International Journal of Machine Learning and Cybernetics manuscript No.
(will be inserted by the editor)

Similarity graph-based max-flow and duality approaches for semi-supervised data classification and image segmentation

Ekaterina Merkurjev

Received: date / Accepted: date

Abstract The max-flow problem entails the computation of a maximum feasible flow from a source to a sink through a network under constraints. Its connection to total variation presents an opportunity to apply the problem to machine learning tasks by incorporating a similarity graph-based setting. In this paper, we integrate max-flow and duality techniques, similarity graph-based frameworks, semi-supervised procedures, class size information and class homogeneity terms to derive three algorithms for machine learning tasks, such as classification, and image segmentation. The first algorithm involves similarity graph-based max-flow incorporating supervised constraints and class size information. The second method involves a duality approach and global minimization of similarity graph-based total variation problems incorporating class size information. The third algorithm involves graph-based convex optimization via max-flow techniques for image segmentation problems involving region parameters, in the case the latter is unknown. An important advantage of the methods is that they require only a small set of labeled samples for good accuracy, in part due to the integration of graph-based and semi-supervised techniques; this is an important advantage due to the scarcity of labeled data. Moreover, some of the proposed algorithms are based on global minimization, and are also able to incorporate class size information, which often improves performance. In addition, the methods perform well on both large and small data sets, the latter of which can result in poor performances for learning methods due to a decreased ability to learn from observed data. The proposed methods are validated using benchmark data sets and are compared favorably to recent methods.

Keywords similarity graph · max-flow · classification · image segmentation · semi-supervised techniques

Ekaterina Merkurjev
Department of Mathematics and CMSE
Michigan State University, MI 48824, USA
E-mail: merkurje@msu.edu

1 Introduction

The maximum flow (max-flow) problem [61], which involves finding a maximum feasible flow from a source to a sink through a flow network under certain constraints, has been studied in many sources, i.e. [54, 36, 37, 107, 62, 14, 24, 56, 60, 32, 42], with efficient algorithms for versions of the problem outlined in, e.g., [51, 72, 94, 55, 53, 69, 65, 77]. Moreover, the problem has been adapted and extended to different areas such as the stereo correspondence [104], image restoration [24] and the disjoint path problem [64]. Connections to total variation have been described in works such as [31], and the theory of the continuous maximum flow problem has been detailed in sources such as [39, 17, 128, 126, 127, 113, 20].

Due to the connection of the max-flow problem to total variation, its setting presents an opportunity to link graph-based learning problems to a modified maximum flow framework on a similarity graph. One advantage of a max-flow setting includes the possibility to develop a global minimization framework through which a global optimum can be found accurately. Thus, this paper will consider reformulations of graph-based learning problems in a similarity graph max-flow setting, with some reformulations being able to incorporate class size information, which often improves accuracy.

In general, machine learning tasks face several challenges. In particular, the success of many existing approaches for learning tasks, such as data classification,

is dependent on a sufficient amount of labeled samples. However, obtaining enough labeled data is difficult as it is time-consuming and expensive, especially in domains where only experts can determine labels; thus, labeled data is scarce. Overall, one of the key limitations of most existing machine learning approaches is their reliance on large labeled sets. In addition, the performance of machine learning methods can be severely affected in case of smaller data sets or data associated with areas of study where the size of the data sets is constrained by the complexity or high cost of experiments. These cases are usually associated with an insufficient number of labeled samples and a decreased ability for machine learning-based models to learn from the observed data, resulting in poor performance. These challenges call for innovative strategies in data science.

Recently, algorithms involving the graphical framework have become some of the most competitive approaches for applications ranging from image processing to social sciences. Such methods have been successful in part due to the many advantages offered by a graphical approach. For example, a similarity graph-based framework provides valuable information about the extent of similarity between data elements via a weighted similarity graph and also yields information about the overall structure of the data. Moreover, a graph setting is able to handle nonlinear structure and it also embeds the dimension of the features in a graph during weight computations, thus reducing the high-dimensionality of the problem. In addition, the graphical framework is able to incorporate diverse types of data, including images.

Inspired by the recent aforementioned successes, we address the aforementioned challenges of machine learning by integrating maximum flow and dual-based techniques, similarity graph-based settings, semi-supervised procedures, class size information and class homogeneity terms, with both labeled and unlabeled data embedded into a graph. In particular, in the experiments, the overwhelming majority of the data embedded into a graph is unlabeled data, which is often much easier and much less costly to obtain than labeled data.

Overall, in this paper, we present three graph-based methods involving maximum flow and dual-based techniques for learning tasks, such as data classification and image segmentation. The experiments on benchmark data sets indicate that the proposed algorithms are highly competitive against other established methods, while using a small amount of labeled data.

There are many advantages of our algorithms:

- The proposed methods require only small amounts of labeled data for accurate classification. In fact, in most cases, a good accuracy can be obtained with at most 1% - 2.5% of the data elements serving as labeled data. This is a crucial advantage due to the scarcity of labeled data for most applications, and due to the reliance of many data classification algorithms on large labeled sets. Our proposed algorithms are able to perform well with a low number of labeled elements in part due to the graph-based setting used and the semi-supervised techniques employed in the procedures.
- In the proposed framework, the data is embedded into a weighted similarity graph, which provides information about the extent of similarity between data elements and the overall structure of data.
- Unlike many classification methods, the proposed algorithms are able to incorporate class size information, which often improves accuracy.
- Several of proposed models are based on global minimization, which avoid local but not global optima. This allows one to accurately find the optimizer.
- The methods perform well for both large and small data sets, the latter of which can result in poor performances for existing learning techniques, due to an often insufficient number of labeled samples.
- The methods are parallelizable during coding.
- Algorithm 3 simultaneously finds both the segmentation result and the region parameters of the data cost terms, when the latter is unknown.

An important machine learning task considered in this paper involves classifying or segmenting data, where the goal is to divide the data into a number of classes or segments. In fact, data classification and segmentation is an integral part of many practical applications, such as medical diagnosis, email spam detection, object detection, video tracking, financial predictions, medical imaging, machine vision and face recognition.

The paper is organized as follows. In Section 2, we present the background, previous work and notation. In Section 3, we derive the three proposed algorithms of this paper (in Sections 3.1, 3.2 and 3.3, respectively). The results of the experiments and a discussion are presented in Section 4. We conclude in Section 5.

2 Background and previous work

2.1 Graph-based framework

The methods developed in this paper integrate a graphical framework, which consists of a graph $G = (V, E)$, where V and E are the vertices and edges, respectively. The vertices of the graph are connected by edges, and a nonnegative weight value is assigned on each edge; this value describes the extent of similarity between

the vertices the edge is connecting. The weight function $w : V \times V \to \mathbb{R}$ is computed so that it assigns smaller values for edges connecting dissimilar vertices, and bigger values for edges connecting similar vertices. *In this paper, we embed each data set in the graphical framework by associating each element of the set with a vertex in the graph.* Naturally, the embedding of data into a graph, as well as the performance of a graph-based algorithm, depends greatly on the edge weights; this section provides more details about graph construction in general, but the exact manner of weight construction for particular data sets is described in Section 6.

The use of the graphical framework offers many advantages. First, a graph-based framework provides valuable information about the extent of similarity between pairs of elements of both labeled and unlabeled data via a weighted similarity graph and also yields information about the overall structure of the data. It also provides a way to handle nonlinearly separable classes and affords the flexibility to incorporate diverse types of data. Moreover, a graph-based setting embeds the dimension of the features in the graph during weight computations, thus reducing the high-dimensionality of the problem. In addition, in image processing, the graphical setting allows one to capture texture more accurately due to the presence of non-local information.

The exact technique of computing the weight value between two elements of data depends on the data set, but first involves feature (attribute) vector construction and a distance metric chosen specifically for the data set at hand. For example, for hyperspectral data, one may choose the feature vector to be the vector of intensity values in the many bands of the image and the distance measure to be the cosine distance. For 3D sensory data, one can take the feature vector to contain both the geometric and color information, and the Euclidean distance as a distance measure; the weights can be calculated using a Gaussian function incorporating normal vectors, e.g., [15]. For text classification, popular feature extraction methods include bag-of-words and term frequency-inverse document frequency, both described in [8]. For biological data tasks, persistent homology [28] can be used for feature construction.

In particular, for image segmentation applications, each node in the graph represents a pixel of the image. The features can be constructed in different ways; one approach is to choose the feature vector to include the intensity values of a neighborhood around a pixel. Then, a chosen weight function, such as a Gaussian weight function, can be used to numerically evaluate the similarity of neighborhoods of two pixels. The weight on an edge of the resulting graph thus represent the degree of similarity between neighborhoods around the pixels in question. One can then sparsify the graph by only connecting two nodes representing two pixels with an edge if their neighborhoods are sufficiently similar.

Once the attribute (feature) information of each instance in the data set is obtained, the weight function w can be computed. There are many choices, but a commonly used weight function is the Gaussian function:

$$w(x_i, x_j) = \exp\left(-\frac{d(x_i, x_j)^2}{\sigma^2}\right), \tag{1}$$

where $d(x_i, x_j)$ represents a distance, such as the Euclidean distance or the cosine similarity measure, between attribute (feature) vectors of data elements x_i and x_j, and $\sigma > 0$. One can also consider the Zelnik-Manor and Perona weight function [95].

For some data, it is more desirable to compute the weights directly by calculating pairwise distances; in this case, the efficiency can be increased by using parallel computing or by reducing the dimension of the data. Then, a graph is often made sparse using, e.g., thresholding or a l nearest neighbors technique, resulting in graph where most of the edge weights are zero; thus, the number of computations is reduced. Overall, a nearest neighbor graph can be computed very efficiently using the *kd*-tree code of the VLFeat library [12].

For very large data sets, one can very efficiently construct an approximation to the full graph using, for example, sampling-based approaches, such as the fast Nyström Extension technique [48, 49, 22].

2.2 Differential operators on graphs

Our definitions of operators on graphs are based on [46, 23]. Consider two Hilbert spaces, $\mathcal{V}$ and $\mathcal{E}$, which are associated with the sets of vertices and edges of $G = (V, E)$, respectively, and the following inner products:

$$\langle u, \gamma\rangle_{\mathcal{V}} = \sum_{x\in V} u(x)\gamma(x),$$
$$\langle \psi, \phi\rangle_{\mathcal{E}} = \frac{1}{2}\sum_{x,y\in V} \psi(x,y)\phi(x,y)w(x,y)^{2a-1},$$

where $a \in [\frac{1}{2}, 1]$. From these definitions, we can define:

$$(\nabla u)_w(x,y) = w(x,y)^{1-a}(u(y) - u(x)).$$

We use the equation $\langle \nabla u, \phi\rangle_{\mathcal{E}} = -\langle u, \mathrm{div}_w\, \phi\rangle_{\mathcal{V}}$ to define the graph-based divergence operator:

$$(\mathrm{div}_w\, \phi)(x) = \frac{1}{2}\sum_{y\in V} w(x,y)^a(\phi(x,y) - \phi(y,x)),$$

where we have exploited symmetry $w(x,y) = w(y,x)$ of the undirected graph in the derivation of the operator. *We use $a = 1$ in the derivations of the paper.*

2.3 Semi-supervised setting

Despite the tremendous accomplishments of machine learning and deep learning, the success behind machine learning algorithms depends on a sufficient amount of labeled samples. However, obtaining enough labeled data is difficult as it is time-consuming and expensive, especially in domains where only experts can accurately determine labels. Therefore, labeled data is scarce.

However, unlabeled data is usually much easier and less costly to obtain than labeled data. Thus, it is advantageous to use a semi-supervised setting, which uses a large amount of unlabeled data and a small amount of labeled data to construct the graph. In fact, the use of unlabeled data for graph construction allows one to obtain important structural information of the data.

A semi-supervised setting involves a collection of labels $\{1, \dots, m\}$ and a small subset of labeled vertices whose ground-truth labels are known. One seeks to correctly label the remaining unlabeled points; a solution is a partition $\Sigma = (\Sigma_1, \dots, \Sigma_n)$ of the set of vertices V, where Σ_i is the set of points assigned to label i. This paper uses a semi-supervised setting, where the vast majority of data embedded into a graph is unlabeled data, which is much easier to obtain than labeled data.

2.4 Previous work

2.4.1 Overview of graph-based learning techniques

In this section, we review recent graph-based and semi-supervised methods, including approaches related to convolutional neural networks, support vector machines, neural networks, label propagation, embedding methods, multi-view and multi-modal algorithms.

Neural networks have also been extended to a graphical framework for the task of semi-supervised learning. For example, [116] describes an attention-based graph neural network. Graph partition neural networks [84], which are extensions of graph neural networks for handling large graphs, are presented in [84]. In [99], graph Markov neural networks are proposed.

Specifically, convolutional neural networks have been extended to a graphical framework for semi-supervised learning. In particular, [73] presents a scalable approach using graph convolutional networks via a convolutional architecture motivated by a localized first-order approximation of spectral graph convolutions. The work [82] develops deeper insights into the graph convolutional neural network model and addresses its fundamental limits. Moreover, a dual graph-based convolutional network approach is described in [135], while a Bayesian graph convolutional network procedure is derived in [130]. In [13], a multi-scale graph convolution model for semi-supervised node classification is presented. In the work [27], generalizations of convolutional neural networks to signals defined on more general domains using two constructions are described; one of them is based on the spectrum of the graph Laplacian matrix.

Moreover, support vector machines are applied to semi-supervised learning using a graphical framework. In [33], graph-based support vector machine methods which emphasize low density regions are derived. In [89, 21], Laplacian support vector machines are formulated. A novel Laplacian twin support vector machine for semi-supervised classification is introduced in [98].

Label and measure propagation methods include [63], which describes a transductive label propagation method that is based on a manifold assumption. Label propagation techniques and the use of unlabeled data to aid labeled data in classification are investigated in [134]. Dynamic label propagation, performing transductive learning through propagation in a dynamic process, is detailed in [117], while semi-supervised learning with measure propagation is shown in [114]. Moreover, the work [132] presents a robust inductive semi-supervised label prediction model for data classification, while [66] proposes a new transductive label propagation algorithm, called Adaptive Neighborhood Propagation (ANP), for semi-supervised data classification problems.

Embedding algorithms are also often used for semi-supervised learning. In [121], it is shown how nonlinear embedding methods for use with shallow semi-supervised learning techniques such as kernel methods can be applied to deep architectures. Other examples include [122].

Examples of multi-view methods and multi-modal algorithms include [92], which proposes a framework via the reformulation of the standard spectral learning model that can be used for multiview clustering and semi-supervised tasks. The work [91] proposes a multi-view learning model which performs semi-supervised classification and local structure learning simultaneously. Multi-modal curriculum learning is described in [58].

Other techniques for graph-based semi-supervised learning and data classification include efficient anchor graph regularization [119] and a Bayesian framework for learning hyperparameters [68]. The work [125] focuses on graph construction for semi-supervised learning and proposes a novel method based on random subspace dimensionality reduction. The work [52] presents a semi-supervised data classification algorithm that learns from dissimilarity and similarity information on labeled and unlabeled data using a novel graph-based encoding of dissimilarity that results in a convex problem. While random graph walks are used in [85], sampling theory for graph signals is utilized in [50]. The work [118]

proposes a bivariate formulation for graph-based semi-supervised learning. Lastly, reproducing kernel Hilbert spaces are integrated into the algorithm in [110].

2.4.2 Maximum flow techniques

Since the methods derived in this paper involve max-flow techniques, this section provides the background on existing work using these techniques. Specifically, the maximum flow problem [61] has been studied in many sources such as [54, 107, 24, 42], as well as papers on push-relabel methods for the max-flow problem [36], maximum flow problems in undirected graphs [37], pseudoflow algorithms for the max-flow problem [62], time bounds for the max-flow problem [14], parametric max-flow [56], max-flow problem of uncertain networks [60], and a study of pseudoflow and push-relabel algorithms for the maximum flow problem [32].

