

Convex Variational Methods on Graphs for Multiclass Segmentation of High-Dimensional Data and Point Clouds

Egil Bae¹  · Ekaterina Merkurjev²

Received: 31 October 2016 / Accepted: 16 February 2017 / Published online: 22 March 2017
© The Author(s) 2017. This article is an open access publication

Abstract Graph-based variational methods have recently shown to be highly competitive for various classification problems of high-dimensional data, but are inherently difficult to handle from an optimization perspective. This paper proposes a convex relaxation for a certain set of graph-based multiclass data segmentation models involving a graph total variation term, region homogeneity terms, supervised information and certain constraints or penalty terms acting on the class sizes. Particular applications include semi-supervised classification of high-dimensional data and unsupervised segmentation of unstructured 3D point clouds. Theoretical analysis shows that the convex relaxation closely approximates the original NP-hard problems, and these observations are also confirmed experimentally. An efficient duality-based algorithm is developed that handles all constraints on the labeling function implicitly. Experiments on semi-supervised classification indicate consistently higher accuracies than related non-convex approaches and considerably so when the training data are not uniformly distributed among the data set. The accuracies are also highly competitive against a wide range of other established methods on three benchmark data sets. Experiments on 3D point clouds acquired by a LaDAR in outdoor scenes demonstrate that the scenes can accurately be segmented into object classes such as vegetation, the ground plane and human-made structures.

Keywords Variational methods · Graphical models · Convex optimization · Semi-supervised classification · Point cloud segmentation

1 Introduction

The graphical framework has become a popular setting for classification [8, 25, 92, 100–102] and filtering [31, 34, 63, 85, 88, 89] of high-dimensional data. Some of the best performing classification algorithms are based on solving variational problems on graphs [3, 10, 14, 16, 17, 25, 39, 44, 45, 66, 68, 82, 86, 101]. In simple terms, these algorithms attempt to group the data points into classes in such a way that pairs of data points with different class memberships are as dissimilar as possible with respect to a certain feature. In order to avoid the computational complexity of working with fully connected graphs, approximations, such as those based on spectral graph theory [10, 38, 66] or nearest neighbors [17, 33, 68], are typically employed. For example, [10] and [66] employ spectral approaches along with the Nyström extension method [36] to efficiently calculate the eigendecomposition of a dense graph Laplacian. Works, such as [17, 24, 33, 39, 68, 99], use the ‘nearest neighbor’ approach to sparsify the graph for computational efficiency. Variational problems on graphs have also become popular for processing of 3D point clouds [30, 33, 45, 55, 60, 62].

When the classification task is cast as the minimization of similarity of point pairs with different class membership, extra information is necessary to avoid the trivial global minimizer of value zero where all points are assigned to the same class. In semi-supervised classification methods, a small set of the data points are given as training data in advance, and their class memberships are imposed as hard constraints in the optimization problem. In unsupervised classification

✉ Egil Bae
Egil.Bae@ffi.no
Ekaterina Merkurjev
kmerkurev@math.msu.edu

¹ Norwegian Defence Research Establishment, P.O. Box 25,
2027 Kjeller, Norway

² Michigan State University, 220 Trowbridge Rd, East Lansing,
MI 48824, USA

methods, one typically enforces the sizes of each class not to deviate too far from each other, examples including the normalized cut [78] and Cheeger ratio cut problems [26].

Most of the computational methods for semi-supervised and unsupervised classification obtain the solution by computing the local minimizer of a non-convex energy functional. Examples of such algorithms are those based on phase fields [10] and the MBO scheme [38,65–67]. PDEs on graphs for semi-supervised classification also include the Eikonal Equation [30] and tug-of-war games related to the infinity Laplacian Equation [34]. Unsupervised problems with class size constraints are inherently the most difficult to handle from an optimization viewpoint, as the convex envelope of the problem has a trivial constant function as a minimizer [16,78,82]. Various ways of simplifying the energy landscape have been proposed [17,43,89]. Our recent work [68] showed that semi-supervised classification problems with two classes could be formulated in a completely convex framework and also presented efficient algorithms that could obtain global minimizers.

Image segmentation is a special classification problem where the objective is to assign each pixel to a region. Algorithms based on energy minimization are among the most successful image segmentation methods, and they have historically been divided into ‘region-based’ and ‘contour-based.’

