of Ranked Ballots

The first thing to know when discussing social choice theory is a basic understanding of ranked ballots. This section will give a formal definition (from [BCE+16]) and explain it with the help of an example.

- $N = \{1, 2, ..., n\}$ is a list of voters.
- A is a finite set of candidates or alternatives of size $m \geq 2$.
- Every voter i casts their ballot as weak linear ordering $\succsim_i$ of A. This order is transitive, meaning if $a \succsim_i b$ and $b \succsim_i c$, then $a \succsim_i c$. It is also reflexive, meaning $\forall a \in A : a \succsim_i a$
- The antisymmetric version of $\succsim_i$ is $\succ_i$
- A profile $P = (\succsim_1, \succsim_2, ..., \succsim_n)$ is the collection of the ballots of the voters.
- Preference aggregation results will use the notation $\succsim$ or $\succ$ depending on the situation.

Ballots used for the experiments in this thesis are not antisymmetric (if $a \succsim_i b$ and $b \succsim_i a$, then $a = b$ for all $a, b \in A$ is not given), but can be seen as complete (meaning

$\forall a \neq b \in A : a \succsim_i b$ or $b \succsim_i a$). This makes them weak orders. An order that is not complete would be called partial order and an order that fulfills transitivity, antisymmetry and completeness is called total order [BCE+16]. In fact the data only consists of top-k orders and it can be assumed that every alternative not in the top-k ranks is seen as equal ranked and ranked below every alternative in the top-k.

Example:
Now to an example, these ballots represent preferences, therefore the candidates could be nearly everything. A possible ranking for colors could be $blue \succ_1 red \succ_1 pink$ and another voter could have e.g. $pink \succ_2 red \succ_2 blue$ and lastly a third voter thinks $red \succ_3 pink \succ_3 blue$. An easy readable form for this can be seen in Table 2.1.

If these ballots are evaluated with Plurality, there would be no winner as all 3 candidates have each 1 point. But using more advanced methods like Borda or Copeland there would be a winner and even a full ranking: $red \succ pink \succ blue$. How these different aggregation methods work will be discussed later.

Rank	Voter 1	Voter 2	Voter 3
1.	*blue*	*pink*	*red*
2.	*red*	*red*	*pink*
3.	*pink*	*blue*	*blue*

Table 2.1: Possible ballots for favorite colors with only 3 candidates and voters.

2.2 Axiomatic Properties of Preference Aggregation Methods

Chapter 3 will discuss some preference aggregation methods. Those methods take multiple ballots as input and their output depend on what type of function they are. For SWFs the outputs are full rankings/ballots. If case of social choice functions (SCFs) a set containing the winning alternatives is returned. How exactly they work will be discussed there, but first some important properties of these methods will be described here.

The following list gives a good overview with descriptions of properties according to [Ren01], [Sch11], [BCE+16] and [Mar96]:

- *Pareto Property*: In a given profile for some $x, y \in A$ if every voter ranks x over y, then y is Pareto dominated. A SCF that fulfills the Pareto Property will never have a Pareto dominated alternative in the set of winners (meaning the first ranked alternatives).
- *Pareto Efficiency* mean that if one alternative dominates every other alternative (is first ranked in every ranking) then it is also the winner of the SCF.

- *Anonymity*: Even if the votes of two voters are exchanged the results stay the same. Or formally for a SCF f that fulfills this property it holds that $f(P) = f(P^*)$ if the profile P^* is obtained from altering the profile P through simply swapping the votes between some voters i and j. This means the ballot of voter i is set to the ballot of voter j ($\succ_i^* = \succ_j$) and the ballot of voter j is set to the one of voter i ($\succ_j^* = \succ_i$).
- *Non-Dictatorship*: A dictatorship has one voter i for whom applies that $\forall a, b \in A$: if $a \succ_i b$ then $a \succ b$ independent of all other voters. For a non-dictatorship this is not the case as long as there are at least two voters.
- *Neutrality*: The result is independent of the labels or names of the alternatives. Formally this means that if between two profiles P and P' every occurrence of the alternatives x, y are swapped then the result $f(P')$ can be obtained from $f(P)$ with swapping x and y.
- *Pairwise Cancellation*: If for every alternative pair $a, b \in A$ there are as many votes that prefer a over b as there are votes that prefer b over a, then all alternatives/-candidates are tied in first rank.
- *Consistency*: If the SCF is applied to two profiles and both result in the same winner(s), then combining them to one profile will still yield these winner(s). If applied to a SWF this rule still only looks at the first ranked alternative in the result. More details to this axiom will follow below.
- *Monotonicity*: If one of the voters gives an alternative a better position, without changing the order of any other alternative, then the new result can not give this alternative a worse position than before the change. For example if one voter first votes $a \succ b \succ c$ and the SWF results in $c \succ a \succ b$ then it is not possible that a vote change to $c \succ a \succ b$ or $a \succ c \succ b$ could result in a rank of c behind a or b.
- *Condorcet Winner*: For two alternatives a, b where $a \neq b$, if a is ranked higher in more rankings than b then this is a pairwise win for a. Now if a wins pairwise against every other alternative, then it is the Condorcet Winner and will be the winner of the SCF if it fulfills the property. This is described in more detail in Section 3.2.
- *Condorcet Loser*: Similar to Condorcet Winner, but if an alternative loses in every pairwise comparison, then it can not win overall.
- *Independence of Clones*: If one election has the alternatives a, b, c and another election the alternatives a_1, a_2, a_3, b, c where the different a_i alternatives are nearly identical to each other (if a person likes/dislikes a_1 then they most probably also like/dislike a_2 and a_3), then the results of the election will not change despite that multiple a_i will be sharing similar ranks. For example if the first election ends with $b \succ a \succ c$ then the second one will end with $b \succ a_1 \sim a_2 \sim a_3 \succ c$ as the a_i will be similarly ranked in every voted ballot.

- *Independence of Irrelevant Alternatives (IIA)* means that the preference of one option a over another option b should not be dependent on different independent alternatives. For example if one voter votes first $a \succ c \succ b$ and then changes the vote $a \succ b \succ c$ then a method that fulfills IIA will not change the resulting order between a and b or a and c.

There exist some rules for aggregation methods regarding these properties. One of them is Arrow's theorem [Arr08] [Ren01]: Pareto efficiency, non-dictatorship and independence of irrelevant alternatives can not be all fulfilled in one SWF. IIA is often chosen as the property that can be dropped. An example for this will be given in Section 3.1 the Borda Method. It can be assumed that also no other aggregation method mentioned will fulfill this property as it is less important in the use case of this thesis. One of the problems with missing the property IIA is that strategic voting is possible, meaning changing the vote from the real preferences to something else in order to change the result in their favor [Ren01].

There are two further properties that can not be both fulfilled by the same SCF, Reinforcement also known as Consistency (seen above) and Condorcet extensions (see below). A SCF f that has the property *Reinforcement* fulfills the property, if for every two profiles p, h the following applies: If $f(p)$ and $f(h)$ share some winning candidates, then these candidates will still win after all the voters of p and h are combined to one profile [BCE+16].

$$f(p) \cap f(h) \neq \emptyset \Rightarrow f(p+h) = f(p) \cap f(h)$$

A *Condorcet extension* means that the SCF will result in the Condorcet winner if and only if one exists. The problem with these two properties is that they are both desirable but can not be both completely fulfilled by a single SCF. For example Borda from Section 3.1 only has the reinforcement property and Section 3.2 will give an example of a popular Condorcet extension [BCE+16].

2.3 Comparison Methods for Incomplete Strict Orders

To give some way of comparing rankings two metrics will be used in this thesis. First how large the intersections between the rankings of two voters are. This means how many candidates are listed in both their rankings. Formally this can be written as for every two rankings r_1 and r_2 the intersection size can be computed with a function $intersection : R^2 \implies \mathbb{N}$ where the inputs are two rankings and the output is the size of the intersection of the two rankings:

$$intersection(r_1, r_2) = |r_1 \cap r_2|$$

As intersection size is an integer from 0 to m it is not easily usable for algorithms that require a certain threshold, however this can be solved if the metric returns a real number between 0 and 1 representing the fraction of the alternatives from the first ranking that are also found in the second one. This version can be called normalized intersection size and will be used in the proposed algorithms of this thesis.

As second method the Kendall tau metric between rankings of voters will be used. This metric gives a value ranging from -1 to 1. -1 means the two rankings are complete opposites, 0 that they don't have any common candidates or equally disagree and agree on the orders. And 1 means they agree completely on the order [Ken45]. There are algorithms to compute this score in $O(|A| \cdot log(|A|))$ as shown in [BGH12].

There are two possible versions that are of relevance for this thesis. The Kendall tau type-a and type b. Both involve calculating how many discordant pairs (D) and how many concordant pairs (C) are found in the two compared ballots. Concordant pairs represent pairs of alternatives a, b for which the first and second ranking both have either $a \succ b$ or $b \succ a$. If one ranking has $b \succ a$ and the other one has $a \succ b$ then it is a discordant pair. And if at least one of the ballots has a and b equally ranked then the pair is neither concordant nor discordant [Ken45].

Tau-a can be computed with:

$$\tau_a = (C - D)/n_0$$

Where n_0 is the number of alternative pairs $n_0 = n(n-1)$ for n the number of alternatives. With this formula many ties lead to a result near 0, which could be exactly what is needed for this thesis [Ken45].

Tau-b in contrast accommodates for ties through replacing n_0 with the number of pairs that are neither equally preferred in the first ranking nor in the second ranking. This is expressed as:

$$\tau_b = (C - D)/\sqrt{n_x * n_y}$$

Where n_x is the number of pairs in the first ranking that are not equally ranked and n_y the same only for the second ranking. This method inflates the result score to be more equally spread between -1 and 1 [Ken45].

Finally a short description of how the performance of Kendall tau can be improved from the trivial run-time of $O(|A|^2)$ to $O(|A| * log(|A|))$. For this it is important to note that the number of discordant pairs are equal to the number of swaps needed with a bubble sort [Ast03] to change the order of the alternatives from the first ranking to the order in the second ranking [FKM+04]. This alone would not improve the run-time, however this number can be computed during a merge sort with a performance of $O(|A| * log(|A|))$.

With this knowledge the calculation of the Kendall tau score can be represented in a different form. First every alternative will be represented by a pair (X, Y) where X is the rank in the first ranking and Y in the second one. Now these alternatives will be sorted by first X and in case of tie second by Y. From this state the ties in the first ranking n_x and the ties in both the first and second ranking n_{xy} can be counted in $O(|A|)$. After this a merge sort is started, which sorts first by Y and for ties by X. This merge sort also calculates the number of swaps for a bubble sort, which correspond to D. Finally the ties in the second ranking n_y can be counted in $O(|A|)$.

This is already everything that is needed to compute Kendall tau. As already mentioned n_0 is the number of pairs overall. As concordant and discordant pairs are never tied in either ranking it can be computed how many of those exist with $C+D = n_0 - n_x - n_y + n_{xy}$. But $C - D$ is required so simply subtracting $2 \cdot D$ will result in the number of concordant pairs minus the number of discordant pairs.

$$C - D = n_0 - n_x - n_y + n_{xy} - 2 \cdot D$$

This can then be used to compute either tau-a or tau-b.

However for this thesis a comparison method would be desirable, that can handle ties and does not just give a higher score when many ties exist like tau-b. For this a little adjustment on tau-a is needed. In the basic version of Kendall tau type-a every tie is rewarded with a value of 0 independent of if the tie occurs in both rankings or only in one of them. This means it is treated as half agreement as it neither rewards 1 like concordant pairs nor treated as -1 like discordant pairs. Here a suggestion from [FKM$^+$04] comes in helpful. They suggest to keep the half agreement if only one of the rankings has the alternatives tied, but they also suggest to treat it as full agreement if both rankings have the alternative pair tied. This means every tied pair in both rankings will also reward 1 point. With this the new tau-a formula can be expressed the following:

$$\tau_a = (C - D + n_{xy})/n_0$$

It would also be possible to express this with saying n_{xy} should already be part of C, but as the formula for $C - D$ is also used for tau-b it was decided to let that part unchanged to not accidentally cause confusion. This is the formula for Kendall tau type-a that will be used throughout this thesis.

CHAPTER 3

Preference Aggregation

The history has brought up many voting systems for different purposes. Some where developed with the goal to be easy understandable and easy executable. An example for such a voting rule is Plurality, a system, where every voter gives one vote to their favorite and the candidate with the most votes wins. This system is often used to select political leaders [You75].

Then there are more complex voting systems. These can take full or partial rankings into account and aggregate these votes to one ranking. Most of them have as goal to be fair and create the best compromise for all voters. One possible use case of them is to compare search engine result rankings and generate an overall ranking.

There are many applications for these SCF (have as result a winner) or SWF (have a ranking as result). Ranging from political votes to marketing, search engines and even biological research can profit from them [Lin10]. This thesis will take a look at some of them and decide which are appropriate for the topic. Here it is important to state that the method must work on weak orders. This is important as in the case of this thesis only the top-k ranks per voter are provided. As not all voters vote for the same k alternatives there normally exist more candidates. With m candidates overall and $m > k$ each voter votes for their top-k candidates through a total order and the remaining $m - k$ candidates are all equally and last ranked. This as a whole is a weak order.

3.1 Borda Method

The first SWF that seems promising is the Borda function. It falls into the category of simple scoring rules, the same group as Plurality. They both use the rankings the voters give to assign scores to the candidates. In case of Plurality the score is 1 for the first place and 0 for all others.

In general scoring rules all have one formal definition in common. Every voter has their total order of alternatives and a scoring rule introduces a sequence of numbers $s_1, s_2, ..., s_m$. Then an alternative that is ranked 1^{st} gets the score s_1, the second ranked alternative s_2 and so on. These scores are then assigned to the alternatives for each voter and the sum is the overall score for each candidate. Plurality has $s_1 = 1$ and all other scores $s_i = 0$ [You95].

For the original Borda function the voters need to provide total orders of the m alternatives. Then the first ranked alternative gets $s_1 = m - 1$ points, the second $s_2 = m - 2$ and the ith gets $s_i = m - i$ points, lets call it a function $score(a, \succsim_j)$. The same scoring happens for every ballot in the profile $P = (\succsim_1, \succsim_2, ..., \succsim_n)$. With this a scoring function can be derived $s : A \implies \mathbb{R}$ where A is the set of all alternatives and $s(a_i)$ gives the sum of all scores for the alternative a_i from all voters: $s(a_i) = \sum_{j=1}^{n} score(a_i, \succsim_j)$. The first place in the resulting orer $\succsim$ is now the alternative with the highest overall score $(s(a))$, and last place the alternative with the lowest overall score [You75].

Example:
This can now be used to aggregate the ranking in Table 2.1 from Section 2.1. $s(blue) = 2 + 0 + 0 = 2$, $s(red) = 1 + 1 + 2 = 4$ and $s(pink) = 0 + 2 + 1 = 3$. This ordered from highest to lowest gives $red \succ pink \succ blue$.

Now if Table 3.1 is considered, there are ballots consisting of weak orders. The original Borda score can't be used directly to solve this problem. However it is not hard to adjust the method to allow indifference between two or more alternatives.

One possibility to achieve this would be to change the score function to return the number of alternatives with a lower rank than a subtracted by the number of alternatives with a higher rank than a like proposed in [Gär73]. This thesis uses a slightly changed version to make it even more similar to the original version. The new scores can be calculated with $m - h(a)$ were m is the number of candidates and $h(a)$ the number of candidates with a higher rank than a. The function that uses this score will be call $borda_{weak}$.

Rank	Voter 1	Voter 2	Voter 3
1.	*blue*	*pink*	*yellow*
2.	*red*	*blue*	*green*
3.	*pink*	*green*	*blue*
4.	*yellow, green*	*red, yellow*	*red, pink*

Table 3.1: Possible ballots for favorite colors with weak orders.

Example:
If this adjusted Borda function is then used for the profile in Table 3.1 the following scores are the result: $borda_{weak}(blue) = 5 + 4 + 3 = 12, borda_{weak}(red) = 4 + 2 + 2 = 8, borda_{weak}(pink) = 3 + 5 + 2 = 10, borda_{weak}(yellow) = 2 + 2 + 5 =$

$9, borda_{weak}(green) = 2 + 3 + 4 = 9$. And finally ordered from highest to lowest the result is $blue \succ pink \succ yellow \sim green \succ red$, where *green* and *yellow* are seen as equally preferred.

So what are the benefits of using the Borda method compared to other SWFs one could ask. The first benefit that comes to mind is that the Borda rule is easy to understand and fast to compute even for large data sets with a linear complexity of $O(n \cdot m)$ where n is the number of voters and m the number of candidates. Additional there are also some axioms that are fulfilled. Namely Pareto property, Pareto efficiency, anonymity, neutrality, pairwise cancellation, consistency and monotonicity (or sometimes similar faithfulness). These properties were already explained in Section 2.2 [Mar96].

Anonymity is trivial to see why Borda fulfills it, as the method ignores all information about the voters. Non-dictatorship is also most of the time the case for preference aggregation methods with the property anonymity as without information about the voters only random choice would allow a dictatorship. Neutrality is similar trivial with the labels of the alternatives being used only to track the score. Pareto property is also self explanatory for Borda. If in every ballot an alternative b is higher ranked than the alternative a then the alternative b is consistently rewarded more points than a and therefore a can not have the most points which is required to be first ranked. Pareto efficiently is even more trivial with an alternative that dominates every other alternative for every voter it will receive the maximum Borda score which guarantees first rank [Mar96].

Monotonicity is also easy to explain on Borda. If one voter changes their vote from $a \succ b \succ c$ to $c \succ a \succ b$ then c gains 2 points or how many ranks it increases and the other alternatives that are overtaken lose one point each. With this it is not possible for c to fall in rank as long as no other changes occur [Mar96].

Finally consistency is similar easy to explain for Borda. If two profiles are evaluated to get the Borda scores of their alternatives and for both profiles the alternative b is in the set of first ranked alternatives then this means b has the highest score in both profiles. For Borda the score of the profiles can simply be added together as recalculation would do nothing different. Now if an alternative b had the highest score in both profiles with x and y the result can not be smaller than two other scores z and v combined as it is already known that $z \leq x$ and $v \leq y$ which implies $z + v \leq x + y$.

With Arrow's theorem it is known that a SCF like Borda is still not perfect for every situation as Pareto efficiency, non-dictatorship and independence of irrelevant alternatives can not be all fulfilled in a SWF. This can be shown for the Borda method, as the difference of the rank between two alternatives matters, and therefore it could change the outcome between a and b in the aggregated ranking if they are near together in one ranking instead of far away. This would not change their pairwise ranking in the vote, but with the Borda method it could make the difference for their pairwise ranking in the result. This can be shown in the example of Table 3.2. [Ren01].

Rank	Voter 1	Voter 2	Voter 3
1.	*blue*	*pink*	*red*
2.	*red*	*blue*	*pink*
3.	*pink*	*red*	*blue*

Table 3.2: Compared to Table 2.1 only Voter 2 changed the positions of *red* and *blue*. But this changes the overall preference of *pink* and *red* to $pink \sim red$ instead of $red \succ pink$.

Furthermore Borda can not be a Condorcet extension as it fulfills consistency and these properties in their standard form can not be both fulfilled by one SCF. [BCE+16]

3.2 Copeland, a Condorcet Method

Copeland developed an alternative method to aggregate rankings. This SWF has as goal to rank the candidates according to how close they are to becoming the Condorcet winner. A Condorcet winner is the alternative that wins every pairwise comparison with every other alternatives. For a pair $a, b \in A$ the pairwise winner is the one which has more individual wins in the rankings than the other, this can be expressed in a score

$$Net_p(a,b) = |\{i \in N | a \succ_i b\}| - |\{i \in N | b \succ_i a\}| \tag{3.1}$$

which represents the pairwise majority margin [BCE+16].

If $Net_p(a,b) > 0$ then a wins pairwise against b, this can be written as $a >^\mu b$. Copeland uses these wins and losses to compute a score for every $x \in A$

$$Copeland(x) = |\{y \in A | x >^\mu y\}| - |\{y \in A | y >^\mu x\}|$$

The ranking is then a sorted list of the alternatives by their Copeland score were the highest score is first place and the lowest score is last place, ties or also written $\sim$ relations are possible. One candidate can at most get a score of $m-1$, this can be reached by at most one candidate and means they are the Condorcet winner. If such a winner exists it will always be ranked first with the Copeland method, therefore it is a Condorcet extension [BCE+16].

Example:
Going back to the example in Table 2.1 the ranking can again be computed as following: $Copeland(blue) = 0 - 2 = -2$, $Copeland(red) = 2 - 0 = 2$ and $Copeland(pink) = 1 - 1 = 0$. Again ordered from highest to lowest the overall ranking is $red \succ pink \succ blue$. Compared to Borda, Copeland does not need to be adjusted for weak orders as Net_p simply ignores ties and if it returns 0 this pairwise comparison results in a difference of 0 in the Copeland score, while > 0 results in $+1$ and < 0 in -1.

Of course being a Condorcet extension does not alone guarantee that the function is useful. One important aspect is the run-time and complexity. Copeland has a run-time of $O(m^2 n)$ with m the number of alternatives and n the number of voters. This is higher than compared to Borda, as the computation of the pairwise majorities needs to compare every alternative with every other alternative. Computing Copeland is TC^0-complete according to [BFH09].

However there are some proven properties of this SWF that can arguably justify the longer run-time. They are already explained in Section 2.2 as always. The Copeland method has the properties monotonicity, anonymity, neutrality, non-dictatorship, pairwise cancellation and Pareto property. And as already mentioned a SCF that is a Condorcet extension will not fulfill reinforcement/consistency, but therefore it fulfills the Condorcet winner and loser criterion [MS97].

Copeland is the method for which pairwise cancellation can be explained the easiest. If for every pair of alternatives a, b there are as many voters that prefer a over b as there are voters that prefer b over a then every alternative has a Copeland score of 0. And as all candidates share the same score, they are all tied in first rank.

For monotonicity it is to mention that it is applicable only if a voter only increases the position of one alternative. If this improved rank also changes the preference relation between any other alternatives, the increased position no longer guarantees to at least hold the previous position as this falls under the property IIA, which is not fulfilled by any of the methods in this thesis. E.g. one voter has initially the preference $a \succ b \succ c$ and the overall Copeland ranking is $c \sim b \succ a$, if this voter then changes his vote to $c \succ b \succ a$ the overall Copeland ranking could then be $b \succ c \succ a$. For this to be possible a and b would be overall equally preferred but b has more $>^\mu$ victories than a and as many as c, the changed vote would then give b one extra pairwise victory in the form $b >^\mu a$ while it is possible that no extra victory for c was won with this change [MS97].

3.3 Kemeny Rule

Kemeny found another way to aggregate rankings that seems to circumvent the rule that a Condorcet extension can not fulfill reinforcement. In fact it is a Condorcet extension that is neutral, anonymous and reinforcing. This works as his rule is a social preference function instead of a SCF, meaning the result is a set of at least one total order, therefore the reinforcement requirement is a much weaker version of the reinforcement property. In this version reinforcement means only that if two profiles results contain the same total order(s), then merging these two profiles will still yield this total order(s) as result with the Kemeny function [BCE+16].

First of all it is necessary to understand how the function works. It uses a form of the Kendall tau metric to compute a similarity of two rankings. This can be defined as a function for which $a, b \in A$ and $r_, r_i$ are rankings: $\delta_{a,b}(r, r_i) = 1$ if r and r_i agree on the preference between a and b and 0 if not. To be precise, this function returns 1 if a pair is a concordant pair (they agree) and 0 if it is a discordant pair (they disagree) [Noe81].

Kemeny now uses the sum of all pairs to compute a score to show agreement of two rankings:

$$d(r, r_i) = \sum_{a,b \in A} \delta_{a,b}(r, r_i)$$

The challenge is then to find a ranking r to maximize the sum of this scoring method used on all rankings r_i in a given profile p. The Kemeny score of a ranking r is written as:

$$d_K(r, p) = \sum_{r_i \in p} \sum_{a,b \in A} \delta_{a,b}(r, r_i)$$

And a ranking r is part of the result of the Kemeny function if and only if its d_K is maximized, meaning there is no other possible r_x with a higher d_K than r [CDK06].

Example:
Again looking at the ballots in Table 2.1 the winner can be computed. But this time the computation is harder. The score for every possible ranking needs to be computed, with the 3 candidates this corresponds to 6 rankings. Table 3.3 shows the calculations. And the ranking with the (in this case unique) highest Kemeny score and therefore the winner is $red \succ pink \succ blue$. This is also the same solution as Copeland and Borda found, but especially with a higher number of candidates and voters such a coincident gets less likely to happen.

Ranking r	Voter r_1	Voter r_2	Voter r_3	sum
$blue \succ red \succ pink$	$1+1+1$	$0+0+0$	$0+0+1$	4
$blue \succ pink \succ red$	$1+1+0$	$0+0+1$	$0+0+0$	3
$red \succ blue \succ pink$	$0+1+1$	$1+0+0$	$1+0+1$	5
$red \succ pink \succ blue$	$0+0+1$	$1+1+0$	$1+1+1$	6
$pink \succ blue \succ red$	$1+0+0$	$0+1+1$	$0+1+0$	4
$pink \succ red \succ blue$	$0+0+0$	$1+1+1$	$1+1+0$	5

Table 3.3: Kemeny method used on the ballots in Table 2.1. calculations represent $\delta_{blue,red} + \delta_{blue,pink} + \delta_{pink,red}$

With Kemeny being a Condorcet extension and having a reinforcement property it is a really good method to choose. Unfortunately it is NP-hard to compute in its basic form for 4 or more alternatives [CDK06]. In fact it is shown to be $P_{\|}^{NP}$-complete. This is the complexity class of sets that are solvable via parallel access to NP [HSV05]. From the basic definition of Kemeny it is easy to show an upper bound for the run-time. With m alternatives there are $m!$ possible rankings. These need to each be compared to all n rankings of the n voters. The comparison is a Kendall tau distance which as already mentioned in Section 2.3 has a run-time of $O(m \cdot log(m))$.

So overall the upper bound is $O(m! \cdot n \cdot m \cdot log(m))$ where $m!$ is the dominant part. As $m!$ is not that nice to show run-time it can be converted to an exponential definition. First of all $m! = 1 \cdot 2 \cdot ... \cdot m \leq m^m$, so m^m is also an upper limit. This can be further changed to be an exponent of 2 as $m = 2^{log_2 m}$ and therefore $m^m = (2^{log_2 m})^m$ which can be expressed as $2^{m \cdot log_2 m}$. With this there is an exponential upper limit for the run-time of $O^*(2^{m \cdot log_2 m})$

Because of this high computational complexity it is often considered to compute an approximation in polynomial time, but this could be seen as computing a different function and not the real Kemeny rule. Other approaches include search based algorithms that depend on heuristics to compute e.g. lower or upper bounds [CDK06].

[BBN14] proposed and tested a few reduction methods that use rules for Kemeny solutions or Condorcet extensions to split the computation into smaller parts. This can decrease the run-time substantially, but it can happen that this reduction barely saves time. They estimated the run-time of their reduction algorithms to $O(|N| \cdot |A|^2)$, where N is the set of voters and A the set of Candidates. However after this reduction an exponential run-time is still problematic depending on the size of the reduced sub problems.

A good summary of some algorithms to compute the exact Kemeny ranking and approximations can be found in [AM12]. They compare over 100 versions of different algorithms and combinations including some branch-and-bound algorithms and local search. To further show that an approximation algorithm is hard to sell as nearly the real one, they even suggest Borda or Copeland as approximations [AM12].

3.4 Schulze Method

The Schulze method is another polynomial-time computable Condorcet extension. But in contrast to Copeland it takes into account how many voters prefer some alternative a over some alternative b [Sch03].

The first step of the method is to calculate a matrix $d[a,b]$, where the value of each pair represents how many voters prefer alternative a to b for all $a, b \in A$. Then a path from every a to every b will be defined with $C(1) = a$ and $C(n) = b$. A path is ordered and only contains $C(i)$ and $C(i+1)$ if $d[C(i), C(i+1)] - d[C(i+1), C(i)] > 0$. The strength of a path is defined as the weakest victory between two $C(i)$ and $C(i+1)$ in the path of length l:

$$strength(path) = min\{d[C(i), C(i+1)] - d[C(i+1), C(i)] | i = 1, .., (l-1)\} \text{ [Sch03]}$$

Then for every $a, b \in A$ the strength of the strongest path from a to b is assigned to $p[a,b]$. $p[a,b] := max\{strength(C(1), .., C(l)) | C(1), .., C(l)$ is a path from alternative a to alternative b $\}$. A strength of 0 means there is no path available [Sch03]. With these strongest path values it is now possible to build a full preference relation R, where $a \succsim b \in R$ if and only if $p[a,b] \geq p[b,a]$. The result is a weak order, where the first ranked candidates are the Schulze winners [CLP18].

alternatives	*red*	*pink*	*blue*
red	0	2	2
pink	1	0	2
blue	1	1	0

Table 3.4: Preference count matrix d for the ballots in Table 2.1.

alternatives	*red*	*pink*	*blue*
red	0	1	1
pink	0	0	1
blue	0	0	0

Table 3.5: Schulze path strength matrix p for the ballots in Table 2.1

Example:
When looking at the simple example in Table 2.1 this method does not change much, but for completeness it will still be demonstrated. Table 3.4 shows the d matrix and Table 3.5 shows the path strength p. In this example the maximum path strength between each alternative is the direct connection between them. The only positive p values are for (*red*,*blue*), (*red*,*pink*) and (*pink*,*blue*). They represent the preference relations and result in the order: $red \succ pink \succ blue$.

Schulze also gave an example algorithm to compute the path strength with a run-time of $O(|A|^3)$. For better understanding Algorithm 3.1 is provided.

The Schulz method also fulfills some important properties: anonymity, neutrality, monotonicity, Pareto Efficiency, Pareto property, non-dictatorship. It also fulfills the extra property independence of clones which means a large number of similar alternatives does not change the result of the election [Sch11].

3.5 Nanson's Rule

Nanson introduced a rule that uses the Borda score repeatedly to decide on a winner. Their method fulfills the Condorcet winner criterion. In general it is only used as SCF to determine a winner, but with a little adjustment it is also possible to build a full ranking [Nio87].