Fast algorithms for versions of the max-flow problem are described in works such as [55, 77, 94], as well as sources involving fast parametric methods for the max-flow problem [51], successive approximation techniques [53], almost-linear-time methods for undirected graphs [69], and algorithms for max-flow in undirected planar graphs [65]. Overall, the max-flow problem has been adapted and extended to areas such as the stereo correspondence problem [104], image restoration [24] and the disjoint path problem [64]. Connections to total variation have been described in, e.g., [31]. Moreover, the theory of the continuous max-flow problem has been detailed in sources such as [126, 113], as well as those describing combinatorial continuous max-flow [39], message-passing techniques for continuous max-flow [17], max-flow procedures for binary labeling [128], a continuous max-flow approach for the max-flow problem [127], and a fast continuous max-flow approach to non-convex multi-labeling problems [20].

The author's prior work in this area [15, 90] involves modifications and adaptations of the maximum flow problem for data classification. In particular, [90] considers the binary case of two classes, while [15] involves interesting applications to 3D point cloud segmentation. The work presented in this paper is inspired by and greatly expands upon the material in these papers.

2.5 Notation

The following notation will be used in this paper. We first embed the data into a similarity graph $G = (V, E)$, where the vertices V of the graph represent the combined unlabeled and labeled data. Let m be the number of classes and w be the weight function $w : V \times V \to \mathbb{R}$. Moreover, let $\{u_i\}_i$ and u be functions such that $u_i : V \to [0, 1]$, for $i = 1 : m$, and $u : V \to \mathbb{R}^m$ such that $u(x) = (u_1(x), ...u_m(x))$. In addition, let L_i and U_i be the lower and upper bounds for the size of class i of V, respectively. When no class size information is available, $L_i = 0$ and $U_i = |V|$. The goal is to divide V into m classes, where class i is denoted by the set V_i.

For the methods presented in the paper, techniques are derived to solve optimization problems involving the aforementioned variable u. Overall, the optimal $u_i(x)$ represents the probability that x belongs to class i.

3 Proposed Work

In this section, we derive our three proposed algorithms in Sections 3.1, 3.2 and 3.3, respectively.

3.1 Similarity graph-based max-flow algorithm incorporating supervised constraints and class size constraints

In this section, we derive a graph-based max-flow method, denoted by Algorithm 1, for learning problems, such as data classification. To derive our model, we start with a dual formulation and then derive equivalent max-flow problems. The notation in Section 2.5 is used.

The proposed algorithm incorporates:

- supervised constraints, i.e. class information of labeled elements.
- class size information, the use of which usually improves prediction accuracy.
- class homogeneity terms, which describe how well data elements fit to particular classes.

Regarding supervised information, the model in this section incorporates the labels of labeled data using the functions $\{k_i\}$ for $i \in \{1, ..., m\}$, where $k_i(x) = 1$ if x is a labeled element of class i and 0 otherwise.

Regarding class size information, there are two types of class size constraints incorporated by this model. The first takes the form of flexible class size constraints

$$S_i^\ell \leq ||V_i|| \leq S_i^u, \quad i \in \{1, ..., m\}, \tag{2}$$

where the size of class i, denoted by $||V_i||$, is constrained to lie between a lower bound S_i^ℓ and an upper bound S_i^u. To avoid imposing absolute upper and lower bounds on the class sizes, one can instead append a piecewise linear penalty term $\sum_{i=1}^m P_\gamma(||V_i||)$ to the optimization problem in question, where $P_\gamma(||V_i||$ is defined as:

$$P_\gamma(||V_i||) = \begin{cases} 0 & \text{if} \quad S_i^\ell \leq ||V_i|| \leq S_i^u \\ \gamma\big(||V_i|| - S_i^u\big) & \text{if} \quad ||V_i|| > S_i^u \\ \gamma\big(S_i^\ell - ||V_i||\big) & \text{if} \quad ||V_i|| < S_i^\ell. \end{cases}$$

(3)

The term $P_\gamma(||V_i||)$ penalizes the size of a class from being out of the range specified by lower and upper bounds. Overall, using class size information usually improves prediction accuracy, and, for most data sets, there is at least some information available about the class sizes, whether in the form of exact values or in the form of upper and lower bounds on the class sizes.

The class homogeneity terms, which measure how well each data element fits with each class, are incorporated using the functions $\{C_i\}$ for $i \in \{1, ..., m\}$, where $C_i(x)$ indicates the cost of assigning x to class i. One should formulate $C_i(x)$ so that it is small if x is likely to belong to class i, and large otherwise. For instance, the terms may be defined using the eigenvectors of the correlation matrix or the graph Laplacian, or using a fit to an expected value of a variable; in particular, the eigenvectors of the graph Laplacian contain information that can be used for classification. It is not necessary to use the terms; one can set $C_i(x) = 0$.

To derive Algorithm 1, we will first consider an optimization problem involving $u = (u_1, ..., u_m) : V \to \mathbb{R}^m$ from Section 2.5. Overall, the optimal $u_i(x)$ will represent the probability that x belongs to class i. Therefore, the optimization problem is equipped with the constraint $u(x) \in \Delta_+^m \quad \forall x$, where

$$\Delta_+^m = \{(z_1, ..., z_m) \in [0,1]^m \ : \ \sum_{i=1}^m z_i = 1\}. \tag{4}$$

Overall, the variable u will be constrained to lie in:

$$\Psi = \{u : V \to \mathbb{R}^m \ \text{ such that } \ u(x) \in \Delta_+^m \quad \forall x\}. \tag{5}$$

The optional class size constraints (2) or penalty term (3) (to be added to the optimization problem) can now be rewritten in terms of the variable u:

$$S_i^\ell \leq \sum_{x \in V} u_i(x) \leq S_i^u, \quad i \in \{1, ..., m\}, \tag{6}$$

$$P_\gamma(u) = \begin{cases} 0 & \text{if } \ S_i^\ell \leq \sum\limits_{x \in V} u_i(x) \leq S_i^u \\ \gamma\big(\sum\limits_{x \in V} u_i(x) - S_i^u\big) & \text{if } \ \sum\limits_{x \in V} u_i(x) > S_i^u \\ \gamma\big(S_i^\ell - \sum\limits_{x \in V} u_i(x)\big) & \text{if } \ \sum\limits_{x \in V} u_i(x) < S_i^\ell. \end{cases} \tag{7}$$

One can then consider the following multiclass classification model, where the optimal $u_i(x)$ represents the probability that x belongs to class i:

$$\min_{u \in \Psi \text{ s.t. } k_i(x) \leq u_i(x) \ \forall x \in V} \Bigg\{ E(u) = \sum_{i=1}^m \Big(\frac{1}{2} \sum_{x,y \in V} w(x,y) |u_i(y) - u_i(x)| + \sum_{x \in V} C_i(x)(u_i(x) - k_i(x)) \Big) \Bigg\}, \tag{8}$$

under optional constraints (6) or penalty term (7).

The first term in this model is a graph-based term resembling a convexified graph cut, which attempts to group data elements in a way such that the elements grouped in different classes are as dissimilar as possible; this is due to the property of the weight function $w : V \to V$ which takes large values for similar data elements and small values for dissimilar data elements.

The second term in this model incorporates the class homogeneity terms and labeled data, while the class size information is incorporated by the optional class size constraints (6) or the penalty term (7).

Note that (8) is a convexified version of a problem with the same terms, but where u is only allowed to take values in $\{0, 1\}$ as in a graph cut formulation; that problem is non-convex due to the binary constraints. It turns out that the convex relaxation indeed closely approximates the non-convex problem, where u takes only binary values. For more information on how the non-convex problem is well approximated by the convex relaxation, please refer to the sources [90, 15].

At the end, the data elements assigned to class i can be obtained using the optimal u^* via:

$$V_i = \{x \ \text{ s.t. } \ \arg\max_j u_j^*(x) = i\}.$$

Overall, it turns out that the dual problem (8) can be equivalently formulated as a 'max-flow' problem:

Theorem 1 *The following max-flow problem can be reformulated as the dual problem* (8) *with optional class size constraints* (6) *or penalty term* (7)*:*

$$\sup_{p_s, p, q, \rho^1, \rho^2} \sum_{x \in V} p_s(x) - \sum_{x \in V} \sum_{i=1}^m k_i(x) p_i(x) + \sum_{i=1}^m \big(\rho_i^1 S_i^\ell - \rho_i^2 S_i^u \big) \tag{9}$$

subject to, for all $i \in \{1, ..., m\}$,

$$|q_i(x,y)| \leq 1, \qquad \forall (x,y) \in E, \tag{10}$$

$$p_i(x) \leq C_i(x), \qquad \forall x \in V, \tag{11}$$

$$\big(\operatorname{div}_w q_i - p_s + p_i\big)(x) = \rho_i^1 - \rho_i^2, \quad \forall x \in V, \tag{12}$$

$$0 \leq \rho_i^1, \rho_i^2 \leq \gamma. \tag{13}$$

No class size information is incorporated by $\gamma = 0$*. The size penalty term* (7) *is incorporated by* $0 < \gamma < \infty$*. Size constraints* (6) *are incorporated by* $\gamma = \infty$*.*

Proof By adding an unconstrained Lagrange multiplier for the flow conservation constraint (12) to optimization problem (9), one obtains the following formulation:

$$\min_{u} \sup_{p_s,p,q,\rho^1,\rho^2} \Big\{ E(u,p_s,p,q,\rho^1,\rho^2) = \\ = \sum_{x\in V} p_s(x) - \sum_{x\in V}\sum_{i=1}^{m} k_i(x)p_i(x) + \sum_{i=1}^{m}\big(\rho_i^1 S_i^\ell - \rho_i^2 S_i^u\big) \\ + \sum_{i=1}^{m}\sum_{x\in V} u_i(x)\big(\operatorname{div}_w q_i - p_s + p_i + \rho_i^2 - \rho_i^1\big)(x)\Big\} \\ \textit{subject to } (10),\ (11) \textit{ and } (13). \tag{14}$$

We then regroup like terms to obtain:

$$\min_{u} \sup_{p_s,p,q,\rho^1,\rho^2} \Big\{ E(u,p_s,p,q,\rho^1,\rho^2) = \\ \sum_{x\in V}\sum_{i=1}^{m} u_i(x)\operatorname{div}_w q_i(x) + \sum_{x\in V}\big((1-\sum_{i=1}^{m} u_i(x))p_s(x)\big) \\ + \sum_{x\in V}\sum_{i=1}^{m}(u_i(x)-k_i(x))p_i(x) + \sum_{i=1}^{m}\rho_i^1\big(S_i^\ell - \sum_{x\in V} u_i(x)\big) \\ + \sum_{i=1}^{m}\rho_i^2\big(\sum_{x\in V} u_i(x) - S_i^u\big)\Big\} \\ \textit{subject to } (10),\ (11) \textit{ and } (13). \tag{15}$$

The optimization problems (14) and (15) satisfy the conditions of the mini-max theorem, described in sources such as Chapter 6, Proposition 2.4 of [45]. In fact, the constraint sets for q, ρ^1, ρ^2 and u are compact and convex, and the energy is convex lower semi-continuous for fixed q and concave upper semi-continuous for fixed u, indicating the existence of at least one primal-dual solution, i.e., saddle point, of finite energy value.

An important distinction between (8) and (9) is the fact that u is unconstrained in (9) and the simplex constraint on u is handled implicitly. This allows one to formulate an algorithm without projections of u, which can restrict step sizes and reduce accuracy.

Now, for a given variable u, one can rewrite the first term of (15) in the following manner:

$$\sup_{q}\Big\{\sum_{x\in V} u_i(x)\operatorname{div}_w \phi(x),\ \text{where } \phi: V\times V,\ \ \|\phi\|_\infty \le 1\Big\} \\ = \frac{1}{2}\sum_{x,y\in V} w(x,y)|u_i(y)-u_i(x)|. \tag{16}$$

In addition, the maximization with respect to p_s of (15) at the point x can be viewed as

$$\sup_{p_s(x)}\big((1-\sum_{i=1}^{n} u_i)p_s\big)(x) \ = \begin{cases} 0 & \text{if } \sum_{i=1}^{n} u_i(x) = 1 \\ \infty & \text{if } \sum_{i=1}^{n} u_i(x) \neq 1. \end{cases} \tag{17}$$

Now, if u does not satisfy the sum to one constraint at x in (8), then the primal-dual energy in (15) would be infinite, contradicting boundedness from above.

In a similar manner, the optimization with respect to p_i in (15) can be expressed as

$$\sup_{p_i(x)\le C_i(x)} (u_i(x)-k_i(x))p_i(x) = \\ \begin{cases} (u_i(x)-k_i(x))C_i(x) & \text{if } u_i(x)\ge k_i(x) \\ \infty & \text{if } u_i(x) < k_i(x), \end{cases} \tag{18}$$

which would make the energy in (15) infinite if u does not satisfy the constraints $u_i(x) \ge k_i(x)$ of (8).

It will now be shown that the flexible class size constraints (6) or the penalty term (7) can be implicitly incorporated via the two variables ρ_1 and ρ_2. In particular, for a given u, the terms of (15) involving ρ^1 and ρ^2 correspond to the following optimization problems:

$$\sup_{0\le\rho_i^1\le\gamma} \rho_i^1\big(S_i^\ell - \sum_{x\in V} u_i(x)\big) = \\ \begin{cases} 0 & \text{if } \sum_{x\in V} u_i(x) \ge S_i^\ell \\ \gamma\big(S_i^\ell - \sum_{x\in V} u_i(x)\big) & \text{if } \sum_{x\in V} u_i(x) < S_i^\ell. \end{cases} \tag{19}$$

$$\sup_{0\le\rho_i^2\le\gamma} \rho_i^2\big(\sum_{x\in V} u_i(x) - S_i^u\big) = \\ \begin{cases} 0 & \text{if } \sum_{x\in V} u_i(x) \le S_i^u \\ \gamma\big(\sum_{x\in V} u_i(x) - S_i^u\big) & \text{if } \sum_{x\in V} u_i(x) > S_i^u. \end{cases} \tag{20}$$

If $\gamma = 0$, the class sizes do not contribute to the energy; thus, the case of no class size information can be incorporated with $\gamma = 0$. When $0 < \gamma < \infty$, the terms (19) and (20) summed together is exactly equal to the size penalty term (7). When $\gamma = \infty$, the constraint set on ρ^1, ρ^2 is no longer compact, but we can apply Sion's generalization of the mini-max theorem [111], which allows either the primal or dual constraint set to be non-compact. It follows that if the size constraints (6) are not satisfied, the energy in (15) would be infinite, contradicting existence of a primal-dual solution.

Combining (16)-(20), one can see that (9) subject to (10)-(13) can be reformulated as (8) with optional class size constraints (6) or the penalty term (7). ■

The optimization problem (9) with constraints (10)-(13) has structural similarities to a max-flow problem over m copies of the graph $G = (V_1, E_1)\times ... \times (V_m, E_m)$, where $(V_i, E_i) = G$ for $i \in \{1, ..., m\}$. Overall, the aim of a maximum flow problem is to maximize the flow from a source vertex to a sink vertex under certain constraints. Moreover, $p_s(x)$ can be viewed as the flow on

the edges from the source to the vertex x in each of the subgraphs $(V_1, E_1), ..., (V_n, E_n)$, all of which have unbounded capacities. In addition, the variables $p_i(x)$ and $C_i(x)$ can be viewed as the flow and capacity on the edge from vertex x in the subgraph (V_i, E_i) to the sink. The constraint (12) is the flow conservation condition. In case of class size constraints, instead of the flow being conserved, there is a constant excess flow of $\rho_i^1 - \rho_i^2$ for each node in the subgraph (V_i, E_i). The energy in (9) is the total amount of flow in the graph.