Region-based methods attempt to find a partition of the image so that the pixels within each region as a whole are as similar as possible. Additionally, some regularity is imposed on the region boundaries to favor spatial grouping of the pixels. The similarity is usually measured in the statistical sense. In the simplest case, the pixels within each region should be similar to the mean intensity of each region, as proposed in the Chan–Vese [23] and Mumford–Shah [71] models. Contour-based methods [50,93] instead seek the best suited locations of the region boundaries, typically at locations of large jumps in the image intensities, indicating the interface between two objects.

More recently, it has been shown that the combination of region and contour-based terms in the energy function can give qualitatively very good results [15,39,48], especially when non-local operators are used in the contour terms [30,39,48]. There now exists efficient algorithms for solving the resulting optimization problems that can avoid getting stuck in a local minimum, including both combinatorial optimization algorithms [12,13,53] and more recent convex continuous optimization algorithms [6,7,15,20,58,75,96–98]. The latter have shown to be advantageous in several aspects, such as the fact that they require less memory and have a greater potential for parallel implementation of graphics processing units (GPUs), but special care is needed in case of non-local variants of the differential operators (e.g., [76]).

Most of the current data segmentation methods [10,14,16,17,44,66,68,82] can be viewed as ‘contour-based,’ since they seek an optimal location of the boundaries of each region. Region-based variational image segmentation models with two classes were generalized to graphs for data segmentation in [59] and for 3D point cloud segmentation in [59,60,84] in a convex framework. The region terms could be constructed directly from the point geometry and/or be constructed from a color vector defined at the points. Concrete examples of the latter were used for experiments on point cloud segmentation. Region terms have also been proposed in the context of Markov Random Fields for 3D point cloud segmentation [2,72,87], where the weights were learned from training data using associate Markov networks. The independent preprint [94] proposed to use region terms for multiclass semi-supervised classification in a convex manner, where the region terms were inferred from the supervised points by diffusion.

Contributions: This paper proposes a convex relaxation and an efficient algorithmic optimization framework for a general set of graph-based data classification problems that exhibits non-trivial global minimizers. It extends the convex approach for semi-supervised classification with two classes given in our previous work [68] to a much broader range of problems, including multiple classes, novel and more practically useful incorporation of class size information, and a novel unsupervised segmentation model for 3D point clouds acquired by a LaDAR.

The same basic relaxation for semi-supervised classification also appeared in the independent preprint [94]. The most major distinctions of this work compared to the preprint [94] are: we also incorporate class size information in the convex framework; we give a mathematical and experimental analysis of the close relation between the convex relaxed and original problems; we propose a different duality-based ‘max-flow’-inspired algorithm; we incorporate information of the supervised points in a different way; and we consider unsupervised segmentation of 3D point clouds.

The contributions can be summarized more specifically as follows:

- We specify a general set of classification problems that are suitable for being approximated in a convex manner. The general set of problems involves minimization of a multiclass graph cut term together with supervised constraints, region homogeneity terms and novel constraints or penalty terms acting on the class sizes. Special cases include semi-supervised classification of high-dimensional data and unsupervised segmentation of 3D point clouds.
- A convex relaxation is proposed for the general set of problems, and its approximation properties are analyzed thoroughly in theory and experiments. This extends the

work on multiregion image segmentation [6, 58, 98] to data clustering on graphs and to cases where there are constraints or penalty terms acting on the class sizes. Since the introduction of either multiple classes or size constraints causes the general problem to become NP-hard, the relaxation can (probably) not be proved to be exact. Instead, conditions are derived for when an exact global minimizer can be obtained from a dual solution of the relaxed problem. The strongest conditions are derived in case there are no constraints on the class sizes, but the theoretical results in both cases show that very close approximations are expected. These theoretical results also agree well with experimental observations.