As already mentioned Nanson uses the Borda score for the rule. After calculation of the score, every alternative with less or equal than the average Borda score gets eliminated. In our case these eliminated alternatives will build the end of the ranking according to their score. Then this gets repeated only without the eliminated candidates in the ballots. This goes on until no alternative remains. From this the run-time can be derived.

Algorithm 3.1: Schulze path strength algorithm [Sch03]

Input: A matrix $d[i,j]$ where the value represents how often candidate i is preferred to j

Output: $p[i,j]$ the path strength from every i to every j

```
p = [[0 for i in 0..len(Candidates)] for j in 0..len(Candidates)]
for i ∈ Candidates do
    for j ∈ Candidates do
        if i ≠ j then
            p[i,j] = d[i,j] − d[j,i];
        end
    end
end
for i ∈ Candidates do
    for j ∈ Candidates do
        if i ≠ j then
            for k ∈ Candidates do
                if i ≠ k and j ≠ k then
                    s = min(p[j,i], p[i,k])
                    if s > p[j,k] then
                        p[j,k] = s
                    end
                end
            end
        end
    end
end
return p
```

As already mentioned in Section. 3.1 Borda can be computed in $O(n \cdot m)$ time with n the number of voters and m the number of alternatives. As Borda is repeated until no alternatives remain and every step will roughly halve the number of alternatives a complete run-time of $O(n \cdot m \cdot log(m))$ can be expected. It is important to note that changing the elimination criterion to less than the average instead of less or equal to the average would already lead to the violation of the Condorcet criterion [Nio87].

Example:
When going back to the example in Table 2.1 the Borda scores were $s(blue) = 2$, $s(red) = 4$ and $s(pink) = 3$. Their average is 3 so every alternative with a score less or equal to 3 gets eliminated. In this case only *red* remains and it is finished. The resulting ranking is again the same with $red \succ pink \succ blue$.

The properties of Nanson's rule are not the most promising as it for example fails monotonicity.

Example:
This can be seen in the example of Table 3.6. The first row shows how many voters voted with the given ranking. With this the first round has the scores: $red = 295, pink = 251, blue = 257, yellow = 197$. Here the average Borda score is 250 and with this *yellow* is eliminated. Then the new scores are $red = 216, pink = 186$ and $blue = 198$ where *red* is the only one above the average of 200 and therefore the winner [Nur04].

If the 12 voters that voted $pink \succ red \succ blue \succ yellow$ instead voted $red \succ pink \succ blue \succ yellow$ this would change the scores to $red = 307, pink = 239, blue = 257, yellow = 197$ which leads to the elimination of *pink* and *yellow*in the first round. Then *red* would loose to *blue* with 149 against 151. And thus increasing the rank of *red* worsened its rank [Nur04].

30	21	20	12	12	5
blue	*pink*	*red*		*red*	*red*
red	*yellow*	*pink*	*red*	*blue*	*blue*
yellow	*blue*	*yellow*	*blue*	*pink*	*yellow*
pink	*red*	*blue*	*yellow*	*yellow*	*pink*

Table 3.6: Example where Nanson can break monotonicity.

Because it is a Condorcet extension it also violates consistency. It at least fulfills anonymity, neutrality, non-dictatorship, Pareto Efficiency and Pareto property [Nur04].

3.6 Maximin Rule

The Maximin method was also originally designed to be a SCF. It uses the pairwise majority margin which was described in the Formula 3.1 as Net_p. As a reminder Net_p is a function $Net_p : A^2 \implies \mathbb{R}$ that takes two alternatives as parameter and returns how many voters prefer the first to the second minus how many voters prefer the second to the first [BCE+16].

Now every alternative gets a score that represents their worst majority margin. This score can be written as $\sigma_x = min_{y \neq x}(Net_p(x, y))$. Then the winner is/are the candidate(s) with the maximum σ_x or written as complete formula: $winner = max_x min_{y \neq x}(Net_p(x, y))$. To get a full ranking with this, the candidates can be ordered by their σ_x score from highest to lowest [BCE+16].

Example:
Again using this method to compute the result for the example in Table 2.1 the minimum majority margins result in the following values: $\sigma_{blue} = -1$, $\sigma_{red} = 1$ and $\sigma_{pink} = -1$. This results for the first time in a different ranking of $red \succ pink \sim blue$.

Rank	Voter 1	Voter 2	Voter 3	Voter 4	Voter 5
1.	*blue*	*blue*	*yellow*	*yellow*	*pink*
2.	*red*	*red*	*red*	*red*	*yellow*
3.	*pink*	*pink*	*pink*	*pink*	*red*
4.	*yellow*	*yellow*	*blue*	*blue*	*blue*

Table 3.7: Minimal σ: $\sigma_{blue} = -1$, $\sigma_{pink} = -3$ (against *red*), $\sigma_{yellow} = -1$ (against *pink*) and $\sigma_{red} = -1$ (against *yellow*). This results in the winners *blue*, *yellow* and *red* despite *blue* losing all pairwise comparisons.

This method fulfills the Condorcet winner property. However it fails the Condorcet loser property, which states that an alternative that loses to every other alternative in pairwise comparison should not be able to win which is possible for Maximin as shown in the example from Table 3.7. One positive aspect of this rule is, that it has a similar run-time as Copeland given the pairwise majority scores [BCE+16].

CHAPTER 4

Data used for the Experiments

One of the goals and prerequisites of this thesis is data collection. Or in other words collecting real world ranking data over a time frame. A good example for such data would be the daily most streamed spotify rankings. On each day there are multiple voters with their individual rankings.

Before introducing the data in the following sections of this chapter, it is necessary to explain in what format the data is collected. For this a data structure based on PREFLIB.ORG was chosen. The so called SOI format for incomplete strict orders. Each file in SOI form is structured the same. First a count of candidates, then a line for each candidate then some information about the rankings including how many voters, the overall weight of the votes and how many unique rankings there are. This is then followed by the rankings. Each one gets its own line starting with a number of how many votes this ranking gets [MW13].

Example:
An example for a standard SOI file with 3 alternatives and 4 voters can be seen here:

```
3
1, red
2, pink
3, blue
4,4,3
1,2,3
1,1,2
2,2,1
```

To use SOI for the collected data one easy removable change had to be done. In the experiments it is important to know who voted how. So instead of only counting how many voters voted for a ranking this number is set to 1 and the voter is written at the beginning of the line separated by a colon. The ballots are still strict and incomplete, meaning they represent a top-k ranking. In such a ballot the first position is strictly preferred to the second and so on and every not listed alternative is less preferred than every listed alternative and equally preferred to every other non listed one [MW13].

An additional change was that every file contains the list of all alternatives in all rankings independent of if they represent the same data point (e.g. the same day). This is important as the rankings only use the number assigned to an alternative. If this index would change between different data points, because different candidates made it into the rankings, it would be harder to compare rankings. For this thesis this changed SOI format is called TSOI for temporal incomplete strict orders.

Example:
Here is an example of how a file *colors.tsoi* could look. It contains three alternatives and four voters ("at","de","us" and "ca") with their top 2 choices:

```
3
1, red
2, pink
3, blue
4,4,3
at:1,2,3
de:1,1,2
us:1,2,1
ca:1,2,1
```

As a way to analyze the data the methods from Section 2.3 will come to use. Especially Kendall tau-b as it gives a more evenly spread view of the data and the intersection size as it also complements the meaning of the tau-b metric. These methods will be used on every pair of voters over all data points and to interpret the data the arithmetic mean of the intersection sizes or tau-b scores between them should give a good overview.

For this chapter Kendall tau type-b was chosen over type-a, as it is easier to look at. However as Kendall tau type-a already indicates some degree of intersection size through values close to 0 in case of low intersection size it will be the version used in Chapter 5 and the experiments.

This chapter takes a look at different data sets. First some basic information about the data is given and then statistical analysis of the Kendall tau scores and intersection size between the rankings of the voters is presented. Furthermore some information for the similarity between rankings of one voter over time can also be useful. For this Kendall tau-b between one ranking and the next older ranking of each voter will be calculated for the whole data sets.

4.1 Eurovision Song Contest

The first collected data set is from the Eurovision Song Contest. These results are provided by [Od19]. It represents the historical jury votes in the finals for each year. This data set will probably have one of the lowest correlation between each data point and therefore the correlation between the voters would be the most important factor of how successful the experiments will be on this data set.

In total 45 years of voting results are available. Together they contain 54 candidates (one region is a candidate not the singer) and 53 regions that act as voters. However at any given year at most 26 candidates received votes and at most 43 regions voted. Despite some changes over time, each region voted for their top 10 in a large part of the data.

However the mean values of the Kendall tau-b distance between the voters seems to be rather low as shown in Figure 4.1. Here and in all following figures a black field means that these voters never voted at the same data point. Pink means low or negative correlation, white neutral correlation and green higher or positive correlation. All this pink and white to weak green does not mean that the different voters never agree, but it means statistics will not give a good estimate of how much they agree at a given data point.

In Table 4.1 it can be seen that for the Eurovision data set the votes of a region at one data point have low correlation to the next data point. The majority of voters have on average a negative Kendall tau-b score between all their consecutive votes and even the maximum reached Kendall tau-b scores are not particularly high. For this it is to note that if a voter skipped one data point their rankings are still compared but instead of with the directly next data point with the next one containing a vote of this voter. There are regions that only voted in one Song Contest, they are exempt from this list and have probably no method to find replacements for them in the experiments as with one data point no really statistical meaningful values can be gained. But they will still stay as part of the data set for completion sake.

voter	avg	min	max
Australia	-0.38	-0.42	-0.33
Turkey	-0.07	-0.43	0.52
Luxembourg	-0.2	-0.57	0.15
Monaco	-0.21	-0.62	0.13
Albania	-0.07	-0.22	0.25
Yugoslavia	-0.15	-0.46	0.29
Estonia	-0.2	-0.49	0.22
Moldova	0.0	-0.49	0.35

Belarus	-0.11	-0.57	0.32
Germany	-0.23	-0.69	0.34
Spain	-0.21	-0.66	0.29
Montenegro	-0.35	-0.6	-0.06
Armenia	-0.08	-0.46	0.32
Switzerland	-0.16	-0.54	0.31
Serbia & Montenegro	0.09	0.05	0.13
San Marino	-0.43	-0.66	-0.21
The Netherlands	-0.2	-0.54	0.33
Serbia	-0.29	-0.55	-0.08
Israel	-0.18	-0.53	0.24
Iceland	-0.28	-0.64	0.27
Andorra	-0.25	-0.37	-0.15
Bulgaria	-0.2	-0.55	0.29
Sweden	-0.25	-0.61	0.35
Poland	-0.32	-0.55	0.05
Russia	-0.3	-0.57	0.13
Austria	-0.19	-0.65	0.48
Romania	-0.21	-0.59	0.16
Denmark	-0.12	-0.41	0.24
Norway	-0.16	-0.55	0.18
Finland	-0.22	-0.54	0.18
Ukraine	-0.25	-0.54	0.08
Cyprus	-0.15	-0.56	0.27
F.Y.R. Macedonia	-0.23	-0.64	0.4
Azerbaijan	-0.19	-0.48	0.18
Slovakia	-0.3	-0.57	0.01
Bosnia & Herzegovina	-0.15	-0.44	0.46
Ireland	-0.26	-0.67	0.16
Malta	-0.22	-0.53	0.02
Greece	-0.2	-0.56	0.27
Portugal	-0.22	-0.62	0.25
Georgia	-0.12	-0.56	0.27
Italy	-0.26	-0.63	0.27
Czech Republic	-0.18	-0.45	0.19
France	-0.21	-0.71	0.25
Slovenia	-0.23	-0.63	0.09
United Kingdom	-0.28	-0.61	0.2
Lithuania	-0.21	-0.53	0.22
Croatia	-0.29	-0.7	0.18
Latvia	-0.25	-0.57	0.08
Belgium	-0.22	-0.54	0.32
Hungary	-0.32	-0.59	0.04

Table 4.1: Statistics of different voters at the Eurovision Song Contest between their votes of one year compared to their votes of the previous year.

4.2 Spotify

Spotify is one of the most known music streaming services as of the time this thesis is written. Their service is available in many countries around the world. They also offer some useful rankings for the regions they provide their service to. All together they have four data sets public available through their API [Spo19]. These will be described in the following sections.

4.2.1 Spotify Daily Charts

This is the biggest data set that will be used for this thesis. It contains 49967 alternatives and 61 regions (including a global version) that act as voters. Not every region has data for every day, but sometimes all of the regions have rankings available for one day. The highest number of alternatives that received votes on one given day is 3690 and this is reached with at most top 200 ballots per region (some exceptions are only top 1). The data was taken from the 1^{st} of January 2017 to the 13^{th} of May 2019, resulting in a total of 860 days or data points.

When looking at the average Kendall tau-b distance for the voters which can be seen in Figure 4.2 it seems a lot more promising with some scores getting above 0.6. But this alone does not guarantee the usefulness in the experiments. Here it could be also helpful to show the mean intersection size between the ballots of the voters as seen in Figure 4.3. This shows one of the weak points of Kendall tau-b as some voters have for example a Kendall tau-b score of 0.3 while at the same time only having 30 out of 200 candidates in common.

4.2.2 Spotify Weekly Charts

Here again each region has their up to top 200 most streamed songs, but this time not daily but per week. Therefore the number of data points goes down to 123 starting with the week of the 23^{rd} of December 2016 until the week of the 26^{th} of April 2019. For one week at most 62 regions had rankings available and at most 3774 candidates made it into the rankings of one week. Overall 32949 alternatives made it into at least one ballot of the total 63 participating voters.

Again the mean Kendall tau-b correlations and the mean intersection sizes between the ballots of the voters can be seen in Figure 4.4 and Figure 4.5. This time they look more similar compared to the statistics for Spotify daily charts. Some of the regions even reach correlation scores of around 0.5 which could proof useful.

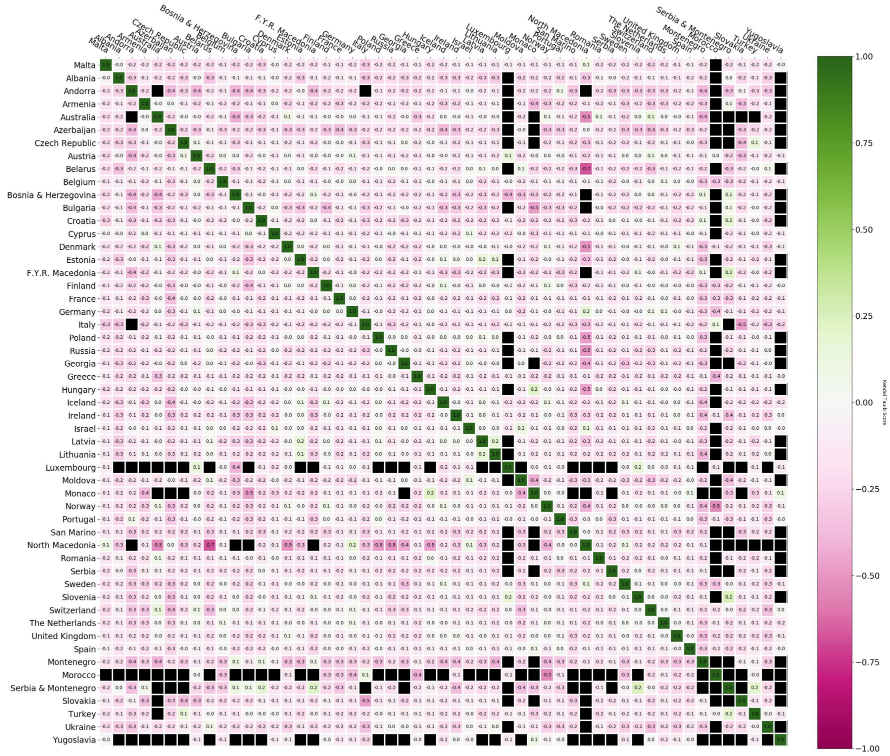


Figure 4.1: Mean Kendall tau-b score between each voting region for the Eurovision Song Contest.

4.2.3 Spotify Viral Daily

The viral data are top 50 candidates, but they depend on social media and not on streaming counts. For viral daily 856 days were collected stating with the 1^{st} of January 2017 up until the 12^{th} of May 2019. In total 86341 candidates made it into at least one ballot of the 61 voting regions. At most 61 regions voted on one day for at most 1966 different candidates.

As these numbers already imply, there is nearly no correlation between the different voters as shown in Figure 4.6.

4.2.4 Spotify Viral Weekly

Now to the viral version of the weekly top 50. Here up to 64 regions voted together for up to 1941 different candidates per week. Overall 64 voters and 68378 candidates are

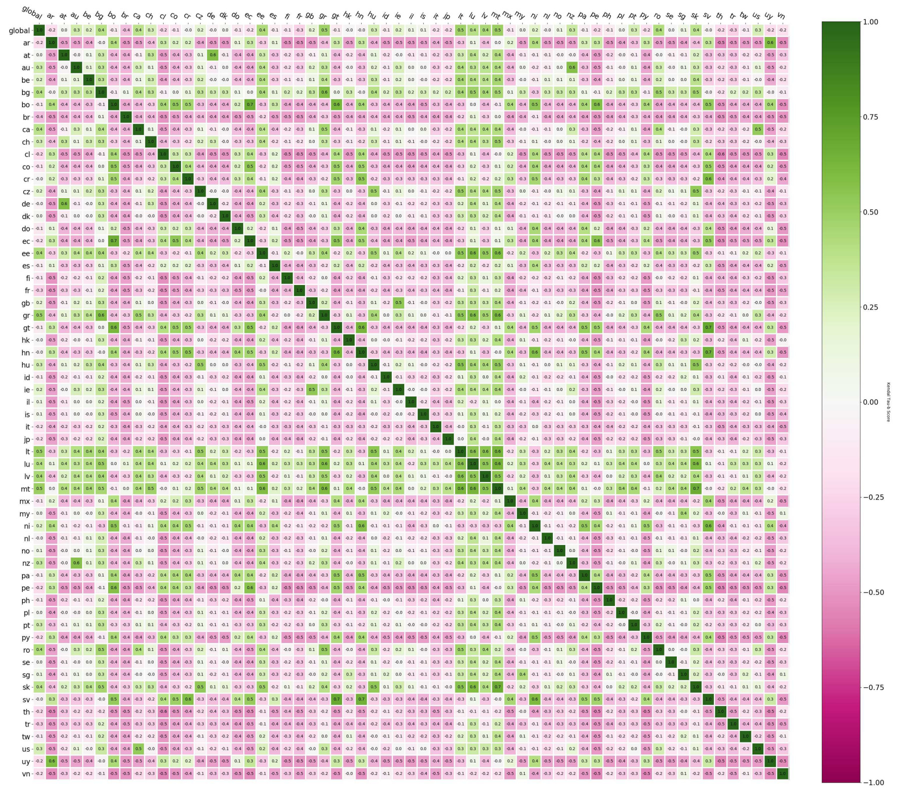


Figure 4.2: Mean Kendall tau-b score between each voting region for the Spotify daily charts.

in this data set. The data starts in the week of the 5^{th} of January 2017 and ends with the week of the 25^{th} of April 2019. As this data set has such a similar low correlation between the voters compared to viral daily data no extra Figure will be provided.

4.2.5 Similarities within the same Voter for Spotify

Here the similarity between data points and their past of all the Spotify data sets are shown in Table 4.2. This data clearly shows that for daily Charts and weekly charts the votes do not change too much between the data points. Especially daily Charts have extreme high Kendall tau-b scores, as even the minimum score between two consecutive data points is almost for all voters positive and the maximum scores are often nearly perfect. Weekly charts are still pretty similar, but not to that extent of the daily data set.

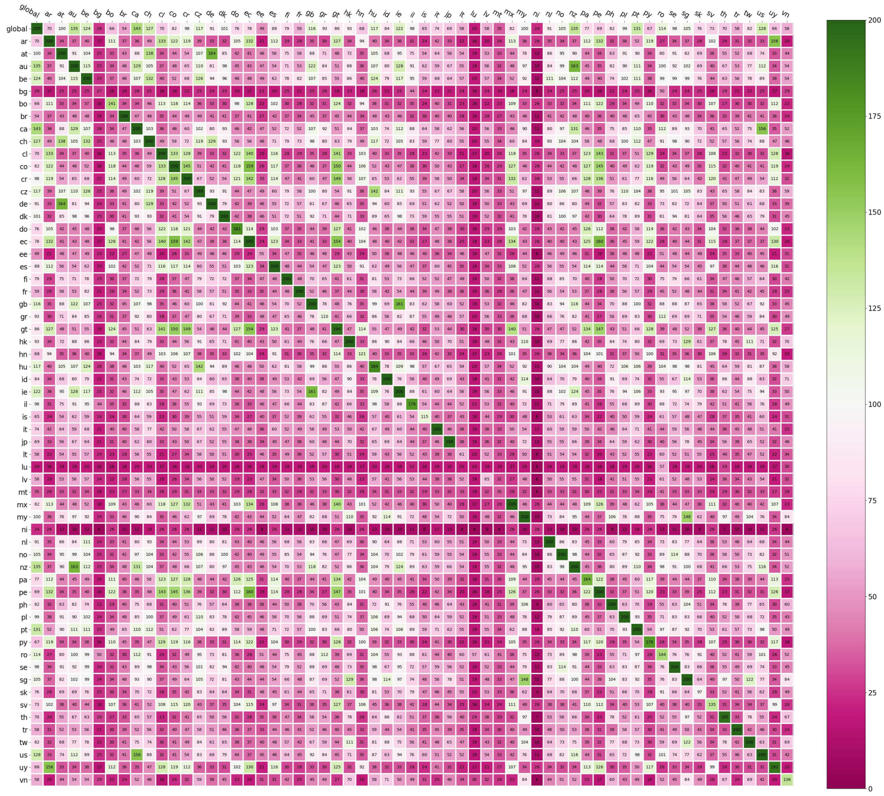


Figure 4.3: Mean intersection size between each voting region for the Spotify daily charts. White already means they share around halve of their alternatives.

The viral daily data set still has acceptable average scores, however the minimum scores are for most voters negative values which could effect the experiments to have more neutral changes as it not as consistent as the non viral data sets. The viral weekly data set on the other hand suggest mostly negative similarity and even their maximum scores are pretty low.

To the Table 4.2 itself, a − means that no data is available to calculate the values. This is most likely because a region does not participate in one of the rankings, like some of them do not have daily rankings available. The reason behind this was not examined for this thesis.

	daily charts			viral daily charts			viral weekly charts			weekly charts		
voter	avg	min	max	avg	min	max	avg	min	max	avg	min	max
de	0.85	0.35	0.97	0.64	-0.23	0.92	-0.18	-0.63	0.36	0.59	0.04	0.8

sv	0.89	0.33	0.95	0.65	0.13	0.87	-0.01	-0.64	0.37	0.79	0.5	0.89
fr	0.84	0.15	0.96	0.61	-0.36	0.94	-0.23	-0.63	0.39	0.58	0.15	0.85
py	0.89	0.59	0.95	0.64	-0.03	0.94	-0.04	-0.65	0.44	0.79	0.57	0.88
tr	0.87	0.43	0.95	0.66	-0.1	0.92	-0.11	-0.59	0.33	0.74	0.5	0.87
gb	0.79	0.0	0.95	0.64	-0.29	0.94	-0.21	-0.64	0.57	0.63	-0.01	0.83
id	0.87	0.48	0.95	0.66	-0.16	0.92	-0.1	-0.55	0.31	0.74	0.52	0.88
ar	0.88	0.42	0.96	0.66	-0.46	0.93	-0.09	-0.62	0.49	0.79	0.58	0.9
hu	0.79	0.2	0.93	0.59	-0.15	0.88	-0.21	-0.66	0.22	0.62	-0.15	0.82
it	0.84	0.34	0.95	0.65	-0.43	0.93	-0.19	-0.6	0.47	0.64	0.25	0.89
mx	0.88	0.29	0.96	0.7	-0.05	0.95	-0.03	-0.58	0.56	0.79	0.55	0.89
cl	0.9	0.29	0.97	0.69	-0.1	0.95	0.01	-0.57	0.65	0.79	0.52	0.91
nz	0.86	-0.02	0.95	0.61	-0.25	0.88	-0.16	-0.67	0.43	0.7	0.14	0.87
bg	0.69	-0.66	0.95	0.63	-0.36	0.9	-0.11	-0.58	0.35	0.59	0.29	0.78
ch	0.82	0.26	0.94	0.55	-0.25	0.89	-0.3	-0.67	0.28	0.59	0.04	0.81
tw	0.73	0.25	0.94	0.57	-0.16	0.9	-0.27	-0.67	0.36	0.62	0.4	0.79
es	0.86	0.42	0.97	0.65	-0.27	0.94	-0.15	-0.59	0.46	0.75	0.46	0.87
be	0.81	0.19	0.94	0.58	-0.19	0.87	-0.23	-0.63	0.28	0.62	0.11	0.84
pl	0.81	0.21	0.94	0.61	-0.32	0.9	-0.2	-0.64	0.37	0.64	-0.02	0.82
at	0.82	0.22	0.95	0.56	-0.17	0.87	-0.28	-0.67	0.26	0.6	0.02	0.83
my	0.85	0.36	0.95	0.66	-0.15	0.9	-0.08	-0.66	0.4	0.7	0.33	0.84
us	0.85	-0.02	0.97	0.65	-0.47	0.96	-0.28	-0.64	0.38	0.61	0.27	0.84
hn	0.89	0.52	0.95	0.65	-0.21	0.98	-0.04	-0.67	0.42	0.78	0.46	0.88
ec	0.89	0.51	0.95	0.68	0.01	0.92	0.04	-0.58	0.48	0.8	0.52	0.9
il	0.81	0.36	0.93	0.61	0.02	0.88	-0.09	-0.43	0.28	0.69	0.35	0.89
gr	0.82	0.3	0.96	0.6	-0.1	0.87	-0.16	-0.63	0.26	0.62	0.27	0.83
cr	0.88	0.44	0.95	0.65	-0.18	0.92	-0.05	-0.67	0.52	0.78	0.43	0.9
gt	0.88	0.4	0.95	0.64	-0.28	0.98	-0.04	-0.67	0.54	0.78	0.52	0.89
au	0.86	-0.12	0.96	0.64	-0.28	0.94	-0.14	-0.66	0.56	0.69	0.17	0.85
lt	0.78	-0.35	0.94	0.58	-0.36	0.88	-0.2	-0.66	0.16	0.59	0.02	0.77
se	0.79	0.01	0.94	0.65	-0.22	0.92	-0.19	-0.62	0.44	0.6	-0.21	0.81
no	0.83	0.17	0.96	0.63	-0.27	0.91	-0.21	-0.68	0.32	0.64	0.06	0.81
ad	-	-	-	-	-	-	-0.6	-0.82	-0.14	0.67	0.3	1.0
sg	0.88	0.27	0.95	0.62	-0.16	0.9	-0.15	-0.65	0.49	0.72	0.46	0.88
fi	0.83	0.01	0.95	0.59	-0.4	0.88	-0.29	-0.65	0.18	0.59	0.05	0.78
cz	0.79	-0.08	0.92	0.58	-0.22	0.87	-0.24	-0.67	0.33	0.63	0.05	0.82
vn	0.8	0.21	0.91	0.67	-0.1	0.88	-0.04	-0.46	0.36	0.63	0.36	0.8
ie	0.8	0.08	0.95	0.6	-0.27	0.91	-0.2	-0.67	0.43	0.64	0.13	0.82
global	0.88	0.13	0.98	0.67	-0.61	0.97	-0.19	-0.65	0.47	0.67	0.25	0.85
do	0.87	0.41	0.95	0.64	-0.09	0.89	-0.06	-0.67	0.5	0.75	0.41	0.87
is	0.82	0.27	0.95	0.59	-0.03	0.9	-0.22	-0.7	0.25	0.63	-0.17	0.82
uy	0.87	0.39	0.95	0.63	-0.12	0.89	-0.08	-0.67	0.36	0.77	0.45	0.89
nl	0.81	0.13	0.96	0.61	-0.34	0.93	-0.21	-0.67	0.51	0.58	0.15	0.81
br	0.86	0.54	0.96	0.68	-0.37	0.95	-0.14	-0.58	0.4	0.76	0.5	0.91

ni	0.82	-0.67	0.95	0.64	0.02	0.84	-0.01	-0.66	0.39	0.79	0.45	0.88
lu	0.79	-0.65	1.0	0.56	-1.0	1.0	-0.34	-0.75	0.36	0.59	0.0	0.81
th	0.85	0.56	0.94	0.7	-0.15	0.89	-0.03	-0.5	0.35	0.65	0.45	0.77
jp	0.86	0.41	0.95	0.53	-0.63	0.89	-0.33	-0.65	0.24	0.7	0.44	0.85
co	0.88	0.17	0.96	0.65	-0.26	0.91	-0.11	-0.6	0.52	0.78	0.48	0.9
dk	0.82	0.08	0.95	0.63	-0.24	0.91	-0.2	-0.63	0.42	0.63	-0.16	0.82
ee	0.78	-0.44	0.95	0.58	-0.29	0.93	-0.22	-0.67	0.28	0.62	0.09	0.82
mt	0.79	-0.66	0.95	0.62	-0.15	1.0	-0.13	-0.68	0.45	0.64	0.16	0.82
pt	0.83	0.24	0.94	0.63	-0.26	0.9	-0.15	-0.52	0.39	0.69	0.36	0.83
pe	0.91	0.46	0.97	0.7	-0.15	0.93	0.06	-0.47	0.56	0.81	0.53	0.91
pa	0.88	0.48	0.95	0.65	-0.14	0.9	-0.02	-0.67	0.45	0.76	0.43	0.88
bo	0.89	0.48	0.95	0.67	0.16	0.87	0.03	-0.67	0.5	0.79	0.5	0.9
hk	0.85	0.25	0.94	0.59	-0.14	0.87	-0.21	-0.66	0.24	0.68	0.4	0.82
lv	0.78	0.17	0.93	0.58	-0.12	0.86	-0.23	-0.68	0.21	0.61	-0.01	0.78
cy	-	-	-	-	-	-	-0.34	-0.71	0.25	0.61	0.26	0.81
ph	0.89	0.47	0.96	0.68	-0.22	0.92	-0.08	-0.56	0.4	0.76	0.46	0.88
ro	0.79	0.28	0.92	0.64	-0.02	0.85	-0.06	-0.45	0.38	0.58	0.27	0.78
ca	0.86	0.21	0.96	0.63	-0.32	0.92	-0.24	-0.65	0.4	0.62	0.0	0.84
mc	-	-	-	-	-	-	-0.67	-1.0	-0.35	-	-	-
sk	0.81	0.17	0.94	0.58	-0.22	0.88	-0.24	-0.68	0.22	0.6	0.11	0.8

Table 4.2: Statistics to show past similarities within the Spotify data sets between days/weeks.