We now formulate a method to solve (9) with constraints (10)-(13) using augmented Lagrangian theory; the technique is efficient, accurate and tolerates a wide range of step sizes since it does not involve any projections of u. Moreover, the convergence of augmented Lagrangian techniques is often guaranteed by theories, such as the ones in [47, 57]. To derive the method, we consider the augmented Lagrangian functional:

$$L = \sum_{x\in V} p_s - \sum_{x\in V}\sum_{i=1}^{m} k_i(x)p_i(x) + \sum_{i=1}^{n}(\rho_i^1 S_i^\ell - \rho_i^2 S_i^u)\big) - \frac{c}{2}\sum_{i=1}^{n}\left\|\mathrm{div}_w\, q_i - p_s + p_i + \rho_i^2 - \rho_i^1\right\|_2^2 + \sum_{x\in V} u_i(x)\big(\mathrm{div}_w\, q_i - p_s + p_i + \rho_i^2 - \rho_i^1\big)(x). \quad (21)$$

One can then formulate a procedure to minimize (21), where one alternatively maximizes (21) for the variables q, p, p_s, ρ_1 and ρ_2 and then updates the Lagrange multiplier u. In particular, our algorithm involves the following steps, where $\|f\|_2^2 = \sum_x f(x)^2$:

$$p_s^{k+1} = \arg\max_{p_s} \sum_{x\in V} p_s - \frac{c}{2}\left\|p_s - A_i^k\right\|_2^2,$$
$$\text{where } A_i^k = {p_i}^k + \mathrm{div}_w\, q_i^k - \frac{u_i^k}{c} + {\rho_i^2}^k - {\rho_i^1}^k. \quad (22)$$

$$q_i^{k+1} = \arg\max_{|q(e)|\le 1\ \forall e\in E} -\frac{c}{2}\left\|\mathrm{div}_w\, q - B_i^k\right\|_2^2,\ \forall i,$$
$$\text{where } B_i^k = {p_s}^{k+1} - {p_i}^k + \frac{u_i^k}{c} - {\rho_i^2}^k + {\rho_i^1}^k. \quad (23)$$

$$p_i^{k+1} = \arg\max_{p_i(x)\le C_i(x)\ \forall x} -\sum_{x\in V} k_i p_i - \frac{c}{2}\left\|p_i - D_i^k\right\|_2^2,\ \forall i,$$
$$\text{where } D_i^k = {p_s}^{k+1} - \mathrm{div}_w\, q_i^{k+1} + \frac{u_i^k}{c} - {\rho_i^2}^k + {\rho_i^1}^k - k_i. \quad (24)$$

$${\rho_i^1}^{k+1} = \arg\max_{0\le\rho_i^1\le\gamma} \sum_{x\in V} \rho_i^1 S_i^\ell - \frac{c}{2}\left\|\rho_i^1 - E_i^k\right\|_2^2,\ \forall i,$$
$$\text{where } E_i^k = p_i^{k+1} + \mathrm{div}_w q_i^{k+1} - \frac{u_i^k}{c} - p_s^{k+1} + {\rho^2}_i^k. \quad (25)$$

$${\rho_i^2}^{k+1} = \arg\max_{0\le\rho_i^2\le\gamma} \sum_{x\in V} -\rho_i^2 S_i^u - \frac{c}{2}\left\|\rho_i^2 - F_i^k\right\|_2^2,\ \forall i,$$
$$\text{where } F_i^k = -p_i^{k+1} - \mathrm{div}_w q_i^{k+1} + \frac{u_i^k}{c} + p_s^{k+1} + {\rho_i^1}^{k+1}. \quad (26)$$

$$u_i^{k+1} = u_i^k - c\,(\mathrm{div}_w\, q_i^{k+1} - p_s^{k+1} + p_i^{k+1} + {\rho_i^2}^{k+1} - {\rho_i^1}^{k+1}). \quad (27)$$

Optimization problems (22) and (24) can be solved easily; the closed form solutions are:

$$p_s^{k+1} = \sum_i (A_i^k + \frac{1}{c})/m, \quad (28)$$

$$p_i^{k+1}(x) = \min\{(D_i^k(x) - \frac{k_i(x)}{c}, C_i(x)\} \quad \forall x, \quad (29)$$

where A_i^k and D_i^k are denoted in (22) and (24).

The optimization problem (23) can be solved by a few steps of the projected gradient method:

$$q_i^{k+1} = \mathrm{Projection}_\eta(q_i^k + c\nabla_w(\mathrm{div}_w q_i^k - B_i^k)),$$
$$\text{where } B_i^k = {p_s}^{k+1} - {p_i}^k + \frac{u_i^k}{c} - {\rho_i^2}^k + {\rho_i^1}^k. \quad (30)$$

In (23), $\mathrm{Projection}_\eta$ is a projection operator which is defined as

$$\mathrm{Projection}_\eta(s(x,y)) = \begin{cases} s(x,y) & \text{if } |s(x,y)| \le \eta, \\ \mathrm{sgn}(s(x,y))\cdot\eta & \text{if } |s(x,y)| > \eta, \end{cases} \quad (31)$$

where sgn is the sign function.

There are extended convergence theories for the augmented Lagrangian method in the case when one of the subproblems is solved inexactly, see e.g. [47, 57]. In our experience, one gradient ascent iteration is enough.

For problems (25) and (26), in case no constraints are given on ρ^1 and ρ^2, the maximizers over the sum of the concave quadratic terms can be computed as the average of the maximizers to each individual term as

$$\mathrm{mean}\big(E_i^k + \frac{S_i^\ell}{c\,||V||}\big), \quad \mathrm{mean}\big(F_i^k - \frac{S_i^u}{c\,||V||}\big), \quad (32)$$

respectively for ρ^1 and ρ^2. Since the objective function is concave and the maximization variable is just a constant, an exact solution to the constrained maximization problems (25) and (26) can now be obtained by a

Algorithm 1

Require: m, V, $w : V \times V \to \mathbb{R}$, $\{C_i\}_{i=1}^m$, $k_i : V \to \{0,1\}\ \forall i \in \{1,...,m\}$, $\{S_i^\ell\}_{i=1}^m$, $\{S_i^u\}_{i=1}^m$, c, η and γ, where m is the number of classes, V is the set of data elements, w is the weight function, $\{C_i\}$ are class homogeneity terms, k_i is a function where $k_i(x) = 1$ if x is a labeled element of class i and 0 otherwise, S_i^ℓ is the lower bound for class i, S_i^u is the upper bound for class i, $c > 0$, $\eta > 0$ and $\gamma \geq 0$. No class size information is incorporated by $\gamma = 0$. The class size penalty term (7) is incorporated by $0 < \gamma < \infty$. Class size constraints (6) are incorporated by $\gamma = \infty$.

Ensure: out $= u$, where $u_i(x)$ is the probability of data element x belonging to class i.

Initialize $k = 0$, $u = 0$, $\rho_i^1 = \rho_i^2 = q_i = 0\ \ \forall i \in \{1,...,m\}$, $p_s = C_m$ and $p_i = p_s\ \ \forall i \in \{1,...,m\}$.

Let $\mathrm{div}_w\, q(x) = \sum_y w(x,y)q(x,y)$.

while Stop criterion not satisfied **do**

Let $k \leftarrow k+1$, and set $p_s^{k+1} = \sum_i ({p_i}^k + \mathrm{div}_w\, q_i^k - \frac{u_i^k}{c} + {\rho_i^2}^k - {\rho_i^1}^k + \frac{1}{c})/m$.

for $i = 1 \to m$ **do**

$q_i^{k+1} = \mathrm{Projection}_\eta(q_i^k + c\nabla_w(\mathrm{div}_w q_i^k - B_i^k))$, where $B_i^k = {p_s}^{k+1} - {p_i}^k + \frac{u_i^k}{c} - {\rho_i^2}^k + {\rho_i^1}^k$.

$p_i^{k+1}(x) = \min\{(D_i^k(x) - \frac{k_i(x)}{c}, C_i(x)\},\ \forall x,$ where $D_i^k(x) = {p_s}^{k+1} - \mathrm{div}_w\, q_i^{k+1} + \frac{u_i^k}{c} - {\rho_i^2}^k + {\rho_i^1}^k$.

${\rho_i^1}^{k+1} = \min\Big(\max\big(\mathrm{mean}(E_i^k + \frac{S_i^\ell}{c\,||V||}), 0\big), \gamma\Big)$, where $E_i^k = p_i^{k+1} + \mathrm{div}_w q_i^{k+1} - \frac{u_i^k}{c} - p_s^{k+1} + {\rho^2}_i^k$.

${\rho_i^2}^{k+1} = \min\Big(\max\big(\mathrm{mean}(F_i^k - \frac{S_i^u}{c\,||V||}), 0\big), \gamma\Big)$, where $F_i^k = -p_i^{k+1} - \mathrm{div}_w q_i^{k+1} + \frac{u_i^k}{c} + p_s^{k+1} + {\rho_i^1}^{k+1}$.

$u_i^{k+1} = u_i^k - c\,(\mathrm{div}_w\, q_i^{k+1} - p_s^{k+1} + p_i^{k+1} + {\rho_i^2}^{k+1} - {\rho_i^1}^{k+1})$.

end for

end while

projection onto that constraint as follows:

$$
\begin{aligned}
&{\rho_i^1}^{k+1} = \min\Big(\max\big(\mathrm{mean}(E_i^k + \frac{S_i^\ell}{c\,||V||}), 0\big), \gamma\Big),\\
&\quad \text{where}\ \ E_i^k = p_i^{k+1} + \mathrm{div}_w q_i^{k+1} - \frac{u_i^k}{c} - p_s^{k+1} + {\rho^2}_i^k.\\
&{\rho_i^2}^{k+1} = \min\Big(\max\big(\mathrm{mean}(F_i^k - \frac{S_i^u}{c\,||V||}), 0\big), \gamma\Big),\\
&\quad \text{where}\ \ F_i^k = -p_i^{k+1} - \mathrm{div}_w q_i^{k+1} + \frac{u_i^k}{c} + p_s^{k+1} + {\rho_i^1}^{k+1}.
\end{aligned}
\tag{33}
$$

Algorithm 1 is parallelizable on GPU. This is due to the fact that the subproblems at each substep can be solved pointwise independently of each other. Moreover, the update formula (30) only necessitates access to the values of neighboring nodes at the previous iteration.

3.2 Global minimization of similarity graph-based problems incorporating class size constraints via a duality approach

In this section, we derive Algorithm 2, which considers a duality approach to solving certain graph-based problems for learning tasks, such as data classification. The notation of Section 2.5 is used. In particular, we first consider the following optimization problem, which is similar to the one studied in the previous section:

$$
\begin{aligned}
\min_{u \in \Psi} \Bigg\{ E(u) = \sum_{i=1}^m \Big(\frac{1}{2} \sum_{x,y \in V} w(x,y)|u_i(y) - u_i(x)| +\\
+ \sum_{x \in V} C_i(x) u_i(x) \Big) \Bigg\},\\
\textit{under optional constraints (6) or penalty term (7).}
\end{aligned}
\tag{34}
$$

The first term in this model is a graph-based term resembling a convexified graph cut, which attempts to group data elements in a way such that the elements grouped in different classes are as dissimilar as possible. The second term incorporates the class homogeneity terms and labeled data, while the class size information is incorporated by optional class size constraints (6) or penalty term (7). Note that (34) is a convexified version of a problem with the same terms but where u is allowed to only take values in $\{0,1\}$; the latter problem is non-convex due to the binary constraints. It turns out that the convex relaxation indeed closely approximates the non-convex problem, where u takes only binary values; please refer to [90] for more information.

At the end, the data elements assigned to class i can be obtained using the optimal u^* via:

$$V_i = \{x \text{ s.t. } \arg\max_j u_j^*(x) = i\}.$$

It turns out that (34) has an interesting equivalent formulation, as shown in the next theorem.

Theorem 2 *The following problem is equivalent to the convex relaxed problem* (34) *under optional class size constraints* (6) *or penalty term* (7)*:*

$$\sup_{\rho^1,\rho^2\in[0,\gamma],|q|_\infty\leq 1}\Big\{E(\rho^1,\rho^2,q)=\sum_i\big(\rho_i^1 S_i^\ell-\rho_i^2 S_i^u\big)+ \\ +\sum_{x\in V}\min_i\Big(C_i(x)+(\mathrm{div}_w\, q_i)(x)+\rho_i^2-\rho_i^1\Big)\Big\}. \quad (35)$$

No size information is incorporated by $\gamma = 0$*. The size penalty term* (7) *is incorporated by* $0 < \gamma < \infty$*. Size constraints* (6) *are incorporated by* $\gamma = \infty$*.*

Proof First, note that, for any arbitrary vector $g = (g_1, ..., g_m) \in \mathbb{R}^m$,

$$\min_{(z_1,...,z_m)\in\Delta_+^m}\sum_{i=1}^m z_i g_i = \min_i(g_1, ..., g_m). \quad (36)$$

Applying (36) to (35), one obtains

$$\sup_{\rho^1,\rho^2\in[0,\gamma],|q|_\infty\leq 1}\Bigg\{E(\rho^1,\rho^2,q)=\sum_i\big(\rho_i^1 S_i^\ell-\rho_i^2 S_i^u\big) \\ +\sum_{x\in V}\Big(\min_{\Sigma}\sum_{i=1}^m u_i(x)\Big(C_i(x)+(\mathrm{div}_w\, q_i)(x)+\rho_i^2-\rho_i^1\Big)\Big)\Bigg\}, \quad (37)$$

where $\Sigma = \{(u_1(x), ..., u_m(x)) \in \Delta_+^m\}$, where Δ_+^m is defined in (4). One can then rewrite (37) as

$$\sup_{\rho^1,\rho^2\in[0,\gamma],|q|_\infty\leq 1}\min_{u\in\Psi}\Big\{E(\rho^1,\rho^2,q,u)= \\ \sum_{x\in V}\sum_{i=1}^m u_i(x)\Big(C_i(x)+(\mathrm{div}_w\, q_i)(x)+\rho_i^2-\rho_i^1\Big)+ \\ +\sum_i\big(\rho_i^1 S_i^\ell-\rho_i^2 S_i^u\big)\Big\}. \quad (38)$$

The above problem satisfies the conditions of the mini-max theorem, described in, e.g., Chapter 6, Proposition 2.4 of [45]. In fact, the constraint sets for q, ρ^1, ρ^2 and u are compact and convex, and the energy is convex lower semi-continuous for fixed q and concave upper semi-continuous for fixed u, indicating the existence of at least one primal-dual solution of finite energy.

Using the mini-max theorem, one can interchange the operators in (38), resulting in:

$$\min_{u\in\Psi}\sup_{\rho^1,\rho^2\in[0,\gamma],|q|_\infty\leq 1}\Big\{E(\rho^1,\rho^2,q,u)= \\ \sum_{x\in V}\sum_{i=1}^m u_i(x)\Big(C_i(x)+(\mathrm{div}_w\, q_i)(x)+\rho_i^2-\rho_i^1\Big)+ \\ +\sum_i\big(\rho_i^1 S_i^\ell-\rho_i^2 S_i^u\big)\Big\}. \quad (39)$$

Rearranging the terms of (39), one obtains

$$\min_{u\in\Psi}\sup_{\rho^1,\rho^2\in[0,\gamma],|q|_\infty\leq 1}\Big\{E(\rho^1,\rho^2,q,u)= \\ \sum_{x\in V}\sum_{i=1}^m u_i(x)\,\mathrm{div}_w\, q_i)(x)+\sum_{x\in V}\sum_{i=1}^m u_i(x)C_i(x)+ \\ \sum_{i=1}^m\rho_i^1\big(S_i^\ell-\sum_{x\in V}u_i(x)\big)+\sum_{i=1}^m\rho_i^2\big(\sum_{x\in V}u_i(x)-S_i^u\big)\Big\}. \quad (40)$$

One can then derive the following formulations:

$$\sup_{0\leq\rho_i^1\leq\gamma}\rho_i^1\big(S_i^\ell-\sum_{x\in V}u_i(x)\big)= \\ \begin{cases}0 & \text{if } \sum_{x\in V}u_i(x)\geq S_i^\ell \\ \gamma\big(S_i^\ell-\sum_{x\in V}u_i(x)\big) & \text{if } \sum_{x\in V}u_i(x)<S_i^\ell.\end{cases} \quad (41)$$

$$\sup_{0\leq\rho_i^2\leq\gamma}\rho_i^2\big(\sum_{x\in V}u_i(x)-S_i^u\big)= \\ \begin{cases}0 & \text{if } \sum_{x\in V}u_i(x)\leq S_i^u \\ \gamma\big(\sum_{x\in V}u_i(x)-S_i^u\big) & \text{if } \sum_{x\in V}u_i(x)>S_i^u.\end{cases} \quad (42)$$

If $\gamma = 0$, the class sizes do not contribute to the energy; thus, the case of no class size information can be incorporated with $\gamma = 0$. When $0 < \gamma < \infty$, the terms (41) and (42) summed together is equal to the size penalty term (7). When $\gamma = \infty$, the constraint set on ρ^1, ρ^2 is no longer compact, but one can apply the Sion's generalization of the mini-max theorem [111], which allows either the primal or dual constraint set to be non-compact. Therefore, if the class size constraints (6) are not satisfied, the energy in (40) would be infinite, contradicting existence of a primal-dual solution.