- The convex relaxed problems are formulated as equivalent dual problems that are structurally similar to the ‘max-flow’ problem over the graph. This extends our work [68] to multiple classes and the work on image segmentation proposed in [95] to data clustering on graphs. We use a conceptually different proof than [68, 95], which relates ‘max-flow’ to another more direct dual formulation of the problem. Furthermore, it is shown that also the size constraints and penalty term can be incorporated naturally in the max-flow problem by modifying the flow conservation condition, such that there should be a constant flow excess at each node.
- As in our previous work [68, 95], an augmented Lagrangian algorithm is developed based on the new ‘max-flow’ dual formulations of the problems. A key advantage compared to related primal–dual algorithms [21] in imaging, such as the one considered in the preprint [94], is that all constraints on the labeling function are handled implicitly. This includes constraints on the class sizes, which are dealt with by separate dual variables indicating the flow excess at the nodes. Consequently, projections onto the constraint set of the labeling function, which tend to decrease the accuracy and put strict restrictions on the step sizes, are avoided.
- We propose an unsupervised segmentation model for unstructured 3D point clouds acquired by a LaDAR within the general framework. It extends the models of [59, 60, 84] to multiple classes and gives concrete examples of region terms constructed purely based on geometrical information of the unlabeled points, in order to distinguish classes such as vegetation, the ground plane and human-made structures in an outdoor scene. We also propose a graph total variation term that favors alignment of the region boundaries along ‘edges’ indicated by discontinuities in the normal vectors or the depth coordinate. In contrast to [2, 41, 72, 87], our model does not rely on any training data.
- Extensive experimental evaluations on semi-supervised classification indicate consistently higher accuracies than related local minimization approaches, and considerably

so when the training data are not uniformly distributed among the data set. The accuracies are also highly competitive against a wide range of other established methods on three benchmark data sets. The accuracies can be improved further if an estimate of the approximate class sizes are given in advance. Experiments on 3D point clouds acquired by a LaDAR in outdoor scenes demonstrate that the scenes can accurately be segmented into object classes such as vegetation, the ground plane and regular structures. The experiments also demonstrate fast and highly accurate convergence of the algorithms, and show that the approximation difference between the convex and original problems vanishes or becomes extremely low in practice.

Organization: This paper starts by formulating the general set of problems mathematically in Sect. 2. Section 3 formulates a convex relaxation of the general problem and analyzes the quality of the relaxation from a dual perspective. Section 4 reformulates the dual problem as a ‘max-flow’ type of problem and derives an efficient algorithm. Applications to semi-supervised classification of high-dimensional data are presented in Sect. 5.1, and applications to segmentation of unstructured 3D point clouds are described in Sect. 5.2, including specific constructions of each term in the general model. Section 5 also presents a detailed experimental evaluation for both applications.

2 Data Segmentation as Energy Minimization Over a Graph

Assume we are given N data points in \mathbb{R}^M . In order to formulate the segmentation of the data points as a minimization problem, the points are first organized in an undirected graph. Each data point is represented by a node in the graph. The edges in the graph, denoted by E , consist of pairs of data points. Weights $w(x, y)$ on the edges $(x, y) \in E$ measure the similarity between data points x and y . A high value of $w(x, y)$ indicates that x and y are similar and a low value indicates that they are dissimilar. A popular choice for the weight function is the Gaussian

$$w(x, y) = e^{-\frac{d(x, y)^2}{\sigma^2}}, \quad (1)$$

where $d(x, y)$ is the distance, in some sense, between x and y . For example, the distance between two 3D points x and y is naturally their Euclidean distance. In order to reduce the computational burden of working with fully connected graphs, one often only considers the set of edges where $w(x, y)$ is largest. For instance, edges may be constructed between each vertex in V and its k -nearest neighbors. More

formally, for each $x \in V$, one constructs an edge $(x, y) \in E$ for the k points with the shortest distance $d(x, y)$ to x . Such a graph can be constructed efficiently by using kd trees in $O(Nk \log(Nk))$ time [9,46]. Note that the number of edges incident to some nodes in the resulting graph may be larger than k , as illustrated in Fig. 2 where $k = 2$, due to symmetry of the undirected graph. The construction of the graph itself provides a basic segmentation of the nodes; for instance in Fig. 2, it can be observed that the graph contains three different connected components. This fact has been exploited in basic graph-based classification methods [1].

In several recent works, the classification problem has been formulated as finding an optimal partition $\{V_i\}_{i=1}^n$ of the nodes V in the graph G . The most basic objective function can be formulated as

$$\begin{aligned} \min_{\{V_i\}_{i=1}^n} & \sum_{i=1}^n \sum_{\substack{(x,y) \in E : \\ x \in V_i, y \notin V_i}} w(x, y), & (2) \\ \text{s.t. } & \cup_{i=1}^n V_i = V, \quad V_k \cap V_l = \emptyset, \quad \forall k \neq l, & (3) \end{aligned}$$

where the constraint (3) imposes that there should be no vacuum or overlap between the subsets $\{V_i\}_{i=1}^n$. If $n = 2$, then (2) is the so-called graph cut [69]. The motivation behind the model (2) is to group the vertices into classes in such a way that pairs of vertices with different class memberships are as dissimilar as possible, indicated by a low value of w .