4.3 I-Phone App Store Rankings

These next data sets were all collected with the service provided by [Oy]. They allow to access ranking data for major app stores. At the beginning of collecting the data their service provided the top 50. This was later extended, but it was decided to stay with top 50 data for this thesis.

In an attempt to get a smaller diversity in apps the I-Phone rankings were chosen, as they need more approval than Android apps. The service provides three different rankings in "free", "paid" and "top grossing" apps. It is also possible to choose a genre and the region, however the data of many regions is regularly not available and therefore 11 regions were chosen for all these data sets. As genres "Games" and "News" were selected. And the time span is the same for all of them with 15^{th} of March 2019 until 15^{th} of May 2019 to a total of 62 days per data set.

4.3.1 Free I-Phone Games

These rankings have a total of 753 alternatives and at most 190 of them made it into the top 50 of one of the regions per day. The mean Kendall tau-b distance and mean intersection size between the regions can be seen in Figure 4.7. From this it is clear that

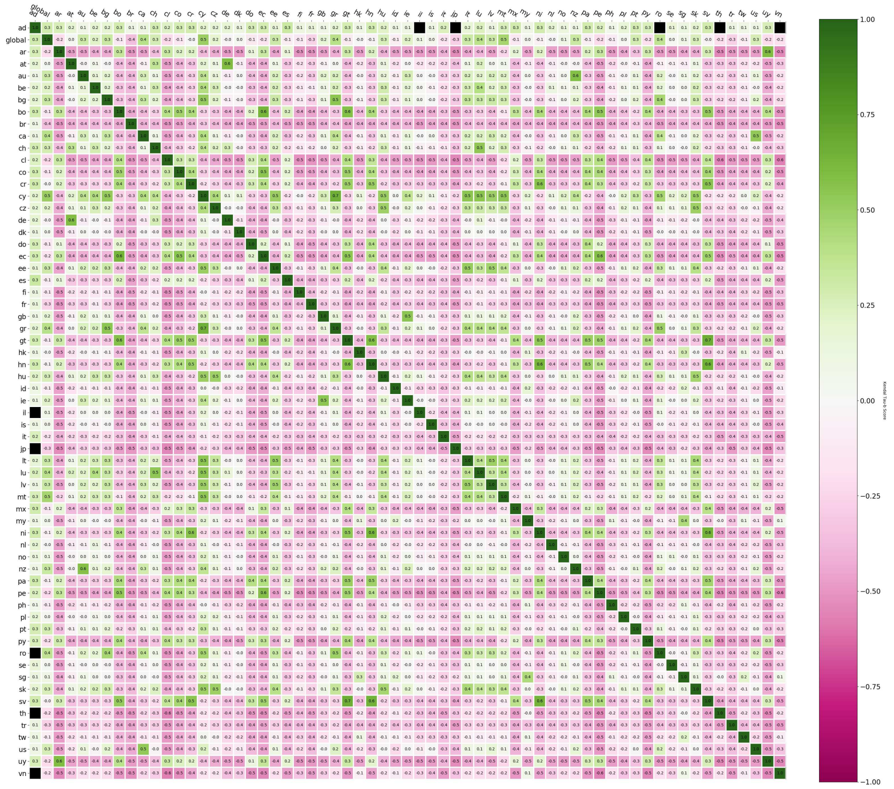


Figure 4.4: Mean Kendall tau-b score between each voting region for the Spotify weekly charts.

Japan represented with "jp" is an outlier that has really low correlation with any other region.

4.3.2 Paid I-Phone Games

Here at most 274 of the total 1154 alternative game apps made it into the rankings on any given day. And the statistics can be seen in Figure 4.8. This data set has much lower mean correlations with really only the "USA" having tau-b values over 0.1 with more than one other country.

4.3.3 Top Grossing I-Phone Games

With a total of 528 and at most 201 alternatives on a single day the statistics for this data set can be seen in Figure 4.9. Compared to the other two I-Phone games charts

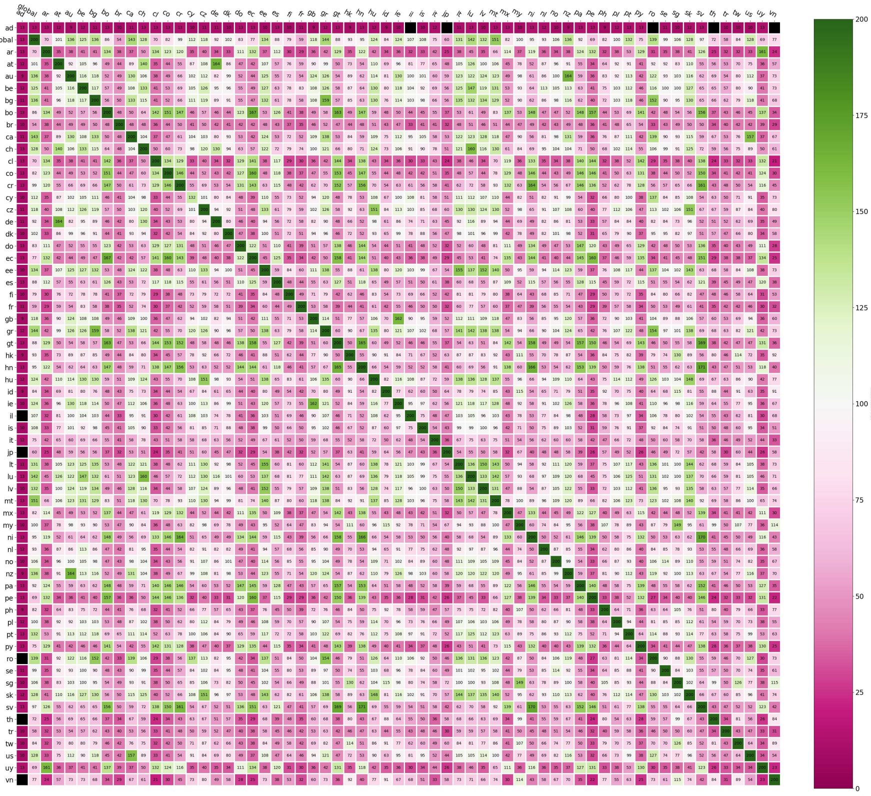


Figure 4.5: Mean intersection size between each voting region for the Spotify weekly charts.

this one has some of the highest correlation between some voters, but "Japan" is still an outlier.

4.3.4 Top 50 Paid I-Phone News Apps

As there are already many data sets with extremely low correlation only the paid news apps were chosen to be included as extra data set. Here at most 166 alternatives made it into the rankings of one day and only 240 apps made it into the whole data set. Figure 4.10 shows the correlation statistics. As the diversity of alternatives is not very high in these ballots the intersection sizes are often relatively high, however the tau-b scores stay near 0. These values could lead to different approaches in the experiments.

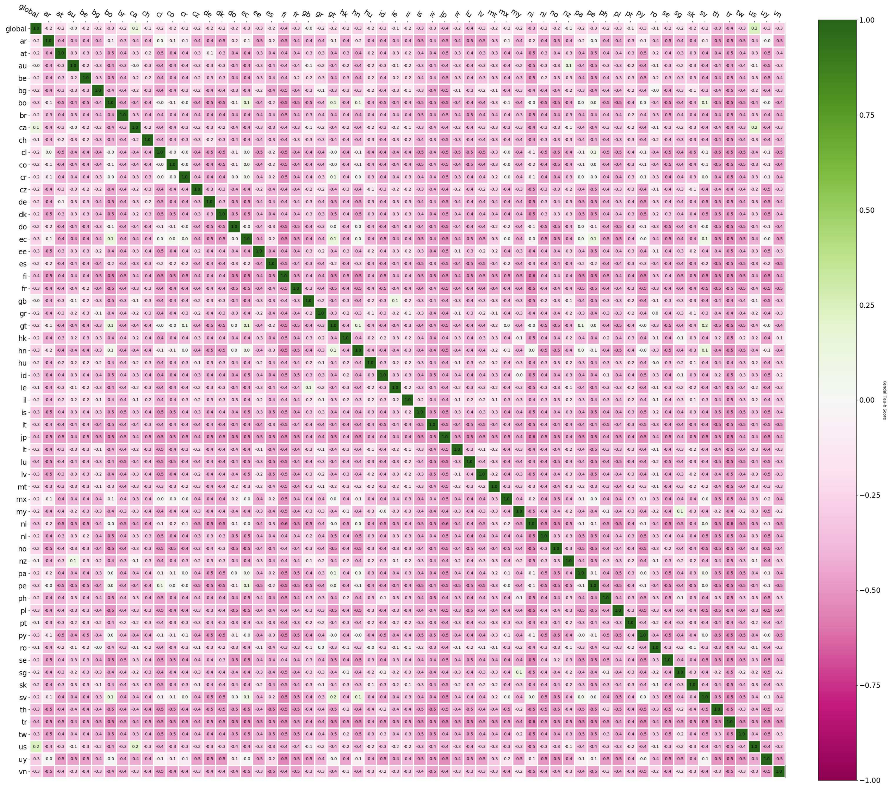
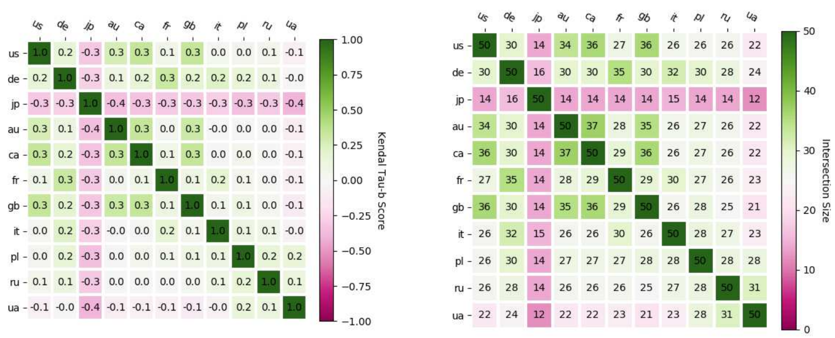
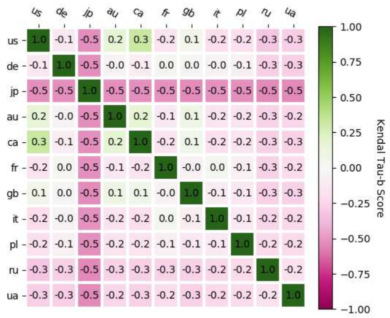


Figure 4.6: Mean Kendall tau-b score between each voting region for the Spotify viral daily charts.



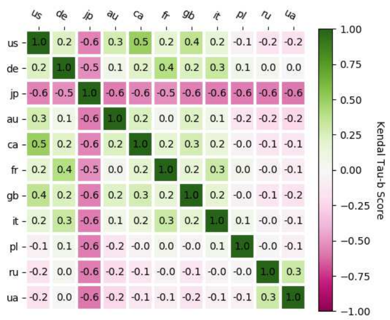
(a) Mean Kendall tau-b score. (b) Mean intersection size.

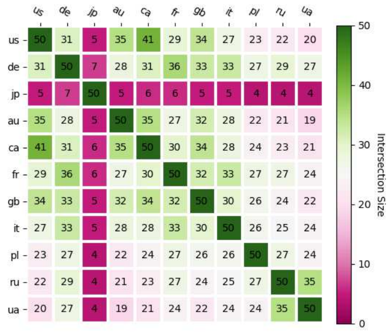
Figure 4.7: Statistics between the regions for free I-Phone Games charts.



(a) Mean Kendall tau-b score. (b) Mean intersection size.

Figure 4.8: Statistics between the regions for paid I-Phone Games charts.





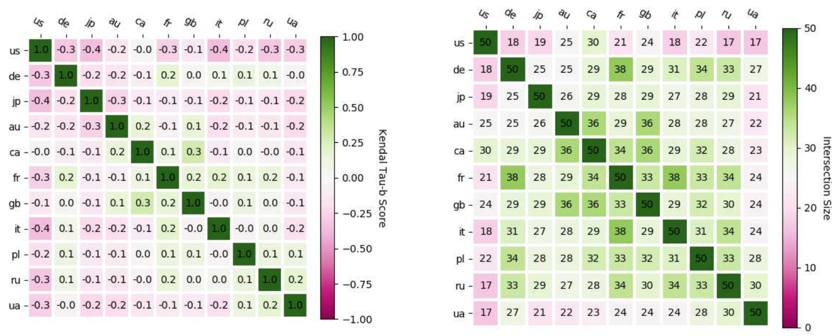
(a) Mean Kendall tau-b score. (b) Mean intersection size.

Figure 4.9: Statistics between the regions for top grossing I-Phone Games.

4.3.5 Past Similarities for I-Phone App Store Data

And to finish the statistical data of the I-Phone App Store charts Table 4.3 gives an overview of the similarities between consecutive data points. Here it is interesting to see that for most voting regions at least one day had the same ranking as the previous day as shown by the Kendall tau-b score of 1.0. Free games, top grossing games and especially paid news apps have mostly promising average scores and not even much too low minimum scores, which could lead to a more consistent result in the later experiments. The average scores of over 0.9 for news apps will hopefully lead to really good results in the experiments.

The data set for paid Games however has a more mixed result. Some high average scores, some low but still positive ones and the minimum scores even have some negative scores. Therefore it can be expected that a method that concentrates on past rankings to find a



(a) Mean Kendall tau-b score. (b) Mean intersection size.

Figure 4.10: Statistics between the regions for paid I-Phone News charts.

replacement will neither perform especially good nor especially bad on this data set.

	free Games			top grossing Games			paid Games			paid News Apps		
voter	avg	min	max	avg	min	max	avg	min	max	avg	min	max
fr	0.63	0.32	1.0	0.68	0.39	1.0	0.49	0.23	1.0	0.97	0.88	1.0
ru	0.75	0.42	1.0	0.72	0.51	1.0	0.66	0.48	1.0	0.97	0.9	1.0
ca	0.59	0.29	0.83	0.61	0.47	0.72	0.49	0.3	0.69	0.91	0.74	1.0
it	0.71	0.37	1.0	0.65	0.44	1.0	0.35	-0.0	1.0	0.94	0.74	1.0
au	0.58	0.34	0.8	0.66	0.48	0.77	0.45	0.32	0.59	0.9	0.72	1.0
ua	0.71	0.28	1.0	0.58	0.28	1.0	0.25	-0.31	1.0	0.82	0.71	1.0
pl	0.72	0.41	1.0	0.54	0.31	1.0	0.29	-0.08	1.0	0.96	0.86	1.0
gb	0.78	0.52	1.0	0.74	0.52	1.0	0.66	0.34	1.0	0.87	0.53	1.0
de	0.7	0.35	1.0	0.73	0.52	1.0	0.56	0.32	1.0	0.89	0.68	1.0
jp	0.71	0.3	1.0	0.6	0.2	1.0	0.74	0.57	1.0	0.94	0.76	1.0
us	0.75	0.52	0.88	0.73	0.61	0.83	0.75	0.62	0.85	0.58	0.38	0.75

Table 4.3: Statistics for the similarity to past data points in the I-Phone data sets.

4.4 Summary of the Data Sets

To give ans overview of all the data sets here is a short summary. From the 3 data sources (Spotify, I-Phone, Song Contest) a total of 9 data sets were collected. Between them all 2253 data points were collected in separate SOI files.

The highest correlations between voters were found in the Spotify daily charts, Spotify weekly charts and top grossing I-Phone games. The similarities for free I-Phone games charts and paid I-Phone News charts also show at least some positive tau-b scores and their intersection sizes are also very promising as a few of the voters share on average over 30 from their 50 alternatives. However they all show no exceedingly high correlations (tau-b of at least 0.6 and green intersection size at the same time). Often a higher tau-b scores mean lower intersection size, which is not optimal as it only means that the voters agree on the few alternatives they share. This can be seen in the Spotify daily charts, where the tau-b scores are often positive, but the intersection size is most of the time less than halve of the 200 alternatives that each voter can vote. And even if both scores between two voters are on rather high (indicated by darker green), the tau-b score between two voters seldom reaches values above 0.6.

The other data sets showed even lower correlation. Therefore Eurovision Song Contest, paid I-Phone games charts and Spotify viral daily/weekly charts are expected to fare worse in the experiments that only use correlation between voters, than the other data sets.

The correlation to past data points seems much more promising in most data sets. There are only two outlier with mostly negative correlation and those are from Song Contest and Spotify viral weekly charts. All the other data sets show at least good correlation between directly following data points with tau-b scores of mostly above 0.6. Some of them like Spotify daily charts and paid I-Phone News charts even manage to reach values above 0.8. Because of this experiments that use previous rankings of voters are expected to have mostly positive effects.

CHAPTER 5

Implicit Delegation Methods

With all the data collected it can be started to design implicit delegation algorithms to be tested on it. As already mentioned implicit delegation means to choose some representative indirectly. It is indirect as the voter does not give the vote themselves but the algorithm derives it from previous votes of the voter themselves and from other voters. For this thesis this means to find a preference ranking for a voter that did not give their vote at one point in time i.e, in the cas of missing preference data.

An implicit delegation method describes how the delegate (replacement) is chosen. It takes as input multiple consecutive data points, or in other words profiles sorted by their date from newest to oldest. From the newest profile some voters and their ballots are missing and these need to be replaced. To make this possible the algorithm also gets a list of the missing voters as input.

With this input the algorithms described in Section 5.1 will start to analyze the given data and assign scores to every possible replacement ranking for each of the missing voters. For this only data from before the newest provided data point is used to find similarities between voters or past data points. Afterwards these rankings will be filtered with the help of their associated scores to remove unsuitable ones. This will be described in more detail in Section 5.2.1.

Finally the remaining replacement rankings for each missing voter can be used to produce the output for the implicit delegation method. For this thesis multiple different outputs are produced that will all be compared and tested in Chapter 6. How these outputs are decided on is described in Section 5.2.2 and Section 5.2.3. If the different methods are a success, they should output replacement rankings for each missing voter that are as close to the removed ones as possible. This is of course only testable in the experiments, as real life applications of these methods would not have the real missing votes available.

The final two sections of this chapter will show how these methods can be used for different goals. The first goal, which is described in Section 5.3, is to allow taking every voter into account, even if one of them did not cast a vote in this particular profile. To achieve this the methods need to only fill in a missing vote if the replacement is with a high probability really similar to what would have been voted, as otherwise the aggregation of the profile would be even less similar to the missing voters preferences. Section 5.4 looks at the goal of prediction top-k alternatives of a single requested voter. Therefore the method for this prediction does not need a list of missing voters and instead only the one voter as input together with the profiles.

5.1 Analysis with Search Algorithms

To start with predicting preferences the given data needs to be analyzed. For this algorithms are needed to traverse and rate the data. These algorithms will use some Kendall tau version together with some normalized intersection size version, which will be discussed later. In this section they will only be called Kendall tau or tau-a and intersection size (i-size) to not distract from the overall algorithms. This part of the implicit delegation algorithms will in the following sections and chapters be called search algorithms, because they search for suitable rankings through analyzing the data.

Each of these search algorithms needs to be executed for every missing voter separately. For every call to them the input profiles are the same and only the requested voter differs. This also means that all of the missing voters are not present in the profile to complete, as a missing ranking should not be able to replace another missing ranking.

5.1.1 Analyzing Similarity to Different Voters

The first method that will be describe here is pretty similar to the different voter analysis from Chapter 4. For this analysis some number of past profiles will be looked at. The newest data points of the input is the one were replacement rankings need to be filled in. Because of this only the closest preceding data points to this requested one will be selected for the analysis. This means if the data is daily and the requested prediction is something like German charts for $Sep.6^{th}2019$, then $Sep.5^{th}2019$, 4^{th}, 3^{rd} and so on will be analyzed if there are German votes available. This way t data points will be selected where t can be selected according to available computing power or required accuracy.

Depending on the data set a higher t could produce a more accurate representation of the similarity or a worse if one of the voters had a strong shift in preferences within the time frame. To avoid misrepresentation of the similarity between two voters older dates can be weighted less than newer dates.

Now a description for the traversal itself is needed. The algorithm compares the voting/ranking of the desired voter to every other ranking in the same profile. As comparison both Kendall tau and intersection size are used and both are stored in separate lists associated with the voter the comparison was with. This is then repeated for every selected data point that contains a ballot of the missing voter. Afterwards the resulting lists are averaged with either the arithmetic mean or some weighted version of it, where later data is weighted less.

The experiments of this thesis use a t value of 15, however one data point is only counted if the missing voter is part of that data point. For the average a weighted mean was chosen to be used in all search algorithms. This weighted version reduces the weight of every data point by 10%. Meaning the i^{th} data point where $i \geq 1$ has a weight of 0.9^{i-1}. The following example will show this.

Example:
To make this all easier to understand here is an example where the input profile list can be seen in Table 5.1. Here the requested voter would be "Germany" and the missing ranking would be from $Sep.6^{th}2019$. If the algorithm described in this section is used the first step is to take the profile from $Sep.5^{th}2019$. From this profile the first comparison could be $ranking_{g5}$ and $ranking_{a5}$, lets say the resulting tau-a score is 0.6 and the intersection size percentage is 0.5. The same comparisons are then done between $ranking_{g5}$ and $ranking_{i5}$ and lets say the results are a tau-a score of 0.8 and an intersection size percentage of 0.9.

These comparisons are then repeated for the profiles from $Sep.4^{th}$ and $Sep.3^{rd}$. After this the results could be between "Germany" and "Austria" the tau-a scores were the already mentioned 0.6, then 0.7 and 0.5. For intersection size the values could be 0.7, 0.7 and 0.6. For the similarity between "Germany" and "Italy" the scores could be like this: tau-a$= 0.8, 0.85, 0.9$ and intersection size$= 0.9, 0.95, 0.95$.

Then the scores need to be averaged with a weight that reduces by 10% for each data point back. Average Tau-a between "Germany" and "Austria" can be calculated with $(0.6 \cdot 1.0 + 0.7 \cdot 0.9 + 0.5 \cdot 0.81)/(1 + 0.9 + 0.81) = 0.603$ and intersection size: $(0.7 \cdot 1.0 + 0.7 \cdot 0.9 + 0.6 \cdot 0.81)/(1 + 0.9 + 0.81) = 0.670$ The same way the average scores for "Germany" and "Italy" can be calculated and result in the following scores: tau-a$= 0.846$ and intersection size$= 0.932$.

These values are then taken as expected scores for the rankings $ranking_{a6}$ (scores from comparison with "Austria") and $ranking_{i6}$ (scores from comparison with "Italy"). The methods described in Section 5.2 will then decide what to do with these rankings. In contrast to the experiments of this thesis this example used only a value of 3 for the number of profiles to analyze (t).

This approach requires $t \cdot n$ comparison operations with t the number of data points to analyze and n the number of voters. The number of voters that is used here only contains the voters that are present in the profile where the missing voters need to be filled in, as comparisons with other voters rankings would be unnecessary. To get a run-time estimate from this the Kendall tau and intersection size metrics need to be considered. A basic intersection size metric can easily be computed in $O(m)$ with m the number of alternatives. However Kendall tau has as already mentioned with an efficient algorithm a

date	Germany	Austria	Italy
$Sep.6^{th}2019$	to-replace	$ranking_{a6}$	$ranking_{i6}$
$Sep.5^{th}2019$	$ranking_{g5}$	$ranking_{a5}$	$ranking_{i5}$
$Sep.4^{th}2019$	$ranking_{g4}$	$ranking_{a4}$	$ranking_{i4}$
$Sep.3^{rd}2019$	$ranking_{g3}$	$ranking_{a3}$	$ranking_{i3}$

Table 5.1: Example for temporal input profiles for different voter search algorithm.

run-time of $O(m \cdot log(m))$. Therefore the overall computation time with these two basic metrics is $O(t \cdot n \cdot m \cdot log(m))$. This method needs to be called once for every missing voter.

The thought behind this analysis is that some voters can have extremely similar preferences and this can be seen in data over some time. This even works if the alternatives that are voted for are changed regularly or for every data point. For example for two people, who most of the time agree on some topic, it can be assumed that they will still agree on that topic at some point soon in the future or any given requested data point.

5.1.2 Analyzing Similarity to Past Rankings of Same Voter

Similar to the previous described method only the last t data points with data from the selected voter before a requested profile are looked at. This method analyzes how many data points back it is the most likely to find a ranking from the missing voter that is as similar to the voters current preferences as possible. Often one data point back will have the best results, however the algorithm here will allow to generate statistics over $1, 2$ and up to d data points back.

For this the voters ranking of one data point will be compared to d data points in the past of it. This is repeated for $t - d$ data points that contain the requested voter. As comparison again some Kendall tau and intersection size metrics are used. These values are then stored in a list corresponding to how far in the past of this data point the comparison ranking was. And at the end the lists will be averaged in some kind to finalize the statistics of up to d possible replacement rankings.

This already makes it easy to see the computation time of $O((t - d) \cdot d)$ and this is multiplied with the run-time of the most demanding metric that is used. For this analysis to make sense the number of used data points t must be larger than d as otherwise the statistics for the different rankings would have few data points to average over. Like for all of the analysis methods, this one also needs to be repeated for every missing voter.

Example:
To make this method easier to understand Table 5.2 shows an example for the required part of an input if only one missing voter is requested. This example uses the following values: $d = 2$ and $t = 5$.

date	Germany	Austria	...
$6^{th}Sep.2019$	to-replace	...	...
$Sep.5^{th}2019$	$ranking_{g5}$	...	...
$Sep.4^{th}2019$	$ranking_{g4}$	...	...
$Sep.3^{rd}2019$	$ranking_{g3}$	...	...
$Sep.2^{nd}2019$	$ranking_{g2}$	...	...
$Sep.1^{st}2019$	$ranking_{g1}$	...	...

Table 5.2: Example for temporal input profiles for past ranking search algorithm.

In the first step $ranking_{g5}$ needs to be compared to the ranking of one day back, in this case $ranking_{g4}$. As example for the results tau-a$= 0.95$ and intersection size (i-size)$= 1.0$ are taken. Afterwards $ranking_{g5}$ is compared to the ranking of two days back $ranking_{g3}$ with the results tau-a$= 0.85$, i-size$= 0.9$. The same is then repeated for $ranking_{g4}$ with $ranking_{g3}$ and $ranking_{g2}$ and then again a last time with $ranking_{g3}$ and its respective previous two data points. As results lists such as these ones for one day back are produced: tau-a$= 0.95, 0.90, 0.85$, i-size$= 1.0, 1.0, 0.95$. And for two days back the following scores are taken: tau-a$= 0.85, 0.9, 0.9$ and i-size$= 0.9, 0.95, 0.9$.

With this the average scores can be calculated. When using the same weighted method for average as above the following scores result for $ranking_{g5}$ which is one day back for the requested profile: tau-a$= 0.904$ and i-size$= 0.985$. And $ranking_{g4}$, which is two days before the given profile, gets the average scores tau-a$= 0.882$ and i-size$= 0.917$ as expected values attached. Now these two rankings are ready to be used in Section 5.2.

For this approach to be useful it is required that the alternatives that are voted for do not change too much over time. If they would change at every data point no similar rankings can be found this way. However in many real life applications the alternatives stay at least similar.

For this method up to 15 data points that include the missing voter are considered for the experiments. As d it was decided to take up to 10 data points back into consideration, therefore the number of analyzed data points t can be taken as 25.

5.1.3 Analyzing Similarity to Past Rankings of Different Voters

This method is pretty similar to the previous one, with the main difference that instead of looking for similarities with rankings of the own past, similarities with past rankings of different voters are searched for. This could find connections where one voter most of the time votes the same as another voter a few data points before. For example one person could daily use a playlist that another person generated the day before. Therefore a time-span t together with a number of how many data points back should be considered d need to be decided on.

The algorithm starts with taking the ranking of the missing voter from the second newest input profile. This ranking is then compared to each ranking of the different voters from one data point in the past of that profile. These comparisons are then repeated for two data points back and so on until d profiles in the past. Then the next newest ranking of the missing voter is taken and the process is repeated until $t-d$ rankings of the requested voter were each compared to the rankings of the other voters from up to d data points back. Only $t-d$ repeats are done to assure that at least a similar amount of comparisons are done for d data points in the past like for 1 data point before the one to compare to. At the end the average expected Kendall tau and intersection size between the currently requested voter and every other voter over the last d data points is calculated and then available and ready to be used for the selection of a suitable ranking.

date	Germany	Austria	Italy
$Sep.6^{th}2019$	to-replace	$ranking_{a6}$	$ranking_{i6}$
$Sep.5^{th}2019$	$ranking_{g5}$	$ranking_{a5}$	$ranking_{i5}$
$Sep.4^{th}2019$	$ranking_{g4}$	$ranking_{a4}$	$ranking_{i4}$
$Sep.3^{rd}2019$	$ranking_{g3}$	$ranking_{a3}$	$ranking_{i3}$
$Sep.2^{nd}2019$	$ranking_{g2}$	$ranking_{a2}$	$ranking_{i2}$
$Sep.1^{st}2019$	$ranking_{g1}$	$ranking_{a1}$	$ranking_{i1}$

Table 5.3: Example of temporal input profiles for past different voter search algorithm.

Example:
Table 5.3 gives an example of input profiles for this method with the missing voter "Germany". Like in the example of the previous search method the value for t is 5 and the value for d is 2. As first step $ranking_{g5}$ needs to be compared to the rankings of one data point back from "Austria" and "Italy". Lets take $ranking_{a4}$ for the first comparison with the tau-a score of 0.6 and i-size of 0.7. Then comes the comparison to $ranking_{i4}$ which results in the tau-a score 0.7 and the intersection size percentage 0.9. After this the comparisons with two days back are done. As results $ranking_{a3}$ scores a tau-a score of 0.4 and an i-size of 0.45. And the comparison between $ranking_{g5}$ and $ranking_{i3}$ results in the tau-a score 0.6 and the i-size 0.8.

