Informatics

Teilüberwachtes Lernen mit Totaler Variation auf Graphen

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Technische Informatik

eingereicht von

Max Geiselbrechtinger, BSc
Matrikelnummer 01609418

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Gerald Matz
Mitwirkung: Projektass. Dipl.-Ing. Thomas Dittrich, BSc

Wien, 5. Dezember 2022

Max Geiselbrechtinger

Gerald Matz

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

Informatics

Semi-Supervised Learning with Total Variation on Graphs

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Computer Engineering

by

Max Geiselbrechtinger, BSc
Registration Number 01609418

to the Faculty of Informatics

at the TU Wien

Advisor: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Gerald Matz
Assistance: Projektass. Dipl.-Ing. Thomas Dittrich, BSc

Vienna, 5th December, 2022

Max Geiselbrechtinger Gerald Matz

Erklärung zur Verfassung der Arbeit

Max Geiselbrechtinger, BSc

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, 5. Dezember 2022

Max Geiselbrechtinger

Danksagung

Ich möchte mich herzlichst bei meinen Betreuern Ao.Univ.Prof. Gerald Matz und Dipl.-Ing. Thomas Dittrich, welche mich stets mit hilfreichem Feedback und stärkender Motivation versorgt haben, für die angenehme und lehrreiche Zusammenarbeit bedanken. Des Weiteren möchte ich mich auch bei meinen Freunden bedanken, die mich in und neben dem Studium begleitet haben und die Zeit so zu einer erlebnisreichen und schönen Reise gemacht haben. Ein ganz besonderer Dank gilt meiner Familie die mich immer Unterstützt hat und die mir in jeder Lebenslage verlässlich und stärkend zur Seite steht.

Kurzfassung

Graphen bieten ein nützliches Modell für viele Probleme da sie es ermöglichen auf flexible und effiziente Art die Datenstruktur zu abstrahieren. Um Gebrauch von den wachsenden Mengen an Daten zu machen ist Klassifizierung eine essentielle Aufgabe die auf eine Partitionierung der Daten in Ähnlichkeitsgruppen abzielt. Allerdings beinhalten viele reale Netzwerke auch Verbindungen die Unähnlichkeiten beschreiben. In Sozialen Netzwerken etwa können Nutzer ihre Sympathie oder Antipathie durch befreunden respektive blockieren ausdrücken. Um diese gegensätzlichen Informationen effektiv zu nutzen ist es nötig spezifische Algorithmen zu konstruieren, die auf vorzeichenbehafteten Graphen arbeiten. Das Hauptaugenmerk der Forschung in der Vergangenheit lag auf dem Clustern von vorzeichenlosen Graphen. Eine vielversprechende Methode des maschinellen Lernens ist empirische Risikominimierung um die Clusterbeschriftung an die bekannten Beschriftungen anzupassen. Um diesen beaufsichtigten Lernprozess zu unterstützen wird ein zusätzliches Ziel hinzugefügt, welches den Algorithmus dazu bringen soll eine Clusterbeschriftung zu generieren, die glatt über die Struktur des Graphen variiert. Es existieren mehrere verschiedene Glattheitsmetriken die für solch eine unbeaufsichtigte Regulierung eingesetzt werden können. Manche davon wurden auch auf vorzeichenbehaftete Graphen erweitert.

In dieser Arbeit zielen wir darauf ab, zwei teilüberwachte Klassifizierungsalgorithmen zu verbessern indem wir die totale Variation als Glattheitsmetrik auf vorzeichenbehafteten Graphen einsetzen. Die totale Variation wurde in mehreren kürzlich entwickelten teilbeaufsichtigten Methoden eingesetzt und hat sich als effektive Zielfunktion herausgestellt. Um die totale Variation wirkungsvoll mit klassischem maschinellen Lernen zu verbinden machen wir Gebrauch von konvexer Optimierung. Insbesondere leiten wir zwei teilbeaufsichtigte Clusterzielfunktionen für vorzeichenbehaftete Graphen her und entwickeln die iterativen Algorithmen um die Optimierungsprobleme effizient zu lösen. Wir präsentieren eine Verhaltensanalyse der Parameter und vergleichen die Clusterperformanz von unseren Methoden mit dem Stand der Technik. Zusätzliche numerische Experimente auf synthetischen Datenmodellen sollen die Fähigkeit unserer Algorithmen, Information über Unähnlichkeit auszunutzen, beurteilen. Die Tests ergaben, dass Algorithmen, welche Gebrauch von der totalen Variation machen, eine überlegene Clusterperformanz aufweisen. Des Weiteren haben wir unsere Methoden auf Daten eines echten sozialen Netzwerks eingesetzt. Unser Algorithmen haben dabei, trotz der widrigen Clusterstruktur, zufriedenstellende Ergebnisse erzielt.

Abstract

Graphs are a natural fit for modeling a multitude of problems as they are flexible and efficient abstractions of complex datasets. To make use of the growing amounts of data, clustering is an essential task that aims at partitioning the data into groups that reflect similarity. However, many real world networks also contain links that describe dissimilarity. In social networks for example users can associate themselves by befriending or blocking each other, which expresses sympathy or antipathy, respectively. To effectively utilize such opposing information in clustering tasks it is necessary to construct specific algorithms that operate on signed graphs. The main focus of research in the past was directed towards clustering on unsigned graphs. A promising machine learning method is to deploy empirical risk minimization to fit the cluster labeling to a set of known labels. To aid this supervised learning process an additional objective is added that should incentivise the algorithm to produce cluster labelings that are smooth with respect to the graph's similarity structure. There exist several different smoothness metrics that can be deployed in such an unsupervised regularization and some have also been extended to signed graphs.

In this thesis we aim to improve two semi-supervised clustering algorithms by utilizing the total variation as a smoothness metric on signed graphs. The total variation has been deployed in several recent semi-supervised methods and has shown to be an effective objective for graph clustering. To combine the total variation with classic machine learning algorithms we make use of convex optimization procedures. In particular we derive two semi-supervised clustering objectives for signed graphs and develop iterative algorithms to efficiently solve the stated optimization problems. We provide an analysis of the behavior of our algorithm's parameters and compare their clustering performance to state of the art methods. Additional numerical experiments are conducted on synthetic data models to asses the ability of our novel algorithms to exploit dissimilarity information. The tests revealed that algorithms which use the total variation indeed produce superior clustering performance in both the unsigned and signed case. Furthermore, we employed our methods on data derived from a real world social network. Our algorithms achieved satisfactory results, despite the problem's unfavorable cluster structure.

Contents

CHAPTER 1

Introduction

Graphs are powerful and versatile models that can be employed for a multitude of real world problems. Examples range from recommender systems for movie or product databases over community detection in social networks to infrastructure analysis such as railway or road congestion detection [LAH07] [LHK10] [LLDM09]. Modeling such problems with graphs can help to exploit sparsity in the data structure and often allows for local formulations that can lead to parallelizable algorithms. In the past, research has been mainly focused on unsigned graphs which are able to encode information about similarity or the absence thereof. Many datasets however, also contain a notion of dissimilarity. For example likes and dislikes in movie recommendations or trust and distrust in social communities. Signed graphs are able to encode such antagonistic relations but the development of theories and tools to exploit these models is a topic of ongoing research [DM20].

In any case, algorithms that perform complex tasks on graphs such as classification of new data points, prediction of missing data attributes or ranking of data elements are hard to designing using conventional programming methodologies. These circumstances have propelled the field of machine learning which constitutes a data driven paradigm for developing algorithms. Rather than coding a fixed procedure that performs a certain task supervised learning focuses on developing algorithms that are able to learn a task from training examples and feedback. However, to deploy supervised learning on sophisticated large scale problems, huge amounts of labeled training examples are required [CSZ06]. A promising solution to restrain the growing need of expensive labeled data is semi-supervised learning which aims to fuse the learning process with additional information about unlabeled data. Employing information of unlabeled data requires the presence of an inherent connection between the structure of the data and the target values. One such assumption that proofed to be suitable in practice states that target values vary smoothly with respect to the underlying data distribution.

In this thesis we will focus on the task of semi-supervised clustering on signed graphs. Clustering is a fundamental task in data analysis and therefore relevant for any scientific field that deals with large quantities of data [VL07]. In data analysis clustering is used to discover grouping or connectivity patterns, it can also be embedded as preprocessing step in more complex pipelines to filter inputs. There exist various ways to define clusters, on unsigned graphs they are usually identified through densely connected subsets of nodes. In the case of signed graphs clustering is often regarded from the standpoint of social balance theory. In [Dav67] Davis defines a graph as k-balanced if there exists a clustering into k partitions such that nodes within each partition are only connected via positive edges and nodes from different partitions are only connected via negative edges. Our aim is to cluster unbalanced graphs (eg. constructed from noisy data) by deploying the total variation as a smoothness metric in combination with semi-supervised learning. For this we assume the number of clusters k to be known in advance.

Pursuing the semi-supervised approach is of particular practical interest since it allows to combine expensive and therefore modestly available labeled data with task inherent domain knowledge. As a concrete example we can consider the detection of communities of similar political partisanship. While conventional polls are laborious and therefore usually only sample a small fraction of the public online social networks provide an abundant amount of information about the relations between individuals. Thus combining sparse training samples with powerful graphical models may provide the basis for an effective algorithm to determine clusters of similar partisanship. Furthermore for tasks of this kind it is crucial to utilize signed graphs in order to model strong opposing concepts like ideological agree- or disagreement.

The core contribution of this work is the derivation of two semi-supervised clustering algorithms for signed graphs. While we utilize classic empirical risk minimization to incorporate prior labeling information in our algorithms we deviate from existing methods as [GZW07] in that we choose the total variation as a smoothness metric. It has been shown that the total variation is a tight continuous relaxation for the discrete clustering problem on unsigned graphs [SB10]. Therefore we expect it to increase the clustering performance when combined with existing semi-supervised methods. Furthermore we present efficient implementations of the algorithms relying on the alternating direction method of multipliers convex optimization scheme. In numerical experiments we analyze the parameters of our algorithms and report their capabilities of effectively utilizing dissimilarity information. Further experiments are conducted to compare the algorithms accuracies with state of the art semi-supervised clustering methods on synthetic datasets. We conclude by clustering a large scale signed graph obtained from an online social network to demonstrate the algorithms capabilities on real world data.

1.1 Notation

Throughout this thesis we adhere to the following notation:

- Italic letters denote scalars (eg. M, n)
- Lowercase bold letters denote vectors (eg. $\boldsymbol{v}$)
- Lowercase letters with a subscript denote the entry of a vector at the position indicated by the subscript (eg. v_i represents the i-th entry of vector $\boldsymbol{v}$)
- Uppercase bold letters denote matrices (eg. $\boldsymbol{K}$)
- Uppercase letters with two letter subscripts denote an entry of the matrix at the position indicated by the subscripts (eg. K_{ij} represents the entry at row i and column j of matrix $\boldsymbol{K}$)
- Uppercase bold letters with subscripts containing a colon refer to row or column slices of a matrix (eg. $\boldsymbol{K}_{i:}$ and $\boldsymbol{K}_{:j}$ represent the i-th row and j-th column of matrix $\boldsymbol{K}$ respectively)
- Italic letters which are followed by a list of arguments enclosed in parenthesis denote scalar functions. Written in bold they denote a vector of scalar functions (eg. $C(r)$, $\boldsymbol{f}(\boldsymbol{x})$)
- Uppercase calligraphic letters denote sets or collections such as tuples (eg. $\mathcal{L}$)
- The identity matrix of size $N \times N$ will be denoted $\boldsymbol{I}_N$ and the vectors of size N containing all ones or all zeros are denoted $\mathbf{1}_N$ and $\mathbf{0}_N$ respectively.
- The operator $\otimes$ denotes the Kronecker product and is used to succinctly formulate block matrix constructions (eg. $\boldsymbol{I}_N \otimes \boldsymbol{A}$ yields a matrix with N blocks of matrix $\boldsymbol{A}$ on the diagonal).

CHAPTER 2

Background

In this chapter we introduce the mathematical concepts used throughout the thesis. Furthermore we summarize the key ideas of statistical learning theory to establish the basis for the derivation of our methods. The last section is devoted to the convex optimization algorithm that we utilize in our implementations.

2.1 Graphs

A graph $\mathcal{G}$ is a 3-tuple $(\mathcal{V}, \mathcal{E}, \boldsymbol{W})$ whose entries are the node set, edge set and the weight matrix. The node set, sometimes referred to as the vertex set, is of the form $\mathcal{V} = \{v_1, \dots, v_N\}$. Unless explicitly stated otherwise the graphs we consider are undirected, simple and contain no self-loops. The edge set is given by $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. Thus the weight matrix has the form $\boldsymbol{W} \in \mathbb{R}^{N \times N}$, where W_{ij} is the weighting of the edge connecting vertices v_i and v_j. If $(v_i, v_j) \notin \mathcal{E}$ then $W_{ij} = 0$.

Furthermore we can introduce signals $x : \mathcal{V} \to \mathbb{R}$ on the vertices of the graph. Such a signal can be compactly represented by a vector $\boldsymbol{x} \in \mathbb{R}^N$ and multiple signals can be aggregated into a graph signal matrix (e.g. $\boldsymbol{X} \in \mathbb{R}^{N \times M}$ for the concatenation of M graph signals). Graph signal processing (GSP) introduces the tools, like filtering and sampling, to manipulate and analyze graph signals. Another key concept in GSP is the graph Fourier transformation (GFT), which constitutes a mapping between the vertex and spectral domain based on the eigendecomposition of the graph Laplacian [SNF+13].

2.1.1 Laplacians

For a graph $\mathcal{G}$ with non-negative edge weights (i.e. $W_{ij} \geq 0$) its Laplacian matrix $\boldsymbol{L}$ is defined as [KSL+10]

$$\begin{gathered}\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}, \\ D_{ii} = \sum_{j=0}^{N} W_{ij}, \quad D_{ij} = 0 \text{ for } i \neq j.\end{gathered} \tag{2.1}$$

The graph Laplacian is symmetric, positive-semidefinite and exhibits some interesting properties about the underlying graph. For instance the multiplicity of the eigenvalue zero indicates the number of connected components of the graph. Furthermore, [SNF+13] directly interprets eigenvalues as the frequency corresponding to a signal that is defined by the graph Laplacians eigenvectors. That is, Shuman et al. consider a signal to be smooth over a graph if it "has similar values at neighboring vertices connected by an edge with a large weight". This connectivity is encoded in the graph Laplacian and reflects the decrease in smoothness for increasing eigenvalues.

For graphs that also possess negative edge weights Kunegis et al. [KSL+10] define the signed Laplacian matrix based on a modified degree matrix

$$\begin{gathered}\bar{\boldsymbol{L}} = \bar{\boldsymbol{D}} - \boldsymbol{W}, \\ \bar{D}_{ii} = \sum_{j=0}^{N} |W_{ij}|, \quad \bar{D}_{ij} = 0 \text{ for } i \neq j.\end{gathered} \tag{2.2}$$

The signed Laplacian is also symmetric and positive-semidefinite. Which can be directly observed from the well known bilinear decomposition

$$\|\boldsymbol{x}\|_{\text{Lap}} = \boldsymbol{x}^T \bar{\boldsymbol{L}} \boldsymbol{x} = \sum_{i=0}^{N} \sum_{j=0}^{N} |W_{ij}|(x_i - S_{ij} x_j)^2. \tag{2.3}$$

Here, $\boldsymbol{x} = (x_1, \ldots, x_N)^T$ is a graph signal and $\boldsymbol{S}$ is the sign matrix of the graph weights (i.e. $S_{ij} = \text{sign}(W_{ij})$). For a connected graph the (non constant) signal $\boldsymbol{x}$ that minimizes the quadratic form in (2.3) is given by the eigenvector corresponding to the second smallest eigenvalue. This fact is used in spectral clustering to obtain graph partitions [VL07]. Furthermore, from the perspective of GSP this suggests that smoothly varying signals represent desirable cluster labelings. We will refer to (2.3) as Laplacian norm albeit in general it is only a semi-norm.