2.1 Size Constraints and Supervised Constraints

Extra assumptions are necessary to avoid the trivial global minimizer of (2), where $V_i = V$ for some i and $V_j = \emptyset$ for all $j \neq i$. There are two common ways to incorporate extra assumptions. In semi-supervised classification problems, the class membership of a small set of the data points is given as training data in advance by the constraints

$$V_i \supseteq T_i, \quad i \in I = \{1, \dots, n\}, \quad (4)$$

where T_i is the set of ‘training’ points known to belong to class i . For notational convenience, the set of all class indices $\{1, \dots, n\}$ is denoted by I in the rest of this paper.

In unsupervised classification problems, one usually assumes that the regions should have approximately equal sizes. The simplest way to achieve this is to impose that each class V_i should have a given size $a_i \in \mathbb{N}$:

$$||V_i|| = a_i, \quad i \in I, \quad (5)$$

where $\sum_{i=1}^n a_i = ||V||$. We focus on the case that the norm $||V_i||$ is the number of nodes in V_i for simplicity. As an alternative, $||V_i||$ could be the sum of degrees of each node in V_i , where the degree of a node is the number of edges incident

to that node. If size constraints are introduced, the problem cannot generally be solved exactly due to NP-hardness. This will be discussed in more detail in Sect. 3.

Usually, a more flexible option is preferred of modifying the energy function such that partitions of equal sizes have lower energy. In case of two classes, the energy (2) becomes $\text{cut}(V_1, V_2) = \sum_{x,y} w(x, y)$, where $x \in V_1$ and $y \in V_2$. Several different ways of normalizing the energy by the class sizes have been proposed, which can be summarized as follows

$$\text{cut}(V_1, V_2) \left(\frac{1}{|V_1|} + \frac{1}{|V_2|} \right), \quad \frac{\text{cut}(V_1, V_2)}{\min(|V_1|, |V_2|)}. \quad (6)$$

The expression on the left is called the ratio cut in case of the norm $|V| = \sum_{x \in V} 1$ and the normalized cut in case of $|V| = \sum_{x \in V} \text{degree}(x)$. The expression on the right is called the Cheeger ratio cut with the norm $|V| = \sum_{x \in V} 1$.

The energy functions (6) are highly non-convex, but ways to simplify the energy landscape have been proposed [16,17,44,82] in order to reduce the number of local minima.

2.2 New Flexible Constraint and Penalty Term on Class Sizes

In this paper, we aim to provide a broader set of constraints and penalty terms acting on the class sizes that can be handled in a completely convex manner. They are designed to achieve the same net result as the ratio energies (6) of promoting classes of equal sizes, but in a completely convex way. They can also promote any other size relations between the class sizes. We will consider flexible size constraints of the form

$$S_i^\ell \leq ||V_i|| \leq S_i^u, \quad i \in I, \quad (7)$$

where $S_i^u \in \mathbb{N}$ is an upper bound on the size of class i and $S_i^\ell \in \mathbb{N}$ is a lower bound. Such types of constraints have previously been proposed for image segmentation in [52]. In case one only knows an estimate of the expected class sizes, such constraints can be used to enforce the sizes to lie within some interval of those estimates. To be well defined, it is obviously required that $\sum_{i=1}^n S_i^\ell \leq ||V||$ and $\sum_{i=1}^n S_i^u \geq ||V||$. Note that if $S_i^\ell = S_i^u = a_i$, then (7) becomes equivalent to (5).