The same is then repeated for $ranking_{g4}$ with first $ranking_{a3}$ and $ranking_{i3}$ and second for two data points back with $ranking_{a2}$ and $ranking_{i2}$. Then finally the comparisons from $ranking_{g3}$ are done and the resulting lists are as following: For one day back with "Austria" the scores are tau-a$= 0.6, 0.55, 0.5$ and i-size$= 0.7, 0.6, 0.55$. The same for "Italy" with tau-a$= 0.7, 0.65, 0.65$ and i-size$= 0.9, 0.9, 0.85$. Additionally the scores for two data points back are the following for "Austria" with tau-a$= 0.4, 0.3, 0.45$ and i-size$= 0.45, 0.45, 0.55$. And for "Italy" the scores are in the following lists: tau-a$= 0.6, 0.6, 0.65$ and i-size$= 0.8, 0.75, 0.8$.

With this the weighted average scores are as followed: The $ranking_{a5}$ as the ranking from "Austria" for one day in the past gets a tau-a score of 0.554 and an i-size of 0.622. Then $ranking_{i5}$ is attached to the tau-a score 0.668 and i-size 0.885. For two data points back the ranking $ranking_{a4}$ has a tau-a value of 0.382 and i-size of 0.480. And the final resulting ranking is $ranking_{i4}$ with an average tau-a of 0.615 and average i-size of 0.783. These rankings will come to use in the methods of the following section.

As this is kind of a combination of the previous two methods it has the highest number of calculations with $(t - d) \cdot d \cdot n$ where n is the number of voters. This does not include the run-time for tau-a and intersection size, therefore this needs to be multiplied by the higher one of their run-times to reach the full run-time. In theory this traversal would also allow to include the other two methods. For this it would need to start with zero data points back for different voters and also do the similarity comparisons for the past rankings of the requested voter. However it was decided that they are handled separately as they all have different data sets on which they could work especially good. With using these three split versions it is easier to distinguish what method worked the best.

Here the same values for d and t are chosen for the experiments as in the previous method with $t = 25$ and $d = 10$. Like in all three methods the t value is not completely fixed, as some data points do not include the missing voter. However the values are correct when computing the run-time.

5.2 Selection of Suitable Rankings

After analyzing past data and generating candidate rankings with their expected Kendall tau score and intersection size it is needed to select a suitable prediction. For this either one of the candidate rankings is taken directly or some form of aggregation is performed on multiple rankings. It would also be possible to use reversed rankings, but this is not tested for this thesis and would mak more sense with complete linear ordered rankings. The methods from this section all produce alternative outputs that will be compared in the following chapter through some experiments. For real usage of these algorithms it would be necessary to choose one of them, because multiple alternative profiles would rarely make much sense.

5.2.1 Selection Criterion

At first a selection criterion is needed that sorts the possible rankings by their chance of usefulness. The simplest way is to just sort by best Kendall tau score and remove any ranking that falls below a certain threshold. However with the given data sets a useful threshold for Kendall tau type-a can be hard to reach, therefore intersection size is used as second criterion where if it meets a given threshold the threshold for the tau-a score is lowered.

After some trial and error it was decided on a first threshold of 0.6 for tau-a scores. Everything that reaches or surpasses this average score is considered as possible ranking.

For all rankings that do not fulfill this, there is still the possibility to be taken into account if their intersection size is on average at least 60% of the rankings of the required voter. Then the threshold for their tau-a score is lowered to 0.3 in the case of Section 5.3 for full aggregation and 0.2 in the case of Section 5.4 for top-k prediction.

Every possible ranking that is still left after this selection criterion can be used for the methods described after this. They will always be sorted by their Kendall tau type-a score from highest to lowest.

Example:
When continuing the examples from the previous section there are the rankings from "Germany" $ranking_{g5}$ and $ranking_{g4}$, the ones from "Italy" $ranking_{i6}$, $ranking_{i5}$ and $ranking_{i4}$ and finally the ballots from "Austria" with $ranking_{a6}$, $ranking_{a5}$ and $ranking_{a4}$. Only two of those rankings have a tau-a score below 0.6 and those are $ranking_{a5}$ and $ranking_{a4}$. From those two $ranking_{a5}$ is the only one with an i-size of at least 0.6 with 0.622. This means for this ranking the threshold for tau-a is lowered and its score of 0.554 meets the lowered threshold. With this a total of 7 rankings are left after this selection criterion. A sorted list of them by tau-a score can be found in Table 5.4. The last column "source" shows from where the ranking is.

ranking	tau-a	i-size	source
$ranking_{g5}$	0.904	0.985	past ranking
$ranking_{g4}$	0.882	0.917	past ranking
$ranking_{i6}$	0.846	0.932	different voter
$ranking_{i5}$	0.668	0.885	past different voter
$ranking_{i4}$	0.615	0.783	past different voter
$ranking_{a6}$	0.603	0.670	different voter
$ranking_{a5}$	0.554	0.622	past different voter

Table 5.4: Remaining candidate rankings after selection criterion.

5.2.2 Copy a Ranking

Now that candidate rankings are available they need to be used to generate an output. This is the easiest possibility to create outputs as simply one ranking per missing voter need to be picked according to their scores. For this thesis this method is split in multiple selections as every analysis source should show its own results.

Therefore one selection chooses the ranking from the search method dor different voters from Section 5.1.1 with the highest tau-a score attached to it, this method will be called **different voter replacement (DVR)**. The next method is called **past replacement (PR)** and it chooses the ranking that has the highest tau-a score from the search method for past rankings of the same voter from Section 5.1.2. The high average tau-a score should lead to a high probability to be a good replacement with high similarities to the real preferences of the voter. And the search method for past rankings of different voters also gets its own copy method with **past different voter replacement (PDVR)** where like in the other two methods also the one with the highest Kendall tau type-a score is chosen. As an output method for a combination of the search methods **highest score replacement (HSR)** is used. It considers all the possible rankings that made it through the selection filter. Then is takes the ranking with the highest overall tau-a score. This method requires that multiple search algorithms are used, which is not necessary for every data set in real applications, but in cases where this is done HSR is expected to produce the best results between these four methods. The experiments will use and compare all four of these methods.

Example:
When looking at the candidate rankings from the example in Table5.4 the following results will be produced: PR picks $ranking_{g5}$, DVR chooses $ranking_{i6}$, PDVR selects $ranking_{i5}$ and last but not least HSR takes the one with the highest chance of high similarity with $ranking_{g5}$. Each of the methods outputs at the end a profile for the 6^{st} of $Sep.$2019 with the rankings it got as input for "Italy" and "Austria" with $ranking_{i6}$ and $ranking_{a6}$ together with their respective chosen ranking as the vote from "Germany". In cases where more than one voter is missing for every missing voter a ranking is chosen according to their own analysis and at the end the output profile will include them all (as long as at least one replacement ranking was suitable per voter).

If it is assumed that the algorithms that search for past replacements from the same voter and from different voters use the same d value a maximum of $O(n \cdot (d+1))$ candidate rankings are possible, where n is the number of voters. The selection of the ranking with the highest score for each method is therefore possible in $O(n \cdot d)$ time.

5.2.3 Aggregate a Ranking from the Candidate Rankings

A computationally more demanding approach is to select up to b best rankings and use one of the aggregation methods to generate a ranking that uses multiple sources. This requires a sorted list by some selection criterion, like the one in Subsection 5.2.1, of the available candidate rankings. From this list the best b rankings are taken if available and combined as a profile to be used as input for the aggregation method that should optimally output a ranking that is more similar to the real preferences of the voter than any individual ranking from the input profile.

Borda seems like a good choice for the SWF to be used as it is fast to compute and has many desirable properties. Furthermore it can be easily altered to allow giving weights to each ranking according to their score. For this a simple weight multiplier has to be applied while adding the points for each ranking. This could produce further improved results.

Example:
For example the profile in Table 5.5 would result in the ranking $pink \succ blue \succ red$ when using the standard Borda version. But when using an altered Borda version with weights according to tau-a scores it can happen that the result changes. For example for $Voter_1$ the weight is 0.9, for $Voter_2$ the weight is 0.3 and for $Voter_3$ the weight is 0.3. Then the Borda score calculation would look like this: $borda_{weight}(blue, weights) = 3 \cdot 0.9 + 2 \cdot 0.3 + 1 \cdot 0.3 = 3.6$, $borda_{weight}(pink, weights) = 2 \cdot 0.9 + 3 \cdot 0.3 + 3 \cdot 0.3 = 3.6$ and $borda_{weight}(red, weights) = 1 \cdot 0.9 + 1 \cdot 0.3 + 2 \cdot 0.3 = 1.8$. With these scores $blue$ and $pink$ are tied in first place with $blue \sim pink \succ red$.

Rank	$Voter_1$	$Voter_2$	$Voter_3$
1.	*blue*	*pink*	*pink*
2.	*pink*	*blue*	*red*
3.	*red*	*red*	*blue*

Table 5.5: Example profile where weights could lead to different result with Borda.

To make them less similar it was decided to set no limit for the number of rankings that are aggregated b in the weighted version and a limit of 5 in the non weighted version. As the analyzing methods have an upper limit of finding $n \cdot d$ candidate rankings this is also the upper limit for the b of the weighted version. The methods will be called **combined replacement (CR)** and **weighted combined replacement (WCR)**.

The resulting rankings would most likely have more alternatives ranked than the normal length of voted rankings, as it is likely that at least some of the input rankings contain different voters and the output ranking of the SWF would contain all of the combined alternatives. Therefore a decision was needed if the result should be cut to the length of the other rankings. This could have either good or bad effects. It either cuts out the less important candidates if the given rankings mostly agree in the same way with the requested ranking. Or it could also cancel out the right alternatives to predict. Without trimming the ranking its increased length compared to the other voters rankings could also lead to unexpected results when aggregating the ranking together with the other shorter rankings of the data point. Because of this and mostly positive effects in limited testing it was decided to trim the result ranking to an appropriate length. However as ties are possible in these rankings the length can still exceed the length of the other rankings as there would be no fair way to remove only some alternatives tied in last rank.

Example:
When going back to the example from Table 5.4 there are seven candidate rankings. WCR would aggregate them all together with their respective tau-a scores as weights. CR would only take 5 of them with the highest tau-a scores. This means $ranking_{a6}$ and $ranking_{a5}$ would not be taken into account for the Borda method. Like in the case of the four copy methods from the previous subsection each of the methods builds

its own profile as a result where their aggregated and then shortened rankings are taken as the votes of "Germany".

The run-time of both variants varies only in their maximum profile size b. First the sorting of the list has a worst case complexity of around $O(n \cdot log(n))$ to $O(n^2)$ depending on the sort algorithm. In this case n represents the number of candidate rankings. The selection of a maximum of b rankings together with the generating of the profile takes $O(b)$.

Then the Borda method will probably take the defining computation time with $O(b \cdot m)$, where m is again the number of alternatives in the profile. In some cases the sorting of the rankings to only select the b best suitable ones could also be the defining factor for the overall run-time. The trimming after using the SWF only takes $O(m)$ and can therefore be nearly ignored. Combined this gives a complexity of $O(max(n \cdot log(n), b \cdot m))$

For the prediction of the top-k of a ranking another alteration could also prove useful. As only the first k alternatives are relevant, the rankings can be trimmed to the length of k before aggregating them. This decreased the time to compute Borda as long as k is smaller than the length of the rankings. However the experiments will still use CR and WCR together with these adjusted methods **trimmed combined replacement (TCR)** and **trimmed weighted combined replacement (TWCR)**.

For these two methods the example would happen nearly the same as the other two methods that use aggregation, except that the chosen candidate rankings would need to be trimmed before aggregating them. In the end they would both also produce their output profiles that contain the requested voters which in the case of the example is "Germany".

5.3 Implicit Delegation of full Rankings

All the described methods can now be combined and used to fill in missing rankings or in the case of the experiments to fill in the removed rankings. In the use case of taking the opinion of missing voters into account, a full ranking needs to be predicted. Therefore the comparison methods need to find similarities between the whole rankings. This can already be achieved by the two methods that were used as placeholders up to now. The Kendall tau type-a that was described in Section 2.3 together with a full intersection size. As mentioned in that section a value from 0 to m would not be optimal, therefore the normalized intersection size with real numbers between 0 and 1 as output is used, where 1 means the second ranking contains all the alternatives of the first one and 0 means it contains none of them.

To give a better understanding of the whole algorithm here is a summary of the composition of the methods. The method starts with accepting the inputs, that include a list of profiles, where the newest one of them needs to be filled in. To show which voters need to be replaced a list of the missing voters is also a required input. Then this input is given to each of the three search methods that analyze the data. These algorithms are called multiple times, once for each voter to replace. Each of them returns multiple candidate rankings for every missing voter together with their expected Kendall tau-a score and intersection size according to the analysis.

Afterwards those rankings are filtered with the help of the selection criterion, which should assure that only rankings with a good chance to be similar to the real ranking are left. This is also done for each of the voters to replace. Then it is time to produce the output profile. For this multiple different output methods are used. This includes the copy methods PR, DVR and PDVR that just pick the ranking with the highest tau-a score of their respective search method, as long as at least one of them made it trough the selection criterion filter. HSR chooses the ranking with the highest score between all the search methods. And finally CR and WCR aggregate up to five or respectively all rankings, that reached the required scores, to each one final replacement ranking that is then cut to the length of the longest other ranking in the profile to complete. These final steps are of course repeated for every one of the missing voters and at the end 6 profiles are returned together with the information which method produced which profile. Figure 5.1 shows a flowchart that nearly represents this program flow with the exception that it ignores the repetitions for the multiple voters and has two additional output methods with TCR and TWCR.

From all these parts the most demanding computation is from the past different voter analysis. As all the parts of the full algorithm are serial the complete run-time only needs to additionally factor in the repetition for every missing voter. This leads to an overall run-time of $O(n_m \cdot (t-d) \cdot d \cdot n \cdot m \cdot log(m))$ where n_m is the number of missing voters, n the number of voters overall, t the number of data points that are considered, d the number of past data points each ranking is compared to and m the number of alternatives in the rankings that are compared with each other by tau-a.

It would be possible to parallelise the algorithm. The easiest part would be to execute the search methods in parallel to each other and for every missing voter. The same could be done for the output methods. And if run-time is crucial it is also possible to make the search algorithms itself parallel, as the different comparisons are independent of each other and only at the end of them some synchronization would be needed to calculate the average values.

5.4 Implicit Delegation of Top-k Alternatives of a Ranking

For this goal only the highest ranked alternatives of a given voter are relevant. Therefore it suffices to use shortened versions of Kendall tau and the normalized intersection size that only look at the top-k alternatives of each ranking. This leads to much faster computation as only a smaller portion of the data is analyzed. Here again intersection size is represented as a fraction of the alternatives from the first ranking that are also

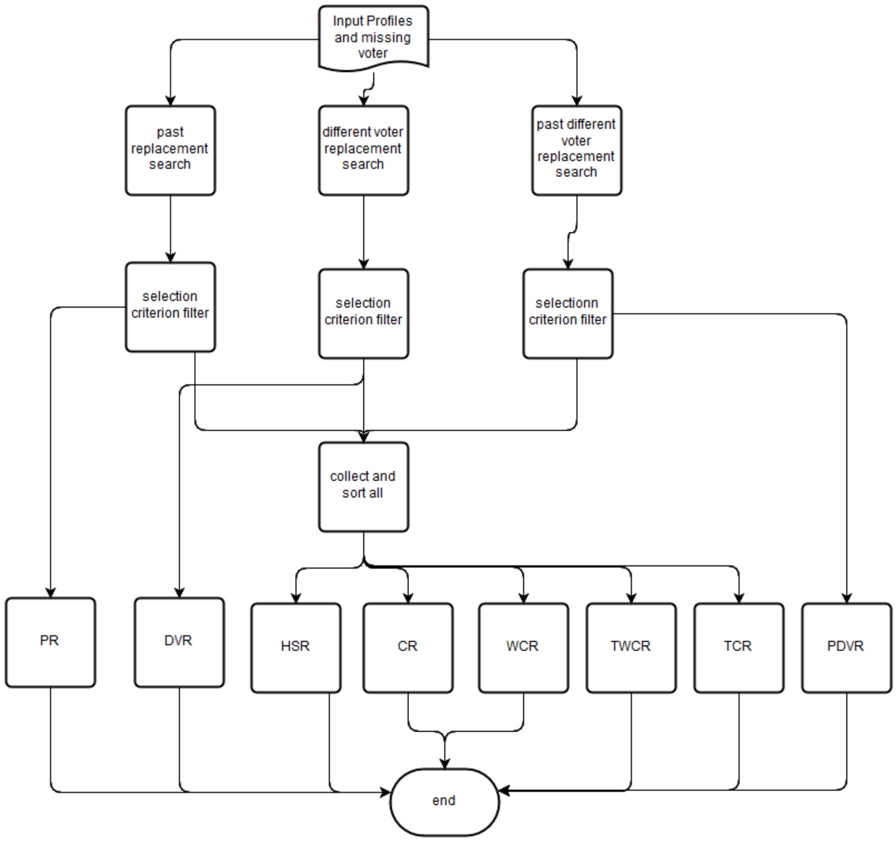


Figure 5.1: Flowchart to show the composition of the implicit delegation algorithm for top-k prediction.

found in the second ranking, but this time limited to the first k ranked alternatives of each ranking.

The program flow of this goal is very similar to the one for the full prediction. One difference is at least in this thesis that each execution only requests one voters ranking to be replaced and it produces two extra output profiles. Otherwise it is nearly the same, just with different Kendall tau type-a and intersection size versions. This means the input is also a list of profiles where the newest one misses one voter that needs to be replaced and that voters name or id also needs to be provided. Then the three search methods do their separate analysis and the resulting rankings are filtered by the selection criterion. And finally the four copy methods and the four aggregation methods select and prepare the eight output profiles. As a reference Figure 5.1 shows a general flowchart of the whole process.

This method has a lower run-time compared to the one in Section 5.3 as it only replaces a single voter. It is also much faster as the used versions of Kendall tau type-a and intersection size ignore a large portion of the alternatives in the rankings. This leads to a run-time of $O(k \cdot log(k))$ instead of $O(m \cdot log(m))$, with m the number of alternatives, for tau-a and $O(k)$ instead of $O(m)$ for i-size. With this the overall run-time can be written as $O((t-d) \cdot d \cdot n \cdot k \cdot log(k))$.

In cases where only the first ranked alternative needs to be predicted the given algorithms could be adjusted to only use intersection size, as Kendall tau makes no sense on one alternative. This case will not be covered in the experiments.

CHAPTER 6

Experiments

Now that the methods for implicit delegation are defined they need to be tested. For this some experiments where developed. These experiments select a subset of the data points of each data set randomly. How many are selected is decided individually for each data set and type of experiment. In general for each data set a chance is defined that defines how likely each experiment is executed. For example a top-k prediction experiment could be executed with a chance of one in three depending on a random number. The random selection is done to avoid any possible bias and is necessary as it would require a much too high run-time to execute all experiments on every data point.

The first type of experiment concentrates on how good the algorithms work to predict the top-k alternatives of single voters. The second type has as goal to predict multiple missing rankings from a single data point. In the following sections they will be described in more detail followed by the results of the experiments for each data set.

6.1 Top-k Prediction Experiments

These experiments use the mechanisms described in the previous chapter for the prediction of the top-k alternatives of rankings from Section 5.4. For this the implicit delegation algorithms are executed for every present voter in every chosen data point. As input all data points up to and including the selected profile are used. From the chosen profile the voter to predict is removed and its name is given as input to indicate it should be predicted. The voter's rankings of the outputs are then compared to the real ranking of the given voter. For this the Kendall tau type-a and intersection size algorithms are used that each only compare the top-k alternatives.

There will also be cases were some methods do not manage to find a replacement and no comparison can be done, those are unsuccessful attempts. This happens because only rankings that reached average scores according to the selection criterion in Section 5.2.1

were considered as replacement, if a search method found no such ranking no replacement can be found. For each successful implicit delegation attempt the intersection size and the tau-a score are saved in lists for the corresponding voter and method. In addition it is also saved how often each method fails to produce suitable results.

Then at the end the average scores for each voter and method are calculated and returned as output, together with a number of how often a result was available vs how often it was attempted to predict the ranking. This is important as not every voter is present in every data point and the fewer data points a voter is present in, the harder the implicit delegation gets. Few data points that contain a given voter can also lead to lower statistical meaning.

6.2 Full Prediction Experiments

Here the implicit delegation algorithm for the prediction of full rankings from Section 5.3 is used. These experiments expect from the algorithm to predict the rankings of around a third of the voters for each data point. For every selected data point the implicit delegation method will be executed multiple times. The input of the calls consists of all profiles up to and including the chosen data point. From the chosen data point a set of voters is removed and their names are used as input list for the missing voters. To not produce any unintended bias the decision which voters are removed is done through splitting the list of available voters from the profile into three brackets, each containing around a third of the voters. For each of these brackets the experiment is executed with those voters as the missing ones. In case of too small data sets it was decided that repeating this selections of three random brackets would be helpful to produce a more averaged out result. Specifically every data set except the four Spotify data sets repeats this selection and prediction three times per selected data point, which results in 9 executions of the algorithm.

The output profiles of the implicit delegation method that include the predicted rankings need then to be aggregated with the different aggregation methods introduced in Chapter 3. For example the resulting profile of the implicit delegation PR is aggregated with Borda, the result is called *PR result*. Then the original profile of the data point is also aggregated with Borda, which will be called *full result* and to get a base value the original profile without the voters of the current bracket is also aggregated and called *incomplete result*. With these three results two Kendall tau-a scores are computed, first between *full result* and *incomplete result*, then between *full result* and *PR result*. If the score from the comparison with *PR result* is higher than the one with *incomplete result* it shows that the implicit delegation worked and achieved its goal to make the result ranking more similar to if the voter had cast a vote.

This is done for every suitable aggregation method and every successfully produced output from the implicit delegation method. An implicit delegation methods output counts as successful if at least one voters ranking could be added to the profile. The experiments also save how high of a percentage of the missing voters each method was able to give a prediction for. As Kemeny has a far to high run-time it was decided to not use it at all. Schulze is used for some experiments, but not for the Spotify data sets, as over 1000 or even over 3000 alternatives at many data points cause a much too long run-time.

The results will show the average scores of each method together with the average base value through the comparison with the *incomplete result*. This base score will be called default. Of course these average scores will be made for each preference aggregation method separately. To give a better context, of how much each implicit delegation method could give a prediction for, an average gain is also provided. This gain value is between 0 and 1 and represents how much of a percentage of the missing voters were predicted.

In addition a second representation of these scores will be given that excludes implicit delegation attempts that failed to predict at least one ranking. This version should show better what the real difference is in case of a successful implicit delegation attempt, as all the default similarities from not successful attempts are not included in the calculation of the average.

6.3 Results

With this the experiments should be clear and the resulting data can be looked at and analyzed. To avoid confusion the columns of the top-k prediction results have the following meanings: "tau-a" meins Kendall tau type-a, "i-size" is the intersection size and "sub" means substitute and shows how often a voter could be predicted compared to how often it was attempted.

For the tables that show the Kendall tau type-a scores for the full prediction experiments the column *default* stands for the base value i.e. the comparison of the full ranking with the ranking where none of the missing voters are present. Therefore this column will always have a gain of 0.0 and will be the same in both tables representing the full prediction results of each data set.

The results will be compared to the correlation results from Chapter 4. As always mentioning this chapter would be too much it can be assumed that mentions of the "analysis of the data set", "the past analysis", "different voter analysis" or similar phrases refer to the correlation tables and figures from that chapter.

6.3.1 Eurovision Song Contest

The first data set to look at is from the Eurovision Song Contest. As the rankings here already only have a length of 10 it was decided to use the top-k experiments to predict the top-3 alternatives. Both types of experiments where conducted without random selection and instead all of the newest 35 data points where selected. This was done as the data set would otherwise be too small. Like for all of the data sets the oldest 10 data points where not considered for the experiments, as there would not be enough past data available to analyze.

The analysis of the data set in Section 4.1 already showed that past rankings of voters have little to no similarity and therefore it is expected that PR will rarely work and will not produce very good results even if it finds something. This is also reflected in Table 6.1, where it is evident that this method rarely found any replacement. The row avg at the bottom which represents a statistical mean of the data of all voters shows how many percent of the attempted implicit delegations found replacements that fulfilled the minimum selection criterion. According to this PR was only able to find a ranking in 2% of all attempts. The average tau-a score is similarly bad to the other methods. And the average intersection size only seems surprisingly high with 40%, however this only means that at least one of the three alternatives was right.

With the search method DVR much more, but still only 9% could be replaced. In this small subset the tau-a score is the highest among all the methods, which is still low but expected of the data set as no particular high similarities between the voters could be seen. The intersection size indicates that on average 2 of the 3 alternatives where predicted, which is also the best between all the methods.

For PDVR it could also be expected to receive these low scores, as for the Eurovision Song Contest past votes have no real connection to new votes. However this method was able to find the highest number of replacements between the three search approaches with 11%, this can probably be attributed to the high number of rankings that are considered.

Then as final method for this table HSR which always uses the statistically best replacement has mostly average results. This is the case because there are many cases where only one possible replacement is available, which often is not optimal. This method and the remaining ones are all capable of replacing the same rankings, as they all only need at least one possible ranking from the search methods. Therefore they were all able to to replace 18% of the attempted rankings.

	PR			DVR			PDVR			HSR		
voter	tau-a	i-size	sub	tau-a	i-size	sub	tau-a	i-size	sub	tau-a	i-size	sub
Albania	-0.11	0.37	3/16	-0.21	0.7	1/16	-	-	0/16	-0.13	0.45	4/16
Andorra	-	-	0/6	0.12	0.8	1/6	-0.19	0.33	3/6	-0.17	0.43	3/6
Armenia	-0.01	0.5	2/13	0.19	0.6	1/13	-0.31	0.2	1/13	0.06	0.53	3/13
Australia	-	-	0/5	-0.31	0.5	1/5	-0.44	0.2	1/5	-0.31	0.5	1/5
Austria	-	-	0/29	0.05	0.62	5/29	-0.35	0.38	5/29	-0.12	0.54	8/29
Azerbaijan	-0.06	0.53	4/12	-	-	0/12	-0.12	0.37	3/12	-0.09	0.46	7/12
Belarus	-0.26	0.27	6/16	0.12	0.62	4/16	-0.16	0.38	5/16	-0.14	0.39	9/16
Belgium	0.03	0.5	1/32	-0.31	0.5	1/32	-0.29	0.4	4/32	-0.23	0.42	5/32
Bosnia&Herzeg	-	-	0/19	-0.06	0.8	1/19	-0.24	0.4	2/19	-0.18	0.53	3/19
Bulgaria	-	-	0/12	-0.18	0.8	1/12	-0.21	0.37	3/12	-0.21	0.37	3/12
Croatia	-	-	0/25	0.2	0.67	6/25	-0.28	0.3	1/25	0.12	0.62	6/25
Cyprus	-	-	0/32	0.06	0.8	1/32	-0.07	0.5	2/32	-0.02	0.6	3/32
Czech Republic	-0.3	0.4	1/8	0.21	0.67	3/8	-0.14	0.4	4/8	0.08	0.6	4/8
Denmark	-0.03	0.6	1/31	-0.06	0.8	1/31	-0.46	0.3	2/31	-0.32	0.47	3/31
Estonia	-	-	0/25	-0.16	0.5	5/25	-0.3	0.45	2/25	-0.2	0.49	7/25
F.Y.R. Macedon	-0.21	0.6	1/18	-0.35	0.3	1/18	-0.28	0.3	1/18	-0.28	0.45	2/18
Finland	-0.62	0.3	1/30	-0.23	0.4	1/30	-0.31	0.43	3/30	-0.41	0.4	4/30
France	-	-	0/35	-0.2	0.5	2/35	-0.25	0.35	4/35	-0.25	0.4	4/35
Georgia	-0.28	0.3	1/12	0.15	0.53	3/12	-0.05	0.5	2/12	-0.07	0.4	4/12
Germany	-	-	0/34	-0.08	0.55	4/34	-0.24	0.45	4/34	-0.16	0.5	8/34

Greece	-	-	0/32	-	-	0/32	-0.34	0.2	2/32	-0.34	0.2	2/32
Hungary	-	-	0/17	-0.24	0.57	3/17	-0.32	0.4	1/17	-0.24	0.57	3/17
Iceland	-	-	0/32	-0.15	0.6	2/32	-0.42	0.37	3/32	-0.3	0.42	4/32
Ireland	-	-	0/34	-	-	0/34	-0.3	0.3	3/34	-0.3	0.3	3/34
Israel	-	-	0/32	-0.35	0.3	1/32	-0.31	0.4	3/32	-0.32	0.38	4/32
Italy	-	-	0/18	-0.26	0.43	3/18	-0.32	0.45	2/18	-0.28	0.45	4/18
Latvia	-0.12	0.4	2/20	0.05	0.66	5/20	-0.32	0.3	1/20	-0.06	0.54	7/20
Lithuania	-	-	0/20	-0.15	0.6	3/20	-0.24	0.35	2/20	-0.17	0.53	4/20
Luxembourg	-	-	0/9	-	-	0/9	-0.46	0.3	1/9	-0.46	0.3	1/9
Malta	-	-	0/29	-	-	0/29	-	-	0/29	-	-	0/29
Moldova	0.04	0.45	2/15	-	-	0/15	-0.14	0.5	1/15	-0.02	0.47	3/15
Monaco	-	-	0/3	-	-	0/3	-0.41	0.3	1/3	-0.41	0.3	1/3
Montenegro	-0.33	0.2	1/11	0.33	0.78	4/11	-0.07	0.5	1/11	0.19	0.68	4/11
North Macedoni	-	-	0/1	-	-	0/1	-	-	0/1	-	-	0/1
Norway	-	-	0/34	-0.05	0.6	1/34	-0.42	0.2	1/34	-0.24	0.4	2/34
Poland	-	-	0/22	-0.11	0.55	4/22	-0.03	0.5	1/22	-0.08	0.5	4/22
Portugal	-	-	0/31	-0.04	0.75	2/31	-0.27	0.43	3/31	-0.18	0.56	5/31
Romania	-	-	0/20	-0.19	0.55	2/20	-0.19	0.3	1/20	-0.19	0.55	2/20
Russia	-	-	0/22	0.04	0.65	2/22	-0.16	0.38	4/22	-0.16	0.4	4/22
San Marino	-	-	0/10	0.06	0.8	1/10	-0.23	0.4	2/10	-0.15	0.55	2/10
Serbia	-	-	0/12	-	-	0/12	-0.45	0.1	1/12	-0.45	0.1	1/12
Serbia&Monten	-	-	0/3	-	-	0/3	-	-	0/3	-	-	0/3
Slovakia	-	-	0/7	0.05	0.6	1/7	-	-	0/7	0.05	0.6	1/7
Slovenia	-	-	0/25	0.11	0.72	6/25	-0.32	0.41	7/25	-0.08	0.59	10/25
Spain	-	-	0/35	-	-	0/35	-0.22	0.37	3/35	-0.22	0.37	3/35
Sweden	-	-	0/35	-0.1	0.5	1/35	-0.24	0.4	2/35	-0.2	0.43	3/35
Switzerland	-	-	0/31	-0.01	0.59	7/31	-0.36	0.45	4/31	-0.1	0.53	9/31
The Netherland	-	-	0/31	-	-	0/31	-0.34	0.3	6/31	-0.34	0.3	6/31
Turkey	-	-	0/27	-	-	0/27	-0.3	0.5	3/27	-0.3	0.5	3/27
Ukraine	-	-	0/15	0.31	0.9	1/15	-	-	0/15	0.31	0.9	1/15
United Kingdom	-	-	0/35	-0.33	0.42	4/35	-0.31	0.38	4/35	-0.29	0.46	7/35
Yugoslavia	-	-	0/7	-	-	0/7	-	-	0/7	-	-	0/7
avg	-0.16	0.4	0.02	-0.02	0.61	0.09	-0.27	0.38	0.11	-0.16	0.48	0.18

Table 6.1: Results with directly copied rankings as replacement on the Song Contest data set for top-3 prediction.