In [DM20] the gradient operator for signed graphs is defined to be a mapping $\nabla_{\mathcal{G}} : \mathbb{R}^N \to \mathbb{R}^{N \times N}$ of the form

$$(\nabla_{\mathcal{G}} \boldsymbol{x})_{ij} = |W_{ij}|^{\frac{1}{p}} (x_i - S_{ij} x_j), \tag{2.4}$$

with $p \in \{1, 2\}$. If we select $p = 2$ this allows the convenient representation of the bilinear Laplacian form as the squared 2-norm of the graph gradient $\boldsymbol{x}^T \bar{\boldsymbol{L}} \boldsymbol{x} = \|\nabla_{\mathcal{G}} \boldsymbol{x}\|_2^2$.

2.1.2 Total Variation

Another metric that comes from image processing and has been extended to signals on general graphs is the total variation. In the case of clustering on signed graphs the signed total variation is defined by [DM20] as

$$\|\boldsymbol{x}\|_{\mathrm{TV}} = \sum_{i=0}^{N} \sum_{j=0}^{N} |W_{ij}| \max(0, x_i - S_{ij} x_j). \tag{2.5}$$

Similar to the Laplacian norm the total variation can be regarded as a metric for smoothness. In contrast to the Laplacian norm it favors graph signals that are mostly constant with sharp transitions. A very favorable representation of the signed total variation was given by [DM20] via the signed graph gradient. For $p = 1$ this representation is

$$\|\boldsymbol{x}\|_{\mathrm{TV}} = \|\nabla_{\mathcal{G}} \boldsymbol{x}\|_{+}, \tag{2.6}$$

where $\|\boldsymbol{X}\|_{+} = \sum_{ij} \max(0, X_{ij})$ is the right sided 1-norm or +-norm. In Chapter 3 we will further review this representation and its application to semi-supervised learning.

2.2 Learning Framework

2.2.1 Supervised Learning

Given a training set of L input-label pairs $\{(x_i, y_i) : x_i \in \mathcal{X}, y_i \in \mathbb{R}\}$ supervised learning aims to find a function $f^* : \mathcal{X} \to \mathbb{R}$ that describes this input-label correspondences. Furthermore the function f^* should also generalize well when applied to inputs that are not contained in the training set. A common way to learn such a function is by minimizing an empirical risk functional of the form

$$f^* = \operatorname*{argmin}_{f} \frac{1}{L} \sum_{i=1}^{L} R(f(x_i), y_i), \tag{2.7}$$

where $R : \mathbb{R}^2 \to \mathbb{R}$ quantifies the loss incurred when $f(x_i)$ deviates from the correct labeling y_i. The choice of the loss function is part of the learning algorithm design and we will review two popular versions later on in this thesis. To prevent the learning algorithm from overfitting the function f^* to the training examples we need to impose constraints on the complexity of the function. This is often enforced by introducing regularization terms into the learning process. A general objective function for supervised learning reads [SSB⁺02]

$$f^* = \operatorname*{argmin}_{f} \frac{1}{L} \sum_{i=1}^{L} R(f(x_i), y_i) + \lambda \Omega(f(x_i)), \tag{2.8}$$

where $\Omega(\cdot)$ is an arbitrary non-negative functional that encodes the intended complexity constraints. The classification rule is given by $\operatorname{sign}(f(x))$.

There exist several different strategies to generalize supervised learning to the multi-class setting [SSB+02]. The one-versus-rest approach for example learns M classifiers that discriminate, as the name implies, each class from the remaining ones. Evaluation is performed by selecting the classifier with the maximum output value. A beneficial feature of the one-versus-rest approach is the possibility to form a confidence measure by comparing the margin between the two largest outputs. The one-versus-one method learns a classifier for each pair of classes, which amounts to a total of $M(M-1)/2$ classifiers. Although the number of required classifiers grows quadratic in the number of classes the individual problems are usually much smaller. Thus the overall learning process may be faster than with the one-versus-rest method for certain problem settings.

Furthermore it is also possible to perform the risk minimization for multiple classes at once, e.g., in [SSB+02] they derived a multi-class formulation for support vector machines. In general we can extend the problem to simultaneously learn M classification functions by introducing a multi-class risk functional R which considers the misclassification costs of each labeled sample for all classes. Additionally it is required to adopt a sparse label encoding to prevent the occurrence of asymmetric costs due to numerical differences of the class labels. A possible multi-class risk functional for least squares could read $R(\boldsymbol{f}(x_i), \boldsymbol{y}_i) = \sum_{j=1}^{M}(1-\delta(1-y_{ij}))(f_j(x_i)-y_{ij})^2$, where $\delta(x)$ is the discrete impulse function that is 1 at 0 and 0 everywhere else. Furthermore, $\boldsymbol{y}_i$ is a M-dimensional vector that has an entry 1 at the index reflecting the class affiliation and all other entries are $-\frac{1}{M-1}$. With the generic multi-class risk functional the supervised learning objective is given by

$$\boldsymbol{f}^* = \underset{\boldsymbol{f}}{\operatorname{argmin}} \frac{1}{L}\sum_{i=1}^{L} R(\boldsymbol{f}(x_i), \boldsymbol{y}_i) + \lambda \sum_{j=1}^{M} \Omega(f_j(x_i)), \tag{2.9}$$

where $\boldsymbol{f} = [f_1, \ldots, f_M]^T$ is a vector of M classification functions. The classification rule for the multi-class case is given by $\operatorname{argmax}_j f_j(x)$.

2.2.2 Kernels

In machine learning kernels are a strong concept that provide various advantages for the design of learning algorithms. In particular a kernel function $k : \mathcal{X}^2 \to \mathbb{R}$ allows the utilization of the powerful linear algebra machinery, specifically inner products, for inputs from an arbitrary feature space $\mathcal{X}$. Another advantage is the possibility to implicitly perform non-linear feature transforms on the input space and thus implement versatile yet scalable algorithms. A popular choice of kernels are Gaussian radial basis functions (RBF kernels). They provide a feature space mapping of infinite dimension with a smoothness constraint on the learnable functions. The RBF kernel has the form

$$k(x, x') = \exp\left(-\frac{\|x-x'\|^2}{2\kappa^2}\right) \tag{2.10}$$

where parameter $\kappa > 0$ controls the smoothness. Larger values of κ promote smoother functions. An in depth theory of kernels and their application in machine learning can be found in [SSB$^+$02]. The individual kernel function evaluations for all given samples can be conveniently packed into the kernel matrix $\boldsymbol{K}$ with $K_{ij} = k(x_i, x_j)$. In the following we will mostly use the matrix form and only resort to kernel function if we want to make an explicit statement about the kernel or its properties.

When working with graphical models often the only information provided are pairwise relations. Therefore it may not be straightforward to compute a similarity measure between any two data elements. For example the RBF kernel requires the ability to compute a distance between all elements of the feature space and thus can not be directly applied on graphs. One possible way to solve this issue is to learn a Euclidean graph embedding from the weight matrix. [CWWK20] give a comprehensive survey on graph representation learning methods. Another option is the application of graph kernels which are constructed from local operators only. In [KL02] Kondor and Lafferty introduce diffusion kernels which utilize the Laplacian matrix to perform a discretized heat diffusion process over the graph structure. The resulting kernel has the form

$$\boldsymbol{K} = e^{-\alpha \boldsymbol{L}}, \tag{2.11}$$

where $\alpha \geq 0$ determines the spread of the diffusion kernel. Similar to the RBF kernel entries in the heat diffusion kernel matrix represent a similarity measure between nodes. In fact heat diffusion kernels are closely related to Gaussian kernels as shown in [KL02]. Albeit circumventing the need for additional representation learning heat diffusion kernels come with some drawbacks. For example the matrix exponentiation is a computational intensive operation that doesn't scale very well to large graphs. Furthermore adding new nodes to the graph requires recomputation of the whole kernel matrix.

We will now review the application of kernels in the supervised learning process. For functions of a reproducing kernel Hilbert space $\mathcal{H}_k$, which is determined through the selected kernel k, we use the norm $\|.\|_k$ as regularizer to obtain the following objective [SSB$^+$02]

$$f^* = \operatorname*{argmin}_{f \in \mathcal{H}_k} \frac{1}{L} \sum_{i=1}^{L} R(f(x_i), y_i) + \lambda \|f\|_k^2. \tag{2.12}$$

The representer theorem [SSB$^+$02, Theorem 4.2] states that the minimizer f^* of this problem is given by a linear combination of the respective kernel

$$f^*(x) = \sum_{i=1}^{L} c_i k(x_i, x) + b. \tag{2.13}$$

This effectively reduces the problem of learning a function to learning a set of L coefficients c_i and a scalar bias term b.

2.2.3 Semi-supervised Learning

Semi-supervised learning grounds on the assumption that geometric properties of the data convey some information about their labeling. For example inputs that are close in feature space might be more likely to belong to the same class rather than inputs that are located far apart from each other. Such assumptions encode domain knowledge and have to be carefully selected to fit the task and data of the specific problem. However, if the assumption holds true, then we can utilize additional unlabeled samples to enhance the learning algorithm.

The extension of the representer theorem in [ZG06] to include U unlabeled samples shows that every objective function of the form

$$f^* = \operatorname*{argmin}_{f \in \mathcal{H}_K} \frac{1}{L} \sum_{i=1}^{L} R(f(x_i), y_i) + \lambda_1 \|f\|_k^2 + \lambda_2 r(f, x_1, \dots, x_N). \tag{2.14}$$

This formulation extends the supervised framework with a regularization term r that may depend on f as well as the labeled and unlabeled data. (2.14) is again minimized by a linear combination of the kernel. In particular the minimizer is similar to that in supervised learning except that it also takes into account all $N = L + U$ data points

$$f^*(x) = \sum_{i=1}^{N} c_i k(x_i, x) + b. \tag{2.15}$$

Although the regularization functional r can be of any form, it is usually beneficial to restrict it to be convex for computational reasons. In Section 3.1 we introduce algorithms that utilize this regularization to promote functions f^* that vary smoothly over similarity graphs. In Chapter 4 we rely on this theoretic result to incorporate the total variation as regularizer when formulating our algorithms.

The generalization of semi-supervised learning to the multi-class case can be achieved in a similar as shown for supervised learning. In particular we again use the generic multi-class risk functional and apply the regularization to all M classification functions. The multi-class objective for semi-supervised learning can be stated as

$$\boldsymbol{f}^* = \operatorname*{argmin}_{\boldsymbol{f}} \frac{1}{L} \sum_{i=1}^{L} R(\boldsymbol{f}(x_i), \boldsymbol{y}_i) + \lambda \sum_{j=1}^{M} \Omega(f_j(x_i)) + \lambda_2 \sum_{j=1}^{M} r(f_j, x_1, \dots, x_N). \tag{2.16}$$

2.3 Alternating Direction Method of Multipliers

The alternating direction method of multipliers (ADMM) is an optimization algorithm from the 1970s that has gained recent popularity due to its parallelizability and its robust convergence properties [BPC+11]. The general idea behind ADMM is to split the objective function in two primal parts whose augmented Lagrangians are alternatingly minimized. The connection between the primal variables is established by updating the dual variable which holds the constraints. Problems that are amenable to the application of ADMM are of the form

$$\begin{aligned} \min \quad & g(\boldsymbol{c}) + h(\boldsymbol{d}) \\ \text{s.t.} \quad & \boldsymbol{A}\boldsymbol{c} + \boldsymbol{B}\boldsymbol{d} = \boldsymbol{e}, \end{aligned} \tag{2.17}$$

where $\boldsymbol{c} \in \mathbb{R}^N$ and $\boldsymbol{d} \in \mathbb{R}^M$ are the primal variables, h and g are convex functions, and the matrices $\boldsymbol{A} \in \mathbb{R}^{P \times N}$, $\boldsymbol{B} \in \mathbb{R}^{P \times M}$ and the vector $\boldsymbol{e} \in \mathbb{R}^P$ encode an affine relation between the primal variables. The iterative update steps of the augmented Lagrangians of this formulation are given by

$$\begin{aligned} \boldsymbol{c}^{t+1} &:= \operatorname*{argmin}_{\boldsymbol{c}} h(\boldsymbol{c}) + \frac{\rho}{2}\|\boldsymbol{A}\boldsymbol{c} + \boldsymbol{B}\boldsymbol{d}^t - \boldsymbol{e} + \boldsymbol{u}^t\|_2^2 \\ \boldsymbol{d}^{t+1} &:= \operatorname*{argmin}_{\boldsymbol{d}} g(\boldsymbol{d}) + \frac{\rho}{2}\|\boldsymbol{A}\boldsymbol{c}^{t+1} + \boldsymbol{B}\boldsymbol{d} - \boldsymbol{e} + \boldsymbol{u}^t\|_2^2 \\ \boldsymbol{u}^{t+1} &:= \boldsymbol{u}^t + \boldsymbol{A}\boldsymbol{c}^{t+1} + \boldsymbol{B}\boldsymbol{d}^{t+1} - \boldsymbol{e}, \end{aligned} \tag{2.18}$$

with the dual variable $\boldsymbol{u}$. With modest assumptions on g and h the ADDM algorithm is guaranteed to converge for all $\rho > 0$, although achieving high accuracy results can require a large amount of iterations. For many practical problems however, ADMM has been found to produce reasonably accurate solutions within a few tens of iterations. As a possible stopping criterion the following expression can be used

$$\|\boldsymbol{r}^t\|_2 \leq \epsilon^{prim} \wedge \|\boldsymbol{s}^t\|_2 \leq \epsilon^{dual}, \tag{2.19}$$

with the primal and dual residuals $\boldsymbol{r}$ and $\boldsymbol{s}$ given by

$$\begin{aligned} \boldsymbol{r}^t &= \boldsymbol{A}\boldsymbol{c}^t + \boldsymbol{B}\boldsymbol{d}^t - \boldsymbol{e}, \\ \boldsymbol{s}^t &= \rho \boldsymbol{A}^T \boldsymbol{B}(\boldsymbol{d}^t - \boldsymbol{d}^{t-1}), \end{aligned} \tag{2.20}$$

which can be related to the suboptimality of the current objective value. Heuristics for the the stopping tolerances ϵ^{prim} and ϵ^{dual} are

$$\begin{aligned} \epsilon^{prim} &= \sqrt{P}\epsilon^{abs} + \epsilon^{rel} \max\left(\|\boldsymbol{A}\boldsymbol{c}^t\|_2, \|\boldsymbol{B}\boldsymbol{d}^t\|_2, \|\boldsymbol{e}\|_2\right), \\ \epsilon^{dual} &= \sqrt{N}\epsilon^{abs} + \epsilon^{rel}\rho\|\boldsymbol{A}^T\boldsymbol{u}^t\|_2, \end{aligned} \tag{2.21}$$

with $\epsilon^{abs} > 0$ and $\epsilon^{rel} > 0$.

CHAPTER 3

State of the Art

Semi-supervised clustering on graphs, also called node classification, aims at finding a node labeling which adheres to the underlying graph structure while taking into account prior label information. In this chapter we will introduce state of the art algorithms for semi-supervised clustering. We will focus on algorithms that operate on signed graphs. However, we will also include seminal works for clustering on unsigned graphs.

For semi-supervised clustering we are given disjoint sets of labeled nodes for each class $\mathcal{L}_m = \{v_i \in \mathcal{V} : y_i = m\}$. The union of all these sets is the label set $\mathcal{L} = \bigcup_{m=1}^{M} \mathcal{L}_m$ with $|\mathcal{L}| = L$ and the set of unlabeled nodes is $\mathcal{U} = \mathcal{V} \setminus \mathcal{L}$ with $|\mathcal{U}| = U$. Usually it holds that $L \ll U$. The target is to estimate the labels for the U unsigned nodes. Since the class labels are not ordinal it is generally beneficial to convert the integer labeling into a one-hot encoding over the M classes. Thus a valid label matrix $\boldsymbol{Y} \in \{0,1\}^{L \times M}$ has exactly one entry of value 1 at the y_i-th index of each row i. Most algorithms estimate a real valued relaxation of the label matrix $\widehat{\boldsymbol{X}} \in \mathbb{R}^{N \times M}$, which we will term class affiliation matrix. The node labeling can be obtained from this matrix, e.g., by selecting the index with the highest class affiliation estimate in each row of $\widehat{\boldsymbol{X}}$.