To avoid imposing absolute upper and lower bounds on the class sizes, we also propose appending a piecewise linear penalty term $\sum_{i=1}^n P_\gamma(||V_i||)$ to the energy function (2), defined as

$$P_\gamma(||V_i||) = \begin{cases} 0 & \text{if } S_i^\ell \leq ||V_i|| \leq S_i^u \\ \gamma (||V_i|| - S_i^u) & \text{if } ||V_i|| > S_i^u \\ \gamma (S_i^\ell - ||V_i||) & \text{if } ||V_i|| < S_i^\ell \end{cases} \quad (8)$$

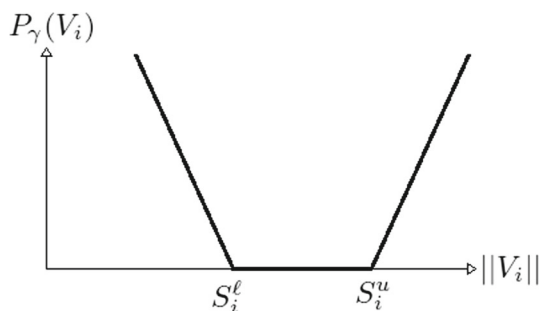


Fig. 1 Illustration of penalty term $P_\gamma(V_i)$

An illustration of $P_\gamma(\|V_i\|)$ is given in Fig. 1. In the limit as $\gamma \rightarrow \infty$, the penalty term becomes an equivalent representation of the hard constraints (7). Note that quadratic or higher-order penalty terms, although they are convex, are not well suited for the convex relaxation, because they tend to encourage non-binary values of the labeling functions. In fact, we believe the set of constraints and penalty terms given here is complete when it comes to being suited for completely convex relaxations.

One major contribution of this paper is an efficient algorithmic framework that handles size constraints of the form (7) and the penalty term (8) naturally, with almost no additional computational efforts.

2.3 Region Homogeneity Terms

The classification problem (2) involves the minimization of an energy on the boundary of the classes. The energy is minimized if the data points on each side of the boundary are as dissimilar as possible. These classification models are therefore similar to edge-based image segmentation models, which align the boundary of the regions along edges in the image where the intensity changes sharply. By contrast, region-based image segmentation models, such as the ‘Chan–Vese’ model, use region homogeneity terms that measure how well each pixel fits with each region, in the energy function.

A graph extension of variational segmentation problems with two classes was formulated in [59,60,62,84], using a non-local total variation term together with a region term promoting homogeneity of a vertex function. The vertex function could be constructed directly from point geometry and/or from external information such as a color vector defined at each point. We extend the general problem to multiple classes and optional constraints as follows:

$$\begin{aligned} \min_{\{V_i\}_{i=1}^n} & \sum_{i=1}^n \sum_{x \in V_i} f_i(x) + \sum_{i=1}^n \sum_{\substack{(x,y) \in E \\ x \in V_i, y \notin V_i}} w(x,y), \\ \text{s.t. } & \cup_{i=1}^n V_i = V, \quad V_k \cap V_l = \emptyset, \quad \forall k \neq l \end{aligned} \tag{9}$$

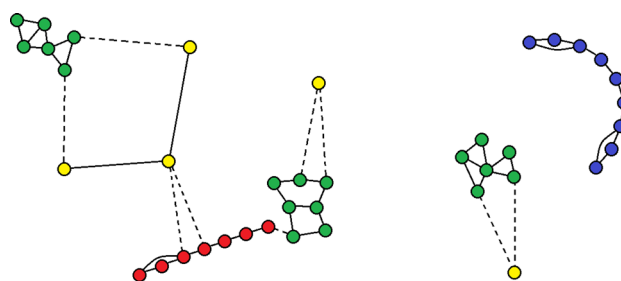


Fig. 2 Example of segmentation of a graph of 2D points (with number of neighbors $k = 2$) into regions of low density (yellow), high degree of correlation of coordinates between neighboring points (red), medium correlation (blue) and low correlation (green). Dashed edges indicate those that contribute to the energy (Color figure online)

under optional supervised constraints (4) and/or size constraints (7)/penalty term (8). In [59,60,62,84], the region terms $f_i(x)$ were defined in terms of a general vertex function f^0 , which could depend on a color vector or the point geometry. Experimental results on point clouds were shown in case f^0 was a color vector defined at each point. In this work, we will give concrete constructions of f_i for point cloud segmentation purely based on the geometry of the 3D points themselves. For example, the eigenvalues and eigenvectors obtained from a local PCA around each point carry useful information for describing the homogeneity within each class. Concrete examples are given in Sect. 5.2. An illustrative example is given in Fig. 2, where each node is a 2D point and the region terms have been constructed to distinguish points with different statistical relations to their neighboring points.

The independent preprint [94], proposed to use region terms in the energy function for semi-supervised classification and the authors, proposed a region term that was inferred from the supervised points by diffusion. In contrast, the region terms in this work do not rely on any supervised points, but are as mentioned only specified and demonstrated for the application of 3D point cloud segmentation.