The implicit delegation methods from Table 6.2 do not show much difference between the weighted and non weighted Borda aggregation. This could be because the searches did not provide many possible rankings to aggregate and in cases with only one possibility the weight has no impact at all. Furthermore all methods have relatively low tau-a scores, which means their weights where probably very similar.

But the trimmed and not trimmed versions show a higher difference. Here it seems like the trimming to the top-3 before aggregating lowered the average intersection size. This is most likely the case because many of the possible rankings only share one alternative in their top-3 with the desired ranking and this one overlap could be different for many of the possible rankings.

If for example the desired ranking contains the alternatives a, b, c and there are three rankings that are aggregated that all have lets say these alternatives in their the top-5 but each one has only one distinct desired alternative in their top-3, then the trimming can lead to the exclusion of all three right alternatives after Borda. The tau-a score is a bit higher, but this is probably only because rankings with lower intersection size get scores closer to 0 as there are less alternative pairs on which they can disagree on the order.

For the full prediction experiments overall only a very small percentage of the missing

voter	WCR			TWCR			TCR			CR		
	tau-a	i-size	sub	tau-a	i-size	sub	tau-a	i-size	sub	tau-a	i-size	sub
Albania	-0.16	0.45	4/16	-0.01	0.17	4/16	-0.01	0.17	4/16	-0.16	0.45	4/16
Andorra	-0.16	0.4	3/6	-0.13	0.17	3/6	-0.12	0.17	3/6	-0.16	0.37	3/6
Armenia	-0.03	0.5	3/13	-0.03	0.23	3/13	-0.04	0.23	3/13	-0.05	0.5	3/13
Australia	-0.39	0.3	1/5	-0.46	0.5	1/5	-0.19	0.6	1/5	-0.34	0.6	1/5
Austria	-0.1	0.5	8/29	-0.06	0.23	8/29	-0.06	0.23	8/29	-0.1	0.5	8/29
Azerbaijan	-0.1	0.46	7/12	-0.07	0.19	7/12	-0.07	0.19	7/12	-0.1	0.46	7/12
Belarus	-0.08	0.52	9/16	-0.11	0.22	9/16	-0.11	0.22	9/16	-0.08	0.52	9/16
Belgium	-0.25	0.44	5/32	-0.21	0.16	5/32	-0.21	0.16	5/32	-0.25	0.44	5/32
Bosnia&Herzeg	-0.17	0.57	3/19	-0.23	0.2	3/19	-0.22	0.2	3/19	-0.15	0.6	3/19
Bulgaria	-0.16	0.4	3/12	-0.09	0.23	3/12	-0.1	0.23	3/12	-0.16	0.43	3/12
Croatia	0.16	0.67	6/25	0.06	0.28	6/25	0.07	0.28	6/25	0.17	0.65	6/25
Cyprus	-0.02	0.6	3/32	0.13	0.27	3/32	0.13	0.27	3/32	-0.02	0.6	3/32
Czech Republic	0.08	0.57	4/8	-0.07	0.28	4/8	-0.05	0.28	4/8	0.06	0.57	4/8
Denmark	-0.25	0.53	3/31	-0.2	0.3	3/31	-0.17	0.3	3/31	-0.26	0.57	3/31
Estonia	-0.24	0.41	7/25	-0.21	0.24	7/25	-0.19	0.24	7/25	-0.22	0.44	7/25
F.Y.R. Macedon	-0.31	0.45	2/18	-0.28	0.2	2/18	-0.28	0.2	2/18	-0.31	0.55	2/18
Finland	-0.36	0.4	4/30	-0.21	0.12	4/30	-0.2	0.12	4/30	-0.34	0.42	4/30
France	-0.22	0.35	4/35	-0.26	0.2	4/35	-0.26	0.2	4/35	-0.21	0.35	4/35
Georgia	0.07	0.53	4/12	0.04	0.28	4/12	0.07	0.28	4/12	0.14	0.53	4/12
Germany	-0.16	0.5	8/34	-0.1	0.18	8/34	-0.1	0.18	8/34	-0.16	0.5	8/34
Greece	-0.34	0.2	2/32	-0.31	0.05	2/32	-0.31	0.05	2/32	-0.34	0.2	2/32
Hungary	-0.15	0.6	3/17	-0.08	0.3	3/17	-0.1	0.27	3/17	-0.13	0.6	3/17
Iceland	-0.25	0.4	4/32	-0.18	0.25	4/32	-0.17	0.25	4/32	-0.24	0.42	4/32
Ireland	-0.3	0.3	3/34	-0.11	0.13	3/34	-0.11	0.13	3/34	-0.3	0.3	3/34
Israel	-0.34	0.4	4/32	-0.37	0.12	4/32	-0.37	0.12	4/32	-0.34	0.4	4/32
Italy	-0.24	0.5	4/18	-0.28	0.15	4/18	-0.28	0.15	4/18	-0.24	0.5	4/18
Latvia	-0.04	0.56	8/20	-0.02	0.28	8/20	-0.03	0.28	8/20	-0.03	0.56	8/20
Lithuania	-0.14	0.53	4/20	-0.04	0.35	4/20	-0.05	0.35	4/20	-0.14	0.53	4/20
Luxembourg	-0.23	0.4	1/9	-0.24	0.3	1/9	-0.24	0.3	1/9	-0.22	0.4	1/9
Malta	-	-	0/29	-	-	0/29	-	-	0/29	-	-	0/29
Moldova	-0.08	0.43	3/15	0.0	0.27	3/15	0.0	0.27	3/15	-0.1	0.43	3/15
Monaco	-0.41	0.3	1/3	-0.32	0.1	1/3	-0.32	0.1	1/3	-0.41	0.3	1/3
Montenegro	0.25	0.78	4/11	0.14	0.4	4/11	0.15	0.4	4/11	0.31	0.78	4/11
North Macedoni	-	-	0/1	-	-	0/1	-	-	0/1	-	-	0/1
Norway	-0.24	0.4	2/34	-0.22	0.15	2/34	-0.22	0.15	2/34	-0.24	0.4	2/34
Poland	-0.09	0.53	4/22	-0.03	0.3	4/22	-0.02	0.3	4/22	-0.06	0.55	4/22
Portugal	-0.17	0.58	5/31	-0.15	0.2	5/31	-0.14	0.2	5/31	-0.15	0.58	5/31
Romania	-0.22	0.45	2/20	-0.1	0.2	2/20	-0.1	0.2	2/20	-0.21	0.5	2/20
Russia	-0.14	0.47	4/22	-0.13	0.35	4/22	-0.13	0.35	4/22	-0.16	0.47	4/22
San Marino	-0.18	0.5	2/10	-0.1	0.2	2/10	-0.1	0.2	2/10	-0.17	0.5	2/10
Serbia	-0.45	0.1	1/12	-0.32	0.1	1/12	-0.32	0.1	1/12	-0.45	0.1	1/12
Serbia&Monten	-	-	0/3	-	-	0/3	-	-	0/3	-	-	0/3
Slovakia	-0.02	0.5	1/7	0.05	0.3	1/7	0.05	0.3	1/7	-0.02	0.5	1/7
Slovenia	-0.03	0.59	10/25	-0.04	0.31	10/25	-0.03	0.31	10/25	-0.03	0.6	10/25
Spain	-0.22	0.37	3/35	-0.2	0.17	3/35	-0.21	0.17	3/35	-0.26	0.37	3/35
Sweden	-0.18	0.47	3/35	-0.38	0.3	3/35	-0.39	0.3	3/35	-0.2	0.5	3/35
Switzerland	-0.12	0.53	9/31	-0.05	0.21	9/31	-0.05	0.21	9/31	-0.12	0.53	9/31
The Netherland	-0.34	0.3	6/31	-0.27	0.08	6/31	-0.27	0.08	6/31	-0.34	0.3	6/31
Turkey	-0.3	0.5	3/27	-0.18	0.2	3/27	-0.18	0.2	3/27	-0.3	0.5	3/27
Ukraine	0.14	0.8	2/15	-0.04	0.25	2/15	-0.05	0.25	2/15	0.13	0.8	2/15
United Kingdom	-0.32	0.43	7/35	-0.25	0.19	7/35	-0.25	0.19	7/35	-0.31	0.41	7/35
Yugoslavia	-	-	0/7	-	-	0/7	-	-	0/7	-	-	0/7
avg	-0.15	0.49	0.18	-0.12	0.23	0.18	-0.12	0.23	0.18	-0.14	0.49	0.18

Table 6.2: Results with aggregated rankings as replacement on the Song Contest data set for top-3 prediction.

voters could be replaced. Therefore the results shown in Table 6.3 give hardly any information. However if only the experiments where at least one of the rankings could be substituted are considered like in Table 6.4 more information can be gained. First of all it can be seen that all implicit delegation methods had only low and mostly negative effects on the results independent of the used aggregation method. This could have been expected, as not many similarities where found in the analysis of the data set and therefore only a few of the missing rankings can be replaced. Additionally most rankings probably only reached similarities that barely cross the required thresholds of the algorithms.

	default	PR	DVR	PDVR	HSR	CR	WCR
Gain	0.0	0.0	0.06	0.01	0.07	0.07	0.07
Borda	0.87	0.87	0.86	0.87	0.86	0.86	0.86
Nanson	0.87	0.87	0.86	0.86	0.86	0.86	0.86
Copeland	0.86	0.86	0.86	0.86	0.86	0.86	0.86
Maximin	0.79	0.79	0.79	0.79	0.78	0.78	0.78
Schulze	0.86	0.86	0.86	0.86	0.85	0.85	0.85

Table 6.3: Average Kendall tau-a scores after implicit delegation of multiple rankings for the Eurovision Song Contest.

	default	PR	DVR	PDVR	HSR	CR	CR
Gain	0.0	0.11	0.15	0.15	0.16	0.16	0.16
Borda	0.87	0.83	0.86	0.82	0.86	0.85	0.86
Nanson	0.87	0.83	0.86	0.82	0.85	0.85	0.85
Copeland	0.86	0.85	0.85	0.85	0.85	0.85	0.85
Maximin	0.79	0.76	0.78	0.75	0.77	0.77	0.77
Schulze	0.86	0.82	0.85	0.85	0.84	0.85	0.85

Table 6.4: Average Kendall tau-a scores of successful implicit delegation attempts of multiple rankings for the Eurovision Song Contest.

6.3.2 Spotify Daily Data Set

This is the largest data set and it also has one of the highest numbers of unique alternatives per data point. Because of this high number of alternatives per data point Schulze was omitted from all the Spotify experiments. With the high number of data points and high computation time per data point only 56 where randomly chosen to perform the full prediction experiment. Every selected data point resulted in three experiments, each for a different subset of missing voters. As the top-10 prediction is computationally much easier those experiments where performed on 110 random data points.

When looking at Table 6.5 it can be immediately seen that a much higher percentage of the implicit delegation executions lead to results, when compared to the Eurovision data set. The lowest success rate for the top-10 prediction was with PDVR and is still at 67%, while PR even reached 99% substitution rate. With the combination of the three search methods the other methods where always able to find some replacement ranking.

Here again the analysis of the data from Chapter 4.2 already gave a good estimate of what to expect from the search implicit delegation methods. As best and nearly perfect method PR shines the most. However the scores of DVR and PDVR are also pretty high and were able to predict on average nearly 8 of the 10 best ranked alternatives with mostly agreeing order. And like expected HSR is dominated by and therefore nearly identical to PR.

If looking at the voters that could be replaced the least with DVR it should not be surprising as the overall analysis of the data set already showed this for some of them. For example "br" could not be replaced a single time and when looking at its correlation row in Fig 4.2 it can be seen that is has on average a negative or extremely close to 0 tau-b score with nearly all other voters. This is also the case for "fi", "fr" and "tr".

	PR			DVR			PDVR			HSR		
voter	tau-a	i-size	sub	tau-a	i-size	sub	tau-a	i-size	sub	tau-a	i-size	sub
ar	0.9	0.96	109/110	0.59	0.84	110/110	0.58	0.83	109/110	0.9	0.95	110/110
at	0.78	0.9	109/110	0.5	0.79	109/110	0.43	0.74	104/110	0.78	0.9	110/110
au	0.86	0.95	109/110	0.56	0.84	108/110	0.52	0.83	107/110	0.86	0.95	110/110
be	0.76	0.91	109/110	0.4	0.72	72/110	0.41	0.7	66/110	0.76	0.9	110/110
bg	0.66	0.83	35/38	0.5	0.75	36/38	0.51	0.76	32/38	0.6	0.81	38/38
bo	0.9	0.96	109/110	0.65	0.85	110/110	0.66	0.85	109/110	0.9	0.96	110/110
br	0.77	0.91	109/110	-	-	0/110	-	-	0/110	0.77	0.91	109/110
ca	0.8	0.93	109/110	0.48	0.75	105/110	0.42	0.72	103/110	0.8	0.93	110/110
ch	0.72	0.89	109/110	0.42	0.73	75/110	0.42	0.72	65/110	0.72	0.89	110/110
cl	0.91	0.96	109/110	0.52	0.78	90/110	0.51	0.76	91/110	0.9	0.96	110/110
co	0.92	0.96	109/110	0.47	0.74	95/110	0.46	0.74	94/110	0.91	0.96	110/110
cr	0.89	0.96	109/110	0.57	0.8	104/110	0.55	0.78	108/110	0.89	0.96	110/110
cz	0.77	0.92	109/110	0.5	0.78	104/110	0.47	0.77	98/110	0.76	0.91	110/110
de	0.79	0.91	109/110	0.49	0.79	107/110	0.43	0.76	104/110	0.79	0.91	110/110
dk	0.81	0.92	109/110	0.31	0.67	25/110	0.3	0.67	24/110	0.8	0.92	110/110
do	0.81	0.94	109/110	0.31	0.67	58/110	0.3	0.65	57/110	0.81	0.94	109/110
ec	0.9	0.97	109/110	0.63	0.86	110/110	0.63	0.85	109/110	0.89	0.97	110/110
ee	0.72	0.88	81/83	0.38	0.73	67/83	0.39	0.72	58/83	0.71	0.88	83/83
es	0.86	0.95	109/110	0.36	0.68	61/110	0.35	0.67	60/110	0.86	0.95	110/110
fi	0.8	0.93	109/110	-	-	0/110	-	-	0/110	0.8	0.93	109/110
fr	0.72	0.88	109/110	-	-	0/110	0.13	0.7	1/110	0.72	0.88	109/110
gb	0.83	0.94	109/110	0.53	0.78	108/110	0.5	0.77	106/110	0.82	0.94	110/110
global	0.84	0.94	109/110	0.45	0.76	106/110	0.44	0.76	97/110	0.84	0.94	110/110
gr	0.77	0.92	109/110	0.49	0.77	102/110	0.44	0.76	98/110	0.77	0.91	110/110
gt	0.9	0.96	109/110	0.67	0.85	110/110	0.65	0.84	109/110	0.9	0.96	110/110
hk	0.8	0.91	109/110	0.45	0.76	87/110	0.44	0.76	86/110	0.8	0.91	110/110
hn	0.87	0.96	109/110	0.59	0.84	110/110	0.57	0.83	109/110	0.87	0.95	110/110
hu	0.79	0.91	109/110	0.51	0.79	108/110	0.46	0.77	105/110	0.79	0.91	110/110
id	0.85	0.94	109/110	0.49	0.81	54/110	0.46	0.78	57/110	0.85	0.94	110/110
ie	0.85	0.95	109/110	0.53	0.78	108/110	0.49	0.77	106/110	0.85	0.95	110/110
il	0.72	0.89	63/64	0.3	0.83	7/64	0.13	0.68	8/64	0.72	0.89	63/64
is	0.79	0.91	109/110	0.24	0.68	13/110	0.21	0.69	13/110	0.79	0.91	110/110
it	0.8	0.92	109/110	0.15	0.61	7/110	0.18	0.64	10/110	0.79	0.91	110/110
jp	0.8	0.93	109/110	0.38	0.72	29/110	0.35	0.74	27/110	0.8	0.93	110/110
lt	0.73	0.89	86/88	0.44	0.78	82/88	0.42	0.77	81/88	0.72	0.89	88/88
lu	0.76	0.91	28/30	0.62	0.82	29/30	0.56	0.8	29/30	0.72	0.88	30/30
lv	0.7	0.89	104/105	0.47	0.77	89/105	0.41	0.75	88/105	0.7	0.89	105/105
mt	0.81	0.93	31/33	0.52	0.77	30/33	0.51	0.77	30/33	0.77	0.91	33/33
mx	0.91	0.96	109/110	0.56	0.8	103/110	0.56	0.8	106/110	0.9	0.96	110/110
my	0.84	0.94	109/110	0.53	0.81	109/110	0.49	0.79	108/110	0.83	0.94	110/110
ni	0.86	0.94	29/31	0.65	0.82	31/31	0.66	0.82	30/31	0.84	0.93	31/31

nl	0.8	0.92	109/110	0.22	0.71	9/110	0.24	0.71	10/110	0.8	0.92	109/110
no	0.84	0.95	109/110	0.31	0.68	33/110	0.28	0.65	29/110	0.84	0.94	110/110
nz	0.84	0.95	109/110	0.56	0.84	107/110	0.53	0.82	106/110	0.84	0.95	110/110
pa	0.86	0.95	109/110	0.45	0.75	95/110	0.44	0.74	92/110	0.85	0.95	110/110
pe	0.94	0.98	109/110	0.61	0.84	110/110	0.63	0.84	109/110	0.93	0.98	110/110
ph	0.87	0.96	109/110	0.32	0.77	30/110	0.3	0.74	30/110	0.86	0.95	110/110
pl	0.74	0.9	109/110	0.44	0.76	42/110	0.45	0.76	38/110	0.74	0.9	110/110
pt	0.77	0.93	109/110	0.33	0.74	48/110	0.34	0.74	45/110	0.77	0.93	110/110
py	0.91	0.97	109/110	0.44	0.78	101/110	0.45	0.79	99/110	0.9	0.97	110/110
ro	0.71	0.87	63/64	0.38	0.73	57/64	0.34	0.68	59/64	0.71	0.87	63/64
se	0.79	0.91	109/110	0.4	0.68	20/110	0.35	0.68	20/110	0.78	0.91	110/110
sg	0.86	0.94	109/110	0.53	0.82	109/110	0.48	0.8	108/110	0.85	0.94	110/110
sk	0.72	0.89	109/110	0.51	0.8	99/110	0.47	0.78	95/110	0.71	0.89	110/110
sv	0.9	0.96	109/110	0.67	0.85	110/110	0.65	0.84	109/110	0.9	0.96	110/110
th	0.77	0.9	79/79	0.31	0.67	6/79	0.29	0.67	7/79	0.77	0.9	79/79
tr	0.87	0.94	109/110	0.16	0.6	1/110	0.11	0.57	4/110	0.87	0.94	109/110
tw	0.65	0.87	109/110	0.41	0.78	62/110	0.4	0.77	60/110	0.65	0.87	110/110
us	0.78	0.92	109/110	0.45	0.73	93/110	0.41	0.72	89/110	0.77	0.91	110/110
uy	0.91	0.97	109/110	0.6	0.85	110/110	0.58	0.84	109/110	0.91	0.96	110/110
vn	0.73	0.9	63/64	0.22	0.57	4/64	0.2	0.57	4/64	0.73	0.9	63/64
avg	0.82	0.93	0.99	0.5	0.79	0.68	0.48	0.77	0.67	0.81	0.93	1.0

Table 6.5: Results with directly copied rankings as replacement on the Spotify daily data set for top-10 prediction.

The implicit delegation methods that use aggregation of Table 6.6 show something interesting. First the trimming before aggregation seems to have a positive effect, which could be attributed to the larger size with top-10 instead of top-3. Secondly the weighted versions namely WCR and TWCR which combine as many possible rankings as are found fare worse than their non weighted counterparts CR and TCR, which only consider up to the best 5 rankings. This seems to suggest that the weights are not enough to decrease the impact of rankings with lower similarity. Overall all four methods performed worse than using a simple copy of some promising ranking. If this persists through all data sets it can probably be assumed that these aggregations have no particularly useful effect, at least when using Borda or the weighted Borda with the tau-a scores as weights.

	WCR			TWCR			TCR			CR		
voter	tau-a	i-size	sub	tau-a	i-size	sub	tau-a	i-size	sub	tau-a	i-size	sub
ar	0.56	0.8	110/110	0.64	0.85	110/110	0.84	0.94	110/110	0.82	0.92	110/110
at	0.42	0.74	110/110	0.59	0.83	110/110	0.68	0.87	110/110	0.61	0.83	110/110
au	0.48	0.79	110/110	0.62	0.87	110/110	0.76	0.92	110/110	0.71	0.88	110/110
be	0.46	0.77	110/110	0.58	0.83	110/110	0.66	0.88	110/110	0.62	0.85	110/110
bg	0.39	0.7	38/38	0.46	0.77	38/38	0.58	0.82	38/38	0.54	0.77	38/38
bo	0.57	0.84	110/110	0.65	0.88	110/110	0.85	0.95	110/110	0.84	0.95	110/110
br	0.6	0.85	109/110	0.63	0.88	109/110	0.67	0.89	109/110	0.66	0.87	109/110
ca	0.48	0.78	110/110	0.63	0.85	110/110	0.7	0.89	110/110	0.66	0.85	110/110
ch	0.37	0.73	110/110	0.56	0.83	110/110	0.63	0.86	110/110	0.53	0.79	110/110
cl	0.56	0.81	110/110	0.66	0.86	110/110	0.84	0.95	110/110	0.82	0.93	110/110
co	0.52	0.77	110/110	0.62	0.83	110/110	0.84	0.95	110/110	0.83	0.94	110/110
cr	0.6	0.81	110/110	0.66	0.84	110/110	0.84	0.94	110/110	0.83	0.93	110/110
cz	0.46	0.77	110/110	0.59	0.85	110/110	0.71	0.89	110/110	0.66	0.86	110/110
de	0.39	0.72	110/110	0.56	0.83	110/110	0.67	0.88	110/110	0.6	0.84	110/110
dk	0.44	0.75	110/110	0.58	0.85	110/110	0.63	0.88	110/110	0.55	0.82	110/110
do	0.54	0.81	109/110	0.62	0.87	109/110	0.71	0.92	109/110	0.69	0.9	109/110
ec	0.55	0.83	110/110	0.63	0.89	110/110	0.84	0.95	110/110	0.83	0.95	110/110
ee	0.46	0.77	83/83	0.58	0.84	83/83	0.65	0.87	83/83	0.61	0.83	83/83
es	0.54	0.8	110/110	0.63	0.85	110/110	0.78	0.92	110/110	0.76	0.91	110/110
fi	0.45	0.78	109/110	0.61	0.88	109/110	0.61	0.88	109/110	0.53	0.82	109/110
fr	0.52	0.78	109/110	0.6	0.83	109/110	0.61	0.84	109/110	0.56	0.8	109/110
gb	0.5	0.76	110/110	0.63	0.84	110/110	0.73	0.9	110/110	0.7	0.87	110/110
global	0.48	0.78	110/110	0.61	0.87	110/110	0.74	0.92	110/110	0.7	0.88	110/110

gr	0.45	0.76	110/110	0.6	0.85	110/110	0.7	0.88	110/110	0.65	0.86	110/110
gt	0.62	0.84	110/110	0.7	0.9	110/110	0.85	0.94	110/110	0.84	0.93	110/110
hk	0.47	0.78	110/110	0.58	0.85	110/110	0.68	0.89	110/110	0.63	0.85	110/110
hn	0.53	0.82	110/110	0.62	0.88	110/110	0.83	0.95	110/110	0.82	0.93	110/110
hu	0.46	0.76	110/110	0.59	0.83	110/110	0.71	0.88	110/110	0.65	0.85	110/110
id	0.56	0.83	110/110	0.65	0.88	110/110	0.77	0.92	110/110	0.75	0.9	110/110
ie	0.49	0.77	110/110	0.62	0.85	110/110	0.76	0.91	110/110	0.71	0.87	110/110
il	0.48	0.8	63/64	0.58	0.86	63/64	0.62	0.88	63/64	0.57	0.84	63/64
is	0.53	0.8	110/110	0.63	0.85	110/110	0.68	0.87	110/110	0.62	0.85	110/110
it	0.55	0.81	110/110	0.63	0.87	110/110	0.66	0.88	110/110	0.62	0.84	110/110
jp	0.57	0.82	110/110	0.64	0.85	110/110	0.72	0.89	110/110	0.69	0.87	110/110
lt	0.48	0.78	88/88	0.56	0.83	88/88	0.66	0.87	88/88	0.64	0.86	88/88
lu	0.46	0.76	30/30	0.6	0.85	30/30	0.71	0.88	30/30	0.69	0.87	30/30
lv	0.48	0.77	105/105	0.58	0.83	105/105	0.64	0.86	105/105	0.59	0.83	105/105
mt	0.48	0.78	33/33	0.62	0.86	33/33	0.74	0.9	33/33	0.68	0.87	33/33
mx	0.59	0.82	110/110	0.65	0.84	110/110	0.84	0.94	110/110	0.83	0.93	110/110
my	0.54	0.81	110/110	0.63	0.87	110/110	0.77	0.93	110/110	0.74	0.9	110/110
ni	0.59	0.82	31/31	0.65	0.84	31/31	0.82	0.92	31/31	0.81	0.91	31/31
nl	0.51	0.79	109/110	0.64	0.88	109/110	0.68	0.9	109/110	0.63	0.85	109/110
no	0.48	0.78	110/110	0.61	0.87	110/110	0.71	0.91	110/110	0.66	0.86	110/110
nz	0.5	0.8	110/110	0.64	0.88	110/110	0.74	0.9	110/110	0.7	0.87	110/110
pa	0.51	0.78	110/110	0.62	0.87	110/110	0.8	0.94	110/110	0.78	0.92	110/110
pe	0.66	0.85	110/110	0.7	0.88	110/110	0.88	0.96	110/110	0.87	0.95	110/110
ph	0.65	0.86	110/110	0.69	0.89	110/110	0.78	0.92	110/110	0.77	0.92	110/110
pl	0.5	0.78	110/110	0.58	0.83	110/110	0.65	0.87	110/110	0.6	0.84	110/110
pt	0.49	0.8	110/110	0.6	0.86	110/110	0.69	0.89	110/110	0.66	0.87	110/110
py	0.47	0.79	110/110	0.57	0.87	110/110	0.83	0.94	110/110	0.82	0.94	110/110
ro	0.43	0.74	63/64	0.56	0.81	63/64	0.6	0.83	63/64	0.56	0.81	63/64
se	0.47	0.76	110/110	0.58	0.84	110/110	0.64	0.87	110/110	0.59	0.83	110/110
sg	0.55	0.83	110/110	0.66	0.87	110/110	0.78	0.93	110/110	0.75	0.9	110/110
sk	0.48	0.77	110/110	0.58	0.83	110/110	0.65	0.86	110/110	0.63	0.85	110/110
sv	0.62	0.82	110/110	0.7	0.87	110/110	0.87	0.95	110/110	0.86	0.94	110/110
th	0.56	0.82	79/79	0.62	0.86	79/79	0.69	0.89	79/79	0.64	0.85	79/79
tr	0.63	0.85	109/110	0.72	0.91	109/110	0.78	0.92	109/110	0.75	0.9	109/110
tw	0.46	0.79	110/110	0.55	0.84	110/110	0.64	0.88	110/110	0.6	0.84	110/110
us	0.51	0.79	110/110	0.63	0.84	110/110	0.71	0.88	110/110	0.65	0.84	110/110
uy	0.53	0.8	110/110	0.62	0.85	110/110	0.85	0.94	110/110	0.84	0.94	110/110
vn	0.51	0.78	63/64	0.59	0.84	63/64	0.62	0.87	63/64	0.57	0.81	63/64
avg	0.51	0.79	1.0	0.62	0.86	1.0	0.73	0.9	1.0	0.69	0.88	1.0

Table 6.6: Results with aggregated rankings as replacement on the Spotify daily data set for top-10 prediction.

For the full prediction experiments it can be seen from Table 6.7 and and Table 6.8 that most of the times all methods could at least replace one missing voter but there seem to be some cases where the random selected combination of missing voter could not be substituted at all. Otherwise both tables are nearly identical.