3.1 Manifold Regularization

Belkin et al. [BNS04] introduce manifold regularization in which they enhance classical supervised learning algorithms such as regularized least squares (RLS) and support vector machines (SVM) with information of unlabeled data. Their approach grounds on the assumption that the data lives on a manifold and the true labeling should vary smoothly across the underlying geometry of the data distribution. To construct an empirical estimate of smoothness on manifolds they chose the discrete Laplace operator. Recall from Section 2.1.1 that minimizing the quadratic form of the Laplacian induces the class affiliation signal to vary smoothly over the weighted graph. Their objective function has the form

$$\min_{f \in \mathcal{H}_k} \quad \frac{1}{L} \sum_{i=1}^{L} R(f(x_i), y_i) + \lambda_1 \|f\|_k^2 + \frac{\lambda_2}{N^2} \boldsymbol{f}^T \boldsymbol{L} \boldsymbol{f}, \tag{3.1}$$

where $\boldsymbol{f}$ is a vector with $f(x_i)$, $i \in \{1, \ldots, N\}$. The empirical risk functional R is either chosen as the squared error or hinge loss for RLS or SVM, respectively. Since manifolds have a local Euclidean structure, the kernel k can be chosen among classic kernels (e.g. polynomial or Gaussian). This method is limited to unsigned graphs and binary clustering. Nevertheless it can be generalized to multi-class settings with the standard one-vs-one or one-vs-rest strategies for supervised classification algorithms.

In [GZW07] Goldberg et al. extended the manifold regularization framework to signed graphs. For the case of binary clustering they simply propose to use the signed Laplacian in the quadratic smoothness regularizer of (3.1). This straightforward adaptation for incorporating dissimilarity however, does not convey to the multi-class setting. Therefore a custom regularization term for dissimilarity edges is devised and combined with a multi-class risk minimization functional. This custom regularization term inspired the definition of the signed total variation and with that also the design of our objective functions in Chapter 4. The multi-class objective function of [GZW07] is given by

$$\begin{aligned} \min_{\boldsymbol{f} \in \mathcal{H}_k} \quad & \frac{1}{L} \sum_{i=1}^{L} R(\boldsymbol{f}(\boldsymbol{x}_i), \boldsymbol{y}_i) + \lambda_1 \sum_{j=1}^{M} \|f_j\|_k^2 + \frac{\lambda_2}{|\mathcal{D}|} \sum_{(s,t) \in \mathcal{D}} \|\boldsymbol{f}(\boldsymbol{x}_s) + \boldsymbol{f}(\boldsymbol{x}_t) - \tfrac{M-2}{M-1} \mathbf{1}\|_+ \\ \text{s.t.} \quad & \sum_{j=1}^{M} f_j(\boldsymbol{x}_i) = 0, \quad i = 1, \ldots, N, \end{aligned} \tag{3.2}$$

with $\mathcal{D}$ being the set of dissimilarity edges. Note that in this formulation $\boldsymbol{f}$ is a vector of M separate classification functions. The class affiliation matrix can be obtained via $\widehat{X}_{ij} = f_j(\boldsymbol{x}_i)$. The term $\frac{M-2}{M-1}\mathbf{1}$ is introduced to make the dissimilarity regularizer comply with the sum to zero label encoding where labels $\boldsymbol{y}_i$ are chosen such that the entry corresponding to the affiliated class is 1 and all other entries are $-\frac{1}{M-1}$. We will investigate this encoding in detail in the next chapter. Their regularizer however does not take into account any similarity information. Furthermore both [BNS04] and [GZW07] make use of convex regularization functionals in order to deploy standard quadratic programming. Since the regularization term of (3.2) is not continuously differentiable it has to be handled via constraints and slack variables. This causes the variables of the quadratic optimization problem to scale with the number of negative edges in the graph.

3.2 Total Variation Minimization

A straight forward objective for determining clusters in graph structured data is based on separation by minimizing edge cuts [VL07]. In general however, this discrete clustering approach leads to NP-hard problems and thus is intractable for most practically relevant cases. This led researchers to focus on establishing tight convex relaxations for the discrete clustering problems in order to utilize powerful linear algebra theory and (convex) optimization solvers.

Bresson et al. [BLUVB13] derived a relaxation of the ratio cut problem that is based on the total variation rather than the quadratic Laplacian form used in spectral clustering. They argue that the total variation metric is more suited for tasks such as clustering as it promotes the formation of larger constant signal areas with few but possibly sharp transitions in between. In contrast to this desired indicator function like behavior the Laplacian norm favors solutions that vary more gradually and thus possibly produces indifferent affiliation signals at the boundary between clusters. Because minimizing the total variation objective alone can lead to degenerate solutions (e.g., most nodes get accumulated in one large cluster) they additionally precondition their algorithm to form clusters of similar sizes. To include information about labeled data a hard constraint is added which simply fixes the respective rows of the class affiliation matrix to equal the indicator vectors of the corresponding classes. The resulting objective is a sum of ratios of convex functions which they solve using a proximal splitting algorithm.

In [BDM18] and [BDHM19] a signed version of the total variation is proposed for binary and multi-class clustering respectively. Recall from Section 2.1.2 the signed total variation is a smoothness metric and can be related to min-cuts on signed graphs. Their basic optimization objective has the form

$$\begin{aligned} \min_{\widehat{\boldsymbol{X}}} \quad & \|\widehat{\boldsymbol{X}}\|_{\mathrm{TV}} \\ s.t. \quad & \widehat{\boldsymbol{X}} \in \mathcal{Q}. \end{aligned} \tag{3.3}$$

With the constraint set $\mathcal{Q}$ enforcing the encoding and prior label information,

$$\begin{aligned} \mathcal{Q} = \{\widehat{\boldsymbol{X}} \in [-1,1]^{N\times M} : & X_{ij} = 1,\, i \in \mathcal{L}_j, \\ & X_{ij} = -1,\, i \in \mathcal{L} \setminus \mathcal{L}_j, \\ & \textstyle\sum_{j=1}^{M} X_{ij} = -M+2,\, i = 1,\dots,N\}. \end{aligned} \tag{3.4}$$

The signed total variation minimization can also be cast into the semi-supervised learning framework from Section 2.2.3. If we consider the semi-supervised multi-class problem with the signed total variation as regularizer we get

$$\begin{aligned} \min_{\boldsymbol{f}\in\mathcal{H}_k} \quad & \frac{1}{L}\sum_{i=1}^{L} R(\boldsymbol{f}(\boldsymbol{x}_i), \boldsymbol{y}_i) + \lambda_1 \sum_{j=1}^{M} \|f_j\|_k^2 + \lambda_2 \sum_{j=1}^{M} \|f_j\|_{\mathrm{TV}} \\ \text{s.t.} \quad & \sum_{j=1}^{M} f_j(\boldsymbol{x}_i) = -M+2, \quad i = 1,\dots,N. \end{aligned} \tag{3.5}$$

Selecting the risk functional to be the characteristic function of the label vector $R(\boldsymbol{f}(\boldsymbol{x}_i), \boldsymbol{y}_i) = \mathcal{X}_{\boldsymbol{y}_i}(\boldsymbol{f}(\boldsymbol{x}_i))$ with the identity kernel function, i.e, $k(x, x') = 1$ if $x = x'$ and 0 else, yields the same solutions as the original problem formulation of (3.4). The model complexity constraints $\lambda_1 \sum_{j=1}^{M} \|f_j\|_k^2$ are not relevant in this problem setting since the characteristic function requires any valid solutions to exactly match all labeled samples regardless, hence λ_1 can be set to 0.

As already mentioned earlier solely relying on the total variation can be detrimental especially when only few labeled nodes are available. To alleviate this problem [BDHM19] implement a regularization procedure which reinforces the weights of unlabeled nodes in the similarity neighborhood of a labeled node. Furthermore they propose an automated tuning scheme to determine a good set of regularization parameters. The resulting convex optimization problem is solved using the ADMM algorithm (cf. Section 2.3) which allows for a distributed implementation to handle large graph instances.

3.3 Diffuse Interface Methods

Another type of semi-supervised graph clustering algorithms is inspired by partial differential equations from the field of material science. In particular the aim is to minimize a discretized version of the Ginzburg-Landau functional which consists of a smoothness and an encoding term that are connected via a diffuse interface scaling parameter. More formally, in the case of two classes, Bertozzi et al. consider the objective

$$\min_{f} \quad \sum_{i=1}^{L} \frac{1}{2}(f(\boldsymbol{x}_i) - y_i)^2 + \frac{\epsilon}{2}\boldsymbol{f}^T \boldsymbol{L} \boldsymbol{f} + \frac{1}{4\epsilon}\sum_{i=1}^{N}(f(\boldsymbol{x}_i)^2 - 1)^2 \tag{3.6}$$

and show that in the limit of the scaling parameter ϵ approaching zero this functional converges to the total variation [BF12]. If we compare the Ginzburg-Landau functional to the semi-supervised learning objective of (2.14) we can identify that they use the squared loss as risk functional. Furthermore they do not introduce any model complexity constraints but rather deploy two regularization terms that are coupled by the scaling parameter ϵ. The first regularization term $\|f\|_{\text{Lap}}$ induces smoothness on the labeling while the second term $\sum_{i=1}^{N}(f(\boldsymbol{x}_i)^2 - 1)^2$ pulls the labels towards either -1 or 1. Their extension to the multi-class case has the form

$$\min_{\boldsymbol{f}} \quad \sum_{i=1}^{L} \frac{1}{2}(\boldsymbol{f}(\boldsymbol{x}_i) - \boldsymbol{y}_i)^2 + \frac{\epsilon}{2}\sum_{j=1}^{M}\|f_j\|_{\text{Lap}} + \frac{1}{2\epsilon}\sum_{i=1}^{N}\left(\prod_{j=1}^{M}\frac{1}{4}\|\boldsymbol{f}(\boldsymbol{x}_i) - \boldsymbol{e}_j\|_1^2\right), \tag{3.7}$$

where M is the number of classes and $\boldsymbol{e}_j$ is the j-th unit vector. The data fidelity and smoothness terms are straightforward multi-class adaptions, only the second regularization term has been modified to drive labels closer to the vertices of the Gibbs-simplex. Mercado et al. [MBS19] extended this scheme to signed graphs by utilizing different combinations of signed and unsigned Laplacians as smoothness metrics in the regularization term.

CHAPTER 4

Semi-supervised learning with Total Variation

In this chapter we present the derivation of our algorithms that utilize the total variation as a regularizer to classic empirical risk minimization, i.e., we solve the problem

$$\min_{f \in \mathcal{H}_k} \quad \sum_{j=1}^{M} \left(\frac{1}{L} \sum_{i=1}^{L} R(f_j(x_i), y_i) + \lambda_1 \|f_j\|_k^2 + \lambda_2 \|f_j\|_{\mathrm{TV}} \right). \tag{4.1}$$

As we have seen in the previous chapter the approach of [BNS04] and its extension to the signed multi-class case in [GZW07] attempt to find cluster affiliation functions that vary smooth over the graph structure. In contrast to Laplacian regularizers the total variation has the desired effect of forming indicator-function-like class affiliations as argued in [BLUVB13]. The theoretic underpinning to solve the resulting optimization problem is given by the extended representer theorem in [ZG06] which allows us to express the minimizer as a linear combination of the kernel.

Because the total variation is not continuously differentiable we derive an appropriate formulation based on variable separation in order to solve the optimization problem with the ADMM algorithm introduced in Section 2.3. In particular we only require the solutions of two separate minimization problems which form the subroutines of the ADMM algorithm.

Semi-supervised multi-class clustering comes down to finding coefficients $\boldsymbol{c}_m$ and a bias term b_m that form the m-th class affiliation estimate (cf. (2.15)). For convenience we combine coefficients and bias into a single vector $\tilde{\boldsymbol{c}}_m = [\boldsymbol{c}_m^T, b_m]^T$ and define the corresponding extended kernel matrix as $\widetilde{\boldsymbol{K}} = [\boldsymbol{K}, \mathbf{1}]$. Now the class affiliation estimate for node $\boldsymbol{x}_i$ and class m is given by $f_m(\boldsymbol{x}_i) = \widehat{X}_{im} = \widetilde{\boldsymbol{K}}_{i:}\tilde{\boldsymbol{c}}_m$.

4.1 Total Variation with Kernels

Applying the representer theorem to the problem of (4.1) yields the objective

$$\min \quad \sum_{j=1}^{M} \left(\frac{1}{L} \sum_{i=1}^{L} R(\widetilde{\boldsymbol{K}}_{i:}\tilde{\boldsymbol{c}}_m, y_i) + \lambda_1 \|\widetilde{\boldsymbol{K}}_{i:}\tilde{\boldsymbol{c}}_m\|_k^2 + \lambda_2 \|\widetilde{\boldsymbol{K}}_{i:}\tilde{\boldsymbol{c}}_m\|_{\mathrm{TV}} \right). \tag{4.2}$$

Recall from Section 2.1.2 that the signed total variation semi-norm can be expression via the +-norm of the graph gradient. Since ADMM allows to separate variables that are related via a linear operator this effectively reduces the total variation to the +-norm of a proxy variable. Fortunately there exists a closed form solution for the augmented Lagrangian of the +-norm. However, the TV-norm in (4.2) contains an additional kernel map that has to be incorporated into the graph gradient operator to allow the effective application of ADMM. In particular we require a linear operator for which the signed graph gradient of the m-th class estimate has the form

$$(\nabla_{\mathcal{G}}\widehat{\boldsymbol{X}}_{:m})_{ij} = |W_{ij}|(\widetilde{\boldsymbol{K}}_{i:}\tilde{\boldsymbol{c}}_m - S_{ij}\widetilde{\boldsymbol{K}}_{j:}\tilde{\boldsymbol{c}}_m) = |W_{ij}|(\widetilde{\boldsymbol{K}}_{i:} - S_{ij}\widetilde{\boldsymbol{K}}_{j:})\tilde{\boldsymbol{c}}_m = (\nabla_{\mathcal{G}}\widetilde{\boldsymbol{K}})_{ij}\tilde{\boldsymbol{c}}_m. \tag{4.3}$$

Therefore we define the signed graph gradient for the kernelized framework as

$$(\nabla_{\mathcal{G}}\widetilde{\boldsymbol{K}})_{ij} = |W_{ij}|(\widetilde{\boldsymbol{K}}_{i:} - S_{ij}\widetilde{\boldsymbol{K}}_{j:}), \tag{4.4}$$

which constitutes a mapping $\nabla_{\mathcal{G}} : \mathbb{R}^{N\times N+1} \to \mathbb{R}^{N^2\times N+1}$. This mapping allows us to establish the connection between total variation and graph gradients for kernelized methods

$$\|\widehat{\boldsymbol{X}}_{:m}\|_{\mathrm{TV}} = \|\widetilde{\boldsymbol{K}}\tilde{\boldsymbol{c}}_m\|_{\mathrm{TV}} = \|\nabla_{\mathcal{G}}\widetilde{\boldsymbol{K}}\tilde{\boldsymbol{c}}_m\|_{+}. \tag{4.5}$$