3 Convex Relaxation of Minimization Problem and Analysis Based on Duality

In this section, the classification problems are formulated as optimization problems in terms of binary functions instead of sets. The binary representations are used to derive convex relaxations. First, some essential mathematical concepts are introduced, such as various differential operators on graphs. These concepts are used extensively to formulate the binary and convex problems and the algorithms.

3.1 Differential Operators on Graphs

Our definitions of operators on graphs are based on the theory in [33,42,91]. More information is found in these papers.

Consider two Hilbert spaces, \mathcal{V} and \mathcal{E} , which are associated with the sets of vertices and edges, respectively, and the following inner products and norms:

$$\begin{aligned} \langle u, \gamma \rangle_{\mathcal{V}} &= \sum_{x \in V} u(x) \gamma(x) d(x)^r, \\ \langle \psi, \phi \rangle_{\mathcal{E}} &= \frac{1}{2} \sum_{x, y \in V} \psi(x, y) \phi(x, y) w(x, y)^{2q-1}, \\ \|u\|_{\mathcal{V}} &= \sqrt{\langle u, u \rangle_{\mathcal{V}}} = \sqrt{\sum_{x \in V} u(x)^2 d(x)^r}, \\ \|\phi\|_{\mathcal{E}} &= \sqrt{\langle \phi, \phi \rangle_{\mathcal{E}}} = \sqrt{\frac{1}{2} \sum_{x, y \in V} \phi(x, y)^2 w(x, y)^{2q-1}}, \\ \|\phi\|_{\mathcal{E}, \infty} &= \max_{x, y \in V} |\phi(x, y)|, \end{aligned} \tag{10}$$

for some $r \in [0, 1]$ and $q \in [\frac{1}{2}, 1]$. Above $d(x)$ is the degree of node x (it’s number of incident nodes) and $w(\cdot, \cdot)$ is the weighting function.

From these definitions, we can define the gradient operator ∇ and the Dirichlet energy as

$$\begin{aligned} (\nabla u)_w(x, y) &= w(x, y)^{1-q} (u(y) - u(x)), \tag{11} \\ \frac{1}{2} \|\nabla u\|_{\mathcal{E}}^2 &= \frac{1}{4} \sum_{x, y \in V} w(x, y) (u(y) - u(x))^2. \tag{12} \end{aligned}$$

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

$$(\text{div}_w \phi)(x) = \frac{1}{2d(x)^r} \sum_{y \in V} w(x, y)^q (\phi(x, y) - \phi(y, x)), \tag{13}$$

where we have exploited symmetry $w(x, y) = w(y, x)$ of the undirected graph in the derivation of the operator.

Using divergence, a family of total variations $TV_w : \mathcal{V} \rightarrow \mathbb{R}$ can now be defined:

$$\begin{aligned} TV_w(u) &= \sup \{ \langle \text{div}_w \phi, u \rangle_{\mathcal{V}} : \phi \in \mathcal{E}, \|\phi\|_{\mathcal{E}, \infty} \leq 1 \} \\ &= \frac{1}{2} \sum_{x, y \in V} w(x, y)^q |u(y) - u(x)|. \end{aligned} \tag{14}$$

The definition of a family of graph Laplacians $\Delta_w = \text{div}_w \nabla : \mathcal{V} \rightarrow \mathcal{V}$ is:

$$(\Delta_w u)(x) = \sum_{y \in V} \frac{w(x, y)}{d(x)^r} (u(y) - u(x)). \tag{15}$$

3.2 Binary Formulation of Energy Minimization Problem

A partition $\{V_i\}_{i=1}^n$ of V satisfying the no vacuum and overlap constraint

$$\cup_{i=1}^n V_i = V, \quad V_k \cap V_l = \emptyset, \quad \forall k \neq l \tag{16}$$

can be described by a binary vector function $u = (u_1, \dots, u_n) : V \mapsto \{0, 1\}^n$ defined as

$$u_i(x) := \begin{cases} 1, & x \in V_i \\ 0, & x \notin V_i \end{cases}, \quad i = 1, \dots, n. \tag{17}$$