Here again PR performed the best as expected. The implicit delegation with DVR and PDVR seem to not have much effect on the resulting score despite both being able to replace more than halve of the missing voters. This could be explained through the analysis of the similarities between the voters which was shown in Section 4.2.1. There the highest scores reached a tau-b score of around 0.6 which is the same as the minimum tau-a score decided on for the implicit delegation itself. This threshold was chosen, as it lead to more neutral results in some testing and everything above is able to have positive effect. Therefore only barely reaching that threshold is also very likely to only produce nearly neutral results. Between the results of the different aggregation methods the implicit delegation methods seem to work very similar.

	default	PR	DVR	PDVR	HSR	CR	WCR
Gain	0.0	0.98	0.55	0.59	0.99	0.99	0.99
Borda	0.62	0.91	0.61	0.63	0.91	0.87	0.85
Nanson	0.62	0.91	0.61	0.63	0.91	0.87	0.84
Copeland	0.62	0.9	0.62	0.64	0.9	0.87	0.83
Maximin	0.64	0.9	0.64	0.64	0.9	0.86	0.84

Table 6.7: Average Kendall tau-a scores after implicit delegation of multiple rankings for the Spotify daily data set.

	default	PR	DVR	PDVR	HSR	CR	WCR
Gain	0.0	1.0	0.55	0.6	0.99	0.99	0.99
Borda	0.62	0.92	0.61	0.63	0.91	0.87	0.85
Nanson	0.62	0.92	0.61	0.63	0.91	0.87	0.84
Copeland	0.62	0.91	0.62	0.64	0.9	0.87	0.83
Maximin	0.64	0.9	0.64	0.64	0.9	0.86	0.84

Table 6.8: Average Kendall tau-a scores of successful implicit delegation attempts of multiple rankings for the Spotify daily data set.

6.3.3 Spotify Weekly Data Set

For the Spotify weekly charts data set 38 data points for top-10 prediction and 21 data points for full prediction were randomly selected.

The previous analysis of the data set serves again as good indicator of the experiments results. Table 6.9 shows that for the top-10 predictions PR performed again the best, but as expected from the data the results are not as good as for Spotify daily charts. This is also the case for the other two search strategies DVR and PDVR, they found substitutes that have on average a high intersection size score, but their tau-a scores are not optimal. This is now the third data set consecutively where PDVR did not show an advantage over DVR and produces lower scores despite looking at much more possible rankings. In this case it even found far less substitutes. Here HSR shines, as it is able to hold the scores of PR while increasing the amount of successful replacement attempts.

If the voters are looked at individually there are many examples where the previous analysis was a strong indicator for voters where DVR did not or only barely find replacements. A few examples are "vn", "tr", "nl", "jp", "it" and "br". Also for PR some info could be gained from the analysis of the whole data set. For example "ro" has the lowest tau-a score in the results and this voter also has one of the lowest average scores in the whole analysis for similarities to past votes.

voter	PR tau-a	PR i-size	PR sub	DVR tau-a	DVR i-size	DVR sub	PDVR tau-a	PDVR i-size	PDVR sub	HSR tau-a	HSR i-size	HSR sub
ad	0.55	0.81	8/9	0.06	0.5	5/9	0.23	0.62	4/9	0.5	0.78	9/9
ar	0.68	0.88	37/38	0.55	0.83	38/38	0.48	0.79	37/38	0.65	0.88	38/38
at	0.4	0.75	31/38	0.45	0.77	37/38	0.18	0.68	16/38	0.5	0.79	38/38
au	0.59	0.84	37/38	0.53	0.79	38/38	0.38	0.74	36/38	0.53	0.8	38/38
be	0.47	0.81	37/38	0.23	0.65	31/38	0.16	0.64	13/38	0.48	0.81	38/38
bg	0.33	0.72	31/38	0.43	0.75	38/38	0.2	0.67	23/38	0.43	0.75	38/38
bo	0.71	0.91	37/38	0.58	0.84	38/38	0.51	0.82	37/38	0.7	0.91	38/38
br	0.63	0.86	37/38	-	-	0/38	-	-	0/38	0.63	0.86	37/38
ca	0.41	0.75	33/38	0.39	0.75	38/38	0.19	0.68	11/38	0.44	0.78	38/38
ch	0.44	0.8	25/38	0.39	0.75	23/38	0.32	0.72	14/38	0.42	0.78	30/38
cl	0.69	0.9	37/38	0.45	0.75	35/38	0.35	0.71	34/38	0.68	0.89	38/38
co	0.71	0.91	37/38	0.38	0.73	32/38	0.34	0.72	26/38	0.71	0.9	38/38
cr	0.68	0.9	37/38	0.48	0.76	38/38	0.37	0.72	37/38	0.68	0.9	38/38
cy	0.43	0.77	37/38	0.46	0.78	38/38	0.27	0.67	24/38	0.49	0.8	38/38
cz	0.44	0.76	37/38	0.47	0.78	38/38	0.3	0.73	30/38	0.46	0.79	38/38
de	0.36	0.73	30/38	0.45	0.76	37/38	0.12	0.59	14/38	0.45	0.77	38/38
dk	0.4	0.76	37/38	0.18	0.65	2/38	-	-	0/38	0.4	0.76	38/38
do	0.53	0.83	37/38	0.28	0.68	23/38	0.15	0.61	13/38	0.51	0.82	37/38
ec	0.69	0.88	37/38	0.6	0.84	38/38	0.48	0.8	37/38	0.69	0.88	38/38
ee	0.42	0.76	37/38	0.36	0.76	27/38	0.22	0.69	18/38	0.44	0.77	38/38
es	0.62	0.88	37/38	0.27	0.7	22/38	0.2	0.68	17/38	0.61	0.87	38/38
fi	0.36	0.74	34/38	-	-	0/38	-	-	0/38	0.36	0.74	34/38
fr	0.38	0.71	34/38	-	-	0/38	-	-	0/38	0.38	0.71	34/38
gb	0.49	0.8	36/38	0.5	0.79	38/38	0.38	0.72	31/38	0.49	0.79	38/38
global	0.5	0.8	33/37	0.39	0.77	37/37	0.34	0.72	19/37	0.45	0.77	37/37
gr	0.43	0.78	36/38	0.45	0.79	38/38	0.29	0.72	25/38	0.47	0.8	38/38
gt	0.68	0.89	37/38	0.61	0.84	38/38	0.43	0.78	37/38	0.67	0.87	38/38
hk	0.51	0.79	37/38	0.39	0.76	30/38	0.4	0.78	12/38	0.48	0.79	38/38
hn	0.66	0.89	37/38	0.54	0.84	38/38	0.38	0.76	37/38	0.65	0.89	38/38
hu	0.42	0.78	37/38	0.46	0.77	38/38	0.27	0.67	33/38	0.44	0.76	38/38
id	0.59	0.85	37/38	0.36	0.74	20/38	0.34	0.76	17/38	0.58	0.84	38/38
ie	0.53	0.82	37/38	0.5	0.79	38/38	0.31	0.71	35/38	0.5	0.8	38/38
il	0.51	0.8	22/23	0.11	0.7	4/23	0.08	0.7	1/23	0.49	0.79	23/23
is	0.42	0.76	36/38	-0.14	0.5	1/38	-	-	0/38	0.42	0.76	36/38
it	0.45	0.75	36/38	0.11	0.53	3/38	0.1	0.56	5/38	0.45	0.75	36/38
jp	0.6	0.85	28/29	-	-	0/29	-	-	0/29	0.6	0.85	28/29
lt	0.41	0.74	37/38	0.34	0.74	38/38	0.17	0.66	15/38	0.38	0.74	38/38
lu	0.39	0.79	34/37	0.37	0.75	37/37	0.24	0.68	24/37	0.38	0.76	37/37
lv	0.36	0.75	36/38	0.4	0.79	29/38	0.27	0.69	23/38	0.35	0.74	37/38
mt	0.42	0.76	37/38	0.37	0.73	38/38	0.3	0.7	18/38	0.44	0.77	38/38
mx	0.69	0.89	37/38	0.5	0.79	33/38	0.43	0.75	32/38	0.69	0.89	38/38
my	0.54	0.83	37/38	0.5	0.8	38/38	0.36	0.74	36/38	0.55	0.84	38/38
ni	0.66	0.89	37/38	0.53	0.8	38/38	0.43	0.78	37/38	0.66	0.89	38/38
nl	0.39	0.77	36/38	-	-	0/38	-	-	0/38	0.39	0.77	36/38
no	0.49	0.78	37/38	0.1	0.63	3/38	-	-	0/38	0.48	0.78	38/38
nz	0.57	0.84	37/38	0.54	0.8	38/38	0.39	0.75	37/38	0.53	0.81	38/38
pa	0.64	0.86	37/38	0.34	0.72	37/38	0.27	0.68	31/38	0.64	0.86	38/38
pe	0.7	0.91	37/38	0.56	0.81	38/38	0.5	0.81	37/38	0.7	0.9	38/38
ph	0.67	0.89	37/38	0.29	0.77	9/38	0.26	0.7	8/38	0.67	0.89	38/38
pl	0.4	0.76	36/38	0.26	0.72	12/38	0.34	0.73	7/38	0.39	0.75	37/38
pt	0.53	0.82	37/38	0.25	0.66	19/38	0.17	0.68	12/38	0.53	0.82	38/38
py	0.67	0.9	37/38	0.38	0.75	32/38	0.3	0.73	28/38	0.66	0.89	38/38
ro	0.27	0.64	18/23	0.37	0.73	23/23	0.16	0.64	7/23	0.29	0.69	23/23
se	0.42	0.77	37/38	0.02	0.4	1/38	-	-	0/38	0.41	0.76	38/38
sg	0.58	0.85	37/38	0.51	0.81	38/38	0.35	0.75	32/38	0.58	0.85	38/38
sk	0.39	0.74	37/38	0.46	0.76	36/38	0.31	0.71	24/38	0.45	0.76	38/38
sv	0.66	0.88	37/38	0.6	0.84	38/38	0.52	0.8	37/38	0.65	0.87	38/38
th	0.56	0.83	28/29	0.1	0.62	4/29	-	-	0/29	0.54	0.81	29/29
tr	0.59	0.85	37/38	-	-	0/38	-	-	0/38	0.59	0.85	37/38
tw	0.52	0.8	37/38	0.33	0.75	26/38	0.38	0.71	11/38	0.51	0.8	38/38
us	0.38	0.73	29/38	0.4	0.75	34/38	0.21	0.67	6/38	0.39	0.74	38/38
uy	0.67	0.87	37/38	0.54	0.83	38/38	0.41	0.77	37/38	0.65	0.87	38/38
vn	0.4	0.74	22/23	-	-	0/23	-	-	0/23	0.4	0.74	22/23
avg	0.52	0.81	0.94	0.44	0.77	0.7	0.34	0.73	0.52	0.52	0.81	0.99

Table 6.9: Results with directly copied rankings as replacement on the Spotify weekly data set for top-10 prediction.

The implicit delegation methods using aggregation in Table 6.10 once more did not provide an advantage over simply copying rankings. And here like for the Spotify daily data the versions that trim to top-10 before aggregating perform better than those without the trimming. It is also further evident, that using the suggested weights is outperformed by limiting the rankings to the best guesses available.

	WCR			TWCR			TCR			CR		
voter	tau-a	i-size	sub	tau-a	i-size	sub	tau-a	i-size	sub	tau-a	i-size	sub
ad	0.35	0.72	9/9	0.39	0.74	9/9	0.41	0.76	9/9	0.35	0.76	9/9
ar	0.35	0.71	38/38	0.49	0.81	38/38	0.6	0.86	38/38	0.56	0.81	38/38
at	0.35	0.7	38/38	0.48	0.79	38/38	0.48	0.8	38/38	0.37	0.72	38/38
au	0.42	0.77	38/38	0.51	0.81	38/38	0.53	0.82	38/38	0.44	0.78	38/38
be	0.33	0.71	38/38	0.43	0.79	38/38	0.4	0.77	38/38	0.32	0.71	38/38
bg	0.28	0.68	38/38	0.41	0.74	38/38	0.42	0.76	38/38	0.29	0.69	38/38
bo	0.37	0.76	38/38	0.51	0.84	38/38	0.64	0.87	38/38	0.61	0.86	38/38
br	0.43	0.76	37/38	0.53	0.83	37/38	0.49	0.83	37/38	0.41	0.76	37/38
ca	0.39	0.74	38/38	0.5	0.78	38/38	0.49	0.79	38/38	0.4	0.75	38/38
ch	0.24	0.71	30/38	0.37	0.75	30/38	0.38	0.77	30/38	0.3	0.73	30/38
cl	0.3	0.71	38/38	0.46	0.82	38/38	0.55	0.85	38/38	0.46	0.77	38/38
co	0.28	0.67	38/38	0.45	0.79	38/38	0.51	0.85	38/38	0.39	0.72	38/38
cr	0.33	0.72	38/38	0.47	0.78	38/38	0.55	0.84	38/38	0.48	0.81	38/38
cy	0.43	0.77	38/38	0.57	0.84	38/38	0.6	0.86	38/38	0.51	0.82	38/38
cz	0.36	0.73	38/38	0.49	0.78	38/38	0.5	0.82	38/38	0.44	0.76	38/38
de	0.35	0.7	38/38	0.45	0.77	38/38	0.44	0.78	38/38	0.36	0.71	38/38
dk	0.35	0.74	38/38	0.38	0.76	38/38	0.37	0.75	38/38	0.35	0.73	38/38
do	0.31	0.71	37/38	0.42	0.79	37/38	0.4	0.8	37/38	0.31	0.7	37/38
ec	0.34	0.74	38/38	0.48	0.84	38/38	0.61	0.88	38/38	0.55	0.83	38/38
ee	0.33	0.74	38/38	0.44	0.79	38/38	0.45	0.81	38/38	0.36	0.75	38/38
es	0.34	0.75	38/38	0.44	0.81	38/38	0.44	0.82	38/38	0.35	0.75	38/38
fi	0.34	0.71	34/38	0.35	0.73	34/38	0.35	0.73	34/38	0.33	0.71	34/38
fr	0.37	0.71	34/38	0.38	0.71	34/38	0.37	0.71	34/38	0.37	0.71	34/38
gb	0.45	0.76	38/38	0.51	0.8	38/38	0.5	0.8	38/38	0.45	0.77	38/38
global	0.38	0.76	37/37	0.52	0.82	37/37	0.55	0.82	37/37	0.46	0.78	37/37
gr	0.39	0.74	38/38	0.57	0.84	38/38	0.62	0.85	38/38	0.51	0.79	38/38
gt	0.38	0.75	38/38	0.54	0.86	38/38	0.66	0.89	38/38	0.59	0.83	38/38
hk	0.41	0.77	38/38	0.5	0.81	38/38	0.49	0.81	38/38	0.44	0.79	38/38
hn	0.32	0.74	38/38	0.5	0.87	38/38	0.62	0.9	38/38	0.51	0.83	38/38
hu	0.37	0.74	38/38	0.48	0.8	38/38	0.51	0.81	38/38	0.44	0.78	38/38
id	0.38	0.76	38/38	0.47	0.81	38/38	0.48	0.83	38/38	0.41	0.77	38/38
ie	0.44	0.78	38/38	0.49	0.81	38/38	0.47	0.82	38/38	0.44	0.79	38/38
il	0.42	0.77	23/23	0.45	0.8	23/23	0.43	0.8	23/23	0.4	0.77	23/23
is	0.4	0.74	36/38	0.42	0.75	36/38	0.41	0.75	36/38	0.4	0.74	36/38
it	0.36	0.71	36/38	0.39	0.73	36/38	0.36	0.72	36/38	0.34	0.71	36/38
jp	0.45	0.77	28/29	0.52	0.82	28/29	0.47	0.8	28/29	0.43	0.77	28/29
lt	0.35	0.74	38/38	0.45	0.77	38/38	0.45	0.78	38/38	0.4	0.77	38/38
lu	0.35	0.76	37/37	0.47	0.8	37/37	0.49	0.81	37/37	0.43	0.79	37/37
lv	0.32	0.73	37/38	0.4	0.76	37/38	0.43	0.78	37/38	0.39	0.75	37/38
mt	0.37	0.72	38/38	0.46	0.79	38/38	0.48	0.8	38/38	0.42	0.75	38/38
mx	0.33	0.72	38/38	0.5	0.82	38/38	0.56	0.83	38/38	0.48	0.77	38/38
my	0.36	0.77	38/38	0.49	0.81	38/38	0.54	0.82	38/38	0.43	0.79	38/38
ni	0.35	0.73	38/38	0.5	0.81	38/38	0.62	0.86	38/38	0.58	0.84	38/38
nl	0.35	0.76	36/38	0.37	0.76	36/38	0.35	0.76	36/38	0.34	0.76	36/38
no	0.4	0.74	38/38	0.45	0.77	38/38	0.43	0.77	38/38	0.4	0.74	38/38
nz	0.42	0.77	38/38	0.49	0.82	38/38	0.47	0.84	38/38	0.41	0.77	38/38
pa	0.26	0.66	38/38	0.45	0.8	38/38	0.51	0.83	38/38	0.36	0.71	38/38
pe	0.38	0.75	38/38	0.55	0.84	38/38	0.66	0.89	38/38	0.63	0.86	38/38
ph	0.46	0.78	38/38	0.54	0.84	38/38	0.48	0.83	38/38	0.44	0.77	38/38
pl	0.32	0.7	37/38	0.36	0.72	37/38	0.35	0.72	37/38	0.33	0.71	37/38
pt	0.34	0.72	38/38	0.47	0.81	38/38	0.45	0.81	38/38	0.34	0.71	38/38
py	0.29	0.74	38/38	0.41	0.82	38/38	0.5	0.84	38/38	0.39	0.78	38/38
ro	0.3	0.72	23/23	0.43	0.77	23/23	0.46	0.77	23/23	0.32	0.73	23/23
se	0.37	0.73	38/38	0.4	0.75	38/38	0.39	0.75	38/38	0.36	0.72	38/38
sg	0.43	0.77	38/38	0.57	0.84	38/38	0.57	0.84	38/38	0.5	0.81	38/38
sk	0.38	0.73	38/38	0.51	0.78	38/38	0.53	0.77	38/38	0.44	0.75	38/38
sv	0.37	0.74	38/38	0.56	0.85	38/38	0.67	0.88	38/38	0.6	0.85	38/38
th	0.44	0.78	29/29	0.49	0.79	29/29	0.46	0.79	29/29	0.42	0.77	29/29
tr	0.41	0.76	37/38	0.49	0.82	37/38	0.44	0.81	37/38	0.38	0.75	37/38
tw	0.36	0.72	38/38	0.45	0.8	38/38	0.46	0.82	38/38	0.39	0.74	38/38
us	0.32	0.7	38/38	0.41	0.76	38/38	0.4	0.76	38/38	0.32	0.7	38/38

uy	0.32	0.74	38/38	0.46	0.8	38/38	0.58	0.84	38/38	0.52	0.81	38/38
vn	0.38	0.73	22/23	0.39	0.74	22/23	0.39	0.74	22/23	0.38	0.73	22/23
avg	0.36	0.74	0.99	0.47	0.8	0.99	0.49	0.81	0.99	0.42	0.76	0.99

Table 6.10: Results with aggregated rankings as replacement on the Spotify weekly data set for top-10 prediction.

The results of the experiments for full prediction of a third of the voters can be seen in the tables 6.11 and 6.12. From this it can be seen, that most of the times at least one of the missing voters could be substituted. The aggregation method Maximin seems to be the least effected by removing voters, this leads to the implicit delegation methods producing the least improvement if tested with Maximin. Still HSR and PR perform the best overall and DVR together with PDVR sometimes even worsen the score despite replacing over 50% of the voters. But this bad performance of DVR is not surprising, as the analysis of the data set already showed mostly low tau-b scores, but some acceptable intersection sizes. This probably lead to many rankings to be accepted because of their intersection size despite relatively low tau-a score.

	default	PR	DVR	PDVR	HSR	CR	WCR
Gain	0.0	0.95	0.64	0.54	0.98	0.98	0.98
Borda	0.62	0.85	0.61	0.61	0.85	0.8	0.8
Nanson	0.62	0.85	0.61	0.61	0.85	0.8	0.8
Copeland	0.63	0.84	0.62	0.62	0.84	0.78	0.78
Maximin	0.64	0.83	0.64	0.62	0.83	0.78	0.78

Table 6.11: Average Kendall tau-a scores after implicit delegation of multiple rankings for the Spotify weekly data set.

	default	PR	DVR	PDVR	HSR	CR	WCR
Gain	0.0	1.0	0.64	0.57	0.98	0.98	0.98
Borda	0.62	0.86	0.61	0.61	0.85	0.8	0.8
Nanson	0.62	0.86	0.61	0.61	0.85	0.8	0.8
Copeland	0.63	0.85	0.62	0.62	0.84	0.78	0.78
Maximin	0.64	0.84	0.64	0.62	0.84	0.79	0.78

Table 6.12: Average Kendall tau-a scores of successful implicit delegation attempts of multiple rankings for the Spotify weekly data set.

6.3.4 Spotify Viral Daily Data Set

Now to the Spotify data sets that only look at viral top-50 rankings. Here 99 data points were selected for top-10 prediction and 48 for full prediction. The selection was done with the same parameters as for Spotify daily and therefore it is completely random that fewer data points were selected.

The experiments on this data set show how much some basic analysis can help with selecting an appropriate implicit delegation method. In Chapter 4.2.3 it was shown that the different voters have low similarities according to Kendall tau-b and this reflects on the results here. The methods DVR and PDVR performed similar with relative low but positive tau-a scores and an average intersection size of just under 60%. The analysis of similarities to past votes showed similar results to the one of weekly charts and this can also be seen in the experiment results, as PR produced pretty close average scores for both data sets. As PR could substitute nearly all rankings and better than the other two methods it is also not surprising that HSR has nearly identical results.

When looking at the voters individually no extreme outliers are found, however for DVR "us" was replaced unusually often. With the previous analysis of the whole data set this could be explained as it is the only voter that has an average tau-b score of at least 0.2 with two other voters. All other voters have only fewer or less similar different voters.

	PR			DVR			PDVR			HSR		
voter	tau-a	i-size	sub	tau-a	i-size	sub	tau-a	i-size	sub	tau-a	i-size	sub
ar	0.6	0.84	99/99	0.36	0.72	43/99	0.3	0.71	36/99	0.6	0.84	99/99
at	0.49	0.79	99/99	0.27	0.67	16/99	0.26	0.68	16/99	0.49	0.79	99/99
au	0.6	0.82	99/99	0.29	0.68	42/99	0.32	0.68	36/99	0.6	0.82	99/99
be	0.54	0.81	99/99	0.27	0.7	2/99	0.28	0.7	2/99	0.54	0.81	99/99
bg	0.56	0.83	99/99	0.22	0.67	10/99	0.18	0.7	4/99	0.56	0.83	99/99
bo	0.62	0.85	99/99	0.36	0.69	56/99	0.34	0.69	52/99	0.62	0.85	99/99
br	0.62	0.84	99/99	-	-	0/99	-	-	0/99	0.62	0.84	99/99
ca	0.61	0.83	99/99	0.36	0.73	59/99	0.35	0.73	42/99	0.6	0.83	99/99
ch	0.48	0.77	99/99	0.19	0.6	6/99	0.24	0.6	5/99	0.48	0.77	99/99
cl	0.66	0.87	99/99	0.26	0.66	40/99	0.19	0.64	30/99	0.66	0.87	99/99
co	0.62	0.83	99/99	0.44	0.71	29/99	0.43	0.71	24/99	0.62	0.83	99/99
cr	0.59	0.85	99/99	0.31	0.68	32/99	0.31	0.67	24/99	0.59	0.85	99/99
cz	0.51	0.79	99/99	0.27	0.63	10/99	0.21	0.58	5/99	0.51	0.79	99/99
de	0.61	0.83	99/99	0.27	0.66	16/99	0.27	0.68	15/99	0.61	0.83	99/99
dk	0.58	0.82	99/99	0.27	0.6	4/99	0.24	0.55	4/99	0.58	0.82	99/99
do	0.56	0.81	99/99	0.3	0.7	25/99	0.26	0.63	22/99	0.56	0.81	99/99
ec	0.66	0.86	99/99	0.38	0.7	54/99	0.4	0.7	40/99	0.66	0.86	99/99
ee	0.53	0.82	99/99	0.29	0.7	2/99	-	-	0/99	0.53	0.82	99/99
es	0.61	0.84	99/99	-	-	0/99	-	-	0/99	0.61	0.84	99/99
fi	0.53	0.79	98/99	-0.37	0.5	1/99	-0.16	0.5	1/99	0.53	0.79	98/99
fr	0.56	0.81	99/99	-	-	0/99	-	-	0/99	0.56	0.81	99/99
gb	0.59	0.81	99/99	0.27	0.69	37/99	0.26	0.67	33/99	0.59	0.81	99/99
global	0.58	0.83	97/97	0.33	0.7	54/97	0.2	0.64	35/97	0.58	0.83	97/97
gr	0.56	0.82	99/99	0.2	0.64	17/99	0.24	0.6	8/99	0.56	0.82	99/99
gt	0.61	0.83	99/99	0.42	0.72	54/99	0.4	0.72	45/99	0.61	0.83	99/99
hk	0.57	0.81	99/99	0.37	0.75	15/99	0.35	0.75	10/99	0.57	0.81	99/99
hn	0.6	0.85	99/99	0.35	0.71	52/99	0.35	0.71	42/99	0.6	0.85	99/99
hu	0.52	0.81	99/99	0.31	0.68	10/99	0.25	0.68	4/99	0.52	0.81	99/99
id	0.64	0.85	99/99	0.2	0.56	7/99	0.19	0.57	4/99	0.64	0.85	99/99
ie	0.57	0.82	99/99	0.29	0.69	35/99	0.29	0.69	34/99	0.57	0.82	99/99
il	0.56	0.83	51/51	0.17	0.6	5/51	0.24	0.62	4/51	0.56	0.83	51/51
is	0.57	0.83	99/99	0.49	0.7	1/99	0.49	0.7	1/99	0.57	0.83	99/99
it	0.59	0.83	99/99	-	-	0/99	-	-	0/99	0.59	0.83	99/99
jp	0.46	0.75	98/99	-	-	0/99	-	-	0/99	0.46	0.75	98/99
lt	0.5	0.8	99/99	0.17	0.59	16/99	0.21	0.62	4/99	0.5	0.8	99/99
lu	0.47	0.8	99/99	0.37	0.77	3/99	0.13	0.6	4/99	0.47	0.8	99/99

lv	0.45	0.79	99/99	0.16	0.56	5/99	0.33	0.7	1/99	0.45	0.79	99/99
mt	0.53	0.81	99/99	-	-	0/99	-0.1	0.3	1/99	0.53	0.81	99/99
mx	0.66	0.87	99/99	0.37	0.72	29/99	0.37	0.7	26/99	0.66	0.87	99/99
my	0.63	0.85	99/99	0.25	0.64	30/99	0.2	0.6	20/99	0.63	0.85	99/99
ni	0.62	0.86	99/99	0.3	0.7	34/99	0.24	0.66	33/99	0.62	0.86	99/99
nl	0.59	0.81	99/99	0.23	0.6	1/99	0.25	0.6	1/99	0.59	0.81	99/99
no	0.61	0.84	99/99	0.24	0.7	3/99	0.27	0.6	2/99	0.61	0.84	99/99
nz	0.58	0.82	99/99	0.31	0.69	41/99	0.3	0.69	37/99	0.58	0.82	99/99
pa	0.59	0.84	99/99	0.25	0.64	31/99	0.28	0.67	21/99	0.59	0.84	99/99
pe	0.65	0.88	99/99	0.35	0.71	46/99	0.36	0.72	39/99	0.65	0.88	99/99
ph	0.6	0.84	99/99	0.08	0.4	2/99	-	-	0/99	0.6	0.84	99/99
pl	0.55	0.8	99/99	-	-	0/99	-	-	0/99	0.55	0.8	99/99
pt	0.6	0.84	99/99	0.27	0.7	3/99	0.23	0.7	2/99	0.6	0.84	99/99
py	0.61	0.85	99/99	0.36	0.7	35/99	0.32	0.7	34/99	0.61	0.85	99/99
ro	0.59	0.84	52/52	0.16	0.6	10/52	0.19	0.62	5/52	0.59	0.84	52/52
se	0.62	0.84	99/99	-	-	0/99	-	-	0/99	0.62	0.84	99/99
sg	0.6	0.82	99/99	0.29	0.67	31/99	0.31	0.69	21/99	0.6	0.82	99/99
sk	0.53	0.82	99/99	0.25	0.66	8/99	0.22	0.63	6/99	0.53	0.82	99/99
sv	0.6	0.85	99/99	0.36	0.69	62/99	0.35	0.7	49/99	0.6	0.85	99/99
th	0.64	0.86	71/71	-	-	0/71	-	-	0/71	0.64	0.86	71/71
tr	0.64	0.85	99/99	-	-	0/99	-	-	0/99	0.64	0.85	99/99
tw	0.53	0.81	99/99	0.07	0.5	1/99	-	-	0/99	0.53	0.81	99/99
us	0.56	0.81	99/99	0.4	0.73	65/99	0.27	0.67	51/99	0.55	0.8	99/99
uy	0.58	0.83	99/99	0.35	0.73	39/99	0.35	0.73	33/99	0.58	0.83	99/99
vn	0.62	0.87	52/52	-	-	0/52	-	-	0/52	0.62	0.87	52/52
avg	0.58	0.83	1.0	0.32	0.69	0.21	0.31	0.68	0.16	0.58	0.83	1.0

Table 6.13: Results with directly copied rankings as replacement on the Spotify viral daily data set for top-10 prediction.

With the aggregated methods shown in Table 6.14 the weighted versions performed for once better than the others. This can probably be attributed to fewer possible rankings and therefore not too many much worse rankings that are added to the aggregation. The trimmed versions performed again a little bit better.