Combining (40) - (42) with (16), one can see that (35) can be reformulated as (34) under optional class size constraints (6) or penalty term (7). ■

A drawback to (35) is the non-smoothness of its objective function. In order to derive an alternate and

Algorithm 2

Require: m, V, $w : V \times V \to \mathbb{R}$, $\{C_i\}_{i=1}^m$, $\{S_i^\ell\}_{i=1}^m$, $\{S_i^u\}_{i=1}^m$, s, α, δ, η and γ, where m is the # of classes, V is the set of data elements, w is the weight function, $\{C_i\}$ are class homogeneity terms, S_i^ℓ is the lower bound for class i, S_i^u is the upper bound for class i, $s < 0$, $\alpha > 0$, $\delta > 0$, $\eta > 0$ and $\gamma \geq 0$. No class size information is incorporated by $\gamma = 0$. The class size penalty term (7) is incorporated by $0 < \gamma < \infty$. Class size constraints (6) are incorporated by $\gamma = \infty$.

Ensure: out $= u$, where $u_i(x)$ is the probability of data element x belonging to class i.

Initialize $k = 0$, $\rho_i^1 = \rho_i^2 = q_i = 0 \ \ \forall i \in \{1, ..., m\}$.

Let $\mathrm{div}_w\, q(x) = \sum_y w(x,y)^a q(x,y)$, $\nabla_w T_i(x,y) = w(x,y)^{1-a}(T_i(y) - T_i(x))$, where $a \in [\frac{1}{2}, 1]$. We usually set $a = 1$.

while Stop criterion not satisfied **do**

Let $k \leftarrow k + 1$.

for $i = 1 \to m$ **do**

$T_i^{k+1} = \Big(e^{(-C_i - \mathrm{div}_w\, q_i^k - \rho_i^{2\,k} + \rho_i^{1\,k})/s}\Big) / \Big(\sum_{i=1}^m e^{(-C_i - \mathrm{div}_w\, q_i^k - \rho_i^{2\,k} + \rho_i^{1\,k})/s}\Big)$.

$q_i^{k+1} = \mathrm{Projection}_\eta(q_i^k - \alpha \nabla_w T_i^{k+1})$, where $\mathrm{Projection}_\eta$ is a projection operator defined in (31).

$\rho_i^{1\,k+1} = \min\Big(\max\big(\rho_i^{1\,k} - \delta(\sum_x T_i^{k+1}(x) - S_i^\ell), 0\big), \gamma\Big)$.

$\rho_i^{2\,k+1} = \min\Big(\max\big(\rho_i^{2\,k} - \delta(S_i^u - \sum_x T_i^{k+1}(x)), 0\big), \gamma\Big)$.

end for

end while

The final variable u is computed via the converged values $q_i^*, \rho_i^{1^*}, \rho_i^{2^*}$:

$$u_i(x) = \begin{cases} 1 & \text{if } i = \underset{j=1,\ldots,m}{\arg\min}\ (C_j + \mathrm{div}_w\, q_j^* + \rho_j^{2^*} - \rho_j^{1^*}) \\ 0 & \text{otherwise} \end{cases}$$

slightly simpler scheme to a max-flow technique of Section 3, we propose to use a smoothing technique, leading to a smoothed primal-dual version of (35).

In particular, to derive the aforementioned technique, we consider the asymptotic functions as defined in [115, 102]. Specifically, the asymptotic function h_∞ of a proper convex function $h(u)$ can be defined as:

$$h_\infty(u) = \lim_{s \to 0^+} sh(\frac{u}{s}). \tag{43}$$

If $h(u) = \log \sum_{j=1}^k e^{u_j}$, then using (43), one obtains $h_\infty(u) = \lim_{s \to 0^+} sh(\frac{u}{s}) = \max_{1 \leq j \leq k} u_j = -\min_{1 \leq j \leq k} -u_j$, where u is a vector in $\mathbb{R}^m$ written as $u = (u_1, ..., u_m)$.

By applying the asymptotic approximation to the *min* operator of (35) using a small s value, we get:

$$\sup_{\rho^1, \rho^2 \in [0,\gamma], |q|_\infty \leq 1} \Bigg\{ E(\rho^1, \rho^2, q) = \sum_i \big(\rho_i^1 S_i^\ell - \rho_i^2 S_i^u\big) - s \sum_{x \in V} \Big(\log \sum_i e^{(-C_i(x) - \mathrm{div}_w\, q_i^k(x) - \rho_i^{2\,k} + \rho_i^{1\,k})/s}\Big) \Bigg\}. \tag{44}$$

Now, using [102], for any $h \in \mathbb{R}^m$,

$$\log \sum_{i=1}^m e^{h_i} = \max_{u \in \Delta_+^m} \Bigg\{ \langle u, h \rangle - \sum_{i=1}^m u_i \log u_i \Bigg\}. \tag{45}$$

Using (45), one can formulate the following smoothed model from (44):

$$\sup_{\rho^1, \rho^2 \in [0,\gamma], |q|_\infty \leq 1} \min_{u \in \Delta_+^m} \Bigg\{ E(\rho^1, \rho^2, q, u) = \sum_{x \in V} \Big(\sum_i u_i(x)(C_i(x) + \mathrm{div}_w\, q_i(x) + \rho_i^2 - \rho_i^1) + s \sum_i u_i(x) \log(u_i(x)) \Big) + \sum_i \big(\rho_i^1 S_i^\ell - \rho_i^2 S_i^u\big) \Bigg\}. \tag{46}$$

As $s \to 0$, (46) approaches (39), which according to the proof of Theorem 2, can be reformulated as (34) under optional class size constraints (6) or penalty term (7). Similar techniques but for non-graphical settings, without class size incorporation, were proposed in [18].

The smoothed formulation in (46) allows one to formulate an efficient and simple algorithm. We propose a projected gradient algorithm, which is detailed as Algorithm 2 and contains similar steps as the techniques proposed by [74, 30]. The projected gradient algorithm is constructed from (44); this method can be viewed as a forward-backward splitting algorithm. Convergence proofs for such methods have been shown in, e.g., [38].

For this algorithm, the labeled points are not explicitly implicitly incorporated as they are for Algorithm 1. However, they can be incorporated by setting $C_i(x)$ to be large if x is a labeled point which is not of class i.

3.3 Graph-based convex optimization in image segmentation involving region parameters

In this section, we derive Algorithm 3 which is useful for image segmentation problems, where the goal is to segment an image into regions. In particular, we consider a model based on a Potts regularity formulation [97], which favors region boundaries of minimal length and contains data cost functions, often depending on parameters. While the number of regions is often known in advance, the region parameters are usually unknown. We will formulate a model which simultaneously finds these parameters and the segmentation regions.

In more detail, we start by briefly considering a continuous setting. Let the goal be to divide an image into m regions, let u_i represent the characteristic function of region i, let ν_i represent an unknown parameter of a data cost term for region i, let X be the set of feasible parameter values, and let $R(x,\nu_i)$ represent the data cost term for region i. As an example, $R(x,\nu_i)$ can take the form of $|\Omega^0(x)-\nu_i|^\kappa$, where $\kappa>0$ and Ω^0 is the original image. In a continuous setting, if Ω is an image, one can consider the following Potts model (see [97]) written in terms of the characteristic functions $\{u_i\}_i$, to be later transferred to a graphical setting:

$$\min_{\{u_i\}_{i=1}^m\in T}\min_{\{\nu_i\}_{i=1}^m\in X}\sum_{i=1}^m\int_\Omega|\nabla u_i|dx+\sum_{i=1}^m\int_\Omega R(x,\nu_i)u_i(x)dx, \tag{47}$$

$$\text{subject to}\quad \sum_i u_i(x)=1\ \ \forall x\in\Omega, \tag{48}$$

where $T=\{u\in BV(\Omega)\text{ such that }u\in\{0,1\}\}$, and BV indicates functions of bounded variation. One can then transfer (47)-(48) to a graphical setting by using graphical operators, and incorporate class size constraints (6) or penalty term (7); in particular, the total variation term can be written in a graphical setting (see (16)):

$$\min_{\{u_i\}_{i=1}^m\in\{0,1\}}\min_{\{\nu_i\}_{i=1}^m\in X}\Big\{E(\{u_i\}_{i=1}^m,\{\nu_i\}_{i=1}^m)=$$
$$=\sum_{i=1}^m\sum_{x,y\in V}w(x,y)|u_i(y)-u_i(x)|+\sum_{i=1}^m\sum_{x\in V}R(x,\nu_i)u_i(x)\Big\}. \tag{49}$$

$$\text{subject to}\ \sum_i u_i(x)=1\ \ \forall x\in V,\text{and optional class size constraints (6) or penalty term (7).} \tag{50}$$

The first term in this model is a graph-based term resembling a graph cut, which attempts to group data elements in a way such that the elements grouped in different classes are as dissimilar as possible. The second term incorporates the data cost terms.

We now assume that the set of feasible data cost term parameters is finite; specifically, the set of values ν_i is restricted to $X=\{g_1,...,g_M\}$. This often occurs in image segmentation problems when X may be the set of quantized gray values. In this case, in order to optimize over a single variable, one may formulate an equivalent *extended model*, where instead of minimizing over m characteristic functions, one minimizes over M characteristic functions, where each function corresponds to a feasible value in X, and $M>m$:

$$\min_u\Big\{E^{\text{extended}}(u)=\sum_{i=1}^M\sum_{x,y\in V}w(x,y)|u_i(y)-u_i(x)| + \sum_{i=1}^M\sum_{x\in V}R(x,g_i)u_i(x)\Big\}, \tag{51}$$

$$\text{subject to}\quad \sum_i u_i(x)=1\ \ \forall x\in V,\quad \sum_{i=1}^M\sup_{x\in V}u_i(x)\le m,$$
$$u_i(x)\in\{0,1\}\ \ \forall x\in V,\ \forall i\quad\text{and optional class size constraints (6) or penalty term (7).} \tag{52}$$

One can show that (51) subject to (52) is equivalent to original model (49) subject to (50) if the feasible values of $\{\nu_i\}_i$ are restricted to a finite set X.

Theorem 3 *Let u^* be an optimal value of (51)-(52), m^* be the # of indices i for which $u_i^*\neq 0$. If $\{i_j\}_{j=1}^{m^*}\subset\{1,...,M\}$ such that $u_{i_j}^*\neq 0$, then $(\{u_{i_j}^*\}_{j=1}^{m^*},\{g_{i_j}\}_{j=1}^{m^*})$ is a global optimum to (49)-(50) with $X=\{g_1,...,g_M\}$.*

Proof First, $m^*\le m$, since otherwise (52) would be violated. Then, let $(\{\tilde u_j\}_{j=1}^m,\{g_{\tilde i_j}\}_{j=1}^m)$ be any other solution to (49)-(50), and define the following function:

$$\hat u_j=0,\ \text{for } j\in\{1,...,M\}\setminus\{\tilde i_1,...,\tilde i_m\}, \tag{53}$$

$$\hat u_{\tilde i_j}=\tilde u_j\ \text{for } j=1,...,m. \tag{54}$$

Therefore, $\hat{u}$ belongs to the feasible set of (51).

Then $E^{\text{extended}}(\hat{u}) =$

$$= \sum_{i=1}^{M} \sum_{x,y\in V} w(x,y)|\hat{u}_i(y) - \hat{u}_i(x)| + \sum_{i=1}^{M} \sum_{x\in V} R(x,g_i)\hat{u}_i(x)$$

$$= \sum_{j=1}^{m} \sum_{x,y\in V} w(x,y)|\hat{u}_{\tilde{i}_j}(y) - \hat{u}_{\tilde{i}_j}(x)| + \sum_{j=1}^{m} \sum_{x\in V} R(x,g_{\tilde{i}_j})\hat{u}_{\tilde{i}_j}(x)$$

$$= \sum_{j=1}^{m} \sum_{x,y\in V} w(x,y)|\tilde{u}_j(y) - \tilde{u}_j(x)| + \sum_{j=1}^{m} \sum_{x\in V} R(x,g_{\tilde{i}_j})\tilde{u}_j(x)$$

$$= E(\{\tilde{u}_j\}_{j=1}^m, \{g_{\tilde{i}_j}\}_{j=1}^m). \quad (55)$$

Since u^* is a global minimizer of E^{extended},

$$E^{\text{extended}}(u^*) \leq E^{\text{extended}}(\hat{u}) = E(\{\tilde{u}_j\}_{j=1}^m, \{g_{\tilde{i}_j}\}_{j=1}^m). \quad (56)$$

Now, $E^{\text{extended}}(u^*) =$

$$= \sum_{i=1}^{M} \sum_{x,y\in V} w(x,y)|u_i^*(y) - u_i^*(x)| + \sum_{i=1}^{M} \sum_{x\in V} R(x,g_i)u_i^*(x)$$

$$= \sum_{j=1}^{m^*} \sum_{x,y\in V} w(x,y)|u_{i_j}^*(y) - u_{i_j}^*(x)| + \sum_{j=1}^{m^*} \sum_{x\in V} R(x,g_{i_j})u_{i_j}^*(x)$$

$$= E(\{u_{i_j}^*\}_{j=1}^{m^*}, \{g_{i_j}\}_{j=1}^{m^*}). \quad (57)$$

Using (56) and (57),

$$E(\{u_{i_j}^*\}_{j=1}^{m^*}, \{g_{i_j}\}_{j=1}^{m^*}) \leq E(\{\tilde{u}_j\}_{j=1}^m, \{g_{\tilde{i}_j}\}_{j=1}^m).$$

Thus, $(\{u_{i_j}^*\}_{j=1}^{m^*}, \{g_{i_j}\}_{j=1}^{m^*})$ is a solution to (49)-(50). ■

Due to the equivalence of the models, we will focus on solving the extended model (51)-(52) which finds the unknown parameters and the optimal regions simultaneously. However, it is nonconvex due to the binary constraints on u. In this case, one can perform a convex relaxation procedure, similarly to [78, 16, 129], by allowing $u(x)$ to take any values in Δ_+^m in (4).

We thus consider the model (51), but allow u to take any value in $[0,1]$. Let μ be a Lagrange multiplier for the second constraint in (52). This results in:

$$\max_{\mu\geq 0} \min_{u} \Bigg\{ \mathcal{L}(u,\mu) = \sum_{i=1}^{M} \sum_{x,y\in V} w(x,y)|u_i(x) - u_i(y)| + \sum_{i=1}^{M} \sum_{x\in V} R(x,g_i)u_i(x) + \mu\Big(\sum_{i=1}^{M} \max_{x\in V} u_i(x) - m \Big) \Bigg\}, \quad (58)$$

$$\text{subject to } \sum_i u_i(x) = 1 \ \ \forall x \in V, \ \ u_i(x) \geq 0 \ \ \forall x \in V, \ \ \forall i,$$

$$\text{under optional class size constraints (6) or penalty term (7)}, \quad (59)$$

where $R(x,g_i)$ is a data cost function for a *potential* class i at x which can be formulated as a class homogeneity term, m is the desired number of classes, and $M > m$ is the number of terms in the discrete parameter set $X = \{g_1, ..., g_M\}$. The third term of (58) instills a penalty on the number of classes. Overall, for some fixed i, the optimal $u_i(x)$ might take a small or zero value for all x so that no data element would be classified into that potential class i. By using many more classes than necessary (i.e. $M > m$), a method which minimizes (58) subject to (59) would simultaneously select the optimal region parameters and optimal u. Similar techniques but for non-graphical settings, without class size incorporation, were proposed in [19].