In other words, $u(x) = e_i$ if and only if $x \in V_i$, where e_i is the unit normal vector which is 1 at the i th component and 0 for all other components. The no vacuum and overlap constraint (16) can be expressed in terms of u as

$$\sum_{i=1}^n u_i(x) = 1, \quad \forall x \in V. \tag{18}$$

Moreover, note that the minimization term of (2) can be naturally related to total variation (14) for $q = 1$. In fact,

$$\sum_{i=1}^n \sum_{\substack{(x, y) \in E \\ x \in V_i, y \notin V_i}} w(x, y) = \sum_{i=1}^n TV_w(u_i). \tag{19}$$

This connection between the two terms was used in several recent works to derive, utilizing the graphical framework, efficient unsupervised algorithms for clustering. For example, [16,89] formulate rigorous convergence results for two methods that solve the relaxed Cheeger cut problem, using non-local total variation. Moreover, [82] provides a continuous relaxation of the Cheeger cut problem, and derives an efficient algorithm for finding good cuts. The authors of [82] relate the Cheeger cut to total variation and then present a split Bregman approach of solving the problem. In [88] the continuum limit of total variation on point clouds was derived.

The general set of problems (9) can now be formulated in terms of u as

$$\min_{u \in \mathcal{B}} E^P(u) = \sum_{i=1}^n \sum_{x \in V} C_i(x) u_i(x) + \sum_{i=1}^n TV_w(u_i) \tag{20}$$

where

$$\mathcal{B} = \left\{ u : V \mapsto \{0, 1\}^n, \sum_{i=1}^n u_i(x) = 1, \quad \forall x \in V \right\} \tag{21}$$

is the set of binary functions indicating the partition. The superscript P stands for ‘primal.’ The optional size constraints (7) can be imposed in terms of u as

$$S_i^\ell \leq \|u_i\| \leq S_i^u, \quad i \in I,$$

where $\|u_i\| = \sum_{x \in V} u_i(x)$. The size penalty term (8) can be imposed by appending the energy function (20) with the term $\sum_{i=1}^n P_\gamma(\|u_i\|)$.

In case of semi-supervised classification, $C_i(x)$ takes the form of

$$C_i(x) = \eta(x) \sum_{i=1}^n |e_i(x) - u_i^0(x)|^2, \quad (22)$$

where u_i^0 is a binary function taking the value of 1 in T_i and 0 elsewhere, and $\eta(x)$ is a function that takes on a large constant value η on supervised points $\cup_{i=1}^n T_i$ and zero elsewhere. If η is chosen sufficiently large, it can be guaranteed that the solution u satisfies the supervised constraints. The algorithm to be presented in this work does not require the selection of an appropriate value for the parameter η , as the ideal case where $\eta = \infty$ can be handled naturally without introducing numerical instabilities.

Region homogeneity terms can be imposed by setting $C_i(x) = f_i(x)$. More generally, region homogeneity terms and supervised data points can be combined by setting

$$C_i(x) = \eta(x) \sum_{i=1}^n |e_i(x) - u_i^0(x)|^2 + f_i(x), \quad (23)$$

The total variation term is defined as in (14) with $q = 1$.

If the number of supervised points is very low and there is no additional region term, the global minimizer of (20) may become the trivial solution where for one of the classes, say k , $u_k(x) = 1$ everywhere, and for the other classes $u_i(x) = 1$ for supervised points of class i and 0 elsewhere. The threshold tends to occur around less than 2.5% of the points. As in our previous work [68], this problem can be countered by increasing the number of edges incident to supervised points in comparison with other points. Doing so will increase the cost of the trivial solution without significantly influencing the desired global minimizer. An alternative, proposed in the preprint [94], is to create region terms in a preprocessing step by diffusing information of the supervised points into their neighbors.

3.3 Convex Relaxation of Energy Minimization Problem

Due to the binary constraints (21), the problem (20) is non-convex. As in several recent works on variational image segmentation [6, 18, 58, 61, 77, 98] and MRF optimization

[2, 11, 51, 54], we replace the indicator constraint set (21) by the convex unit simplex

$$\mathcal{B}' = \left\{ u : V \mapsto [0, 1]^n, \sum_{i=1}^n u_i(x) = 1, \forall x \in V \right\}. \quad (24)$$

Hence, we are interested in solving the following convex relaxed problem

$$\min_{u \in \mathcal{B}'} E^P(u) = \sum_{i=1}^n \sum_{x \in V} C_i(x) u_i(x) + \sum_{i=1}^n TV_w(u_i). \quad (25)$$

under optional size constraints (7) or penalty term (8). In case $n = 2$ and no size constraints, the relaxation is exact, as proved for image segmentation in [22, 80] and classification problems on graphs in [59, 68]. In case $n > 2$, the problem becomes equivalent to a multiway cut problem, which is known to be NP-hard [28]. In case size constraints are imposed, the problem becomes NP-hard even when $n = 2$, as it becomes equivalent to a knapsack problem [64] in the special case of no TV term.