	WCR			TWCR			TCR			CR		
voter	tau-a	i-size	sub	tau-a	i-size	sub	tau-a	i-size	sub	tau-a	i-size	sub
ar	0.39	0.75	99/99	0.49	0.8	99/99	0.5	0.8	99/99	0.45	0.76	99/99
at	0.43	0.77	99/99	0.45	0.77	99/99	0.43	0.77	99/99	0.41	0.76	99/99
au	0.43	0.75	99/99	0.5	0.77	99/99	0.47	0.78	99/99	0.42	0.75	99/99
be	0.44	0.76	99/99	0.48	0.78	99/99	0.46	0.77	99/99	0.43	0.75	99/99
bg	0.43	0.79	99/99	0.49	0.82	99/99	0.46	0.81	99/99	0.41	0.77	99/99
bo	0.41	0.76	99/99	0.5	0.8	99/99	0.5	0.81	99/99	0.45	0.77	99/99
br	0.43	0.75	99/99	0.5	0.79	99/99	0.47	0.78	99/99	0.42	0.74	99/99
ca	0.45	0.76	99/99	0.53	0.8	99/99	0.53	0.8	99/99	0.46	0.77	99/99
ch	0.42	0.74	99/99	0.44	0.76	99/99	0.42	0.76	99/99	0.4	0.75	99/99
cl	0.41	0.74	99/99	0.53	0.81	99/99	0.52	0.81	99/99	0.45	0.75	99/99
co	0.45	0.74	99/99	0.52	0.78	99/99	0.52	0.8	99/99	0.47	0.76	99/99
cr	0.41	0.76	99/99	0.49	0.8	99/99	0.48	0.81	99/99	0.42	0.77	99/99
cz	0.44	0.75	99/99	0.48	0.77	99/99	0.46	0.76	99/99	0.43	0.75	99/99
de	0.41	0.74	99/99	0.51	0.79	99/99	0.46	0.78	99/99	0.39	0.73	99/99
dk	0.43	0.75	99/99	0.49	0.79	99/99	0.46	0.79	99/99	0.42	0.75	99/99
do	0.41	0.74	99/99	0.48	0.77	99/99	0.47	0.78	99/99	0.43	0.75	99/99
ec	0.43	0.75	99/99	0.53	0.81	99/99	0.53	0.82	99/99	0.48	0.77	99/99
ee	0.44	0.77	99/99	0.47	0.78	99/99	0.44	0.78	99/99	0.43	0.76	99/99
es	0.42	0.76	99/99	0.5	0.8	99/99	0.47	0.79	99/99	0.42	0.75	99/99
fi	0.44	0.75	98/99	0.47	0.77	98/99	0.45	0.76	98/99	0.43	0.75	98/99
fr	0.44	0.76	99/99	0.48	0.78	99/99	0.45	0.77	99/99	0.42	0.74	99/99
gb	0.4	0.73	99/99	0.49	0.77	99/99	0.48	0.78	99/99	0.41	0.74	99/99
global	0.38	0.72	97/97	0.49	0.8	97/97	0.46	0.79	97/97	0.37	0.71	97/97
gr	0.44	0.76	99/99	0.5	0.79	99/99	0.46	0.79	99/99	0.41	0.75	99/99
gt	0.41	0.74	99/99	0.49	0.78	99/99	0.5	0.79	99/99	0.45	0.75	99/99
hk	0.46	0.77	99/99	0.51	0.79	99/99	0.47	0.79	99/99	0.43	0.76	99/99
hn	0.42	0.77	99/99	0.49	0.8	99/99	0.49	0.81	99/99	0.45	0.78	99/99
hu	0.44	0.77	99/99	0.48	0.79	99/99	0.44	0.79	99/99	0.42	0.76	99/99
id	0.45	0.76	99/99	0.54	0.81	99/99	0.5	0.81	99/99	0.44	0.75	99/99

ie	0.44	0.76	99/99	0.48	0.78	99/99	0.48	0.78	99/99	0.47	0.77	99/99
il	0.44	0.77	51/51	0.5	0.8	51/51	0.47	0.78	51/51	0.41	0.75	51/51
is	0.5	0.79	99/99	0.53	0.8	99/99	0.51	0.8	99/99	0.48	0.78	99/99
it	0.46	0.78	99/99	0.52	0.8	99/99	0.49	0.8	99/99	0.44	0.77	99/99
jp	0.38	0.71	98/99	0.41	0.73	98/99	0.39	0.72	98/99	0.36	0.7	98/99
lt	0.42	0.75	99/99	0.46	0.77	99/99	0.42	0.77	99/99	0.4	0.75	99/99
lu	0.41	0.76	99/99	0.43	0.78	99/99	0.41	0.77	99/99	0.4	0.76	99/99
lv	0.38	0.75	99/99	0.41	0.77	99/99	0.39	0.77	99/99	0.36	0.74	99/99
mt	0.41	0.74	99/99	0.46	0.78	99/99	0.43	0.77	99/99	0.39	0.73	99/99
mx	0.42	0.76	99/99	0.52	0.82	99/99	0.51	0.82	99/99	0.44	0.77	99/99
my	0.42	0.76	99/99	0.52	0.82	99/99	0.48	0.81	99/99	0.4	0.75	99/99
ni	0.43	0.76	99/99	0.49	0.8	99/99	0.49	0.81	99/99	0.45	0.77	99/99
nl	0.46	0.77	99/99	0.52	0.8	99/99	0.49	0.8	99/99	0.45	0.77	99/99
no	0.42	0.74	99/99	0.51	0.79	99/99	0.47	0.79	99/99	0.41	0.74	99/99
nz	0.44	0.74	99/99	0.5	0.77	99/99	0.49	0.77	99/99	0.45	0.75	99/99
pa	0.43	0.76	99/99	0.49	0.8	99/99	0.49	0.81	99/99	0.44	0.77	99/99
pe	0.42	0.75	99/99	0.51	0.81	99/99	0.53	0.83	99/99	0.47	0.77	99/99
ph	0.44	0.75	99/99	0.52	0.81	99/99	0.48	0.8	99/99	0.42	0.75	99/99
pl	0.43	0.74	99/99	0.47	0.77	99/99	0.44	0.76	99/99	0.41	0.74	99/99
pt	0.46	0.76	99/99	0.53	0.8	99/99	0.49	0.79	99/99	0.43	0.74	99/99
py	0.45	0.77	99/99	0.51	0.8	99/99	0.5	0.81	99/99	0.46	0.78	99/99
ro	0.45	0.77	52/52	0.52	0.79	52/52	0.48	0.79	52/52	0.42	0.75	52/52
se	0.44	0.76	99/99	0.52	0.8	99/99	0.48	0.79	99/99	0.42	0.74	99/99
sg	0.43	0.75	99/99	0.52	0.79	99/99	0.49	0.8	99/99	0.42	0.74	99/99
sk	0.46	0.78	99/99	0.48	0.79	99/99	0.46	0.78	99/99	0.44	0.77	99/99
sv	0.4	0.76	99/99	0.5	0.8	99/99	0.52	0.8	99/99	0.46	0.77	99/99
th	0.5	0.78	71/71	0.56	0.82	71/71	0.53	0.82	71/71	0.48	0.78	71/71
tr	0.49	0.78	99/99	0.54	0.82	99/99	0.51	0.81	99/99	0.47	0.76	99/99
tw	0.45	0.76	99/99	0.49	0.78	99/99	0.47	0.77	99/99	0.44	0.76	99/99
us	0.4	0.74	99/99	0.46	0.76	99/99	0.44	0.76	99/99	0.38	0.73	99/99
uy	0.41	0.75	99/99	0.48	0.79	99/99	0.47	0.81	99/99	0.43	0.77	99/99
vn	0.48	0.79	52/52	0.52	0.83	52/52	0.48	0.83	52/52	0.45	0.79	52/52
avg	0.43	0.76	1.0	0.5	0.79	1.0	0.47	0.79	1.0	0.43	0.75	1.0

Table 6.14: Results with aggregated rankings as replacement on the Spotify viral daily data set for top-10 prediction.

The following two tables show how few rankings could be replaced by DVR and PDVR for the full prediction experiments. Otherwise nothing really surprising can be seen. But Maximin shares the base case this time with the other aggregation methods and the increase through the implicit delegation is still the lowest between them. This could further suggest that the methods work a bit worse for Maximin.

	default	PR	DVR	PDVR	HSR	CR	WCR
Gain	0.0	1.0	0.04	0.03	1.0	1.0	1.0
Borda	0.6	0.83	0.6	0.6	0.83	0.78	0.79
Nanson	0.6	0.83	0.6	0.6	0.83	0.78	0.79
Copeland	0.6	0.81	0.59	0.6	0.81	0.77	0.78
Maximin	0.6	0.78	0.6	0.6	0.78	0.73	0.74

Table 6.15: Average Kendall tau-a scores after implicit delegation of multiple rankings for the Spotify viral daily data set.

	default	PR	DVR	PDVR	HSR	CR	WCR
Gain	0.0	1.0	0.09	0.1	1.0	1.0	1.0
Borda	0.6	0.83	0.6	0.6	0.83	0.78	0.79
Nanson	0.6	0.83	0.6	0.6	0.83	0.78	0.79
Copeland	0.6	0.81	0.59	0.59	0.81	0.77	0.78
Maximin	0.6	0.78	0.6	0.6	0.78	0.73	0.74

Table 6.16: Average Kendall tau-a scores of successful implicit delegation attempts of multiple rankings for the Spotify viral daily data set.

6.3.5 Spotify Viral Weekly Data Set

Spotify viral weekly charts is the smallest of the Spotify data sets and the experiments chose randomly 28 and 16 data points for top-10 and full prediction.

As the analysis already showed this data set had no high likelihood for any of the implicit delegation methods to work well. This is good reflected in Table 6.17. All methods were only able to substitute a small percentage of the requested rankings. Neither a high similarity to past rankings nor a high similarity to different voters could be seen. This probably lead to the selection of mostly rankings with a high average intersection size, which seem to have worked consistently over all methods, but their tau-a scores are not great.

The past analysis of this data set showed for all voters really low and mostly negative scores. When looking at a few that reached the highest scores in the overall analysis for past similarities it can be seen that they are also part of the few voters that could at least sometimes be substituted with PR. Some examples for this are "mx" and "bo".

	PR			DVR			PDVR			HSR		
voter	tau-a	i-size	sub	tau-a	i-size	sub	tau-a	i-size	sub	tau-a	i-size	sub
ad	-	-	0/23	-	-	0/23	-	-	0/23	-	-	0/23
ar	0.38	0.68	5/28	0.34	0.76	5/28	0.3	0.55	2/28	0.34	0.76	5/28
at	-	-	0/28	0.3	0.67	3/28	-	-	0/28	0.3	0.67	3/28
au	-	-	0/28	0.28	0.7	12/28	-	-	0/28	0.28	0.7	12/28
be	-	-	0/28	-	-	0/28	-	-	0/28	-	-	0/28
bg	-	-	0/28	-	-	0/28	-	-	0/28	-	-	0/28
bo	0.27	0.64	5/28	0.3	0.67	11/28	0.21	0.62	5/28	0.3	0.67	11/28
br	-0.34	0.3	1/28	-	-	0/28	-	-	0/28	-0.34	0.3	1/28
ca	-	-	0/28	0.33	0.69	15/28	-	-	0/28	0.33	0.69	15/28
ch	-	-	0/28	-	-	0/28	-	-	0/28	-	-	0/28
cl	0.19	0.65	6/28	-0.03	0.57	3/28	-	-	0/28	0.17	0.64	7/28
co	0.21	0.62	5/28	0.51	0.8	5/28	0.23	0.57	3/28	0.51	0.8	5/28
cr	0.28	0.68	4/28	0.29	0.68	4/28	0.09	0.5	1/28	0.2	0.62	5/28
cy	-	-	0/28	-	-	0/28	-	-	0/28	-	-	0/28
cz	-	-	0/28	-	-	0/28	-	-	0/28	-	-	0/28
de	-	-	0/28	0.3	0.67	3/28	-	-	0/28	0.3	0.67	3/28
dk	-0.12	0.4	1/28	-	-	0/28	-	-	0/28	-0.12	0.4	1/28
do	-	-	0/28	0.18	0.56	5/28	-	-	0/28	0.18	0.56	5/28
ec	0.29	0.64	5/28	0.3	0.7	10/28	0.19	0.6	4/28	0.3	0.7	10/28
ee	-	-	0/28	-	-	0/28	-	-	0/28	-	-	0/28
es	-0.08	0.55	2/28	-	-	0/28	-	-	0/28	-0.08	0.55	2/28
fi	-	-	0/28	-	-	0/28	-	-	0/28	-	-	0/28
fr	-	-	0/28	-	-	0/28	-	-	0/28	-	-	0/28

gb	0.17	0.6	3/28	0.29	0.68	9/28	-	-	0/28	0.29	0.68	9/28
global	-0.05	0.5	2/28	0.33	0.73	14/28	-	-	0/28	0.28	0.7	16/28
gr	-	-	0/28	-	-	0/28	-	-	0/28	-	-	0/28
gt	0.33	0.68	5/28	0.3	0.67	13/28	0.29	0.64	5/28	0.3	0.67	13/28
hk	-	-	0/28	0.14	0.55	2/28	-	-	0/28	0.14	0.55	2/28
hn	0.36	0.7	3/28	0.24	0.68	11/28	0.23	0.65	4/28	0.24	0.68	11/28
hu	-	-	0/28	-	-	0/28	-	-	0/28	-	-	0/28
id	-	-	0/28	-	-	0/28	-	-	0/28	-	-	0/28
ie	0.14	0.53	3/28	0.29	0.68	9/28	-	-	0/28	0.24	0.63	9/28
il	-	-	0/15	-	-	0/15	-	-	0/15	-	-	0/15
is	-	-	0/28	-	-	0/28	-	-	0/28	-	-	0/28
it	-0.16	0.5	1/28	-	-	0/28	-	-	0/28	-0.16	0.5	1/28
jp	-	-	0/28	-	-	0/28	-	-	0/28	-	-	0/28
lt	-	-	0/28	-	-	0/28	-	-	0/28	-	-	0/28
lu	-	-	0/28	-	-	0/28	-	-	0/28	-	-	0/28
lv	-	-	0/28	-	-	0/28	-	-	0/28	-	-	0/28
mc	-	-	0/7	-	-	0/7	-	-	0/7	-	-	0/7
mt	-	-	0/28	-	-	0/28	-	-	0/28	-	-	0/28
mx	0.31	0.7	5/28	0.28	0.72	5/28	-0.18	0.4	1/28	0.27	0.72	5/28
my	-	-	0/28	0.42	0.7	2/28	-	-	0/28	0.42	0.7	2/28
ni	0.29	0.6	2/28	0.27	0.72	6/28	-0.01	0.5	1/28	0.27	0.72	6/28
nl	-	-	0/28	-	-	0/28	-	-	0/28	-	-	0/28
no	-	-	0/28	-	-	0/28	-	-	0/28	-	-	0/28
nz	-	-	0/28	0.28	0.7	12/28	-	-	0/28	0.28	0.7	12/28
pa	0.14	0.5	1/28	0.16	0.57	4/28	0.1	0.7	1/28	0.16	0.57	4/28
pe	0.33	0.76	5/28	0.31	0.69	8/28	0.29	0.7	5/28	0.31	0.69	8/28
ph	0.11	0.55	4/28	-	-	0/28	-	-	0/28	0.11	0.55	4/28
pl	0.23	0.6	2/27	-	-	0/27	-	-	0/27	0.23	0.6	2/27
pt	-	-	0/28	-	-	0/28	-	-	0/28	-	-	0/28
py	0.15	0.64	5/28	0.06	0.65	6/28	-	-	0/28	0.03	0.6	6/28
ro	-	-	0/15	-	-	0/15	-	-	0/15	-	-	0/15
se	-	-	0/28	-	-	0/28	-	-	0/28	-	-	0/28
sg	-	-	0/28	0.28	0.62	4/28	-	-	0/28	0.28	0.62	4/28
sk	-	-	0/28	-	-	0/28	-	-	0/28	-	-	0/28
sv	0.29	0.64	5/28	0.25	0.69	14/28	0.24	0.7	5/28	0.25	0.69	14/28
th	-	-	0/22	-	-	0/22	-	-	0/22	-	-	0/22
tr	-	-	0/28	-	-	0/28	-	-	0/28	-	-	0/28
tw	-	-	0/28	-	-	0/28	-	-	0/28	-	-	0/28
us	-	-	0/28	0.31	0.71	20/28	-	-	0/28	0.31	0.71	20/28
uy	0.18	0.6	5/28	0.32	0.75	6/28	-	-	0/28	0.32	0.75	6/28
vn	-	-	0/15	-	-	0/15	-	-	0/15	-	-	0/15
avg	0.22	0.63	0.05	0.28	0.69	0.13	0.22	0.63	0.02	0.26	0.67	0.14

Table 6.17: Results with directly copied rankings as replacement on the Spotify viral weekly data set for top-10 prediction.

The aggregated substitutes seem to not have changed the results much, as they most likely did not aggregate many possible rankings together. From these results shown in Table 6.18 no real information to differentiate between the four methods can be gained.

	WCR			TWCR			TCR			CR		
voter	tau-a	i-size	sub	tau-a	i-size	sub	tau-a	i-size	sub	tau-a	i-size	sub
ad	-	-	0/23	-	-	0/23	-	-	0/23	-	-	0/23
ar	0.21	0.64	5/28	0.33	0.74	5/28	0.35	0.72	5/28	0.24	0.7	5/28
at	0.3	0.67	3/28	0.3	0.67	3/28	0.3	0.67	3/28	0.3	0.67	3/28
au	0.28	0.7	12/28	0.28	0.7	12/28	0.28	0.7	12/28	0.28	0.7	12/28
be	-	-	0/28	-	-	0/28	-	-	0/28	-	-	0/28
bg	-	-	0/28	-	-	0/28	-	-	0/28	-	-	0/28
bo	0.31	0.69	11/28	0.32	0.68	11/28	0.33	0.67	11/28	0.35	0.72	11/28
br	-0.34	0.3	1/28	-0.34	0.3	1/28	-0.34	0.3	1/28	-0.34	0.3	1/28
ca	0.34	0.71	15/28	0.32	0.66	15/28	0.32	0.66	15/28	0.34	0.71	15/28
ch	-	-	0/28	-	-	0/28	-	-	0/28	-	-	0/28
cl	0.1	0.6	7/28	0.11	0.6	7/28	0.12	0.63	7/28	0.1	0.6	7/28
co	0.35	0.76	5/28	0.46	0.8	5/28	0.47	0.78	5/28	0.24	0.68	5/28
cr	0.22	0.64	5/28	0.24	0.68	5/28	0.26	0.64	5/28	0.26	0.68	5/28
cy	-	-	0/28	-	-	0/28	-	-	0/28	-	-	0/28

cz	-	-	0/28	-	-	0/28	-	-	0/28	-	-	0/28
de	0.3	0.67	3/28	0.3	0.67	3/28	0.3	0.67	3/28	0.3	0.67	3/28
dk	-0.12	0.4	1/28	-0.12	0.4	1/28	-0.12	0.4	1/28	-0.12	0.4	1/28
do	0.26	0.66	5/28	0.29	0.7	5/28	0.31	0.68	5/28	0.25	0.66	5/28
ec	0.33	0.72	10/28	0.38	0.72	10/28	0.37	0.72	10/28	0.31	0.73	10/28
ee	-	-	0/28	-	-	0/28	-	-	0/28	-	-	0/28
es	-0.08	0.55	2/28	-0.08	0.55	2/28	-0.08	0.55	2/28	-0.08	0.55	2/28
fi	-	-	0/28	-	-	0/28	-	-	0/28	-	-	0/28
fr	-	-	0/28	-	-	0/28	-	-	0/28	-	-	0/28
gb	0.24	0.68	9/28	0.28	0.7	9/28	0.29	0.7	9/28	0.25	0.68	9/28
global	0.32	0.7	16/28	0.29	0.68	16/28	0.29	0.68	16/28	0.31	0.71	16/28
gr	-	-	0/28	-	-	0/28	-	-	0/28	-	-	0/28
gt	0.27	0.65	13/28	0.35	0.69	13/28	0.36	0.68	13/28	0.3	0.67	13/28
hk	0.14	0.55	2/28	0.14	0.55	2/28	0.14	0.55	2/28	0.14	0.55	2/28
hn	0.21	0.69	11/28	0.24	0.69	11/28	0.24	0.69	11/28	0.24	0.71	11/28
hu	-	-	0/28	-	-	0/28	-	-	0/28	-	-	0/28
id	-	-	0/28	-	-	0/28	-	-	0/28	-	-	0/28
ie	0.3	0.66	9/28	0.28	0.64	9/28	0.29	0.67	9/28	0.3	0.66	9/28
il	-	-	0/15	-	-	0/15	-	-	0/15	-	-	0/15
is	-	-	0/28	-	-	0/28	-	-	0/28	-	-	0/28
it	-0.16	0.5	1/28	-0.16	0.5	1/28	-0.16	0.5	1/28	-0.16	0.5	1/28
jp	-	-	0/28	-	-	0/28	-	-	0/28	-	-	0/28
lt	-	-	0/28	-	-	0/28	-	-	0/28	-	-	0/28
lu	-	-	0/28	-	-	0/28	-	-	0/28	-	-	0/28
lv	-	-	0/28	-	-	0/28	-	-	0/28	-	-	0/28
mc	-	-	0/7	-	-	0/7	-	-	0/7	-	-	0/7
mt	-	-	0/28	-	-	0/28	-	-	0/28	-	-	0/28
mx	0.23	0.64	5/28	0.34	0.72	5/28	0.41	0.72	5/28	0.28	0.72	5/28
my	0.42	0.7	2/28	0.42	0.7	2/28	0.42	0.7	2/28	0.42	0.7	2/28
ni	0.29	0.68	6/28	0.29	0.73	6/28	0.28	0.72	6/28	0.3	0.72	6/28
nl	-	-	0/28	-	-	0/28	-	-	0/28	-	-	0/28
no	-	-	0/28	-	-	0/28	-	-	0/28	-	-	0/28
nz	0.28	0.7	12/28	0.28	0.7	12/28	0.28	0.7	12/28	0.28	0.7	12/28
pa	0.21	0.62	4/28	0.21	0.65	4/28	0.24	0.62	4/28	0.24	0.62	4/28
pe	0.34	0.74	8/28	0.37	0.71	8/28	0.37	0.71	8/28	0.34	0.74	8/28
ph	0.11	0.55	4/28	0.11	0.55	4/28	0.11	0.55	4/28	0.11	0.55	4/28
pl	0.23	0.6	2/27	0.23	0.6	2/27	0.23	0.6	2/27	0.23	0.6	2/27
pt	-	-	0/28	-	-	0/28	-	-	0/28	-	-	0/28
py	0.19	0.67	6/28	0.21	0.73	6/28	0.26	0.72	6/28	0.17	0.62	6/28
ro	-	-	0/15	-	-	0/15	-	-	0/15	-	-	0/15
se	-	-	0/28	-	-	0/28	-	-	0/28	-	-	0/28
sg	0.28	0.62	4/28	0.28	0.62	4/28	0.28	0.62	4/28	0.28	0.62	4/28
sk	-	-	0/28	-	-	0/28	-	-	0/28	-	-	0/28
sv	0.26	0.69	14/28	0.3	0.72	14/28	0.3	0.7	14/28	0.25	0.71	14/28
th	-	-	0/22	-	-	0/22	-	-	0/22	-	-	0/22
tr	-	-	0/28	-	-	0/28	-	-	0/28	-	-	0/28
tw	-	-	0/28	-	-	0/28	-	-	0/28	-	-	0/28
us	0.32	0.7	20/28	0.31	0.69	20/28	0.3	0.7	20/28	0.32	0.71	20/28
uy	0.2	0.7	6/28	0.29	0.77	6/28	0.28	0.73	6/28	0.21	0.7	6/28
vn	-	-	0/15	-	-	0/15	-	-	0/15	-	-	0/15
avg	0.26	0.68	0.14	0.28	0.68	0.14	0.29	0.68	0.14	0.27	0.68	0.14

Table 6.18: Results with aggregated rankings as replacement on the Spotify viral weekly data set for top-10 prediction.

The experiments for the full prediction did also not produce many results. Table 6.19 shows that the implicit delegation attempts did barely change anything overall. When looking at only the successful attempts in Table 6.20 it can be seen that PDVR could not replace a single ranking. And also that only Maximin was noticeable influenced by the implicit delegation, but more negative than positive. The results with PR seem to have worsened the similarity to the full profile rather strong when using Maximin.

	default	PR	DVR	PDVR	HSR	CR	WCR
Gain	0.0	0.01	0.03	0.0	0.03	0.03	0.03
Borda	0.59	0.59	0.59	0.59	0.59	0.59	0.59
Nanson	0.59	0.59	0.59	0.59	0.59	0.59	0.59
Copeland	0.58	0.58	0.58	0.58	0.58	0.58	0.58
Maximin	0.56	0.56	0.56	0.56	0.56	0.56	0.56

Table 6.19: Average Kendall tau-a scores after implicit delegation of multiple rankings for the Spotify viral weekly data set.

	default	PR	DVR	PDVR	HSR	CR	WCR
Gain	0.0	0.1	0.07	-	0.09	0.09	0.09
Borda	0.59	0.59	0.59	-	0.59	0.59	0.59
Nanson	0.59	0.59	0.59	-	0.59	0.59	0.59
Copeland	0.58	0.58	0.58	-	0.58	0.58	0.58
Maximin	0.56	0.44	0.56	-	0.54	0.55	0.55

Table 6.20: Average Kendall tau-a scores of successful implicit delegation attempts of multiple rankings for the Spotify viral weekly data set.

6.3.6 Free I-Phone Games Data Set

Now to the I-Phone App store data sets. These are smaller than the Spotify ones and contain the least voters between all the data sets used for this thesis. Because of this it was decided to use all but the last 10 data points for the top-10 prediction experiments for all I-Phone data sets. This results in 52 profiles each. The full prediction experiments still used random selection to decide on the data points where the experiments are performed on. In the case of free games 15 data points were selected. As every data point corresponds to 9 individual implicit delegations a total of 135 experiments were conducted.

Not surprisingly the analysis of the data helped predict the outcome really good. Table 6.21 contains the direct copy methods where PR got good scores and the different voter replacement methods only have acceptable but not bad or negative results. And like the correlation tables showed "jp" has nearly no similarity to the other voters and therefore DVR and PDVR could not find a single replacement for this voter.

voter	PR tau-a	PR i-size	PR sub	DVR tau-a	DVR i-size	DVR sub	PDVR tau-a	PDVR i-size	PDVR sub	HSR tau-a	HSR i-size	HSR sub
au	0.49	0.78	48/52	0.42	0.78	43/52	0.39	0.76	41/52	0.49	0.78	50/52
ca	0.57	0.81	50/52	0.4	0.79	47/52	0.32	0.75	46/52	0.56	0.8	52/52
de	0.69	0.88	48/50	0.31	0.7	50/50	0.24	0.69	47/50	0.69	0.88	50/50
fr	0.71	0.88	47/49	0.33	0.69	39/49	0.25	0.66	36/49	0.7	0.88	49/49
gb	0.75	0.91	48/50	0.3	0.74	50/50	0.3	0.75	47/50	0.74	0.91	50/50
it	0.74	0.89	50/52	0.27	0.72	31/52	0.25	0.69	33/52	0.74	0.89	50/52
jp	0.58	0.85	50/52	-	-	0/52	-	-	0/52	0.58	0.85	50/52
pl	0.69	0.89	47/49	0.27	0.71	28/49	0.23	0.67	31/49	0.69	0.89	47/49
ru	0.7	0.88	48/50	0.21	0.73	23/50	0.18	0.65	33/50	0.7	0.88	48/50
ua	0.73	0.88	45/47	0.22	0.73	15/47	0.2	0.7	13/47	0.73	0.88	45/47
us	0.71	0.9	50/52	0.3	0.74	52/52	0.24	0.72	49/52	0.71	0.9	52/52
avg	0.67	0.87	0.96	0.32	0.74	0.68	0.27	0.71	0.68	0.67	0.87	0.98

Table 6.21: Results with directly copied rankings as replacement on the free I-Phone games data set for top-10 prediction.

The methods in Table 6.22 followed again the pattern of trimmed versions perfom better than non trimmed ones and not weighted methods better than their weighted counterpart, but all results are close to each other and show no improvement over simply copying the rankings.

voter	WCR tau-a	WCR i-size	WCR sub	TWCR tau-a	TWCR i-size	TWCR sub	TCR tau-a	TCR i-size	TCR sub	CR tau-a	CR i-size	CR sub
au	0.32	0.72	50/52	0.4	0.78	50/52	0.48	0.79	50/52	0.44	0.76	50/52
ca	0.4	0.74	52/52	0.48	0.78	52/52	0.54	0.79	52/52	0.52	0.8	52/52
de	0.36	0.75	50/50	0.47	0.82	50/50	0.53	0.84	50/50	0.46	0.79	50/50
fr	0.41	0.76	49/49	0.51	0.81	49/49	0.56	0.83	49/49	0.49	0.8	49/49
gb	0.37	0.75	50/50	0.49	0.83	50/50	0.55	0.86	50/50	0.5	0.81	50/50
it	0.44	0.79	50/52	0.52	0.81	50/52	0.54	0.83	50/52	0.49	0.8	50/52
jp	0.45	0.79	50/52	0.51	0.83	50/52	0.45	0.82	50/52	0.41	0.78	50/52
pl	0.39	0.79	47/49	0.47	0.84	47/49	0.49	0.85	47/49	0.41	0.79	47/49
ru	0.49	0.79	48/50	0.58	0.84	48/50	0.54	0.84	48/50	0.51	0.82	48/50
ua	0.45	0.77	45/47	0.54	0.83	45/47	0.53	0.82	45/47	0.48	0.77	45/47
us	0.43	0.8	52/52	0.52	0.84	52/52	0.54	0.85	52/52	0.51	0.83	52/52
avg	0.41	0.77	0.98	0.5	0.82	0.98	0.52	0.83	0.98	0.48	0.8	0.98

Table 6.22: Results with aggregated rankings as replacement on the free I-Phone games data set for top-10 prediction.