To solve the problem (58), one can apply an augmented Lagrangian technique, the convergence of which is often guaranteed by theories, such as the ones in [47, 57]. The technique is also accurate and tolerates a wide range of step sizes. One can apply this procedure to solve (58) by alternating between the following two steps until convergence, where $\mathcal{L}$ is defined in (58):

$$1) \quad u^{n+1} = \arg\min_u \mathcal{L}(u, \mu^n)$$

$$\text{s.t. } \sum_i u_i(x) = 1, \quad u_i(x) \geq 0 \ \ \forall x \in V \ \ \forall i.$$

$$\text{under optional class size constraints (6) or penalty term (7)}, \quad (60)$$

$$2) \quad \mu^{n+1} = \max\Big(0, \mu^n + \lambda\big(\Big[\sum_i \max_{x\in V} u_i^{n+1}(x)\Big] - m\big)\Big). \quad (61)$$

We now show that the first subproblem (60) can be rewritten as a primal-dual problem with a max-flow setting. In particular, we show the connection between the label cost constraint in (51) and adding an additional flexible flow constraint in the max-flow setting. Before going into detail in Theorem 4, we define:

$$\Theta_i^{\mu} = \{r_i : V \to \mathbb{R} \text{ such that } \sum_{x\in V} |r_i(x)| \leq \mu\}. \quad (62)$$

Theorem 4 *The following primal-dual problem with a max-flow setting can be written as the optimization problem in* (60) *:*

$$\min_u \sup_{p_s,p,q,r,\rho^1,\rho^2} \Big\{ \sum_{x\in V} p_s(x) + \sum_{i=1}^{m} \big(\rho_i^1 S_i^{\ell} - \rho_i^2 S_i^u\big) + \sum_{i=1}^{m} \sum_{x\in V} u_i(x)\big(\operatorname{div}_w q_i - p_s + p_i - r_i + \rho_i^2 - \rho_i^1\big)(x) \Big\} \quad (63)$$

subject to, for all $i \in \{1, ..., m\}$,

$$|q_i(x,y)| \leq 1, \qquad \forall (x,y) \in E, \tag{64}$$

$$p_i(x) \leq R(x, g_i), \qquad \forall x \in V, \tag{65}$$

$$\sum_{x \in V} |r_i(x)| \leq \mu^n, \tag{66}$$

$$0 \leq \rho_i^1, \rho_i^2 \leq \gamma. \tag{67}$$

Proof By regrouping like terms, one obtains:

$$\min_u \sup_{p_s,p,q,r,\rho^1,\rho^2} \Big\{ E(p_s, p, q, r, \rho^1, \rho^2, u) =$$
$$\sum_{x \in V} \sum_{i=1}^m u_i(x) \operatorname{div}_w q_i(x) + \sum_{x \in V} ((1 - \sum_{i=1}^m u_i(x)) p_s(x))$$
$$+ \sum_{x \in V} \sum_{i=1}^m u_i(x) p_i(x) - \sum_{x \in V} u_i(x) r_i(x)$$
$$+ \sum_{i=1}^m \rho_i^1 \big(S_i^\ell - \sum_{x \in V} u_i(x)) + \sum_{i=1}^m \rho_i^2 \big(\sum_{x \in V} u_i(x) - S_i^u \big) \Big\}$$
$$\text{subject to (64)-(67)} \tag{68}$$

For a given variable u, the maximization of the term involving r can be written as:

$$\sup_{r_i(x) \in \Theta_i^{\mu^n}} - \sum_{x \in V} u_i(x) r_i(x) = \mu^n \max_{x \in V} u_i(x). \tag{69}$$

For a given variable u, the maximization of the term involving p_i can be written as:

$$\sup_{p_i(x) \leq R(x, g_i)} u_i(x) p_i(x) = \begin{cases} u_i(x) R(x, g_i) & \text{if } u_i(x) \geq 0 \\ \infty & \text{if } u_i(x) < 0. \end{cases} \tag{70}$$

An important distinction between (60) and (63) is the fact that u is unconstrained in (63) and the simplex constraint on u is handled implicitly. This allows one to formulate an algorithm without projections of u, which can restrict step sizes and reduce accuracy.

The rest of the proof follows similarly to that of Theorem 1. In particular, we use the mini-max theorem, described in sources such as Chapter 6, Proposition 2.4 of [45], and Sion's generalization of the mini-max theorem [111], as done in the proof of Theorem 1. In addition, keeping the u variable constant, the optimization of the q_i, p_s, ρ^1 and ρ^2 terms can be written as (16), (17), (19) and (20), respectively. Overall, the simplex and non-negativity constraint on u in (60) and the optional class size constraints (6) or penalty term (7) are implicitly incorporated into the primal-dual formulation (63) in the manner shown in the proof of Theorem 1. Specifically, the positivity constraint on u can be obtained from (70), the simplex constraint on u can be obtained from (17), and the class size constraints are implicitly incorporated by (19) and (20).

Therefore, (63) subject to (64)-(67) can be rewritten as the first subproblem in (60). ■

As stated previously, the first subproblem (60) of the scheme (60) - (61) can be approached via the augmented Lagrangian technique, as in Section 3. The technique is efficient, accurate and tolerates a wide range of step sizes since it does not involve any projections of u. Moreover, the convergence of augmented Lagrangian techniques is often guaranteed by theories, such as the ones in [47, 57]. To derive the method, we consider the augmented Lagrangian functional:

$$L = \sum_{x \in V} p_s + \sum_{i=1}^n (\rho_i^1 S_i^\ell - \rho_i^2 S_i^u))$$
$$- \frac{c}{2} \sum_{i=1}^n \left\| \operatorname{div}_w q_i - p_s + p_i - r_i + \rho_i^2 - \rho_i^1 \right\|_2^2$$
$$+ \sum_{x \in V} u_i(x) \big(\operatorname{div}_w q_i - p_s + p_i - r_i + \rho_i^2 - \rho_i^1 \big)(x). \tag{71}$$

One can then formulate a procedure to minimize (71), where one alternatively maximizes (71) for each of the variables q, p, p_s, r, ρ_1 and ρ_2 separately, and then updates the Lagrange multiplier u.

In practice, the alternating scheme for the first subproblem (60) becomes the following, where we perform these steps for N_{iter} iterations, where N_{iter} is a small number (we use 5), and $\|f\|_2^2 = \sum_x f(x)^2$:

$$p_s^{k+1} = \arg\max_{p_s(x)} \sum_{x \in V} p_s - \frac{c}{2} \sum_i \left\| p_s - A_i^k \right\|_2^2,$$
$$\text{where } A_i^k = {p_i}^k + \operatorname{div}_w q_i^k - r_i^k - \frac{u_i^k}{c} + {\rho_i^2}^k - {\rho_i^1}^k. \tag{72}$$

$$q_i^{k+1} = \arg\max_{|q(e)| \leq 1 \ \forall e \in E} - \frac{c}{2} \left\| \operatorname{div}_w q - B_i^k \right\|_2^2, \ \forall i,$$
$$\text{where } B_i^k = {p_s}^{k+1} - {p_i}^k + r_i^k + \frac{u_i^k}{c} - {\rho_i^2}^k + {\rho_i^1}^k. \tag{73}$$

$$p_i^{k+1} = \arg\max_{p_i(x) \leq C_i(x) \ \forall x} - \frac{c}{2} \left\| p_i - D_i^k \right\|_2^2, \ \forall i,$$
$$\text{where } D_i^k = {p_s}^{k+1} + r_i^k - \operatorname{div}_w q_i^{k+1} + \frac{u_i^k}{c} - {\rho_i^2}^k + {\rho_i^1}^k. \tag{74}$$

$${\rho_i^1}^{k+1} = \arg\max_{0 \leq \rho_i^1 \leq \gamma} \sum_{x \in V} \rho_i^1 S_i^\ell - \frac{c}{2} \left\| \rho_i^1 - E_i^k \right\|_2^2, \ \forall i,$$
$$\text{where } E_i^k = p_i^{k+1} + \operatorname{div}_w q_i^{k+1} - \frac{u_i^k}{c} - r_i^k - p_s^{k+1} + {\rho^2}_i^k. \tag{75}$$

Algorithm 3

Require: m, V, $w : V \times V \to \mathbb{R}$, $\{R(x, g_i)\}_{i=1}^m$, $k_i : V \to \{0, 1\}\ \forall i \in \{1, ..., m\}$, $\{S_i^\ell\}_{i=1}^m$, $\{S_i^u\}_{i=1}^m$, μ_{initial}, c, λ, η, γ, N_{iter}, where m is the # of classes, V is the set of data elements, w is the weight function, $\{R(x, g_i)\}_{i=1}^m$ are the data cost terms, $k_i(x) = 1$ if x is a labeled element of class i and 0 otherwise, S_i^ℓ and S_i^u are lower and upper bounds for class i, $\mu_{\text{initial}} > 0$, $c > 0$, $\lambda > 0$, $\eta > 0$, $\gamma \geq 0$ and N_{iter} is a small number (we choose 5). No class size information is incorporated by $\gamma = 0$. The class size penalty term (7) is incorporated by $0 < \gamma < \infty$. Class size constraints (6) are incorporated by $\gamma = \infty$.

Ensure: out $= u$, where $u_i(x)$ is the probability of data element x belonging to class i.

Initialize $n = 1$, $\mu^1 = \mu_{\text{initial}}$ and let $\text{div}_w\, q(x) = \sum_y w(x, y)q(x, y)$.

while Stop criterion not satisfied **do**

Initialize $u = \rho_i^1 = \rho_i^2 = q_i = r_i = 0\ \ \forall i \in \{1, ..., m\}$, $p_s = R(x, g_m)$, $p_i = p_s\ \ \forall i \in \{1, ..., m\}$.

for $k = 1 \to N_{iter}$ **do**

Set $p_s^{k+1} = \sum_i (p_i{}^k + \text{div}_w\, q_i^k - r_i^k - \frac{u_i^k}{c} + \rho_i^{2\,k} - \rho_i^{1\,k} + \frac{1}{c})/m$.

for $i = 1 \to m$ **do**

$q_i^{k+1} = \text{Projection}_\eta(q_i^k + c\nabla_w(\text{div}_w q_i^k - B_i^k))$, where $B_i^k = p_s{}^{k+1} - p_i{}^k + r_i^k + \frac{u_i^k}{c} - \rho_i^{2\,k} + \rho_i^{1\,k}$.

$p_i^{k+1}(x) = \min\{(D_i^k(x), R(x, g_i)\},\ \forall x$, where $D_i^k(x) = p_s{}^{k+1} + r_i^k - \text{div}_w\, q_i^{k+1} + \frac{u_i^k}{c} - \rho_i^{2\,k} + \rho_i^{1\,k}$.

$\rho_i^{1\,k+1} = \min\Big(\max\big(\text{mean}(E_i^k + \frac{S_i^\ell}{c\,||V||}), 0\big), \gamma\Big)$, where $E_i^k = p_i^{k+1} + \text{div}_w q_i^{k+1} - \frac{u_i^k}{c} - r_i^k - p_s^{k+1} + \rho^{2\,k}{}_i$.

$\rho_i^{2\,k+1} = \min\Big(\max\big(\text{mean}(F_i^k - \frac{S_i^u}{c\,||V||}), 0\big), \gamma\Big)$, where $F_i^k = -p_i^{k+1} - \text{div}_w q_i^{k+1} + r_i^k + \frac{u_i^k}{c} + p_s^{k+1} + \rho_i^{1\,k+1}$.

$r_i^{k+1} = \underset{r_i(x) \in \Theta_i^{\mu^n}}{\arg\max} -\frac{c}{2}\left\|r_i - G_i^k\right\|_2^2$, where $G_i^k = \text{div}_w\, q_i^{k+1} - p_s^{k+1} + p_i^{k+1} - \frac{u_i^k}{c} + \rho_i^{2\,k+1} - \rho_i^{1\,k+1}$.

$u_i^{k+1} = u_i^k - c\,(\text{div}_w\, q_i^{k+1} - p_s^{k+1} + p_i^{k+1} + \rho_i^{2\,k+1} - \rho_i^{1\,k+1} - r_i^{k+1})$.

end for

end for

Set $u^{n+1} = u^{N_{iter}}$ and $\mu^{n+1} = \max\Big(0, \mu^n + \lambda(\Big[\sum_i \max_{x \in V} u_i^{n+1}(x)\Big] - m)\Big)$.

end while

$$\rho_i^{2\,k+1} = \underset{0 \leq \rho_i^2 \leq \gamma}{\arg\max} \sum_{x \in V} -\rho_i^2 S_i^u - \frac{c}{2}\left\|\rho_i^2 - F_i^k\right\|_2^2,\ \forall i,$$
$$\text{where}\ \ F_i^k = -p_i^{k+1} - \text{div}_w q_i^{k+1} + r_i^k + \frac{u_i^k}{c} + p_s^{k+1} + \rho_i^{1\,k+1}. \tag{76}$$

$$r_i^{k+1} = \underset{r_i(x) \in \Theta_i^{\mu^n}}{\arg\max} -\frac{c}{2}\left\|r_i - G_i^k\right\|_2^2,\ \forall i,$$
$$\text{where}\ \ G_i^k = \text{div}_w\, q_i^{k+1} - p_s^{k+1} + p_i^{k+1} - \frac{u_i^k}{c} + \rho_i^{2\,k+1} - \rho_i^{1\,k+1} \tag{77}$$

$$u_i^{k+1} = u_i^k - H_i^k,\ \text{where}$$
$$H_i^k = c\,(\text{div}_w\, q_i^{k+1} - p_s^{k+1} + p_i^{k+1} + \rho_i^{2\,k+1} - \rho_i^{1\,k+1} - r_i^{k+1}), \tag{78}$$

After N_{iter} iterations of the above, we set $u^{n+1} = u^{N_{iter}}$.

For (72) and (74), the closed form solutions are:

$$p_s^{k+1} = \sum_i (A_i^k + \frac{1}{c})/m, \tag{79}$$

$$p_i^{k+1}(x) = \min\{(D_i^k(x), R(x, g_i)\},\ \forall x, \tag{80}$$

where A_i^k and D_i^k are denoted in (72) and (74).

The optimization problem (73) can be solved by a few steps of the projected gradient method:

$$q_i^{k+1} = \text{Projection}_\eta(q_i^k + c\nabla_w(\text{div}_w q_i^k - B_i^k)),$$
$$\text{where}\ \ B_i^k = p_s{}^{k+1} - p_i{}^k + r_i^k + \frac{u_i^k}{c} - \rho_i^{2\,k} + \rho_i^{1\,k}. \tag{81}$$

where Projection_η is a projection operator in (31).

There are extended convergence theories for the augmented Lagrangian method in the case when one of the subproblems is solved inexactly, see e.g. [47, 57]. In our experience, one gradient ascent iteration is enough.

Consider now (75) and (76). In case no constraints are given on ρ^1 and ρ^2, the maximizers over the sum

of the concave quadratic terms can be computed as the average of the maximizers to each individual term as

$$\rho_i^{1\,k+1} = \text{mean}\big(E_i^k + \frac{S_i^\ell}{c\,||V||}\big), \tag{82}$$

$$\rho_i^{2\,k+1} = \text{mean}\big(F_i^k - \frac{S_i^u}{c\,||V||}\big). \tag{83}$$

with E_i^k and F_i^k defined in (75) and (76), respectively. Since the objective function is concave and the maximization variable is just a constant, an exact solution to the constrained maximization problem can now be obtained by a projection onto that constraint as follows:

$$\rho_i^{1\,k+1} = \min\Big(\max\big(\text{mean}(E_i^k + \frac{S_i^\ell}{c\,||V||}), 0\big), \gamma\Big), \tag{84}$$

$$\rho_i^{2\,k+1} = \min\Big(\max\big(\text{mean}(F_i^k - \frac{S_i^u}{c\,||V||}), 0\big), \gamma\Big). \tag{85}$$

Here, E_i^k and F_i^k are defined in (75) and (76).