4.2 Class encoding

We will adopt the class encoding of [GZW07] which is chosen according to the sum-to-zero constraint. In particular the affiliation y_i of node v_i is reflected by the M-dimensional vector label vector $\boldsymbol{y}_i$ with all entries being $-\frac{1}{M-1}$ except for a 1 at the m-th position. This encoding however, can lead to a potentially harmful penalization of the objective in the presence of dissimilarity edges. To showcase this problem we partition the edge set into $\mathcal{E}_{sim}$ and $\mathcal{E}_{dis}$ which contain only similarity and dissimilarity edges respectively. This allows us to decompose the total variation of the class affiliation matrix into

$$\|\widehat{\boldsymbol{X}}\|_{\mathrm{TV}} = \sum_{(i,j)\in\mathcal{E}_{sim}} |W_{ij}|\|\widehat{\boldsymbol{X}}_{i:} - \widehat{\boldsymbol{X}}_{j:}\|_{+} + \sum_{(i,j)\in\mathcal{E}_{dis}} |W_{ij}|\|\widehat{\boldsymbol{X}}_{i:} + \widehat{\boldsymbol{X}}_{j:}\|_{+}. \tag{4.6}$$

If we now examine the two scenarios where $\widehat{\boldsymbol{X}}_{i:} = \widehat{\boldsymbol{X}}_{j:}$ and $\widehat{\boldsymbol{X}}_{i:} \neq \widehat{\boldsymbol{X}}_{j:}$ for the similarity case (for simplicity we neglect the edge weights in this analysis) we get

$$\|\widehat{\boldsymbol{X}}_{i:} - \widehat{\boldsymbol{X}}_{j:}\|_{+} = \begin{cases} 0 & \text{if } \widehat{\boldsymbol{X}}_{i:} = \widehat{\boldsymbol{X}}_{j:}, \\ \frac{M}{M-1} & \text{if } \widehat{\boldsymbol{X}}_{i:} \neq \widehat{\boldsymbol{X}}_{j:}. \end{cases} \tag{4.7}$$

Minimizing this enforces similarity between the class affiliation of nodes v_i and v_j as intended. However, for the dissimilarity case we have

$$\|\widehat{\boldsymbol{X}}_{i:} + \widehat{\boldsymbol{X}}_{j:}\|_+ = \begin{cases} 2 & \text{if } \widehat{\boldsymbol{X}}_{i:} = \widehat{\boldsymbol{X}}_{j:}, \\ \frac{2(M-2)}{M-1} & \text{if } \widehat{\boldsymbol{X}}_{i:} \neq \widehat{\boldsymbol{X}}_{j:}. \end{cases} \tag{4.8}$$

For $M > 2$ this class encoding erroneously penalizes the case where the class affiliation of nodes v_i and v_j do not equal. A solution to this issue proposed in [GZW07] is to introduce a constant offset for the dissimilarity edges of the form

$$\|\widehat{\boldsymbol{X}}_{i:} + \widehat{\boldsymbol{X}}_{j:} - \tfrac{M-2}{M-1}\mathbf{1}\|_+ = \begin{cases} \frac{M}{M-1} & \text{if } \widehat{\boldsymbol{X}}_{i:} = \widehat{\boldsymbol{X}}_{j:}, \\ 0 & \text{if } \widehat{\boldsymbol{X}}_{i:} \neq \widehat{\boldsymbol{X}}_{j:}. \end{cases} \tag{4.9}$$

The offset not only eliminates the issue with penalizing the desired case it also shifts the intensity of penalization to match with the similarity case. We will show how to incorporate this constant shift when deriving the ADMM update for the TV-norm at the end of the next section.

4.3 Regularized Least Squares with Total Variation

Similarly to Goldberg et al. [GZW07] we start of with the regularized least squares formulation and augment it with an additional regularization term. In our case we use the total variation of the class estimates. This leads us to the following optimization problem

$$\begin{aligned} \min_{\boldsymbol{f} \in \mathcal{H}_k} \quad & \frac{1}{L}\sum_{i=1}^{L} \|\boldsymbol{f}(\boldsymbol{x}_i) - \boldsymbol{y}_i)\|_2^2 + \lambda_1 \sum_{j=1}^{M} \|f_j\|_k^2 + \lambda_2 \sum_{j=1}^{M} \|f_j\|_{\text{TV}} \\ \text{s.t.} \quad & \sum_{j=1}^{M} f_j(\boldsymbol{x}_i) = 0, \quad i = 1, \dots, N. \end{aligned} \tag{4.10}$$

Rewriting this in vectorized version and applying the representer theorem by expressing the class affiliation estimate as linear combinations of the learnable coefficients and the kernel matrix yields the convex optimization problem

$$\begin{aligned} \min \quad & \sum_{m=1}^{M} \left(\tfrac{1}{L}\|\boldsymbol{J}\widetilde{\boldsymbol{K}}\tilde{\boldsymbol{c}}_m - \boldsymbol{Y}_{:m}\|_2^2 + \lambda_1 \tilde{\boldsymbol{c}}_m^T \bar{\boldsymbol{K}}\tilde{\boldsymbol{c}}_m + \lambda_2 \|\widetilde{\boldsymbol{K}}\tilde{\boldsymbol{c}}_m\|_{\text{TV}} \right) \\ \text{s.t.} \quad & \sum_{m=1}^{M} \widetilde{\boldsymbol{K}}\tilde{\boldsymbol{c}}_m = \mathbf{0}. \end{aligned} \tag{4.11}$$

Without loss of generality we assume that the first L nodes are the labeled nodes. Thus we introduce $\boldsymbol{J}$ as a short formulation for the first L rows of the $N \times N$ identity matrix. $\boldsymbol{Y}$ is the $L \times M$ label matrix with all $-\frac{1}{M-1}$ entries except for 1 at Y_{ij} if a label indicates that node i belongs to class j. Furthermore $\bar{\boldsymbol{K}}$ is the kernel matrix $\boldsymbol{K}$ with an additional row and column of zeros appended to form a square $(N+1) \times (N+1)$ matrix.

To derive an ADMM admissible form we first incorporate the equality constraints into the objective function and further apply identity (4.5) to the total variation. We have

$$\min \quad \sum_{m=1}^{M} \left(\tfrac{1}{L} \|\boldsymbol{J}\widetilde{\boldsymbol{K}}\tilde{\boldsymbol{c}}_m - \boldsymbol{Y}_{:m}\|_2^2 + \lambda_1 \tilde{\boldsymbol{c}}_m^T \bar{\boldsymbol{K}} \tilde{\boldsymbol{c}}_m + \mathcal{X}_{\mathcal{Q}}(\widetilde{\boldsymbol{C}}) + \lambda_2 \|\nabla_{\mathcal{G}} \widetilde{\boldsymbol{K}} \tilde{\boldsymbol{c}}_m\|_+ \right), \tag{4.12}$$

where $\mathcal{X}_{\mathcal{Q}}$ is the indicator function of the set $\mathcal{Q} = \{\widetilde{\boldsymbol{C}} \in \mathbb{R}^{(N+1)\times M} : \widetilde{\boldsymbol{K}}\widetilde{\boldsymbol{C}}\mathbf{1} = \mathbf{0}\}$ which contains all extended coefficients that meet the sum-to-zero constraint.

After introducing auxiliary variables $\boldsymbol{D}_m$ and the affine constraints that relate them to the original coefficients the problem

$$\begin{aligned} \min \quad & \sum_{m=1}^{M} \left(\tfrac{1}{L} \|\boldsymbol{J}(\widetilde{\boldsymbol{K}}\tilde{\boldsymbol{c}}_m - \boldsymbol{Y}_{:m})\|_2^2 + \lambda_1 \tilde{\boldsymbol{c}}_m^T \bar{\boldsymbol{K}} \tilde{\boldsymbol{c}}_m + \mathcal{X}_{\mathcal{Q}}(\widetilde{\boldsymbol{C}}) + \lambda_2 \|\boldsymbol{D}_m\|_+ \right) \\ \text{s.t.} \quad & \nabla_{\mathcal{G}} \widetilde{\boldsymbol{K}} \tilde{\boldsymbol{c}}_m = \boldsymbol{D}_m, \quad m = 1 \ldots M, \end{aligned} \tag{4.13}$$

is in the desired form (cf. (2.17). This leads to the update steps

$$\begin{aligned} \boldsymbol{C}^{t+1} &= \underset{\boldsymbol{C}}{\operatorname{argmin}} \ \tfrac{1}{L} \|\boldsymbol{J}(\widetilde{\boldsymbol{K}}\boldsymbol{C} - \boldsymbol{Y})\|_{\mathrm{F}}^2 + \lambda_1 \operatorname{tr}(\boldsymbol{C}^T \bar{\boldsymbol{K}} \boldsymbol{C}) + \mathcal{X}_{\mathcal{Q}}(\boldsymbol{C}) + \tfrac{\varrho}{2} \|\nabla_{\mathcal{G}} \widetilde{\boldsymbol{K}} \boldsymbol{C} - \boldsymbol{D}^t + \boldsymbol{U}^t\|_{\mathrm{F}}^2, \\ \boldsymbol{D}^{t+1} &= \underset{\boldsymbol{D}}{\operatorname{argmin}} \ \lambda_2 \|\boldsymbol{D}\|_+ + \tfrac{\varrho}{2} \|\nabla_{\mathcal{G}} \widetilde{\boldsymbol{K}} \boldsymbol{C}^{t+1} - \boldsymbol{D} + \boldsymbol{U}^t\|_{\mathrm{F}}^2, \end{aligned} \tag{4.14}$$

for which we will now derive closed form solutions.

The $\boldsymbol{C}$-update amounts to solving a quadratic function subject to affine constraints. This problem can be minimized by solving the respective KKT system as described in [BBV04]. To derive the solution we first introduce $\boldsymbol{c}$ as the vectorization of $\boldsymbol{C}$ in order to cast the problem into the standard quadratic given by

$$\begin{aligned} \min \quad & \tfrac{1}{2} \boldsymbol{c}^T \boldsymbol{P} \boldsymbol{c} + \boldsymbol{q}^T \boldsymbol{c} + \boldsymbol{r} \\ \text{s.t.} \quad & \boldsymbol{G}\boldsymbol{c} = \boldsymbol{h}. \end{aligned} \tag{4.15}$$

The matrices $\boldsymbol{P} \in \mathbb{R}^{(N+1)M \times (N+1)M}$, $\boldsymbol{G} \in \mathbb{R}^{N \times (N+1)M}$ and vectors $\boldsymbol{q} \in \mathbb{R}^{(N+1)M}$, $\boldsymbol{h} \in \mathbb{R}^{N}$ are given by

$$\begin{aligned} \boldsymbol{P} &= \boldsymbol{I}_M \otimes \left(\tfrac{2}{L} \widetilde{\boldsymbol{K}}^T \boldsymbol{J}^T \boldsymbol{J} \widetilde{\boldsymbol{K}} + 2\lambda_1 \bar{\boldsymbol{K}} \right), \\ \boldsymbol{q} &= \left(\boldsymbol{I}_M \otimes \left(-\tfrac{2}{L} \widetilde{\boldsymbol{K}}^T \boldsymbol{J} \right) \right) \boldsymbol{y}, \\ \boldsymbol{G} &= \mathbf{1}_M^T \otimes \widetilde{\boldsymbol{K}}, \\ \boldsymbol{h} &= \mathbf{0}_N. \end{aligned} \tag{4.16}$$

The KKT system of the augmented Lagrangian for the $\boldsymbol{C}$-update has the form

$$\begin{bmatrix} \boldsymbol{P} + \rho \boldsymbol{A}^T \boldsymbol{A} & \boldsymbol{G}^T \\ \boldsymbol{G} & \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{c} \\ \boldsymbol{\nu} \end{bmatrix} = \begin{bmatrix} \rho \boldsymbol{A}^T \boldsymbol{v} - \boldsymbol{q} \\ \boldsymbol{h} \end{bmatrix}. \tag{4.17}$$

Here $\boldsymbol{\nu}$ represents the dual variable for the Lagrangian of the affine constraints. To simplify the notation we introduced $\boldsymbol{A} = \boldsymbol{I}_M \otimes (\nabla_{\mathcal{G}} \widetilde{\boldsymbol{K}})$ and $\boldsymbol{v} = \boldsymbol{d} - \boldsymbol{u}$, with $\boldsymbol{d}$ and $\boldsymbol{u}$ being the vectorizations of $[\boldsymbol{D}_1, \dots, \boldsymbol{D}_M]$ and the ADMM duals $[\boldsymbol{U}_{\boldsymbol{1}}, \dots, \boldsymbol{U}_M]$ respectively.

Let $\boldsymbol{S} = \boldsymbol{P} + \rho \boldsymbol{A}^T \boldsymbol{A}$ then the solution for $\boldsymbol{c}$ can be obtained via the Schur complement (cf. [BBV04]) as

$$\boldsymbol{c} = (\boldsymbol{S}^{-1} - \boldsymbol{S}^{-1}\boldsymbol{G}^T(\boldsymbol{G}\boldsymbol{S}^{-1}\boldsymbol{G}^T)^{-1}\boldsymbol{G}\boldsymbol{S}^{-1})(\rho \boldsymbol{A}^T \boldsymbol{v} - \boldsymbol{q}) - \boldsymbol{S}^{-1}\boldsymbol{G}^T(\boldsymbol{G}\boldsymbol{S}^{-1}\boldsymbol{G}^T)^{-1}\boldsymbol{h}. \tag{4.18}$$

Since $\boldsymbol{S}$ is block diagonal and all blocks are equal, solving for $\boldsymbol{c}$ requires inverting two $N \times N$ matrices, which can be precomputed and cached. However the complete factorization of the KKT system has no special form and can grow rather large. Therefore we utilized a numeric linear algebra package that can store matrices in a spares format to be able to perform the $\boldsymbol{C}$-update even for large N.

The minimization of the augmented Lagrangian for the $\boldsymbol{D}$-update results in the evaluation of the proximal operator of the +-norm. As the norm consists of a separable sum of maximum operations we can use the results for scalar proximal operators. According to [CP11] (Table 2, ii) the solution is given by the element wise one-sided soft thresholding operation with the argument $\boldsymbol{V}_m = \nabla_{\mathcal{G}} \widetilde{\boldsymbol{K}} \tilde{\boldsymbol{c}}_m + \boldsymbol{U}_m$ for all $m = 1 \dots M$ with

$$soft_{\left[0, \frac{\lambda_2}{\rho}\right]}(V_{ij}) = \begin{cases} V_{ij} & \text{if } V_{ij} < 0, \\ V_{ij} - \frac{\lambda_2}{\rho} & \text{if } V_{ij} > \frac{\lambda_2}{\rho}, \\ 0 & \text{else.} \end{cases} \tag{4.19}$$

To incorporate the constant shift for dissimilarity edges as described in (4.9) we can use the translation rule for proximal operators [CP11] (Table 1, i). Thus we construct a constant shift matrix $\boldsymbol{\Delta}$ that only targets the negative edges. The shift matrix is given by

$$\Delta_{ij} := |W_{ij}|(1 - S_{ij})\tfrac{1}{2}\tfrac{M-2}{M-1}. \tag{4.20}$$

The translation rule states to subtract $\boldsymbol{\Delta}$ prior to applying the one-sided soft thresholding operation and adding it back to the result afterwards. With this we have derived all the necessary update steps to apply the ADMM algorithm on regularized least squares with total variation. A listing of the complete algorithm is given in Section 4.5.