The full prediction results in the following tables have again no real conclusive differences between the aggregation methods except that Borda seems to be effected slightly better by the implicit delegation methods that use Borda aggregation itself than the other preference aggregations. Here it could be important to mention that compared to the previous data sets far less voters were missing in these experiments as overall only 11 are available and a third of this are 3 or 4.

	default	PR	DVR	PDVR	HSR	CR	WCR
Gain	0.0	1.0	0.41	0.41	1.0	1.0	1.0
Borda	0.74	0.86	0.73	0.73	0.86	0.81	0.82
Nanson	0.74	0.86	0.72	0.73	0.86	0.8	0.81

Copeland	0.74	0.86	0.72	0.73	0.86	0.79	0.8
Maximin	0.73	0.85	0.73	0.72	0.85	0.76	0.78
Schulze	0.73	0.85	0.72	0.72	0.85	0.77	0.78

Table 6.23: Average Kendall tau-a scores after implicit delegation of multiple rankings for the free I-Phone games data set.

	default	PR	DVR	PDVR	HSR	CR	WCR
Gain	0.0	1.0	0.51	0.52	1.0	1.0	1.0
Borda	0.74	0.86	0.72	0.73	0.86	0.81	0.82
Nanson	0.74	0.86	0.72	0.72	0.86	0.8	0.81
Copeland	0.74	0.86	0.71	0.72	0.86	0.79	0.8
Maximin	0.73	0.85	0.73	0.72	0.85	0.76	0.78
Schulze	0.73	0.85	0.72	0.72	0.85	0.77	0.78

Table 6.24: Average Kendall tau-a scores of successful implicit delegation attempts of multiple rankings for the free I-Phone games data set.

6.3.7 Paid I-Phone Games Data Set

This seems to be the first data set where the analysis did not help to predict everything. The top-10 seem to have a higher similarity between the voters and data points than the overall rankings which lead to the results in Table 6.25. Like with most of the data sets PR produced the best results which are then picked up by HSR for similar results with more success rate. The interesting part is that DVR and PDVR produced higher tau-a scores than for free games despite the analysis seeing less similarities. Therefore it could prove useful to adjust the analysis to the required parts, meaning if only top-k are relevant, then only analyzing the similarity between the top-k and not the full rankings should be more accurate.

But the analysis nevertheless provided good indications of the results. For example "jp", "ua" and "ru" had the lowest tau-b scores for similarities with other voters and this resulted in the implicit delegation method DVR to fail in finding any substitute for those three voters.

	PR			DVR			PDVR			HSR		
voter	tau-a	i-size	sub	tau-a	i-size	sub	tau-a	i-size	sub	tau-a	i-size	sub
au	0.68	0.86	50/52	0.55	0.79	52/52	0.54	0.8	50/52	0.68	0.86	52/52
ca	0.72	0.88	50/52	0.55	0.78	52/52	0.55	0.76	50/52	0.71	0.87	52/52
de	0.67	0.87	48/50	0.24	0.61	15/50	0.23	0.58	12/50	0.65	0.87	50/50
fr	0.72	0.85	47/49	0.35	0.67	49/49	0.32	0.66	47/49	0.71	0.84	49/49
gb	0.73	0.89	48/50	0.23	0.62	17/50	0.22	0.62	15/50	0.71	0.88	50/50
it	0.61	0.84	50/52	0.35	0.67	49/52	0.32	0.66	50/52	0.6	0.83	52/52
jp	0.75	0.9	50/52	-	-	0/52	-	-	0/52	0.75	0.9	50/52
pl	0.51	0.76	47/49	0.12	0.5	1/49	0.13	0.6	2/49	0.51	0.76	47/49
ru	0.82	0.91	48/50	-	-	0/50	-	-	0/50	0.82	0.91	48/50
ua	0.46	0.7	43/47	-	-	0/47	0.16	0.5	2/47	0.46	0.7	43/47

us	0.85	0.92	50/52	0.56	0.77	52/52	0.54	0.76	50/52	0.84	0.92	52/52
avg	0.69	0.85	0.96	0.45	0.72	0.52	0.43	0.71	0.5	0.68	0.85	0.98

Table 6.25: Results with directly copied rankings as replacement on the paid I-Phone games data set for top-10 prediction.

As seen in Table 6.26 the weighted versions have again lower average scores. This time TWCR is worse than CR, which would indicate a higher number of possible rankings with low score.

	WCR			TWCR			TCR			CR		
voter	tau-a	i-size	sub	tau-a	i-size	sub	tau-a	i-size	sub	tau-a	i-size	sub
au	0.6	0.83	52/52	0.62	0.84	52/52	0.66	0.85	52/52	0.67	0.85	52/52
ca	0.67	0.86	52/52	0.66	0.83	52/52	0.7	0.87	52/52	0.71	0.88	52/52
de	0.42	0.78	50/50	0.49	0.81	50/50	0.53	0.84	50/50	0.48	0.8	50/50
fr	0.5	0.76	49/49	0.55	0.78	49/49	0.57	0.79	49/49	0.56	0.78	49/49
gb	0.48	0.79	50/50	0.56	0.82	50/50	0.61	0.87	50/50	0.58	0.85	50/50
it	0.39	0.75	52/52	0.48	0.8	52/52	0.53	0.8	52/52	0.45	0.76	52/52
jp	0.5	0.81	50/52	0.54	0.83	50/52	0.51	0.83	50/52	0.49	0.81	50/52
pl	0.43	0.73	47/49	0.46	0.74	47/49	0.42	0.73	47/49	0.4	0.71	47/49
ru	0.73	0.9	48/50	0.72	0.88	48/50	0.74	0.89	48/50	0.74	0.9	48/50
ua	0.43	0.67	43/47	0.45	0.69	43/47	0.44	0.69	43/47	0.43	0.67	43/47
us	0.73	0.83	52/52	0.74	0.85	52/52	0.8	0.9	52/52	0.81	0.9	52/52
avg	0.54	0.79	0.98	0.57	0.81	0.98	0.6	0.83	0.98	0.58	0.81	0.98

Table 6.26: Results with aggregated rankings as replacement on the paid I-Phone games data set for top-10 prediction.

For the full prediction experiments 15 random data points were selected. The results here are more in line with the analysis from Section 4.3. All methods have pretty low effects on the similarity and DVR and PDVR were not often able to replace rankings.

	default	PR	DVR	PDVR	HSR	CR	WCR
Gain	0.0	0.85	0.08	0.12	0.85	0.85	0.85
Borda	0.66	0.78	0.65	0.66	0.78	0.75	0.75
Nanson	0.66	0.78	0.65	0.65	0.78	0.75	0.75
Copeland	0.66	0.75	0.65	0.66	0.75	0.73	0.73
Maximin	0.68	0.76	0.68	0.68	0.76	0.74	0.74
Schulze	0.66	0.73	0.65	0.66	0.73	0.71	0.71

Table 6.27: Average Kendall tau-a scores after implicit delegation of multiple rankings for the paid I-Phone games data set.

	default	PR	DVR	PDVR	HSR	CR	WCR
Gain	0.0	0.85	0.31	0.35	0.85	0.85	0.85
Borda	0.66	0.78	0.65	0.65	0.78	0.75	0.75

Nanson	0.66	0.78	0.65	0.65	0.78	0.75	0.75
Copeland	0.66	0.75	0.64	0.66	0.75	0.73	0.73
Maximin	0.68	0.76	0.68	0.67	0.76	0.74	0.74
Schulze	0.66	0.73	0.65	0.65	0.73	0.71	0.71

Table 6.28: Average Kendall tau-a scores of successful implicit delegation attempts of multiple rankings for the paid I-Phone games data set.

6.3.8 Top Grossing I-Phone Games Data Set

Top grossing games were a bit more promising in the overall analysis than free games, but this seems to be another example for a data set, where the top-10 similarities are not perfectly described by the full analysis. All three search methods performed worse than on the free games data set. But the general range for the resulting scores in Table 6.29 is not far from what was expected. For example "us" and "ca" showed high similarity to each other and DVR found for both of them always a substitute. But "gb" also looked like it could easily have some similar substitutes and DVR could not even replace this voter a single time.

	PR			DVR			PDVR			HSR		
voter	tau-a	i-size	sub	tau-a	i-size	sub	tau-a	i-size	sub	tau-a	i-size	sub
au	0.64	0.85	50/52	-	-	0/52	-	-	0/52	0.64	0.85	50/52
ca	0.35	0.76	50/52	0.3	0.7	52/52	0.14	0.64	15/52	0.34	0.76	52/52
de	0.5	0.88	48/50	0.31	0.71	43/50	0.23	0.69	35/50	0.49	0.87	50/50
fr	0.62	0.86	47/49	0.3	0.67	49/49	0.19	0.67	43/49	0.61	0.86	49/49
gb	0.65	0.83	48/50	-	-	0/50	-	-	0/50	0.65	0.83	48/50
it	0.54	0.86	50/52	0.29	0.69	50/52	0.22	0.65	40/52	0.53	0.85	52/52
jp	0.44	0.76	50/52	-	-	0/52	-	-	0/52	0.44	0.76	50/52
pl	0.48	0.82	47/49	-	-	0/49	-	-	0/49	0.48	0.82	47/49
ru	0.61	0.84	48/50	-	-	0/50	-0.14	0.6	1/50	0.61	0.84	48/50
ua	0.51	0.81	45/47	-	-	0/47	-	-	0/47	0.51	0.81	45/47
us	0.52	0.82	50/52	0.3	0.7	52/52	0.14	0.65	17/52	0.51	0.82	52/52
avg	0.53	0.83	0.96	0.3	0.69	0.44	0.19	0.66	0.27	0.53	0.82	0.98

Table 6.29: Results with directly copied rankings as replacement on the top grossing I-Phone games data set for top-10 prediction.

Table 6.30 shows the results of the methods that use the Borda aggregation. This time no real difference between them could be observed.

	WCR			TWCR			TCR			CR		
voter	tau-a	i-size	sub	tau-a	i-size	sub	tau-a	i-size	sub	tau-a	i-size	sub
au	0.56	0.83	50/52	0.56	0.83	50/52	0.57	0.83	50/52	0.58	0.84	50/52
ca	0.36	0.76	52/52	0.39	0.78	52/52	0.38	0.78	52/52	0.35	0.76	52/52
de	0.43	0.82	50/50	0.45	0.8	50/50	0.45	0.81	50/50	0.45	0.83	50/50
fr	0.46	0.79	49/49	0.51	0.81	49/49	0.51	0.83	49/49	0.48	0.8	49/49
gb	0.61	0.83	48/50	0.6	0.81	48/50	0.61	0.83	48/50	0.61	0.83	48/50
it	0.46	0.83	52/52	0.46	0.81	52/52	0.46	0.82	52/52	0.45	0.83	52/52
jp	0.4	0.73	50/52	0.42	0.74	50/52	0.4	0.74	50/52	0.39	0.74	50/52
pl	0.38	0.77	47/49	0.4	0.76	47/49	0.36	0.76	47/49	0.35	0.76	47/49
ru	0.48	0.8	48/50	0.48	0.79	48/50	0.49	0.79	48/50	0.48	0.8	48/50
ua	0.42	0.78	45/47	0.41	0.78	45/47	0.39	0.77	45/47	0.4	0.79	45/47
us	0.48	0.81	52/52	0.49	0.79	52/52	0.49	0.8	52/52	0.49	0.8	52/52

avg	0.46	0.8	0.98	0.47	0.79	0.98	0.46	0.8	0.98	0.46	0.8	0.98

Table 6.30: Results with aggregated rankings as replacement on the top grossing I-Phone games data set for top-10 prediction.

The full prediction was performed on 15 randomly selected data points and the results are shown in the tables 6.31 and 6.32. Here Maximin got the lowest tau-a score when comparing the full profile with one where the missing voters are removed (default case). However the methods with positive effect like PR have the highest improvement with Maximin, but still a lower overall score than the other aggregation methods.

	default	PR	DVR	PDVR	HSR	CR	WCR
Gain	0.0	1.0	0.48	0.54	1.0	1.0	1.0
Borda	0.73	0.86	0.72	0.73	0.86	0.84	0.84
Nanson	0.73	0.86	0.72	0.73	0.86	0.84	0.84
Copeland	0.74	0.85	0.73	0.75	0.85	0.83	0.83
Maximin	0.65	0.83	0.65	0.66	0.83	0.8	0.81
Schulze	0.74	0.84	0.74	0.76	0.84	0.82	0.82

Table 6.31: Average Kendall tau-a scores after implicit delegation of multiple rankings for the top grossing I-Phone games data set.

	default	PR	DVR	PDVR	HSR	CR	WCR
Gain	0.0	1.0	0.53	0.57	1.0	1.0	1.0
Borda	0.73	0.86	0.72	0.73	0.86	0.84	0.84
Nanson	0.73	0.86	0.72	0.73	0.86	0.84	0.84
Copeland	0.74	0.85	0.73	0.75	0.85	0.83	0.83
Maximin	0.65	0.83	0.65	0.66	0.83	0.8	0.81
Schulze	0.74	0.84	0.74	0.76	0.84	0.82	0.82

Table 6.32: Average Kendall tau-a scores of successful implicit delegation attempts of multiple rankings for the top grossing I-Phone games data set.

6.3.9 Paid I-Phone News Data Set

The last data set is from the paid I-Phone News charts and there is again some difference between the analysis for similarities over the full rankings compared to the experiments with top-10 similarities. The method PR is not as good as the overall analysis could have implied but it is far from bad. A bit surprisingly DVR and PDVR were not able to substitute even a single ranking. Because of this HSR is identical to PR. This data set is probably the best example to show that the overall analysis can show extremely high similarities like in the case of "au" with 0.9 but sill produce a worse tau-a score than another voter which showed a far lower promise like "us" with a tau-b score of 0.58. However the intersection sizes are still extremely high overall, which means the statistics were not useless in indicating the results.

voter	PR			DVR			PDVR			HSR		
	tau-a	i-size	sub	tau-a	i-size	sub	tau-a	i-size	sub	tau-a	i-size	sub
au	0.43	0.82	50/52	-	-	0/52	-	-	0/52	0.43	0.82	50/52
ca	0.5	0.86	50/52	-	-	0/52	-	-	0/52	0.5	0.86	50/52
de	0.51	0.85	46/49	-	-	0/49	-	-	0/49	0.51	0.85	46/49
fr	0.76	0.95	48/50	-	-	0/50	-	-	0/50	0.76	0.95	48/50
gb	0.54	0.84	45/49	-	-	0/49	-	-	0/49	0.54	0.84	45/49
it	0.65	0.91	47/49	-	-	0/49	-	-	0/49	0.65	0.91	47/49
jp	0.57	0.91	50/52	-	-	0/52	-	-	0/52	0.57	0.91	50/52
pl	0.85	0.96	45/47	-	-	0/47	-	-	0/47	0.85	0.96	45/47
ru	0.85	0.96	48/50	-	-	0/50	-	-	0/50	0.85	0.96	48/50
ua	0.92	0.98	43/45	-	-	0/45	-	-	0/45	0.92	0.98	43/45
us	0.46	0.66	50/52	-	-	0/52	-	-	0/52	0.46	0.66	50/52
avg	0.64	0.88	0.95	-	-	0.0	-	-	0.0	0.64	0.88	0.95

Table 6.33: Results with directly copied rankings as replacement on the paid I-Phone news data set for top-10 prediction.

The results in Table 6.34 are a bit of an outlier, as the weighted methods outperform the non weighted ones. But the trimmed versions still performed better than the not trimmed ones. It could be that the weights worked better in this case because all the input rankings are from the same source with the method PR. However the difference is still so low that no extreme benefit stems from one over the other and the copy methods still work the best.

voter	WCR			TWCR			TCR			CR		
	tau-a	i-size	sub	tau-a	i-size	sub	tau-a	i-size	sub	tau-a	i-size	sub
au	0.37	0.8	50/52	0.39	0.81	50/52	0.36	0.8	50/52	0.34	0.79	50/52
ca	0.43	0.83	50/52	0.45	0.84	50/52	0.41	0.84	50/52	0.41	0.82	50/52
de	0.42	0.8	46/49	0.44	0.83	46/49	0.4	0.82	46/49	0.4	0.79	46/49
fr	0.56	0.92	48/50	0.59	0.88	48/50	0.6	0.92	48/50	0.57	0.93	48/50
gb	0.45	0.81	45/49	0.5	0.82	45/49	0.45	0.81	45/49	0.43	0.81	45/49
it	0.48	0.87	47/49	0.51	0.86	47/49	0.46	0.84	47/49	0.47	0.86	47/49
jp	0.48	0.89	50/52	0.52	0.88	50/52	0.46	0.87	50/52	0.44	0.88	50/52
pl	0.62	0.92	45/47	0.64	0.93	45/47	0.64	0.93	45/47	0.61	0.92	45/47
ru	0.63	0.92	48/50	0.59	0.84	48/50	0.6	0.88	48/50	0.66	0.92	48/50
ua	0.76	0.94	43/45	0.79	0.95	43/45	0.82	0.96	43/45	0.81	0.95	43/45
us	0.5	0.71	50/52	0.51	0.7	50/52	0.5	0.7	50/52	0.51	0.71	50/52
avg	0.52	0.85	0.95	0.54	0.85	0.95	0.52	0.85	0.95	0.51	0.85	0.95

Table 6.34: Results with aggregated rankings as replacement on the paid I-Phone news data set for top-10 prediction.

For the full prediction experiments of this data set 17 random data points were selected. The results in Table 6.35 and Table 6.36 are as could be expected from the previous analysis. The method PR worked extremely well, DVR and PDVR worked only rarely and the rest showed no improvement over PR. And finally all aggregation methods have similar results.

	default	PR	DVR	PDVR	HSR	CR	WCR
Gain	0.0	1.0	0.01	0.05	1.0	1.0	1.0
Borda	0.73	0.94	0.73	0.73	0.94	0.92	0.91
Nanson	0.73	0.94	0.73	0.73	0.94	0.91	0.91
Copeland	0.74	0.95	0.74	0.74	0.95	0.91	0.9
Maximin	0.73	0.94	0.73	0.73	0.94	0.9	0.89
Schulze	0.74	0.94	0.74	0.74	0.94	0.91	0.89

Table 6.35: Average Kendall tau-a scores after implicit delegation of multiple rankings for the paid I-Phone news data set.

	default	PR	DVR	PDVR	HSR	CR	WCR
Gain	0.0	1.0	0.3	0.32	1.0	1.0	1.0
Borda	0.73	0.94	0.72	0.73	0.94	0.92	0.91
Nanson	0.73	0.94	0.71	0.73	0.94	0.91	0.91
Copeland	0.74	0.95	0.72	0.75	0.95	0.91	0.9
Maximin	0.73	0.94	0.73	0.73	0.94	0.9	0.89
Schulze	0.74	0.94	0.72	0.74	0.94	0.91	0.89

Table 6.36: Average Kendall tau-a scores of successful implicit delegation attempts of multiple rankings for the paid I-Phone news data set.

6.4 Summary of the Results

After this limited testing on 9 real world preference data sets it can be seen that the method PR is often really good in predicting ballots for voters. In seven of the nine data sets it resulted in the highest average Kendall tau type-a scores between the three analysis methods. As expected HSR produced the best results as long as good individual results between the methods were available.

For some of the data sets (Spotify daily and weekly, I-Phone games free and paid) DVR was able to find acceptable replacements when considering the top-k. However there was no data set, for which this method was able to on average improve the similarity after a full prediction of multiple missing voters. This does not definitely mean that DVR is never useful for this purpose, only that none of the tested data sets were suitable for this.

In general it could be observed that if PR produced high tau-a scores for the top-k prediction experiments, it also worked well for the full predictions. The method PDVR produced most of the time the worst results. Considering it is also the most demanding method it can probably be recommended to not use it at all in most cases and only sometimes to improve the results of HSR.

The implicit delegation methods CR, WCR and additionally for top-k TCR and TWCR all failed to improve the results. This does not mean no aggregation of potential replacement rankings can improve the similarity to the original, it simply means no such method could be found.

When comparing the results of CR, WCR for the full prediction no definite advantage of one over the other was observed. However for top-k prediction the weighted version produced slightly worse results. This can probably be attributed to a higher number of partly less likely similar rankings to be combined for WCR. The weights were an attempt to lessen the impact of rankings with low similarity, but the chosen weights seem to not have enough impact. For example a ranking that got an average tau-a score of 0.5 has half as much impact as a ranking with a score of 1.0. Maybe some more wide spread weights could produce better results.

When comparing the trimmed versions TCR and TWCR to the not trimmed versions CR, WCR it is clear that for top-10 predictions the trimming is of advantage. However for the one data set with top-3 experiments, namely the Eurovision data set, the trimming to top-3 rankings before aggregating worsened the results. It is likely that the selected k and the original size of the rankings both have an impact on how the trimming effects the results.

Now to the differences between the aggregation methods for the full prediction experiments. It seems like the tested SWFs are all similar effected by the implicit delegation methods. As single very small outlier Maximin has the lowest Kendall tau-a scores after the implicit delegations. But the difference is so small, that it can probably be assumed that in most cases the different implicit delegation methods perform nearly equally good or bad with all the tested SWFs.

CHAPTER 7

Conclusion

To sum up, multiple preference aggregation methods were defined together with their properties and run-time complexity. This gives a good overview of popular SWFs. As Kemeny has exponential run-time it was decided to not use it within the experiments. Schulze was also removed from the tests for the largest data sets. But with Borda, Nanson, Copeland, Maximin and sometimes Schulze a wide variety of SWFs is considered.

After this theoretical background was shown the data sets that are used in the testing were presented. For these data sets a new data type was developed on the basis of SOI from PREFLIB.ORG to allow temporal not anonym data collections. For the data the main requirements were to have data over a time-frame from real world applications. For each of the data sets a basic analysis was performed, that compares different voters to each other and another analysis that compared for each voter every ballot with the ranking in the data point before it. This analysis can be used to predict which methods work good on a given data set.

Then the different implicit delegation methods were described and presented together with a worst case run-time estimate. For the different methods that analyze the data, some descriptions were given for suitable data sets. E.g. PR works the best on data where not many alternatives are changed between the data points and the voters do not change their opinion very much between each data point.

Last but not least the experiments were explained and their results presented. Here it was evident that the past analysis gave a pretty accurate prediction for the method PR, as it worked acceptable to good for the data sets, where higher and positive scores were give, and gave only very few and not great results, where low or negative scores were given.

It could also be observed that the methods that depend on different voters DVR and PDVR did only rarely produce good results. This could also be expected as the correlation analysis between voters was in general low. But they were not useless, as they were often able to predict at least a good portion of the alternatives that made it into the top-k.

Overall HSR produced the best results. However it is dependent on the three analysis methods and if performance is a requirement it could be a good decision to only use the method that shows the most promise in the overall analysis. For PDVR there is no direct analysis, partly because the results would best be represented in a 3-dimensional matrix, which is not good and meaningful representable on paper. However it is more of an edge case and probably works best if the correlation between the voters in general is high.

For the implicit delegation methods that combine different rankings no improvement over a simple copy of the best possible solution was observed. Therefore these methods can probably be seen as failure and more research is required to test different approaches. For example different weights for WCR or TWCR could lead to better results. The number of rankings to combine or the minimum requirements for these rankings could also need to be adjusted. However some potentially useful results could be observed. For the prediction of the top-k it can be beneficial to trim the input rankings to an appropriate length before aggregating them. The length needs to be at least k and should be higher if k is very small like 3.

Finally it could be observed that the different implicit delegation methods perform roughly the same for the full prediction with different SWFs. This means they can be used independently of the given function, at least with the tested ones. More testing with other methods could be conducted to see if there exist some SWFs for which the methods perform different, but as the tested ones fit such a wide spectrum it is likely that other preference aggregation methods effect the results similar.

Acronyms

CR combined replacement. 48–50, 58, 59, 61, 63, 65, 66, 68–71, 73–81

DVR different voter replacement. 46, 47, 50, 56, 59, 60, 62–64, 66, 67, 69, 70, 73–80, 83

HSR highest score replacement. 47, 50, 56, 59, 60, 63, 64, 66, 67, 69, 70, 73–81, 84

IIA Independence of Irrelevant Alternatives. 8, 15

PDVR past different voter replacement. 47, 50, 56, 59, 60, 62–64, 66, 67, 69, 70, 72–80, 83, 84

PR past replacement. 46, 47, 50, 54, 56, 59, 60, 62–64, 66, 67, 69, 70, 72–80, 83

SCF social choice function. 6–8, 11, 13–15, 18, 20

SWF social welfare function. 2, 6–8, 11, 13–15, 47–49, 81, 83, 84

TCR trimmed combined replacement. 49, 50, 58, 61, 65, 68, 71, 74, 76, 77, 79, 81

TWCR trimmed weighted combined replacement. 49, 50, 58, 61, 65, 68, 71, 74, 76, 77, 79, 81, 84

WCR weighted combined replacement. 48–50, 58, 59, 61, 63, 65, 66, 68–71, 73–81, 84

Bibliography

[AM12] Alnur Ali and Marina Meilă. Experiments with Kemeny Ranking: What works when? *Mathematical Social Sciences*, 64(1):28–40, 2012.

[Arr08] Kenneth J Arrow. Arrow's Theorem. *The New Palgrave Dictionary of Economics: Volume 1–8*, pages 241–244, 2008.

[ASS10] Kenneth J Arrow, Amartya Sen, and Kotaro Suzumura. *Handbook of Social Choice and Welfare*, volume 2. Elsevier, 2010.

[Ast03] Owen Astrachan. Bubble Sort: An Archaeological Algorithmic Analysis. *SIGCSE Bulletin (Association for Computing Machinery, Special Interest Group on Computer Science Education)*, pages 1–5, 03 2003.

[BBN14] Nadja Betzler, Robert Bredereck, and Rolf Niedermeier. Theoretical and Empirical Evaluation of Data Reduction for Exact Kemeny Rank Aggregation. *Autonomous Agents and Multi-Agent Systems*, 28(5):721–748, 2014.

[BCE+16] Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D Procaccia. *Handbook of Computational Social Choice*. Cambridge University Press, 2016.

[BDW84] Charles Blackorby, David Donaldson, and John A Weymark. Social Choice with Interpersonal Utility Comparisons: A Diagrammatic Introduction. *International Economic Review*, pages 327–356, 1984.

[BFH09] Felix Brandt, Felix Fischer, and Paul Harrenstein. The Computational Complexity of Choice Sets. *Mathematical Logic Quarterly*, 55(4):444–459, 2009.

[BGH12] Franz J Brandenburg, Andreas Gleißner, and Andreas Hofmeier. Comparing and Aggregating Partial Orders with Kendall Tau Distances. In *International Workshop on Algorithms and Computation*, pages 88–99. Springer, 2012.

[CDK06] Vincent Conitzer, Andrew Davenport, and Jayant Kalagnanam. Improved Bounds for Computing Kemeny Rankings. In *In In Proceedings of the 21st National Conference on Artificial Intelligence (AAAI*, volume 6, pages 620–627. AAAI Press, 2006.

[CELM07] Yann Chevaleyre, Ulle Endriss, Jérôme Lang, and Nicolas Maudet. A Short Introduction to Computational Social Choice. In *International Conference on Current Trends in Theory and Practice of Computer Science*, pages 51–69. Springer, 2007.

[CLP18] Theresa Csar, Martin Lackner, and Reinhard Pichler. Computing the Schulze Method for Large-Scale Preference Data Sets. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pages 180–187, 2018.

[EH89] Jon Elster and Aanund Hylland. *Foundations of Social Choice Theory*. Cambridge University Press, 1989.

[FKM+04] Ronald Fagin, Ravi Kumar, Mohammad Mahdian, D. Sivakumar, and Erik Vee. Comparing and Aggregating Rankings with Ties. In *Proceedings of the Twenty-third ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '04, pages 47–58, New York, NY, USA, 2004. ACM.

[Gär73] Peter Gärdenfors. Positionalist Voting Functions. *Theory and Decision*, 4(1):1–24, 1973.

[HSV05] Edith Hemaspaandra, Holger Spakowski, and Jörg Vogel. The Complexity of Kemeny Elections. *Theoretical Computer Science*, 349(3):382 – 391, 2005.

[Ken45] Maurice G Kendall. The Treatment of Ties in Ranking Problems. *Biometrika*, 33(3):239–251, 1945.

[Lin10] Shili Lin. Rank Aggregation Methods. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(5):555–570, 2010.

[Mar96] Thierry Marchant. Valued Relations Aggregation with the Borda Method. *Journal of Multi-Criteria Decision Analysis*, 5(2):127–132, 1996.

[MS97] Vincent R Merlin and Donald G Saari. Copeland Method II: Manipulation, Monotonicity, and Paradoxes. *Journal of Economic Theory*, 72(1):148–172, 1997.

[MW13] Nicholas Mattei and Toby Walsh. Preflib: A Library for Preferences http://www. preflib.org. In *International Conference on Algorithmic DecisionTheory*, pages 259–270. Springer, 2013.

[Nio87] Emerson MS Niou. A Note on Nanson's Rule. *Public Choice*, 54(2):191–193, 1987.

[Noe81] Gottfried E Noether. Why Kendall Tau? *Teaching Statistics*, 3(2):41–43, 1981.

[Nur04] Hannu Nurmi. Monotonicity and its Cognates in the Theory of Choice. *Public Choice*, 121(1-2):25–49, 2004.

[Od19] Stephan Okhuijsen / @datagraver. Eurovision Song Contest Scores 1975-2019. `https://data.world/datagraver/eurovision-song-contest-scores-1975-2019`, 2019. Accessed: 2019-06-02.

[Oy] AppFollow.fi Oy. Appfollow. `https://appfollow.io`. Accessed: 2019-07-12.

[Ren01] Philip J Reny. Arrow's Theorem and the Gibbard-Satterthwaite Theorem: A Unified Approach. *Economics Letters*, 70(1):99–105, 2001.

[Sch03] Markus Schulze. A new Monotonic and Clone-Independent Single-Winner Election Method. *Voting matters*, 17(1):9–19, 2003.

[Sch11] Markus Schulze. A new Monotonic, Clone-Independent, Reversal Symmetric, and Condorcet-Consistent Single-Winner Election method. *Social Choice and Welfare*, 36(2):267–303, 2011.

[Spo19] AB Spotify. Spotify Charts. `https://spotifycharts.com/regional`, 2019. Accessed: 2019-05-15.

[SSA+02] Kōtarō Suzumura, Amartya Kumar Sen, Kenneth Joseph Arrow, et al. *Handbook of Social Choice and Welfare*, volume 1. Gulf Professional Publishing, 2002.

[vB11] Michelle R vanDellen and Evander Baker. Implicit Delegation of Responsibility: Joint Self-Control in Close Relationships. *Social Psychological and Personality Science*, 2(3):277–283, 2011.

[You75] H Peyton Young. Social Choice Scoring Functions. *Society for Industrial and Applied Mathematics (SIAM) Journal on Applied Mathematics*, 28(4):824–838, 1975.

[You95] Peyton Young. Optimal Voting Rules. *Journal of Economic Perspectives*, 9(1):51–64, 1995.