The optimization problem (77) involving r_i can be addressed by the projection of G^k to the L_1-ball Θ_i^μ defined in (62) using a fast projection algorithm [44, 86].

Just like Algorithm 1, Algorithm 3 has the desired property of being parallelizable on GPU. This is due to the fact that the subproblems (72)- (77) can be solved pointwise independently of each other. Moreover, the update formula (81) for q only necessitates access to the values of neighboring nodes at the previous iterate.

In addition, we note that Algorithm 3 can be applied in an unsupervised setting and in a semi-supervised setting. For semi-supervised applications, one can incorporate labeled data by including the $\{k_i\}$ terms as in (24) in Algorithm 1, or by setting the term $R(x, g_i)$ to be large if x is a labeled point which is not of class i.

4 Experiments

For each data set, in the experiments, we randomly select a percentage of the data set to use as labeled data. The labels of the unlabeled points are initialized by creating a Voronoi diagram with the labels of the labeled points as the seed points. Every point is assigned the label of the labeled point in its Voronoi cell.

The results for experiments are in Figure 1 and Table 1. In Figure 1, the framework is semi-supervised, with only a small amount of labeled data used. In Table 1, the framework is unsupervised, with no labeled data used. In addition, for Algorithm 1, we use $\eta = 10$, $\gamma = 10$, and $c = 0.05$. For Algorithm 2, $s = 0.01$, $\delta = 0.05$, $\eta = 100$, $\alpha = 0.001$ and $\gamma = 100$. For Algorithm 3, $\mu_{\text{initial}} = 10000$, $c = 0.1$, $\lambda = 100$, $\eta = 50$ and $\gamma = 100$.

Regarding the class sizes, we set the upper bounds $\{U_i\}_i$ and lower bounds $\{L_i\}_i$ so that the amount of data elements classified into a class does not deviate more than 15% from the true class size.

Overall, for each data set, we average the accuracy over 500 to 10000 experiments, where each experiment involves a different set of labeled elements. Moreover, we ensure that each data set element is included in an experiment as a labeled point of the data set.

All experiments were run on a 2.2 GHz Intel Core i7 computer. The nearest neighbors were calculated using the *kd*-tree code in the VLFeat library [12].

4.1 Data Sets

- **Berkeley Segmentation Data Set**. This data set [88] is a data set of 500 images, both in color format and gray format, of different, mostly outside, scenes. For each image that we used, we take the intensities of a 7×7 neighborhood of each point to be the feature (attribute) vector. This vector is used to compute the weight matrix $W = \{w(x, y)\}$ using the formula in (1), $\sigma = 1$ and 15 nearest neighbors. We do not preprocess the images in any way.
- **MNIST**. This data set [76] is a data set of 70,000 grayscale 28×28 images of handwritten digits; the set consists of 10 classes. The weight matrix $W = \{w(x, y)\}$ is constructed using 15 nearest neighbors, Euclidean distance measure, the formula in (1) and $\sigma = 1$, without any preprocessing of the images.
- **Fashion MNIST.** This data set [3] is a dataset of 70,000 28×28 grayscale images of clothing items; the set contains 10 classes. The weight matrix is constructed using 15 nearest neighbors, Euclidean distance measure, the formula in (1) and $\sigma = 1$, without any preprocessing of the images.
- **Opt-Digits.** This data set [6] is a set of grayscale images of 5620 handwritten digits; it contains 10 classes. The dimension reduction performed on the data produced a 64-dimensional feature vector for each image, where each value is an integer in the range of 0 to 16. The weight matrix is constructed using 20 nearest neighbors, Euclidean distance measure, the formula in (1) and $\sigma = 1$.
- **WebKB.** This data set [41] is a set of 4199 webpages from Cornell, Texas, Washington and Wisconsin universities, and other miscellaneous pages, to be divided into: project, course, faculty and student. It is preprocessed as in [29], and tfidf term weighting [29] is used to represent the features, which are then normalized to unitary length. To construct the weights, we use a pairwise document similarity mea-

Table 1 Classification Accuracy in Percent for Algorithms 1 and 2

% of data used as labeled elements		2.5%	5%	10%	15%	20%
MNIST	Alg. 1	97.39	97.61	97.70	97.80	97.90
	Alg. 2	97.12	97.32	97.51	97.65	97.83

% of data used as labeled elements		2.5%	5%	10%	15%	20%
Fashion MNIST	Alg. 1	82.05	83.17	84.19	84.66	85.22
	Alg. 2	82.02	83.11	84.07	84.53	85.02

% of data used as labeled elements		2.5%	5%	10%	15%	20%
Pendigits	Alg. 1	97.31	98.39	98.85	99.01	99.21
	Alg. 2	97.24	98.31	98.78	99.00	99.17

% of data used as labeled elements		1%	2.5%	5%	10%	20%
Landsat	Alg. 1	85.96	87.43	88.61	89.82	90.71
	Alg. 2	85.86	87.37	88.54	89.20	90.69

% of data used as labeled elements		2.5%	5%	10%	15%	20%
Optdigits	Alg. 1	98.49	98.52	98.63	98.76	98.86
	Alg. 2	98.31	98.37	98.51	98.68	98.79

% of data used as labeled elements		2.5%	5%	10%	15%	20%
Reuters	Alg. 1	92.08	95.23	95.40	95.51	95.86
	Alg. 2	91.58	94.90	95.13	95.38	95.55

% of data used as labeled elements		5%	10%	20%	30%	40%
WebKB	Alg. 1	81.64	84.47	86.57	88.33	89.16
	Alg. 2	81.54	84.25	86.23	88.16	89.09

% of data used as labeled elements		2.5%	5%	10%	15%	20%
20 Newsgroups	Alg. 1	73.74	76.65	80.63	81.72	83.19
	Alg. 2	72.94	76.44	80.20	81.39	82.89

Fig. 1 Image Segmentation Results for Algorithm 3 (Unsupervised Graph-Based Setting). Algorithm 3 simultaneously finds the segmentation result and the region parameters of data cost terms, when the latter is unknown.

sure [93] as a distance measure, the weight formula (1) with $\sigma = 1$, and 30 nearest neighbors.

– **Reuters.** This data set [9] is a collection of 7674 documents with 8 classes: 'crude', 'earn', 'acq', 'grain', 'interest', 'money-fx', 'ship' and 'trade'. Tfidf term weighting [29] is used to represent the features, which are then normalized to unitary length. To construct the weight matrix, we use a pairwise document similarity measure [93] as a distance measure, the formula in (1) with $\sigma = 1$, and 30 nearest neighbors.

– **20 Newsgroups.** This data set [1] is a set of approximately 20,000 newsgroup documents, to be divided into 20 classes. Tfidf term weighting [29] is used to represent the feature vectors, which are then normalized to unitary length. To construct the weights, we use cosine similarity as a distance measure, the formula in (1) with $\sigma = 1$, and 30 nearest neighbors.

– **Landsat.** This data set [10] is a set of 6435 elements containing multi-spectral values of pixels in 3×3 neighbourhoods in a satellite image, and the classification is associated with the central pixel. There are

6 classes. Each of the elements of the data contains the pixel values in the four spectral bands of each pixel in the 3×3 neighbourhood; therefore, each feature vector consists of 36 values. The weights are constructed using (1) with $\sigma = 1$, Euclidean distance, and 20 nearest neighbors.

- **Pendigits.** This data set [7] of 10 classes is a set of 10992 images of handwritten digits. Each image is represented by a vector of 16 values, each between 0 and 100. The weight matrix is constructed using (1) with $\sigma = 1$, and 30 nearest neighbors.

4.2 Comparison to Recent Methods

We compare our methods to many recent algorithms, most of which are semi-supervised and from the last 5 years. The results are in Table 2 and Figure 2.

In particular, we compare the results of all data sets to the following label propagation methods:

- weighted nonlocal Laplacian (WNLL) [108]
- centered kernel method (CKM) [87]
- sparse label propagation (SLP) [67]
- p-Laplace learning (p-Laplace) [101].

We also compare the results of all data sets to the following graph embedding methods and popular GNN-based models for graph semi-supervised learning tasks:

- graph neural network (GNN) [59]
- graph convolutional network (GCN) [73]
- Planetoid [123].

From Table 2 and Figure 2, we see that the results of the proposed methods compare favorably to those of recent algorithms. In particular, from Table 2, we see that when the methods are compared to classical graph embedding methods and popular GNN-based models such as GCN [73], GNN [59] and Planetoid [123], the proposed methods outperform the aforementioned models. In particular, for MNIST, Pendigits and Opt-Digits, the results of the proposed techniques are 6%-19% higher in accuracy than that of the comparison methods in Table 2. For Reuters data, the result is 5%-14% higher than that of other methods in Table 2. For the Landsat and 20 Newsgroups data sets, the results are 2%-14% higher in accuracy than that of the comparison methods in Table 2. Moreover, from Figure 2, we can also see that the results of the proposed methods compare favorably to those of recent algorithms. For example, the results of the proposed methods on WebKB data have an 2.1%-14% improvement from comparison methods.

Moreover, for the Optdigits data set, in addition to the aforementioned algorithms, we also compare to the following semi-supervised procedures: TV-based multiclass graph partitioning (TVRF) [124] and high-dense graph learning (HiDEGL) with four different versions [120]. The result of TVRF is from the paper [120]. For labeled data, we use 140 labeled points.

For the Pendigits data, in addition to the aforementioned algorithms, we also compare to the following semi-supervised methods: semi-supervised predictive clustering trees (SSL-PCT) [80], semi-supervised deep classification (SDC) [43], direct kernel mapping (1-Kernel-LP-map) [131], kernel mapping- induced label reconstruction (1-Kernel-LP-recons) [131], discriminative sparse flexible manifold embedding (Sparse FME) [133], semi-supervised self-organizing map (SS-SOM) [26]. The results for 1-Kernel-LP-recons, 1-Kernel-LP-map and Sparse FME methods are from [131]. All methods consider 2.5% of the data set as labeled.

For MNIST, in addition to the aforementioned methods, we also compare to the following (mostly semi-supervised) methods: label propagation adaptive resonance theory (LPART) [71], Ensembles [35], batch semi-supervised self-organizing map (Batch SS-SOM) [25], continual learning [79], active learning [2], and successive subspace learning (Pixel Hop) [34]. The results for Ensembles is from [35]. All methods consider 2.5% of the data as labeled, except Batch SS-SOM and active learning, which consider 5% of the data as labeled.

For Fashion MNIST, in addition to the aforementioned methods, we also compare to semi-supervised deep classification (SDC) [43], batch semi-supervised self-organizing map (Batch SS-SOM) [25], a convolutional neural network (LeNet-5) [75], Wasserstein adversarial active learning (WAAL) [109], and continual learning [79]. The result for WAAL is from [109]. All methods consider 2.5% of data as labeled, except Batch SS-SOM, which considers 5% of data as labeled.

For Landsat, in addition to the aforementioned methods, we also compare to the following semi-supervised methods: a modified particle competition and cooperation models (MPCCM-mcCPU and MPCCM-GPU) [112], 2 versions of active labeling (with Tri-Training or STDP) [96], and 2 versions of a framework based on local cores for self-labeled semi-supervised classification (LC-SSC with Tri-Training or STDP) [81]. Most results are from [81]. All methods use only 64 labels in total.

For Reuters, in addition to the aforementioned methods, we also compare to semi-supervised Grid-TCTN [103], semi-supervised GE-TCTN [40], multinomial Naive Bayes (Mult-nb) [11], linear kernel support vector machine (SVC-tfidf) [11], ExtraTrees with 200 trees (Globebig-tfidf) [11], and ExtraTrees with 200 trees using a 100-dimensional word2vec embedding (w2v-tfidf) [11].

Table 2 Comparison to Semi-Supervised Graph Embedding and GNN-based Models

Data Set	% Labeled Data	Proposed (Alg 1)	Proposed (Alg 2)	GCN [73]	GNN [59]	Planetoid [123]
MNIST	0.7%	**95.95**	95.90	88.71	85.92	86.62
Pendigits	1%	**91.49**	91.17	83.28	76.36	85.82
Opt-Digits	1%	**97.53**	97.13	83.16	78.35	87.98
WebKB	10%	84.47	84.25	**85.37**	80.13	84.04
Fashion MNIST	0.7%	**79.81**	79.69	76.58	73.29	78.14
Landsat	1%	**85.96**	85.86	83.57	80.35	83.41
Reuters	2.5%	**92.08**	91.58	85.16	78.35	87.03
20 Newsgroups	2.5%	**73.74**	72.94	70.26	60.86	66.11

Table 3 Timing Results for Algorithms 1 and 2 (in seconds)

Data Set	Number of classes	Number of elements	Timing in seconds	Timing in seconds
MNIST	10	70000	28.28 s	13.39 s
Pendigits	10	10992	7.23 s	4.45 s
Opt-Digits	10	5620	2.88 s	1.38 s
WebKB	4	4199	1.11 s	0.59 s
Fashion MNIST	10	70000	22 s	11.63 s
Landsat	6	6435	2.67 s	1.36 s
Reuters	8	7674	6.04 s	2.82 s
20 Newsgroups	20	18820	33.34 s	15.28 s

Table 4 Timing Results for Algorithm 3 (in seconds)

Data set	Number of pixels in the image	Construction of nearest neighbor graph	Timing in seconds for Algorithm 3
Berkeley Segmentation data (one image)	154401	$\sim$ 90 s	$\sim$ 35 s

Table 5 Comparison of Timing to that of Other Algorithms

	MNIST	Fashion MNIST	Other data sets from Section 6.1
Pre-algorithm computations (construction of graph weights) for Algorithms 1 and 2 (proposed)	$\sim$ 1.5 min	$\sim$ 1.5 min	0.03 min-1 min (except for the 20 Newsgroups data)
Pre-algorithm computations (training) for LeNet-5 [4]	$\sim$ 25 min	$\sim$ 25 min	not included in paper
Pre-algorithm computations (training) for for PixelHop [34]	$\sim$ 15 min	$\sim$ 15 min	not included in paper
Pre-algorithm computations (training) for LIBSVM [5]	$\sim$ 15 min	$\sim$ 15 min	not included in paper

Table 6 Friedman's Test and Ranking for Table 2

Statistic	p-value	Result
42.77778	<0.00001	H0 is rejected

(a) Friedman's Test

Algorithm 1	Algorithm 2	GCN	Planetoid	GNN
1.125	2.125	3.25	3.50	5.00

(b) Ranking

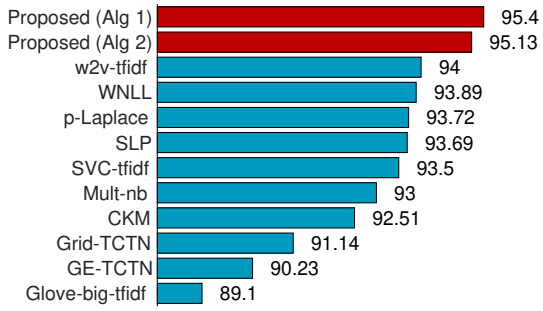
Table 7 Friedman's Test and Ranking for Figure 2