4.4 Support Vector Machine with Total Variation

Again we start off with a similar formulation as presented in [GZW07] and augment it with the total variation of the class estimates. For the SVM formulation we substitute the empirical risk functional of (4.1) with the hing-loss and arrive at the following problem

$$\begin{aligned} \min_{\boldsymbol{f}\in\mathcal{H}_k} \quad & \tfrac{1}{L}\sum_{i=1}^{L}\|\boldsymbol{f}(\boldsymbol{x}_i)-\boldsymbol{y}_i\|_+ + \lambda_1\sum_{j=1}^{M}\|f_j\|_{\mathrm{k}}^2 + \lambda_2\sum_{j=1}^{M}\|f_j\|_{\mathrm{TV}} \\ \text{s.t.} \quad & \sum_{j=1}^{M} f_j(\boldsymbol{x}_i) = 0, \quad i = 1,\dots,N. \end{aligned} \tag{4.21}$$

Note that the multi-class SVM formulation used in [GZW07] only considers misclassification costs, i.e., they exclude the +-norm terms where the $\boldsymbol{y}_i$ entry equals 1. However, we will also introduce a cost for deviations from the correct class labeling. This is not in conformity with standard empirical risk minimization but it prevents us from having to introduce a separate graph gradient for each of the M classes. With the additional cost term and expansion of the total variation according to (4.5) the objective function reads

$$\begin{aligned} \sum_{m=1}^{M}\Bigg(& \tfrac{1}{L}\sum_{l\in\mathcal{L}\setminus\mathcal{L}_m}(\widetilde{\boldsymbol{K}}_{l:}\tilde{\boldsymbol{c}}_m + \tfrac{1}{M-1})_+ + \tfrac{1}{L}\sum_{l\in\mathcal{L}_m}(\widetilde{\boldsymbol{K}}_{l:}\tilde{\boldsymbol{c}}_m - 1)_+ + \lambda_1\tilde{\boldsymbol{c}}_m^T\bar{\boldsymbol{K}}\tilde{\boldsymbol{c}}_m + \\ & +\lambda_2\sum_{(i,j)\in\mathcal{E}}|W_{ij}|(\widetilde{\boldsymbol{K}}_{i:}\tilde{\boldsymbol{c}}_m - S_{ij}\widetilde{\boldsymbol{K}}_{j:}\tilde{\boldsymbol{c}}_m)_+\Bigg)\,. \end{aligned} \tag{4.22}$$

We will now extend the original graph such that the hinge-loss terms of the labeled nodes are generated by the graph gradient. This will effectively reduce the problem to a similar structure as for *RLS-TV* and thus allows us to solve it using the variable update procedure derived in the previous section.

4.4.1 Augmented Graph

First we introduce M anchor nodes that are connected to the labeled nodes. In particular the m-th anchor node will be connected to all nodes within the label set that are affiliated to class m (i.e., $v \in \mathcal{L}_m$). The edges are weighted by $\frac{1}{2\lambda_2 L}$ to include the empirical risk normalization and compensate for the regularization parameter of the TV term. So for each labeled node v_l the gradient of the extended graph will contain M additional terms of the form $\frac{1}{2\lambda_2 L}(\widetilde{\boldsymbol{K}}_{l:} - \widetilde{\boldsymbol{K}}_{a:})$ where v_a is the appropriate anchor node. Furthermore due to the undirectedness of the graph the gradient also includes the M mirrored terms $\frac{1}{2\lambda_2 L}(\widetilde{\boldsymbol{K}}_{a:} - \widetilde{\boldsymbol{K}}_{l:})$. Although the mirrored terms are not explicitly contained in the original objective (4.22) they do not influence the solution since the sum-to-zero constraints enforce a similar behavior anyways.

In order to fully resemble the SVM costs of (4.22) we need that $\widetilde{\boldsymbol{K}}_{a:}\tilde{\boldsymbol{c}}_m = 1$ if anchor node v_a represents class m and $\widetilde{\boldsymbol{K}}_{a:}\tilde{\boldsymbol{c}}_m = -\frac{1}{M-1}$ otherwise. This is achieved by introducing an orthogonal extension of the kernel matrix

$$\widetilde{\boldsymbol{K}}^{\perp} = \begin{bmatrix} \boldsymbol{K} & \boldsymbol{0} & \boldsymbol{1} \\ \boldsymbol{0} & \boldsymbol{I}_M & \boldsymbol{0} \end{bmatrix}. \tag{4.23}$$

Furthermore we need to extend the optimization coefficients $\boldsymbol{C}$ with additional anchor coefficients in the following way

$$\boldsymbol{C}^{\forall} = \tfrac{M}{M-1}\boldsymbol{I}_M - \tfrac{1}{M-1}\boldsymbol{1}, \qquad \widetilde{\boldsymbol{C}}^{\perp} = \begin{bmatrix} \boldsymbol{C} \\ \boldsymbol{C}^{\forall} \\ \boldsymbol{b}^T \end{bmatrix}. \tag{4.24}$$

Since the anchor coefficients $\boldsymbol{C}^{\forall}$ have to be fixed during the optimization procedure we introduce an additional constraint to keep them at their respective values. With the augmented graph and the extended definitions for kernel and coefficients we are now in the situation to rewrite the optimization problem as

$$\min \quad \sum_{m=1}^{M} \lambda_1 \tilde{\boldsymbol{c}}_m^T \bar{\boldsymbol{K}} \tilde{\boldsymbol{c}}_m + \mathcal{X}_{\mathcal{Q}}(\widetilde{\boldsymbol{C}}) + \mathcal{X}_{\mathcal{P}}(\widetilde{\boldsymbol{C}}) + \sum_{m=1}^{M} \lambda_2 \|\widetilde{\boldsymbol{K}}^{\perp} \tilde{\boldsymbol{c}}_m^{\perp}\|_{\mathrm{TV}}, \tag{4.25}$$

where $\mathcal{X}$ is the indicator function, $\mathcal{Q} = \{\widetilde{\boldsymbol{C}} \in \mathbb{R}^{(N+1)\times M} : \widetilde{\boldsymbol{K}}\widetilde{\boldsymbol{C}}\boldsymbol{1} = \boldsymbol{0}\}$ is the set of all coefficients that meet the sum-to-zero constraints and $\mathcal{P} = \{[\boldsymbol{C}_1^T, \boldsymbol{C}_2^T, \boldsymbol{C}_3^T]^T \in \mathbb{R}^{(N+M+1)\times M} : \boldsymbol{C}_2 = \boldsymbol{C}^{\forall}\}$ denotes the set of coefficients whose anchors are fixed to the values defined in (4.24). Separating the variables as in the previous section yields the following ADMM-admissible formulation

$$\begin{aligned} \min \quad & \sum_{m=1}^{M} \lambda_1 \tilde{\boldsymbol{c}}_m^T \bar{\boldsymbol{K}} \tilde{\boldsymbol{c}}_m + \mathcal{X}_{\mathcal{Q}}(\widetilde{\boldsymbol{C}}) + \mathcal{X}_{\mathcal{P}}(\widetilde{\boldsymbol{C}}^{\forall}) + \sum_{m=1}^{M} \lambda_2 \|\boldsymbol{D}_m\|_+ \\ \text{s.t.} \quad & \nabla_{\mathcal{G}} \widetilde{\boldsymbol{K}}^{\perp} \tilde{\boldsymbol{c}}_m^{\perp} = \boldsymbol{D}_m, \quad m = 1, \dots, M. \end{aligned} \tag{4.26}$$

Besides the absence of the data fidelity term and the addition of constraint $\mathcal{P}$ this problem has the same form as (4.13). This allows us to utilize the same update steps as presented in (4.14) with only slight modifications.

Since set $\mathcal{P}$ contains only a single point it requires us to simply reset the anchor coefficients to their specific values at each ADMM iteration. The absence of the data fidelity term alters the upper-left matrix of the KKT system in (4.18). Therefore we have to set $\boldsymbol{P} = \boldsymbol{I}_M \otimes 2\lambda_1 \bar{\boldsymbol{K}}$ and $\boldsymbol{q} = \boldsymbol{0}$ for the $\boldsymbol{C}$-update.

The $\boldsymbol{D}$-update stays similar to that derived for *RLS-TV* in (4.19) except that it is applied to the augmented graph $\mathcal{G}'$ and the extended kernel matrix $\widetilde{\boldsymbol{K}}^{\perp}$. With these modifications we effectively reduced *SVM-TV* to the *RLS-TV* problem setting and can adopt the ADMM update framework derived in Section 4.3.

4.5 Algorithm Summary

Algorithm 4.1 contains both the *RLS-TV* as well as the *SVM-TV* variant. For *RLS-TV* the statements enclosed within parentheses have to be skipped. Recall from Section 2.3

that we have to select the augmented Lagrangian parameter ρ and set appropriate stopping tolerances. In the experiments we found that setting $\rho = 1$ and the tolerances of the stopping rule (2.19) to $\epsilon^{abs} = 10^{-3}$ and $\epsilon^{rel} = 10^{-3}$ produces sufficiently accurate solutions while converging within 50 iterations for most problem instances. We also implemented the varying penalty strategy presented in [BPC$^+$11] to speed up convergence. This is not explicitly listed in Algorithm 4.1.

The bulk of Algorithm 4.1 is devoted to the initialization and construction of the KKT system for the quadratic objective of the $\tilde{\boldsymbol{c}}_m$ variables. In the loop the basic ADMM scheme is applied (i.e., update primal variables $\boldsymbol{C}$ → update primal variables $\boldsymbol{D}$ → update dual variables $\boldsymbol{U}$). After the algorithm reaches the stopping criterion (2.19) or the maximum number of iterations t_{max} is reached the final clustering can be obtained by selecting the index of the largest entry of each row in the class affiliation matrix $\widehat{\boldsymbol{X}}$ as class label, i.e,

$$\hat{y}_i = \underset{m \in \{1,\ldots,M\}}{\operatorname{argmax}} \widehat{X}_{im}. \tag{4.27}$$

Algorithm 4.1: RLS-TV (SVM-TV) algorithm

```
**Data:** $\mathcal{G}, \boldsymbol{K}, \boldsymbol{Y}, \lambda_1, \lambda_2$
**Result:** $\widehat{\boldsymbol{X}}$
$t \leftarrow 0$ $\rho \leftarrow 1$
(SVM-TV: augment graph $\mathcal{G}$ as in Section 4.4.1)
(SVM-TV: extend kernel $\boldsymbol{K}$ as in (4.23))
$\boldsymbol{P} \leftarrow \boldsymbol{I}_M \otimes (\frac{2}{|\mathcal{L}|}\boldsymbol{K}^T\boldsymbol{J}\boldsymbol{K} + 2\lambda_1\boldsymbol{K})$
$\boldsymbol{q} \leftarrow (\boldsymbol{I}_M \otimes (-\frac{2}{|\mathcal{L}|}\boldsymbol{K}^T\boldsymbol{J}))\boldsymbol{y}$
$\boldsymbol{G} \leftarrow \boldsymbol{1}_M^T \otimes \boldsymbol{K}$
$\boldsymbol{h} \leftarrow \boldsymbol{0}$
(SVM-TV: set $\boldsymbol{P} \leftarrow \boldsymbol{I}_M \otimes 2\lambda_1\boldsymbol{K}$ and $\boldsymbol{q} \leftarrow \boldsymbol{0}$)
$\boldsymbol{KKT} \leftarrow$ construct from $\boldsymbol{P}, \boldsymbol{q}, \boldsymbol{G}, \boldsymbol{h}$ according to (4.17)
$\boldsymbol{C}^0 = \boldsymbol{0}$
(SVM-TV: set anchor coefficients of $\boldsymbol{C}$ as in (4.24))
$\boldsymbol{D}^0 \leftarrow \nabla_{\mathcal{G}}\boldsymbol{C}^0$
$\boldsymbol{U}^0 \leftarrow \boldsymbol{0}$
**while** *criterion (2.19) not satisfied* **do**
    $\boldsymbol{C}^{t+1} \leftarrow \boldsymbol{KKT}^{-1}(\rho\nabla_{\mathcal{G}}^T(\boldsymbol{D}^t - \boldsymbol{U}^t) - \boldsymbol{q})$
    (SVM-TV: enforce anchor coefficients in $\boldsymbol{C}^{t+1}$ as in (4.24))
    $\boldsymbol{V} \leftarrow \nabla_{\mathcal{G}}\boldsymbol{C}^{t+1} + \boldsymbol{U}^t$
    $\boldsymbol{D}^{t+1} \leftarrow \mathit{soft}_{[0,\lambda_2/\rho]}(\boldsymbol{V} - \boldsymbol{\Delta}) + \boldsymbol{\Delta}$
    $\boldsymbol{U}^{t+1} \leftarrow \boldsymbol{U}^t + \nabla_{\mathcal{G}}\boldsymbol{C}^{t+1} - \boldsymbol{D}^{t+1}$
    $t \leftarrow t + 1$
**end**
$\widehat{\boldsymbol{X}} \leftarrow \boldsymbol{K}\boldsymbol{C}^t$
```

CHAPTER 5

Experiments

In this chapter we perform numerical experiments to quantify the behavior of the algorithms derived in Chapter 4 as well as to assess their accuracy in comparison to existing state of the art algorithms. For a succinct presentation of the results we introduce the following abbreviations. Our methods from Section 4.3 and Section 4.4 will be termed *RLS-TV* and *SVM-TV*, respectively. The naming of the manifold regularization algorithms from Belkin et. al [BNS04] is set to *LapRLS* and *LapSVM*. For the regularized TV minimization algorithm of [BDHM19] we use *TVMinReg* and the unsigned TV minimization algorithm of [BLUVB13] we term *TVBresson*.

For the evaluation we generate two-dimensional Euclidean data according to a two moon and a multi spiral model. The formulation of the two moon model is taken from [BDM18] and is based on randomly sampling a cluster label $l_i \in \{-1, 1\}$ for each of the N nodes. The coordinates of node v_i are then computed as follows

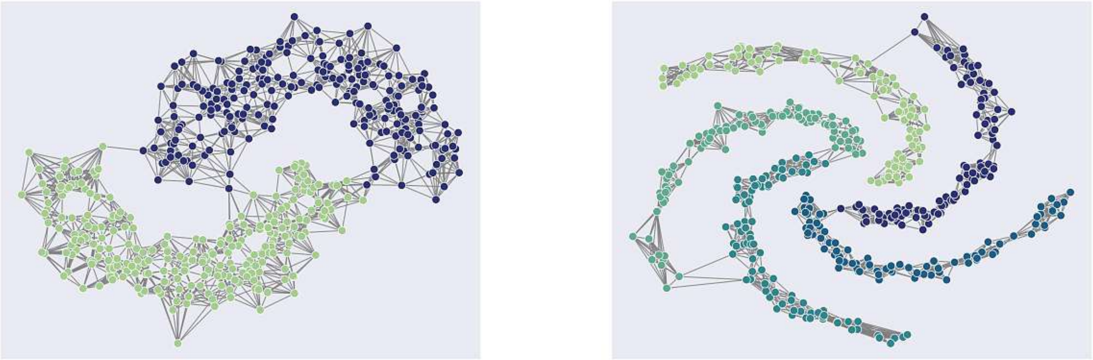
$$\boldsymbol{x}_i = \begin{pmatrix} \frac{l_i}{2} \\ 0 \end{pmatrix} + \begin{pmatrix} \cos\phi_i \\ l_i \sin\phi_i \end{pmatrix} + \boldsymbol{w}_i. \tag{5.1}$$

Where $\phi_i \sim \mathcal{U}(0, \pi)$ is a random angle and $\boldsymbol{w}_i \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$ is two dimensional white Gaussian noise. The multi spiral model for M clusters is given by [BDHM19] as

$$\boldsymbol{x}_i = \frac{1}{4} \begin{pmatrix} (M^2\psi_i/\pi + 4)\cos(\psi_i + l_i 2\pi/M) \\ (M^2\psi_i/\pi + 4)\sin(\psi_i + l_i 2\pi/M) \end{pmatrix} + \boldsymbol{w}_i, \tag{5.2}$$

where the cluster labels are randomly drawn from the set $l_i \in \{0, \ldots, M-1\}$ and $\psi_i \sim \mathcal{U}(0, 4\pi/M)$ is again a random angle.

A graph is then constructed from these coordinates by connecting each node to its k-nearest neighbors, hence the name KNN graph. In order to form an undirected graph we take the maximum of the resulting adjacency matrix with its transpose. The weights of the graph edges are determined by the distance between each pair of adjacent nodes. We used the Gaussian weight function $W_{ij} = \exp\left(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2/(2\kappa^2)\right)$ in our experiments. Thus, for nodes that are close the weight approaches 1 in contrast to nodes that are far apart where the weight is tending to 0. Graphs constructed in this fashion are similarity graphs where the presence and strength of a connection represents closeness of the data points in the feature space. Examples of synthetic similarity graphs derived from the two moon and multi spiral model are depicted in Fig. 5.1.