Statistic	p-value	Result
61.69737	<0.00001	H0 is rejected

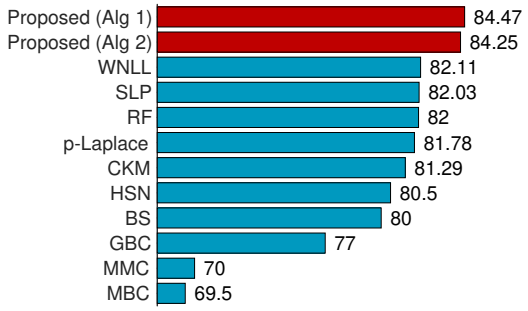
(a) Friedman's Test

Algorithm 1	Algorithm 2	WNLL	pLaplace	SLP	CKM
1.00	2.00	3.57	3.71	4.86	5.86

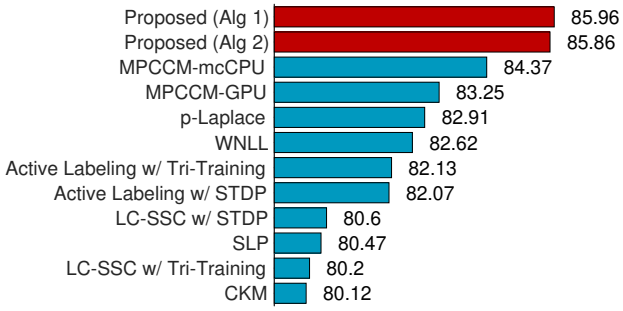
(b) Ranking



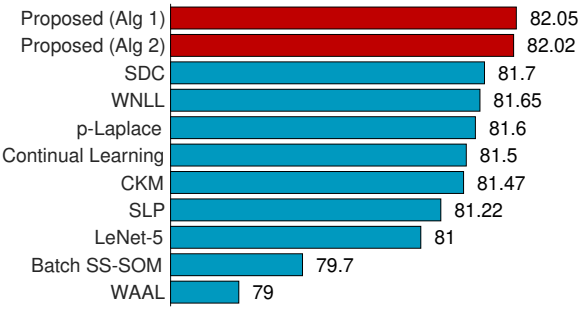
(a) Reuters (≈ 10% labeled data)



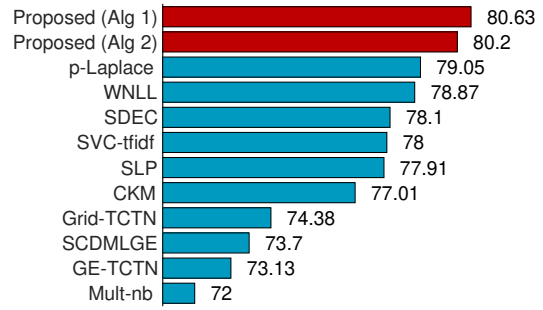
(b) WebKB (≈ 10% labeled data)



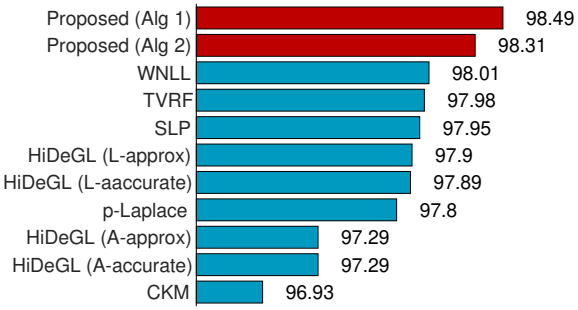
(c) Landsat (≈ 1% labeled data)



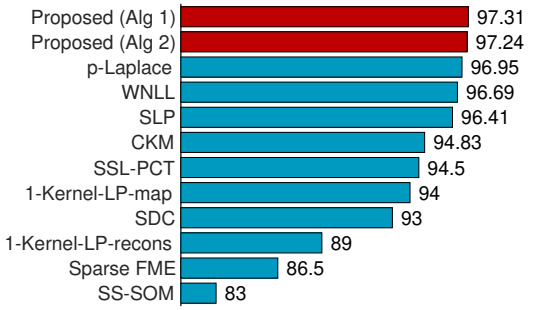
(d) Fashion MNIST (≈ 2.5% labeled data)



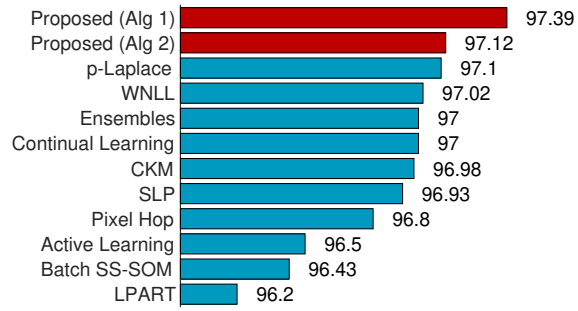
(e) 20 Newsgroups (≈ 10% labeled data)



(f) Optdigits (≈ 2.5% labeled data)



(g) Pendigits (≈ 2.5% labeled data)



(h) MNIST (≈ 2.5% labeled data)

Fig. 2 Comparison to Other Recent Methods (Accuracy in Percent)

The result of Grid-TCTN and GE-TCTN is from [40]. All methods consider 10% of the data as labeled.

For the WebKB data set, we also compare our algorithms to a merged multi-class classifier (MMC) [106], graph-based classification (GBC) [105], merged binary classifier (MBC) [106], hybrid support vector machine and Naïve Bayes classifier (HSC) [105], boosting (BS) [70] and random forests (RF) [70]. The proposed methods and the last two methods use around 420 labeled points, while others use 560 labeled points.

For 20 Newsgroups, we also compare to the following (mostly semi-supervised) methods: semi-supervised Grid-TCTN [103], GE-TCTN [40], semi-supervised deep embedded clustering (SDEC) [100], semi-supervised clustering with deep metric learning and graph embedding (SCDMLGE) [83], multinomial Naïve Bayes (Mult-nb) [11], and linear kernel support vector machine (SVC-tfidf) [11]. All methods consider 10% of data as labeled.

4.3 Timing

The timing for Algorithms 1, 2 and 3 is included in Tables 2-3. The proposed algorithms are very fast; all experiments were performed on a 2.2 GHz Intel Core i7 computer. Overall, Table 2 involves Algorithms 1 and 2 and includes the timing needed for everything but the construction of the weight matrix. For all but 20 Newsgroups, the time to construct the weight matrix is only 0.03 - 1 minutes; for 20 Newsgroups, 6 minutes are needed. Table 3 includes the timing for Algorithm 3. Table 4 compares the timing of the proposed methods to that of several recent algorithms.

4.4 Analysis and Influence of the Parameters

As is the case for the vast majority of classification algorithms, the values of the parameters in the proposed techniques influence their accuracy. In particular, some parameters should be kept large or small, otherwise the accuracy can be suboptimal. For example, the parameter c in Algorithms 1 and 3 must be kept small but not too small (otherwise, converge will be slower) as it is a projected gradient descent step; usually, it is in the range of 0.01-0.1. The parameters δ and α in Algorithm 2 should also be kept small- we usually choose them to be 0.05 or smaller. In addition, the parameter s in Algorithm 2 is a smoothing parameter and must be kept small, usually 0.05 or smaller, otherwise the asymptotic approximation of the technique would not be valid. However, the parameters η and γ must not be small, as they involve projections onto $[-\eta, \eta]$ and $[0, \gamma]$, respectively; usually, they should be at least 10, otherwise a lot of information is lost. Moreover, for Algorithm 3, we choose μ_{initial} to be a large parameter, so that (close to) the desired number of segments in the image is attained in the result. Since μ_{initial} is a large parameter, λ should not be small, otherwise very little change in μ will be made during the computations. Lastly, N_{iter} in Algorithm 3 should be a small integer, such as 5, as that is enough to obtain accurate results.

4.5 Statistical Tests

We performed statistical tests on experiments in Table 2 and Figure 2; the results are in Table 6 and Table 7. The null hypothesis H0, which states that the means of the results of the algorithms are the same, is rejected due to a very low p-value. Overall, we see that the proposed methods are consistently ranked the best.

5 Conclusion

This paper derives three algorithms for machine learning learning problems, such as data classification and image segmentation, using maximum flow and duality techniques, similarity graph-based frameworks, semi-supervised procedures, class size information and class homogeneity terms. The third method in particular is tailored for image segmentation problems involving region parameters, in the case the latter is unknown. The new algorithms offer several advantages, including requiring only small amounts of labeled data for good accuracy, in part due to an integration of graph-based and semi-supervised techniques; this feat is important due to the scarcity of labeled data. The proposed algorithms are also based on global minimization frameworks, which allows one to avoid local but not global minima. The methods are also able to incorporate class size information which often improves performance. In addition, the new methods can be used successfully on both large and small data sets, the latter of which can result in poor performances for machine learning methods due a decreased ability for machine learning-based models to learn from the observed data. Moreover, the algorithms are parallelizable during coding. Overall, the three methods form powerful approaches to some of the most important problems of machine learning, and address some of its challenges.

In the future, it would be interesting to investigate a variety of class homogeneity terms for data classification and applications such as 3D point clouds and hyperspectral imaging. Using class homogeneity terms is likely to improve the classification accuracy compared to that of models based primarily on boundary

terms and help avoid trivial global minimizers. For instance, class homogeneity may be defined in terms of the eigendecomposition of the covariance matrix or the graph Laplacian matrix. Moreover, we plan to implement OpenMP directive-based parallelism in our algorithms and optimize the OpenMP implementations.

Funding This work is supported in part by NSF grant DMS-2052983.

Conflicts of Interest The authors declare that they have no conflict of interest.

Data Availability Statement The links to all the data sets analyzed in this paper are included in this paper via citations, and the data is also available at the repository at `https://github.com/kmerkurev/Data`.

References

1. 20 Newsgroups Data Set. `http://qwone.com/~jason/20Newsgroups/`.
2. Accelerate machine learning with active learning. `https://becominghuman.ai/accelerate-machine-learning-with-active-learning-96cea4b72fdb`.
3. Fashion MNIST Data Set. `https://github.com/zalandoresearch/fashion-mnist`.
4. LeNet-5 in 9 lines of code using Keras. `https://medium.com/@mgazar/lenet-5-in-9-lines-of-code-using-keras-ac99294c8086`.
5. LIBSVM – A Library for Support Vector Machines. `https://www.csie.ntu.edu.tw/~cjlin/libsvm/`.
6. Optical Recognition of Handwritten Digits Data Set. `https://archive.ics.uci.edu/ml/datasets/optical+recognition+of+handwritten+digits`.
7. Pen-Based Recognition of Handwritten Digits Data Set. `https://archive.ics.uci.edu/ml/datasets/Pen-Based+Recognition+of+Handwritten+Digits`.
8. Quick introduction to bag-of-words (bow) and tf-idf for creating features from text. `https://www.analyticsvidhya.com/blog/2020/02/quick-introduction-bag-of-words-bow-tf-idf/`.
9. Reuters Data Set. `https://www.cs.umb.edu/~smimarog/textmining/datasets/`.
10. Statlog Data Set. `https://archive.ics.uci.edu/ml/datasets/Statlog+(Landsat+Satellite)`.
11. Text classification with word2vec. `http://nadbordrozd.github.io/blog/2016/05/20/text-classification-with-word2vec/`.
12. VLFeat Library. `https://www.vlfeat.org`.
13. S. Abu-El-Haija, A. Kapoor, B. Perozzi, and J. Lee. N-GCN: Multi-scale graph convolution for semi-supervised node classification. *Uncertainty in Artificial Intelligence*, pages 841–851, 2020.
14. R.K. Ahuja, J.B. Orlin, and R.E. Tarjan. Improved time bounds for the maximum flow problem. *SIAM Journal on Computing*, 18(5):939–954, 1989.
15. E. Bae and E. Merkurjev. Convex variational methods on graphs for multiclass segmentation of high-dimensional data and point clouds. *Journal of Mathematical Imaging and Vision*, 58(3):468–493, 2017.
16. E. Bae and X.-C. Tai. Efficient global minimization for the multiphase Chan-Vese model of image segmentation. 5681:28–41, 2009.
17. E. Bae, X.-C. Tai, and J. Yuan. Maximizing flows with message-passing: Computing spatially continuous min-cuts. In *Energy Minimization Methods in Computer Vision and Pattern Recognition - 10th International Conference, Hong Kong, China, January 13-16, 2015. Proceedings*, pages 15–28, 2014.
18. E. Bae, J. Yuan, and X.-C. Tai. Global minimization for continuous multiphase partitioning problems using a dual approach. *International Journal of Computer Vision*, 92(1):112–129, 2011.
19. E. Bae, J. Yuan, and X.-C. Tai. Simultaneous convex optimization of regions and region parameters in image segmentation models. In *Innovations for Shape Analysis*, pages 421–438. Springer, 2013.
20. E. Bae, J. Yuan, X.-C. Tai, and Y. Boykov. A fast continuous max-flow approach to non-convex multi-labeling problems. In *Efficient Algorithms for Global Optimization Methods in Computer Vision*, pages 134–154. 2014.
21. M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006.
22. S. Belongie, C. Fowlkes, F. Chung, and J. Malik. Spectral partitioning with indefinite kernels using the Nyström extension. In *European Conference on Computer Vision*, pages 531–542, 2002.
23. A.L. Bertozzi and Y. van Gennip. Gamma-convergence of graph Ginzburg-Landau functionals. *Advanced in Differential Equations*, 17(11–12):1115–1180, 2012.
24. Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26:359–374, 2001.
25. P. Braga, H.R. Medeiros, and H.F. Bassani. Deep categorization with semi-supervised self-organizing maps. In *International Joint Conference on Neural Networks*, pages 1–7, 2020.
26. P.H.M Braga and H.F. Bassani. A semi-supervised self-organizing map for clustering and classification. In *International Joint Conference on Neural Networks*, pages 1–8, 2018.
27. J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun. Spectral networks and locally connected networks on graphs. *International Conference on Learning Representation*, 2014.
28. Z. Cang, L. Mu, and G.-W. Wei. Representability of algebraic topology for biomolecules in machine learning based scoring and virtual screening. *PLoS Computational Biology*, 14(1):e1005929, 2018.
29. A. Cardoso. Datasets for single-label text categorization. `http://web.ist.utl.pt/~acardoso/datasets/`, 2007.
30. A. Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, 20(1):89–97, 2004.
31. A. Chambolle and J. Darbon. On total variation minimization and surface evolution using parametric maximum flows. *International Journal of Computer Vision*, 84(3):288, 2009.
32. B.G. Chandran and D.S. Hochbaum. A computational study of the pseudoflow and push-relabel algorithms for the maximum flow problem. *Operations Research*, 57(2):358–376, 2009.

33. O. Chapelle and A. Zien. Semi-supervised classification by low density separation. In *International Conference on Artificial Intelligence and Statistics*, volume 2005, pages 57–64, 2005.
34. Y. Chen and C.J. Kuo. PixelHop: A successive subspace learning (SSL) method for object recognition. *Journal of Visual Communication and Image Representation*, 70:102749, 2020.
35. Y. Chen, Y. Yang, M. Zhang, and C.-C. J. Kuo. Semi-supervised learning via feedforward-designed convolutional neural networks. In *IEEE International Conference on Image Processing*, pages 365–369. IEEE, 2019.
36. B.V. Cherkassky and A.V. Goldberg. On implementing the push-relabel method for the maximum flow problem. *Algorithmica*, 19(4):390–410, 1997.
37. P. Christiano, J.A. Kelner, A. Madry, D.A. Spielman, and S.-H. Teng. Electrical flows, Laplacian systems, and faster approximation of maximum flow in undirected graphs. In *Forty-Third Annual ACM Symposium on Theory of Computing*, pages 273–282, 2011.
38. P. L. Combettes and V. R. Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling & Simulation*, 4(4):1168–1200, 2005.
39. C. Couprie, L. Grady, H. Talbot, and L. Najman. Combinatorial continuous maximum flow. *SIAM Journal on Imaging Sciences*, 4(3):905–930, 2011.
40. F.P. Coutinho. *Construção Automática de Funções de Proximidade para Redes de Termos usando Evolução Gramatical*. PhD thesis, Universidade de São Paulo, 2019.
41. M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery. Learning to extract symbolic knowledge from the world wide web. In *Fifteenth National Conference on Artificial Intelligence*, pages 509–516. AAAI Press, 1998.
42. G. Dantzig and D.R. Fulkerson. On the max flow min cut theorem of networks. *Linear Inequalities and Related Systems*, 38:225–231, 2003.
43. B.V.A. de Lima, A.D.D. Neto, L.E.M. Silva, V.P. Machado, and J.G.C. Costa. Semi-supervised classification using deep learning. In *Brazilian Conference on Intelligent Systems*, pages 717–722. IEEE, 2019.
44. J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. Efficient projections onto the $l1$-ball for learning in high dimensions. In *Proceedings of the 25th International Conference on Machine learning*, pages 272–279, 2008.
45. I. Ekeland and R. Téman. *Convex analysis and variational problems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1999.
46. A. Elmoataz, O.Lezoray, and S. Bougleux. Nonlocal discrete regularization on weighted graphs: a framework for image and manifold processing. *IEEE Transactions On Image Processing*, 17:1047–1060, 2008.
47. J.E. Esser. *Primal dual algorithms for convex models and applications to image restoration, registration and nonlocal inpainting*. UCLA, 2010.
48. C. Fowlkes, S. Belongie, F. Chung, and J. Malik. Spectral grouping using the Nyström method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):214–225, 2004.
49. C. Fowlkes, S. Belongie, and J. Malik. Efficient spatiotemporal grouping using the Nyström method. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–I. IEEE, 2001.
50. A. Gadde, A. Anis, and A. Ortega. Active semi-supervised learning using sampling theory for graph signals. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 492–501, 2014.
51. G. Gallo, M.D. Grigoriadis, and R.E. Tarjan. A fast parametric maximum flow algorithm and applications. *SIAM Journal on Computing*, 18(1):30–55, 1989.
52. A.B. Goldberg, X. Zhu, and S. Wright. Dissimilarity in graph-based semi-supervised classification. In *Artificial Intelligence and Statistics*, pages 155–162, 2007.
53. A.V. Goldberg and R.E. Tarjan. Solving minimum cost flow problems by successive approximation. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing*, May 1987.
54. A.V. Goldberg and R.E. Tarjan. A new approach to the maximum-flow problem. *Journal of the ACM*, 35(4):921–940, 1988.
55. A.V. Goldberg and R.E. Tarjan. Efficient maximum flow algorithms. *Communications of the ACM*, 57(8):82–89, 2014.
56. D. Goldfarb and W. Yin. Parametric maximum flow algorithms for fast total variation minimization. *SIAM Journal on Scientific Computing*, 31(5):3712–3743, 2009.
57. T. Goldstein, X. Bresson, and S. Osher. Global minimization of Markov random fields with applications to optical flow. *Inverse Problems & Imaging*, 6(4):623, 2012.
58. C. Gong, D. Tao, S.J. Maybank, W. Liu, G. Kang, and J. Yang. Multi-modal curriculum learning for semi-supervised image classification. *IEEE Transactions on Image Processing*, 25(7):3249–3260, 2016.
59. W. Hamilton, Z. Ying, and J. Leskovec. Inductive representation learning on large graphs. *Advances in Neural Information Processing Systems*, 30, 2017.
60. S. Han, Z. Peng, and S. Wang. The maximum flow problem of uncertain network. *Information Sciences*, 265:167–175, 2014.
61. T. E. Harris and F. S. Ross. Fundamentals of a method for evaluating rail net capacities. *Research Memorandum*, RM-1573, 1955.
62. D.S. Hochbaum. The pseudoflow algorithm: A new algorithm for the maximum-flow problem. *Operations Research*, 56(4):992–1009, 2008.
63. A. Iscen, G. Tolias, Y. Avrithis, and O. Chum. Label propagation for deep semi-supervised learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5070–5079, 2019.
64. A. Itai, Y. Perl, and Y. Shiloach. The complexity of finding maximum disjoint paths with length constraints. *Networks*, 12(3):277–286, 1982.
65. G.F. Italiano, Y. Nussbaum, P. Sankowski, and C. Wulff-Nilsen. Improved algorithms for min cut and max flow in undirected planar graphs. In *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing*, pages 313–322, 2011.
66. L. Jia, Z. Zhang, L. Wang, W. Jiang, and M. Zhao. Adaptive neighborhood propagation by joint $l2$, 1-norm regularized sparse coding for representation and classification. In *IEEE 16th International Conference on Data Mining*, pages 201–210. IEEE, 2016.
67. A. Jung, A. O. Hero III, A. C. Mara, S. Jahromi, A. Heimowitz, and Y.C. Eldar. Semi-supervised learning in network-structured data via total variation minimization. *IEEE Transactions on Signal Processing*,

67(24):6256–6269, 2019.
68. A. Kapoor, H. Ahn, Y. Qi, and R.W. Picard. Hyperparameter and kernel learning for graph based semi-supervised classification. In *Advances in Neural Information Processing Systems*, pages 627–634, 2006.
69. J.A. Kelner, Y.T. Lee, L. Orecchia, and A. Sidford. An almost-linear-time algorithm for approximate max flow in undirected graphs, and its multicommodity generalizations. In *Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 217–226, 2014.
70. Z.H. Kilimci, S. Akyokus, and S.I. Omurca. The effectiveness of homogenous ensemble classifiers for Turkish and English texts. In *2016 International Symposium on Innovations in Intelligent Systems and Applications*, pages 1–7, 2016.
71. T. Kim, I. Hwang, G.-C. Kang, W.-S. Choi, H. Kim, and B.-T. Zhang. Label propagation adaptive resonance theory for semi-supervised continuous learning. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4012–4016. IEEE, 2020.
72. D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589, 2014.
73. T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.
74. K. C. Kiwiel. Proximal minimization methods with generalized bregman functions. *SIAM Journal on Control and Optimization*, 35:1142–1168, 1995.
75. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
76. Y. LeCun and C. Cortes. The MNIST database of handwritten digits. `http://yann.lecun.com/exdb/mnist/`.
77. Y.T. Lee and A. Sidford. Path finding methods for linear programming: Solving linear programs in $\tilde{O}(\sqrt{rank})$ iterations and faster algorithms for maximum flow. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 424–433. IEEE, 2014.
78. J. Lellmann, J. Kappes, J. Yuan, F. Becker, and C. Schnörr. Convex multi-class image labeling by simplex-constrained total variation. In *Scale Space and Variational Methods in Computer Vision*, volume 5567 of *LNCS*, pages 150–162. Springer, 2009.
79. T. Lesort, H. Caselles-Dupré, M. Garcia-Ortiz, A. Stoian, and D. Filliat. Generative models from the perspective of continual learning. In *International Joint Conference on Neural Networks*, pages 1–8, 2019.
80. J. Levatić, M. Ceci, D. Kocev, and S. Džeroski. Semi-supervised classification trees. *Journal of Intelligent Information Systems*, 49(3):461–486, 2017.
81. J. Li, Q. Zhu, Q. Wu, and D. Cheng. An effective framework based on local cores for self-labeled semi-supervised classification. *Knowledge-Based Systems*, 197:105804, 2020.
82. Q. Li, Z. Han, and X.-M. Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
83. X. Li, H. Yin, K. Zhou, and X. Zhou. Semi-supervised clustering with deep metric learning and graph embedding. *World Wide Web*, 23(2):781–798, 2020.
84. R. Liao, M. Brockschmidt, D. Tarlow, A. Gaunt, R. Urtasun, and R. S. Zemel. Graph partition neural networks for semi-supervised classification. *International Conference on Learning Representations*, 2018.
85. F. Lin and W. W. Cohen. Semi-supervised classification of network data using very few labels. In *2010 International Conference on Advances in Social Networks Analysis and Mining*, pages 192–199. IEEE, 2010.
86. J. Liu and J. Ye. Efficient euclidean projections in linear time. In *Proceedings of the 26th International Conference on Machine Learning*, pages 657–664, 2009.
87. X. Mai and R. Couillet. A random matrix analysis and improvement of semi-supervised learning for large dimensional data. *Journal of Machine Learning Research*, 19:1–27, 2018.
88. D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings of the 8th Internationall Conference on Computer Vision*, volume 2, pages 416–423, July 2001.
89. S. Melacci and M. Belkin. Laplacian support vector machines trained in the primal. *Journal of Machine Learning Research*, 12(3), 2011.
90. E. Merkurjev, E. Bae, A.L. Bertozzi, and X.-C. Tai. Global binary optimization on graphs for classification of high-dimensional data. *Journal of Mathematical Imaging and Vision*, 52(3):414–435, 2015.
91. F. Nie, G. Cai, and X. Li. Multi-view clustering and semi-supervised classification with adaptive neighbours. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.
92. F. Nie, J. Li, and X. Li. Parameter-free auto-weighted multiple graph learning: A framework for multiview clustering and semi-supervised classification. In *International Joint Conference on Artificial Intelligence*, pages 1881–1887, 2016.
93. M. Oghbaie and M. M. Zanjireh. Pairwise document similarity measure based on present term set. *Journal of Big Data*, 5(1):52, 2018.
94. J.B. Orlin. Max flows in $O(nm)$ time, or better. In *Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing*, pages 765–774, 2013.
95. P. Perona and L. Zelnik-Manor. Self-tuning spectral clustering. *Advances in Neural Information Processing Systems*, 17:1601–1608, 2004.
96. N. Piroonsup and S. Sinthupinyo. Analysis of training data using clustering to improve semi-supervised self-training. *Knowledge-Based Systems*, 143:65–80, 2018.
97. R.B. Potts. Some generalized order-disorder transformations. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 48, pages 106–109. Cambridge University Press, 1952.
98. Z. Qi, Y. Tian, and Y. Shi. Laplacian twin support vector machine for semi-supervised classification. *Neural Networks*, 35:46–53, 2012.
99. M. Qu, Y. Bengio, and J. Tang. GMNN: Graph Markov neural networks. *International Conference on Machine Learning*, pages 5241–5250, 2019.
100. Y. Ren, K. Hu, X. Dai, L. Pan, S.C. Hoi, and Z. Xu. Semi-supervised deep embedded clustering. *Neurocomputing*, 325:121–130, 2019.
101. M. F. Rios, J. Calder, and G. Lerman. Analysis and algorithms for ℓ_p-based semi-supervised learning on graphs. *Applied and Computational Harmonic Analysis*, 60:77–122, 2022.
102. R.T. Rockafellar. *Convex Analysis*. Number 28. Princeton University Press, 1970.

103. R.G. Rossi, S.O. Rezende, and A. de Andrade Lopes. Term network approach for transductive classification. *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 497–515, 2015.
104. S. Roy and I.J. Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. In *IEEE Proceedings of International Conference on Computer Vision*, pages 492–499, 1998.
105. A.I. Saleh, M.F. Al Rahmawy, and A.E. Abulwafa. A semantic based web page classification strategy using multi-layered domain ontology. *World Wide Web*, 20(5):939–993, 2017.
106. A.I. Saleh, A.I. El Desouky, and S.H. Ali. Promoting the performance of vertical recommendation systems by applying new classification techniques. *Knowledge-Based Systems*, 75:192–223, 2015.
107. A. Schrijver. On the history of the transportation and maximum flow problems. *Mathematical Programming*, 91(3):437–445, 2002.
108. Z. Shi, S. Osher, and W. Zhu. Weighted nonlocal Laplacian on interpolation from sparse data. *Journal of Scientific Computing*, 73(2):1164–1177, 2017.
109. C. Shui, F. Zhou, C. Gagné, and B. Wang. Deep active learning: Unified and principled method for query and training. In *International Conference on Artificial Intelligence and Statistics*, pages 1308–1318, 2020.
110. V. Sindhwani, P. Niyogi, and M. Belkin. Beyond the point cloud: from transductive to semi-supervised learning. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 824–831, 2005.
111. M. Sion. On general minimax theorems. *Pacific Journal of Mathematics*, 8:171–176, 1958.
112. R.M. Souza and F. Breve. Parallelization of the particle competition and cooperation approach for semi-supervised learning. In *Workshop de Visão Computacional*, pages 402–406, 2015.
113. G. Strang. Maximum flows and minimum cuts in the plane. *Advances in Mechanics and Mathematics*, III:1–11, 2008.
114. A. Subramanya and J. Bilmes. Semi-supervised learning with measure propagation. *Journal of Machine Learning Research*, 12:3311–3370, 2011.
115. M. Teboulle. A unified continuous optimization framework for center-based clustering methods. *Journal of Machine Learning Research*, 8(Jan):65–102, 2007.
116. K. K. Thekumparampil, C. Wang, S. Oh, and L.-J. Li. Attention-based graph neural network for semi-supervised learning. *arXiv preprint arXiv:1803.03735*, 2018.
117. B. Wang, Z. Tu, and J. K. Tsotsos. Dynamic label propagation for semi-supervised multi-class multi-label classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 425–432, 2013.
118. J. Wang, T. Jebara, and S.-F. Chang. Semi-supervised learning using greedy max-cut. *Journal of Machine Learning Research*, 14(Mar):771–800, 2013.
119. M. Wang, W. Fu, S. Hao, D. Tao, and X. Wu. Scalable semi-supervised learning by efficient anchor graph regularization. *IEEE Transactions on Knowledge and Data Engineering*, 28(7):1864–1877, 2016.
120. Z. Wang, L. Wang, R. Chan, and T. Zeng. Large-scale semi-supervised learning via graph structure learning over high-dense points. *arXiv preprint arXiv:1912.02233*, 2019.
121. J. Weston, F. Ratle, H. Mobahi, and R. Collobert. Deep learning via semi-supervised embedding. In *Neural networks: Tricks of the trade*, pages 639–655. 2012.
122. W. Yang, Z.and Cohen and R. Salakhudinov. Revisiting semi-supervised learning with graph embeddings. In *International Conference on Machine Learning*, pages 40–48, 2016.
123. Z. Yang, W. Cohen, and R. Salakhudinov. Revisiting semi-supervised learning with graph embeddings. In *International Conference on Machine Learning*, pages 40–48, 2016.
124. K. Yin and X.-C. Tai. An effective region force for some variational models for learning and clustering. *Journal of Scientific Computing*, 74(1):175–196, 2018.
125. G. Yu, G. Zhang, C. Domeniconi, Z. Yu, and J. You. Semi-supervised classification based on random subspace dimensionality reduction. *Pattern Recognition*, 45(3):1119–1135, 2012.
126. J. Yuan, E. Bae, and X.-C. Tai. A study on continuous max-flow and min-cut approaches. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2217–2224, 2010.
127. J. Yuan, E. Bae, X.-C. Tai, and Y. Boykov. A continuous max-flow approach to Potts model. In *European Conference on Computer Vision*, volume 6316 of *LNCS*, pages 379–392, 2010.
128. J. Yuan, E. Bae, X.-C. Tai, and Y. Boykov. A spatially continuous max-flow and min-cut framework for binary labeling problems. *Numerische Mathematik*, 126(3):559–587, 2013.
129. C. Zach, D. Gallup, J.-M. Frahm, and M. Niethammer. Fast global labeling for real-time stereo using multiple plane sweeps. In *Vision, Modeling and Visualization Workshop*, volume 6, page 7, 2008.
130. Y. Zhang, S. Pal, M. Coates, and D. Ustebay. Bayesian graph convolutional neural networks for semi-supervised classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):5829–5836, 2019.
131. Z. Zhang, L. Jia, M. Zhao, G. Liu, M. Wang, and S. Yan. Kernel-induced label propagation by mapping for semi-supervised classification. *IEEE Transactions on Big Data*, 5(2):148–165, 2018.
132. Z. Zhang, F. Li, L. Jia, J. Qin, L. Zhang, and S. Yan. Robust adaptive embedded label propagation with weight learning for inductive classification. *IEEE Transactions on Neural Networks and Learning Systems*, 29(8):3388–3403, 2017.
133. Z. Zhang, Y. Zhang, F. Li, M. Zhao, L. Zhang, and S. Yan. Discriminative sparse flexible manifold embedding with novel graph for robust visual representation and label propagation. *Pattern Recognition*, 61:492–510, 2017.
134. X. Zhu and Z. Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical report, Carnegie Mellon University, 2002.
135. C. Zhuang and Q. Ma. Dual graph convolutional networks for graph-based semi-supervised classification. In *Proceedings of the 2018 World Wide Web Conference*, pages 499–508, 2018.