(a) Two moon model

(b) Multi spiral model

Figure 5.1: KNN graph examples

Throughout the experiments, if not explicitly stated otherwise, we fix the number of nodes $N = 500$ and the number of clusters for the multi spiral model to $M = 5$. The clusters of both models are of equal size. The number of neighbors for the KNN graphs is fixed with $k = 10$. The parameter for the Gaussian weight function of the similarity edges is set to $\kappa = 0.6$ and the noise variance is set to $\sigma^2 = 0.3$. Sampling of labels is performed such that at least one label per cluster is know and the remaining labels are drawn uniformly among the clusters. Finally results are reported via mean error rates that are calculated by dividing the number of all falsely estimated labels by the total number of nodes and results are averaged over 100 Monte Carlo iterations.

5.1 Parameter Selection

To identify suitable parameters for the *RLS-TV* and *SVM-TV* algorithms we performed a grid search over the regularization parameters λ_1 and λ_2. These experiments were conducted on unsigned KNN graphs. The RBF kernel width was set to 0.6 in accordance with the Gaussian edge weight parameter of the KNN graph. The search process was performed on a logarithmic grid with 10^p for $p \in \{-9, -8, \ldots, 1\}$. We repeated the parameter search for each model with $|\mathcal{L}| \in \{M, 10M\}$ labels and report the mean error rate in Fig. 5.2.

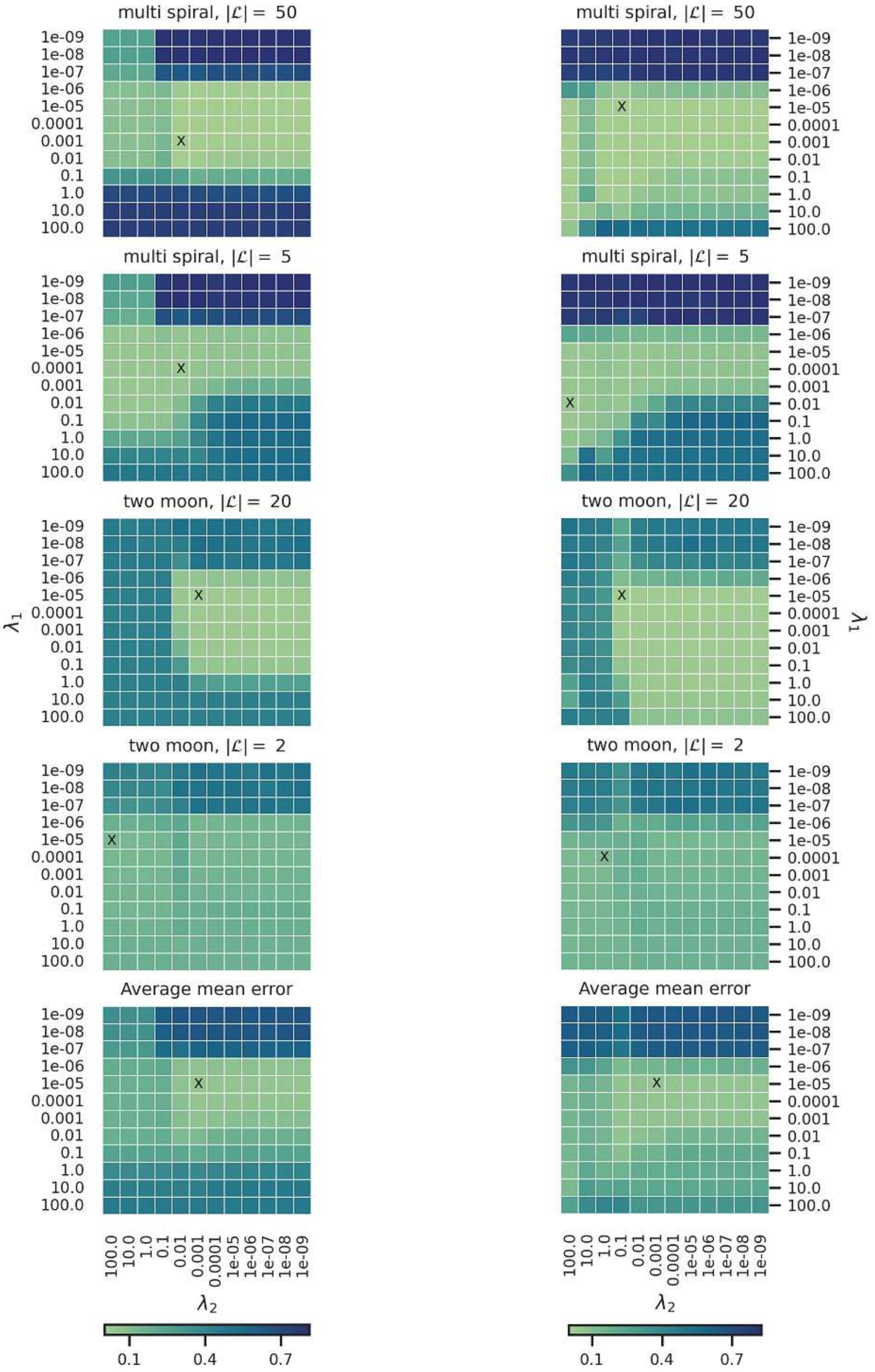


Figure 5.2: Mean error rates for *RLS-TV* (left) and *SVM-TV* (right) averaged over 100 realizations of the two moon model with $|\mathcal{L}| \in \{2, 20\}$ and 100 realizations of the multi spiral model with $M = 5$ classes and $|\mathcal{L}| \in \{5, 50\}$, and the average over all of those cases combined. For each combination of algorithm and model, the position with minimal error rate is marked with the symbol 'X'.

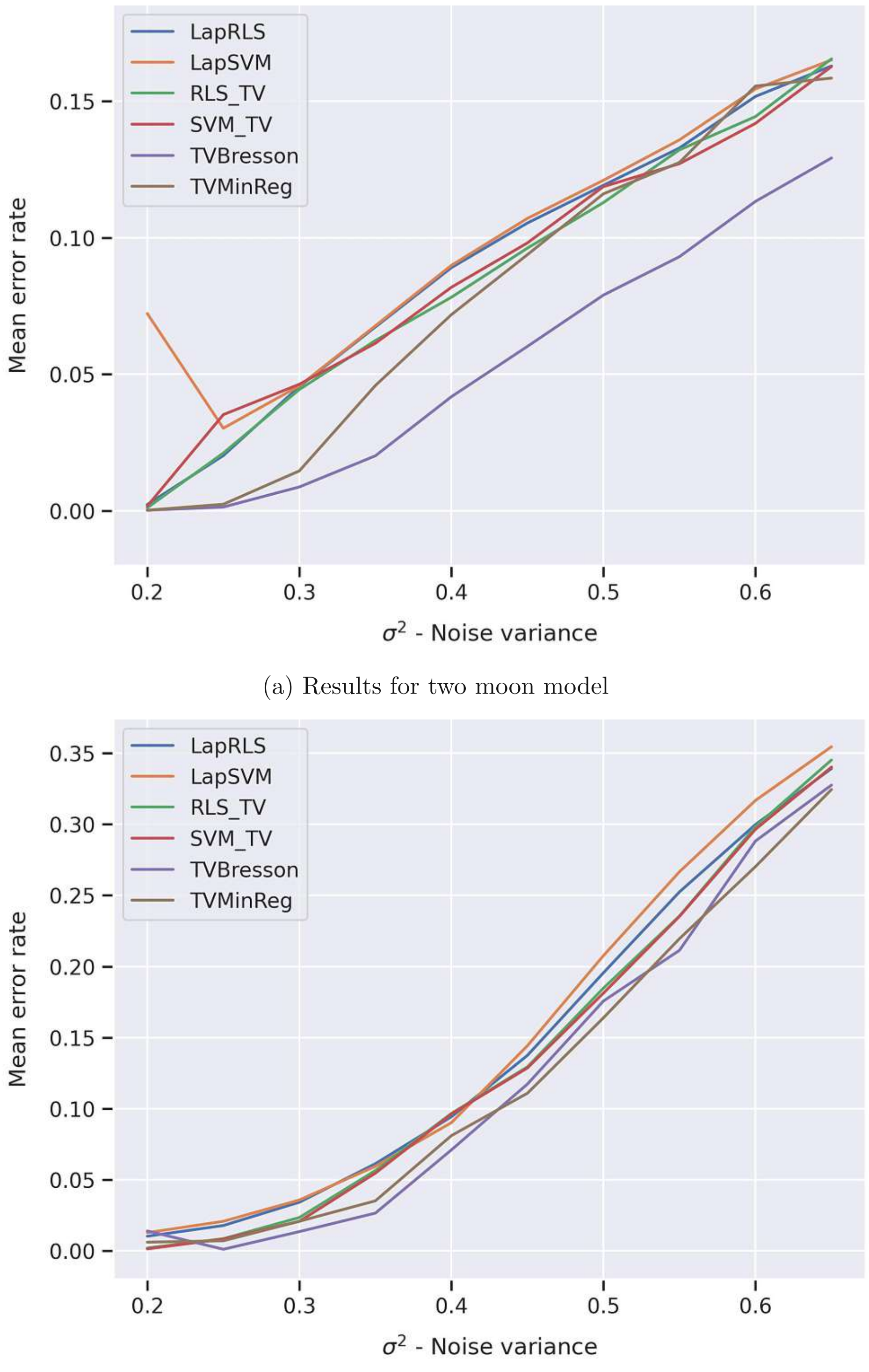
The last row of Fig. 5.2 shows the mean error rates over all model and label configurations for *RLS-TV* and *SVM-TV*, respectively. For both algorithms the results contain a uniform region of low error rate in the middle-right of the explored parameter space. The size of the regions possibly indicate good generalization abilities for unseen problem instances (e.g. different M and $|\mathcal{L}|$ settings). The parameters for *RLS-TV* and *SVM-TV* are chosen such that the average mean error rate over all configurations is minimized, which for both algorithms is given by $\lambda_1 = 10^{-5}, \lambda_2 = 10^{-3}$.

The parameters for the algorithms *LapRLS* and *LapSVM* were selected by the same grid search procedure as described above. The only exception is that we had to set the RBF kernel width to 0.1 to prevent numerical issues with the quadratic solver. The parameters resulted in $\lambda_1 = 10^{-9}, \lambda_2 = 10^{0}$ for *LapRLS* and $\lambda_1 = 10^{-8}, \lambda_2 = 10^{-2}$ for *LapSVM*.

For the *TVMinReg* algorithm the single parameter x_{min} is selected to be 0.9 as proposed by the authors in [BDM18] and [BDHM19].

5.2 Comparison on Unsigned Graphs

To compare the different algorithms we report their mean error rate over increasing noise variances in the data models. Fig. 5.3a and Fig. 5.3b show that the algorithms scale in a comparable fashion for both cases. However, *TVBresson* outperforms its contendents by a margin of about three percent on the two moon model and on the multi spiral model it's performance is in a close match with *TVMinReg*. In both experiments of Fig. 5.3 the pure total variation minimization approaches show to fit the data best. This might be due to their transductive methodology. In contrast to that the semi-supervised learning methods attempt to learn a general function that mimics the dataset. This technique is likely to fail when the variance in the data increases significantly and no additional samples are provided to the algorithms.



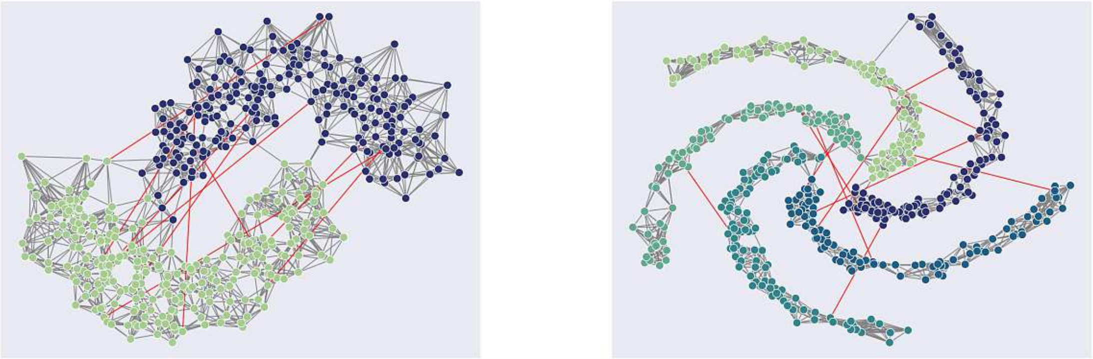
(a) Results for two moon model

(b) Results for multi spiral model

Figure 5.3: Mean error rate for varying noise variances σ^2 with two moon and multi spiral model, $|\mathcal{L}| = 10$.

5.3 Signed Graphs

In this section we augment the KNN graphs with negatively weighted edges and examine the ability of *RLS-TV* and *SVM-TV* to learn from dissimilarity information. In particular, we use the same strategy as [GZW07], which is as follows: We introduce a set of dissimilarity edges $\mathcal{D}$ which are sampled randomly by an oracle. The oracle chooses v_i and v_j among different clusters and connects them with a negative edge of weight $W_{ij} = -w_{dis}$. To prevent the propagation of label information by the oracle the selected nodes mustn't be contained in the label set (i.e. $v_i \notin \mathcal{L} \wedge v_j \notin \mathcal{L}$). Fig. 5.4 depicts examples of KNN graphs that are augmented with $|\mathcal{D}| = 10$ dissimilarity edges drawn in red.



(a) Two moon model with dissimilarity

(b) Multi spiral model with dissimilarity

Figure 5.4: KNN graph examples with ten dissimilarity edges

5.3.1 Effects of Dissimilarity Edges

As a first analysis we consider how the number of dissimilarity edges and their magnitude affects the accuracy of *RLS-TV* and *SVM-TV*. Therefore we perform a logarithmic search for $w_{dis} = 10^{p_1}$ with $p_1 \in \{-4, -3, \ldots, 2\}$ and $|\mathcal{D}| = 3^{p_2}$ with $p_2 \in \{0, \ldots, 7\}$. For the plots in Fig. 5.5 and Fig. 5.6 we set the number of known labels to $|\mathcal{L}| = 10$ and increased the noise variance of the models to $\sigma^2 = 0.5$. The plots for the two moon and multi spiral model are presented in Fig. 5.5 and Fig. 5.6 respectively. The surface plots show how the mean error rates vary for different dissimilarity settings.

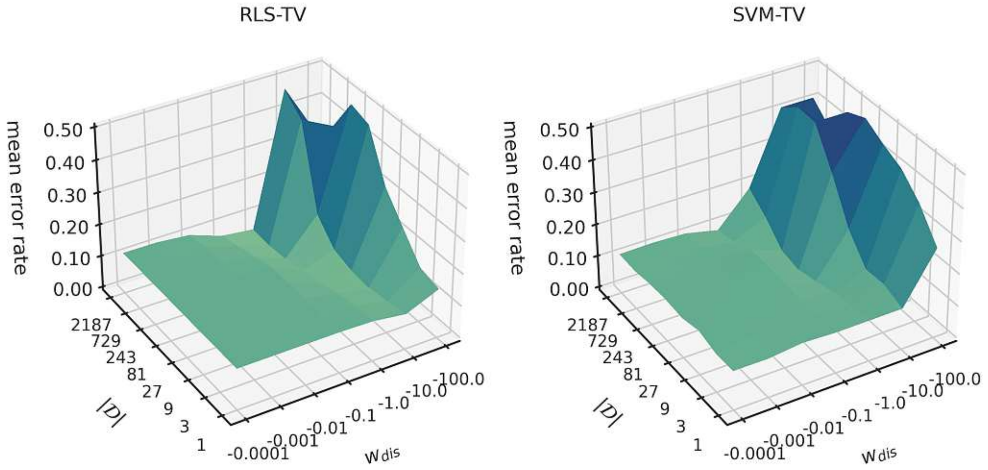


Figure 5.5: Mean error rates for *RLS-TV* and *SVM-TV* over different combinations of $|\mathcal{D}|$ and w_{dis} on the two moon model.

The unsigned KNN graphs for two moon and multi spiral model contain around 3000 edges and the highest edge weights do not exceed 1. In light of this information it seems reasonable that the minimum error rates are achieved for dissimilarity edge weights with $w_{dis} = -1$. According to the ridges in Fig. 5.5 and Fig. 5.6 dissimilarity edge counts between 100 and 1000 yield good performance improvements. This suggests that excessively increasing the weight or number of the negative edges might incur numerical instabilities or diminish focus on similarity information. In the best cases the error rate for *RLS-TV* and *SVM-TV* is about 2.5 times lower than in the absence of any dissimilarity edges.

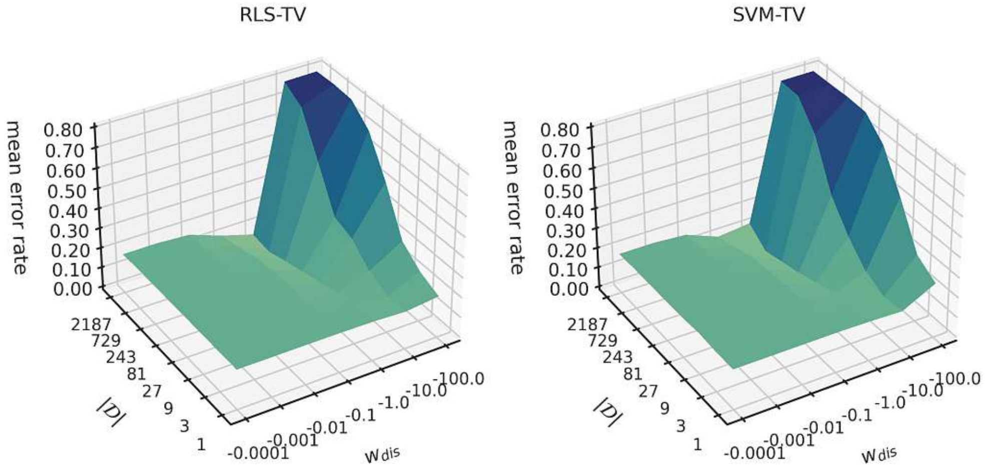
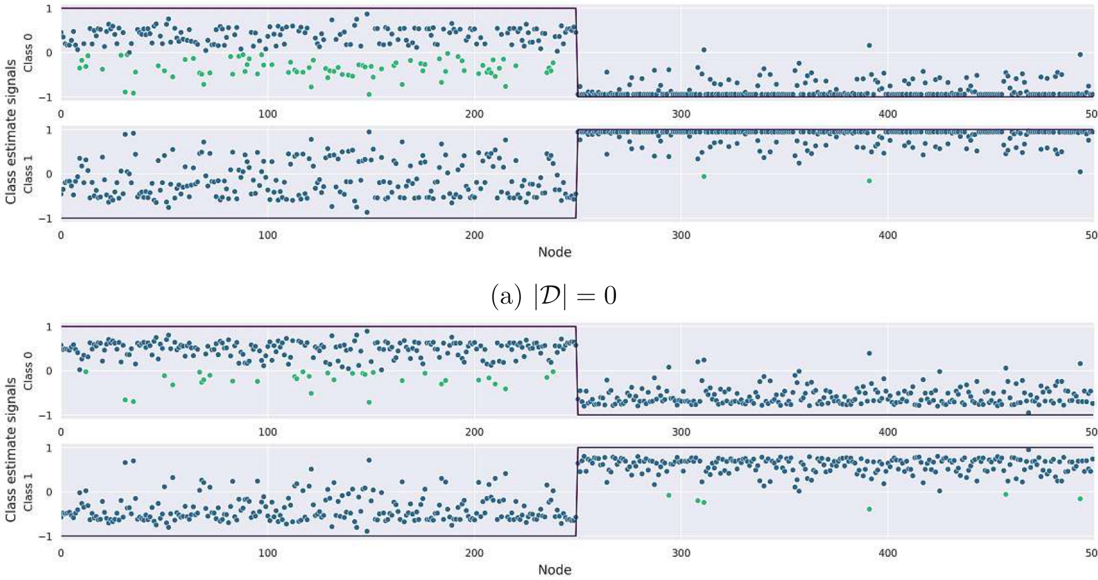


Figure 5.6: Mean error rates for *RLS-TV* and *SVM-TV* over different combinations of $|\mathcal{D}|$ and w_{dis} on the multi spiral model.



(a) $|\mathcal{D}| = 0$

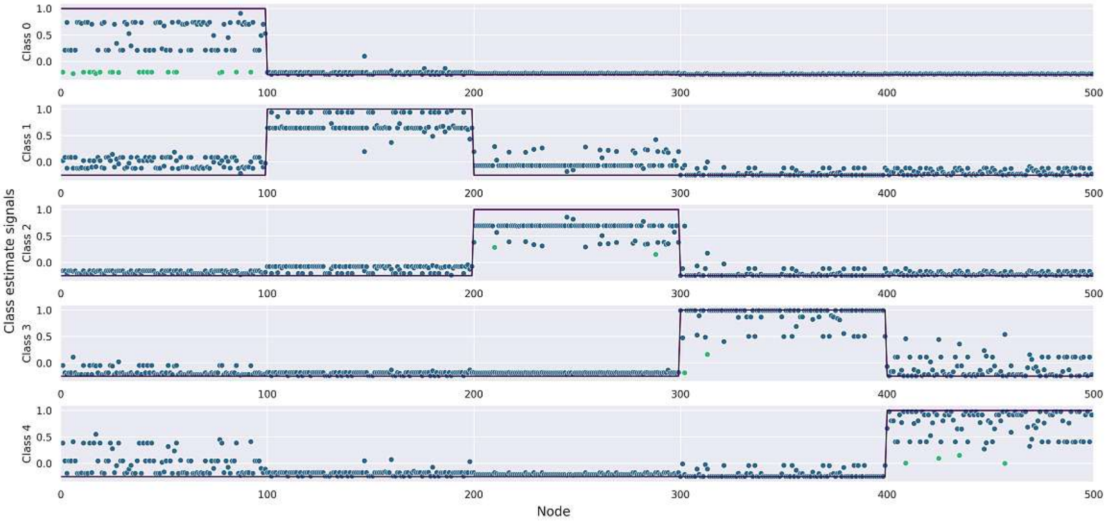
(b) $|\mathcal{D}| = 100$ with $w_{dis} = -1$

Figure 5.7: Estimated class affiliation signals for two moon dataset to showcase dissimilarity affecting *RLS-TV*. Green nodes are attributed to a wrong class and the purple lines indicate ground truth signals.

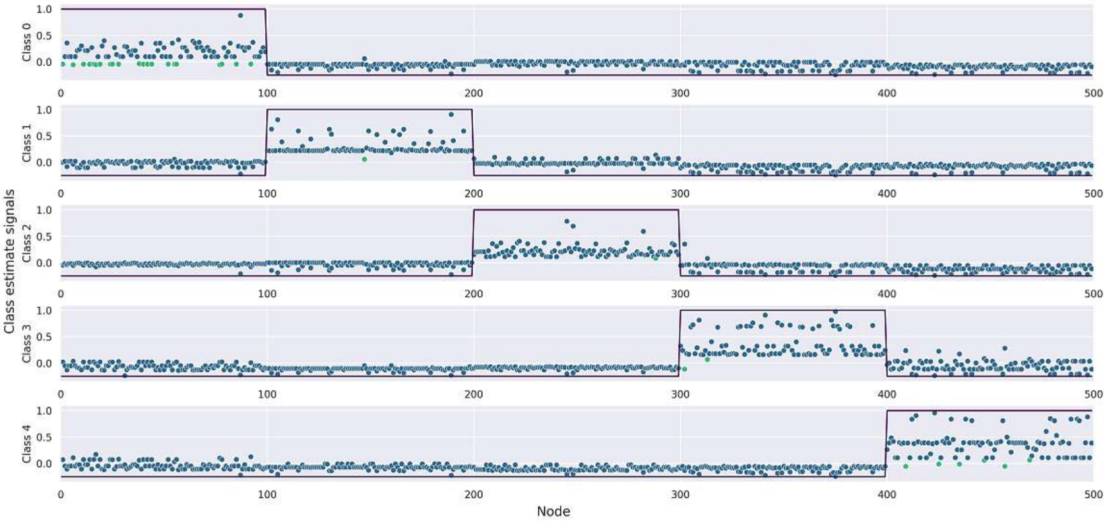
While conducting the experiments we observed that introducing dissimilarity in the models can impede the discrimination abilities of *RLS-TV* and *SVM-TV*. To quantify this effect we plotted the complete class affiliation estimates for $|\mathcal{D}| = 0$ and $|\mathcal{D}| = 100$ with $w_{dis} = -1$. Since the observations were similar for both algorithms we only include the results for *RLS-TV*.

Fig. 5.7 shows how including dissimilarity information increases the accuracy of the prediction (observe that Fig. 5.7a contains less green nodes than Fig. 5.7b). However, it also has the effect of pulling the class estimate signal towards zero. In the case of multiple classes, as shown in Fig. 5.8, this detrimental effect is even stronger.

Because the total variation constitutes a convex relaxation of the underlying discrete problem a valid solution is obtained by setting all class estimate signals to zero. The experiments suggest that the algorithms *RLS-TV* and *SVM-TV* are not able to fully utilize additional dissimilarity information but rather introduce a trade off with similarity information. Another possible explanation is that the classification functions are based on RBF kernels which only register similarity information, therefore regularizing with dissimilarity information could confuse the learning process through opposing incentives.



(a) $|\mathcal{D}| = 0$



(b) $|\mathcal{D}| = 100$ with $w_{dis} = -1$

Figure 5.8: Estimated class affiliation signals for multi spiral dataset to showcase dissimilarity affecting *RLS-TV*. Green nodes are attributed to a wrong class and the purple lines indicate ground truth signals.

5.3.2 Parameter Validation

To validate our parameter selection of Section 5.1, which was based on unsigned graphs, we conducted the same grid search procedure on signed graphs aswell. The aim of these simulations is to evaluate if the previously selected parameters are still appropriate in the presence of dissimilarity information. In Fig. 5.9 we can identify large regions of low error rate in the middle left of all cases. This is very similar to the unsigned results. Indeed, for *RLS-TV* we can identify the same optimal parameters as compared with Fig. 5.2. Only the *SVM-TV* algorithm would benefit from reducing λ_1 and λ_2 by around 1% lower error rates. Nevertheless, it seems that our algorithms in combination with the parameter selected in Section 5.1 are able to generalize to problems on signed graphs.

5.3.3 Comparison on Signed Graphs

To conclude our experiments on the effect of dissimilarity edges we perform the same comparison as in Section 5.2 with signed KNN graphs. The plots in Fig. 5.10 show the mean error rates for increasing noise variances on the two moon and multi spiral models augmented with $|\mathcal{D}| = 100$ dissimilarity edges of weight $w_{dis} = -1$. Note that for the two moon experiments we used the algorithms *LapRLSd* and *LapSVMd* from [GZW07] which extend *LapRLS* and *LapSVM* with the signed Laplacian (cf. Chapter 3). For the multi spiral model we did not use *LapRLSd* and *LapSVMd* because of their computational complexity and only modest accuracy gains. The results for *TVBresson* are obtained on the unsigned KNN graphs since this algorithm is not capable of exploiting dissimilarity information.

For both data models in Fig. 5.10 we can observe that the additional information from signed edges decreases the error rate of *RLS-TV*, *SVM-TV* and *TVMinReg* significantly over all noise levels. The dissimilarity aware algorithms *RLS-TV* and *TVMinReg* even outperform the best unsigned algorithm *TVBresson* in both experiments. The results suggest that our algorithms are both comparable to state of the art semi-supervised clustering methods on signed graphs.

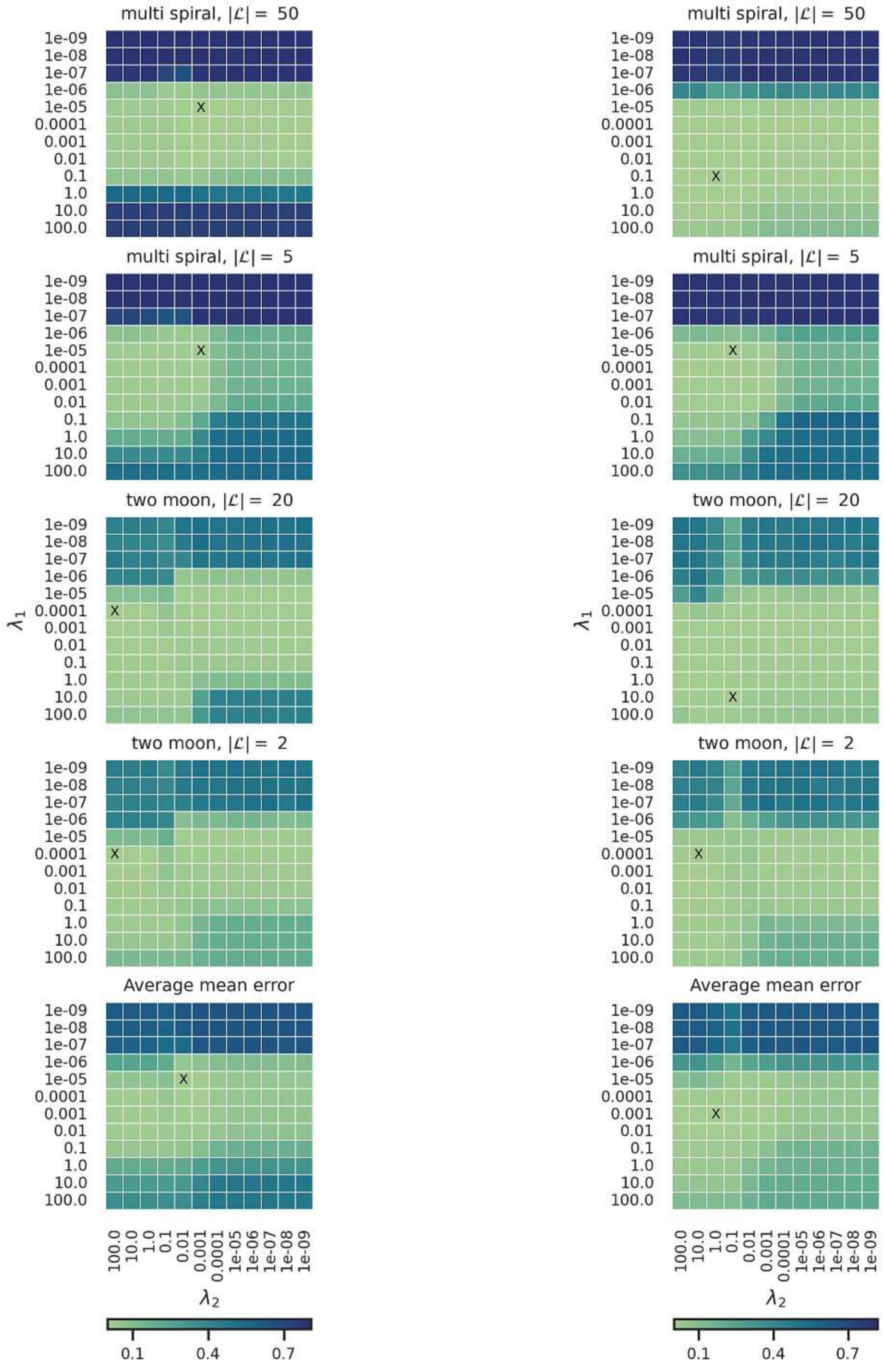
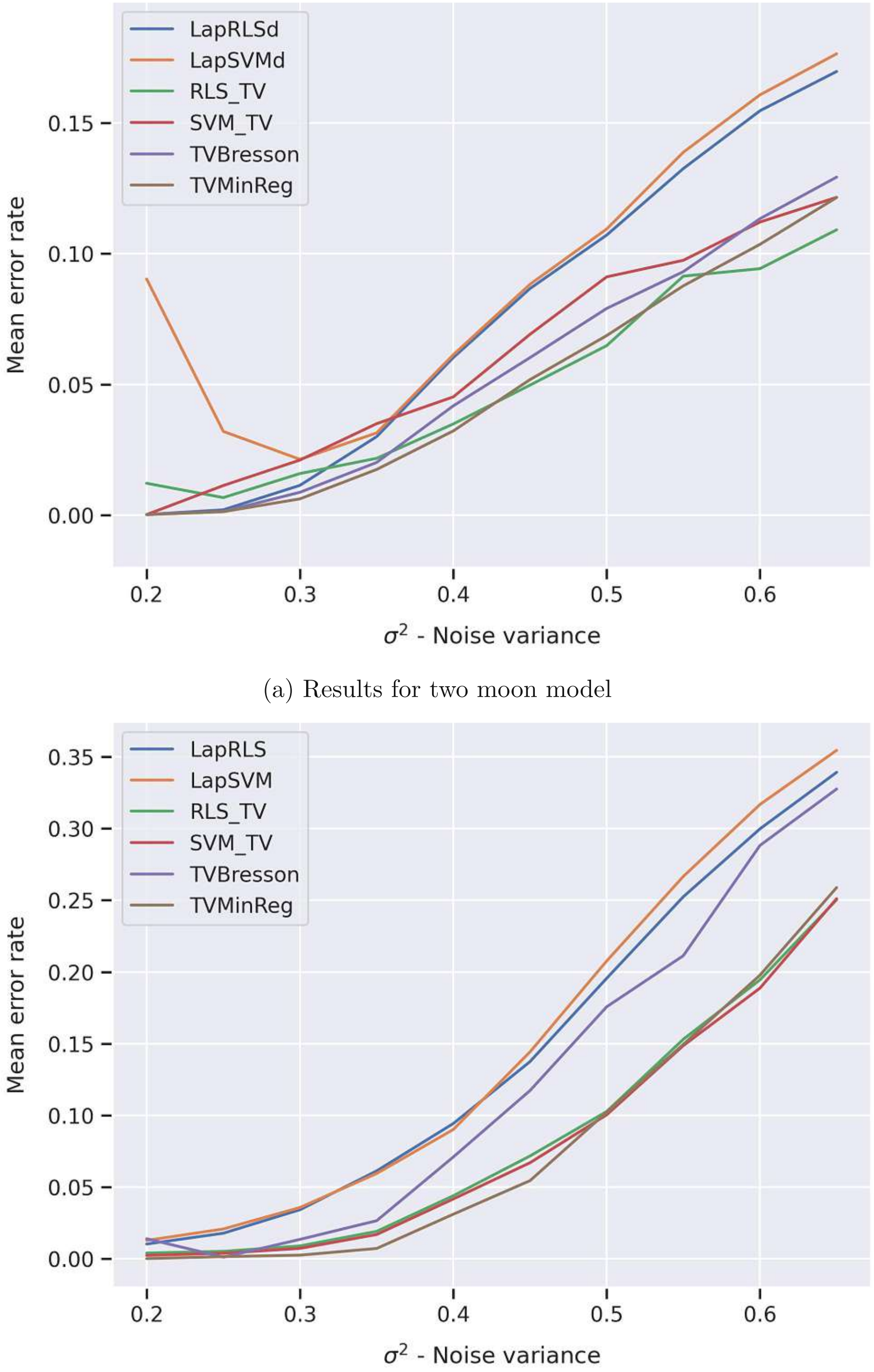


Figure 5.9: Mean error rates for *RLS-TV* (left) and *SVM-TV* (right) averaged over 100 realizations with $|\mathcal{D}| = 100$ and $w_{dis} = -1$ of the two moon model with $|\mathcal{L}| \in \{2, 20\}$ and the multi spiral model with $M = 5$ classes and $|\mathcal{L}| \in \{5, 50\}$, and the average over all of those cases combined. For each combination of algorithm and model, the position with minimal error rate is marked with the symbol 'X'.



(a) Results for two moon model

(b) Results for multi spiral model

Figure 5.10: Mean error rate for varying noise variances σ^2 with two moon and multi spiral model, $|\mathcal{L}| = 10$, $|\mathcal{D}| = 100$ and $w_{dis} = -1$.

5.4 Wikipedia Elections

In this section we will apply our algorithms on a real-world data set. We consider the Wikipedia adminship elections (*wiki-Elec* dataset) from [LHK10]. This dataset consists of elections in which the Wikipedia community decides if a given user should be promoted as administrator. To construct a signed graph from the elections we follow [MBS19] where only users which have been up for voting are considered. Furthermore two users are connected with an undirected edge if they appear in at least one common election. The edge weight is set to 1 or -1 if the average of all votes between the two users was supporting or opposing respectively. From the resulting signed network the largest connected component is extracted and yields the final graph. In summary the graph has 2325 nodes (users) of which 52.6% have received the adminship status. The nodes are connected by more than 100.000 edges of which around 77% are positively weighted. The ground truth labels are established by the actual election results and are sampled in an uniform random fashion by ensuring that at least one label per class is selected.

In order to assess the performance of our algorithms we clustered the *wiki-Elec* graph for different amounts of known labels. The main parameters were kept as derived from the parameter search in Section 5.1. However due to the lack of a Euclidean node embedding we can not deploy the RBF kernel as in the previous experiments. Therefore we used the heat diffusion kernel as presented in Section 2.2.2 with the parameter $\alpha = 1$ arbitrarily selected. The resulting kernel matrix is densely populated and thus leads to computational intensive processes. As we observed that convergence was slow while increasing iterations did not substantially enhance accuracy, we reduced the stopping tolerances of the ADMM algorithm to $\epsilon^{abs} = 10^{-2}$ and $\epsilon^{rel} = 10^{-2}$.

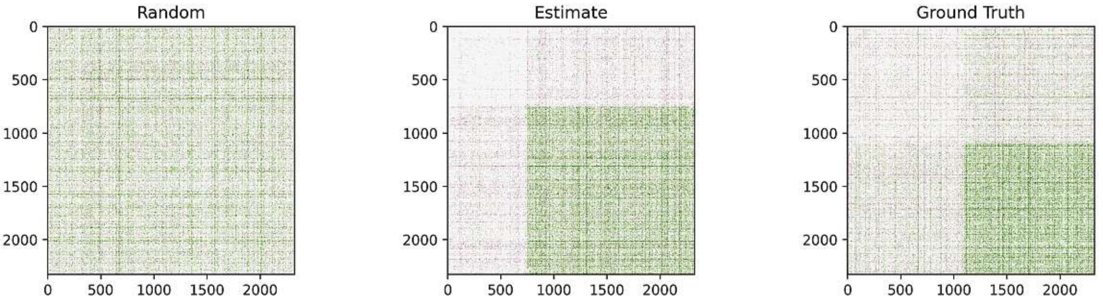


Figure 5.11: Weight matrix of Wikipedia Elections graph sorted by cluster labels, green (red) cells represent positive (negative) weights.

Fig. 5.11 depicts visualizations of the graph's weight matrix where the entries are sorted according to example clusterings. The left most clustering is obtained by uniform randomly sampling labels $\{0, 1\}$ for each node. The middle clustering is obtained by *RLS-TV* with $|\mathcal{L}| = 233$. The right most clustering presents the actual election results. It can be observed that the true clustering is quite unbalanced (i.e. it contains negative edges within and positive edges across the clusters). Examining the result of *RLS-TV* it appears as though the algorithm forms an outlier cluster with generally very sparsely connected nodes and an inlier cluster with strongly positive connected nodes.

We compare our results to the diffuse interface method of Mercado et al. [MBS19] which we reviewed in Chapter 3. We selected their two best methods *GL*(L_{SN}) and *GL*(L_{AM}), which utilize different combinations of signed and unsigned graph Laplacians. Table 5.1 reports the respective results of [MBS19, Table 2] which are averaged over 10 realizations of randomly selected labels. The entries *RLS-TV* and *SVM-TV* in Table 5.1 report the mean accuracies of our algorithms for varying percentages of labeled nodes over 100 Monte Carlo iterations. Note that the results in Table 5.1 are training set accuracies, i.e., correct predicted unlabeled nodes divided by the total number of unlabeled nodes. This metric emphasizes the improvement for increasing prior knowledge. Despite the fact that the *GL*(L_{AM}) method outperforms our algorithms for all label configurations, *RLS-TV* and *SVM-TV* yield accuracies of more than 84% for only 10% of labeled nodes. Furthermore their improvement is scaling with additional labeled nodes in a comparable fashion to *GL*(L_{SN}).

Algorithm \Labels	1%	5%	10%	15%
RLS-TV	0.811	0.813	0.841	0.849
SVM-TV	0.813	0.831	0.841	0.849
GL(L_{SN})	0.806	0.842	0.851	0.852
GL(L_{AM})	**0.879**	**0.885**	**0.887**	**0.887**

Table 5.1: Mean classification accuracies of the algorithms for different percentages of labeled nodes on the *wiki-Elec* dataset.

CHAPTER 6

Conclusion

In this thesis we reviewed the concepts of semi-supervised learning and its relation to smoothness metrics on graphical models. Furthermore we presented state of the art methods in which the total variation is used as a performant objective for clustering on graphs. Hence, we proposed to utilize the total variation regularizer in empirical risk minimization for clustering on signed graphs. To showcase the effectiveness of this approach we developed the two algorithms *RLS-TV* and *SVM-TV*. We further employed a variable separation scheme that allows us to formulate our convex optimization objectives in an ADMM admissible form. The required numerical update procedures were derived and the full algorithm implementation was presented.

In a collection of experiments on graphs derived from synthetic data models we showed that combining the total variation with empirical risk minimization indeed leads to superior results when compared with previous methods that rely on the quadratic Laplacian form for regularization. We conducted experiments on unsigned and signed KNN graphs to visualize parameter trade-offs and analyze detrimental effects of large negative edge weights. The thorough tests showed that *RLS-TV* and *SVM-TV* exhibit comparable accuracy with state of the art algorithms. We concluded the experiments with clustering a signed graph obtained from real world dataset. Although our methods retrieved qualitatively reasonable clusterings for the severely unbalanced social network they could not outperform novel diffuse interface methods.

6.1 Future Work

During the work on this thesis we encountered several interesting questions for which we could not find the capacity to thoroughly examine them. Therefore we present them in the following as proposals for future research on the topic of semi-supervised clustering on signed graphs.

A major criterion of modern machine learning algorithms is scalability with the ever growing amount of data. Although our implementations demonstrated to be capable of handling large scale problems there remains space for improvement. In particular the factorization of the KKT system constitutes a computationally intensive process. Since ADMM is an iterative scheme relying on an approximate solution for the KKT system (eg. through gradient descent methods) might decrease computational demands while preserving sufficient over all accuracy of the results.

Our algorithms rely on the basic heuristic that each node should be assigned to the cluster with the maximum entry in the estimated class affiliation matrix. A promising method that borrows from spectral clustering is to view the rows of the class affiliation matrix as Euclidean vectors and utilize a k-means algorithm to retrieve the final cluster labeling. This approach might significantly boost clustering accuracy while incurring only a marginal increase in computation.

Throughout the experiments we could not identify any significant performance differences between *RLS-TV* and *SVM-TV*. It would be interesting to test if the support vector implementation is beneficial for noisy sets of labels or for severely imbalanced clusters in order to justify its slightly more complex structure.

Finally our formulated objectives do not incorporate constraints on the cluster sizes. While introducing less bias to the learning process this can lead to degenerate solutions especially in cases where only few labels are present. There exist several possible regularization techniques in the literature to effectively overcome this issue. However, merging such methods with our algorithms might not be a trivial task.

List of Figures

List of Algorithms

List of Tables

Bibliography

[BBV04] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[BDHM19] Peter Berger, Thomas Dittrich, Gabor Hannak, and Gerald Matz. Semi-supervised multiclass clustering based on signed total variation. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4953–4957. IEEE, 2019.

[BDM18] Peter Berger, Thomas Dittrich, and Gerald Matz. Semi-supervised clustering based on signed total variation. In *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 793–797. IEEE, 2018.

[BF12] Andrea L Bertozzi and Arjuna Flenner. Diffuse interface models on graphs for classification of high dimensional data. *Multiscale Modeling & Simulation*, 10(3):1090–1118, 2012.

[BLUVB13] Xavier Bresson, Thomas Laurent, David Uminsky, and James Von Brecht. Multiclass total variation clustering. *Advances in Neural Information Processing Systems*, 26, 2013.

[BNS04] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from examples. 2004.

[BPC$^+$11] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

[CP11] Patrick L Combettes and Jean-Christophe Pesquet. Proximal splitting methods in signal processing. In *Fixed-point algorithms for inverse problems in science and engineering*, pages 185–212. Springer, 2011.

[CSZ06] Oliver Chapelle, Bernhard Schölkopf, and Alexander Zien. *Semi-Supervised Learning*. MIT press, 2006.

[CWWK20] Fenxiao Chen, Yun-Cheng Wang, Bin Wang, and C-C Jay Kuo. Graph representation learning: a survey. *APSIPA Transactions on Signal and Information Processing*, 9, 2020.

[Dav67] James A Davis. Clustering and structural balance in graphs. *Human relations*, 20(2):181–187, 1967.

[DM20] Thomas Dittrich and Gerald Matz. Signal processing on signed graphs: Fundamentals and potentials. *IEEE Signal Processing Magazine*, 37(6):86–98, 2020.

[GZW07] Andrew B Goldberg, Xiaojin Zhu, and Stephen Wright. Dissimilarity in graph-based semi-supervised classification. In *Artificial Intelligence and Statistics*, pages 155–162. PMLR, 2007.

[KL02] Risi Imre Kondor and John Lafferty. Diffusion kernels on graphs and other discrete structures. In *Proceedings of the 19th international conference on machine learning*, volume 2002, pages 315–322, 2002.

[KSL$^+$10] Jérôme Kunegis, Stephan Schmidt, Andreas Lommatzsch, Jürgen Lerner, Ernesto W De Luca, and Sahin Albayrak. Spectral analysis of signed graphs for clustering, prediction and visualization. In *Proceedings of the 2010 SIAM international conference on data mining*, pages 559–570. SIAM, 2010.

[LAH07] Jure Leskovec, Lada A Adamic, and Bernardo A Huberman. The dynamics of viral marketing. *ACM Transactions on the Web (TWEB)*, 1(1):5–es, 2007.

[LHK10] Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. Signed networks in social media. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 1361–1370, 2010.

[LLDM09] Jure Leskovec, Kevin J Lang, Anirban Dasgupta, and Michael W Mahoney. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics*, 6(1):29–123, 2009.

[MBS19] Pedro Mercado, Jessica Bosch, and Martin Stoll. Node classification for signed social networks using diffuse interface methods. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 524–540. Springer, 2019.

[SB10] Arthur Szlam and Xavier Bresson. Total variation, cheeger cuts. In *ICML*, 2010.

[SNF$^+$13] David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE signal processing magazine*, 30(3):83–98, 2013.

[SSB+02] Bernhard Schölkopf, Alexander J Smola, Francis Bach, et al. *Learning with kernels: support vector machines, regularization, optimization, and beyond.* MIT press, 2002.

[VL07] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[ZG06] Xiaojin Zhu and Andrew Goldberg. Semi-supervised regression with order preferences. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 2006.