In this paper, we are interested in using the convex relaxation (25) to solve the original problem approximately. Under certain conditions, the convex relaxation gives an exact global minimizer of the original problem. For instance, it can be straight forwardly shown that

Proposition 1 *Let u^* be a solution of the relaxed problem (25), with optional size constraints (7) or penalty term (8). If $u^* \in \mathcal{B}$, then u^* is a global minimizer of the original non-convex problem (20).*

Proof Let $E^P(u)$ be the energy function defined in (25) with or without the size penalty term (8). Since $\mathcal{B} \subset \mathcal{B}'$ it follows that $\min_{u \in \mathcal{B}'} E^P(u) \leq \min_{u \in \mathcal{B}} E^P(u)$. Therefore, if $u^* = \arg \min_{u \in \mathcal{B}'} E^P(u)$ and $u^* \in \mathcal{B}$ it follows that $E(u^*) = \min_{u \in \mathcal{B}} E^P(u)$. The size constraints (7) can be regarded as a special case by choosing $\gamma = \infty$. \square

If the computed solution of (25) is not completely binary, one way to obtain an approximate binary solution that exactly indicates the class membership of each point is to select the binary function as the nearest vertex in the unit simplex by the threshold

$$u^T(x) = e_\ell(x), \quad \text{where } \ell = \arg \max_{i \in I} u_i(x). \quad (26)$$

As an alternative to the threshold scheme (26), binary solutions of the convex relaxation (25) can also be obtained from a dual solution of (25), which has a more solid theoretical justification if some conditions are fulfilled. The dual problem also gives insight into why the convex relaxation is expected

to closely approximate the original problem. This is the topic of the next section.

3.4 Analysis of Convex Relaxation Through a Dual Formulation

We will now derive theoretical results which indicate that the multiclass problem (20) is closely approximated by the convex relaxation (25). The following results extend those given in [6] from image domains to graphs. In contrast to [6], we also incorporate size constraints or penalty terms in the analysis. In fact, the strongest results given near the end of the section are only valid for problems without such size constraints/terms. This observation agrees well with our experiments, although in both cases very close approximations are obtained.

We start by deriving an equivalent dual formulation of (25). Note that this dual problem is different from the ‘max-flow’ type dual problem on graphs proposed in our previous work [68] in case of two classes. Its main purpose is theoretical analysis, not algorithmic development. In fact, its relation to flow maximization will be the subject of the next section. Dual formulations on graphs have also been proposed in [45] for variational multiscale decomposition of graph signals.

Theorem 1 *The convex relaxed problem (25) S can equivalently be formulated as the dual problem*

$$\sup_{q, \rho^1, \rho^2} \sum_{x \in V} \min_{i \in I} \left(C_i(x) + (\operatorname{div}_w q_i)(x) + \rho_i^2 - \rho_i^1 \right) + \left(\rho_i^1 S_i^\ell - \rho_i^2 S_i^u \right), \tag{27}$$

subject to

$$(q_1, \dots, q_n) \in S_\infty^n, \tag{28}$$

$$\rho_i^1, \rho_i^2 \in [0, \gamma], \quad i = 1, \dots, n, \tag{29}$$

where the above set of infinity norm spheres is defined as

$$S_\infty^n = \{ (q_1, \dots, q_n) : \mathcal{E} \mapsto \mathbb{R}^n \text{ s.t. } \|q_i\|_{\mathcal{E}, \infty} \leq 1 \quad \forall i \}. \tag{30}$$

No size information is incorporated by choosing $\gamma = 0$. The size penalty term (8) is incorporated by choosing $0 < \gamma < \infty$. Size constraints (7) are incorporated by choosing $\gamma = \infty$.

Proof By using the definition of total variation (14), the problem (25) with size penalty term (8) can be expressed in primal–dual form as

$$\min_{u \in \mathcal{B}'} \sup_{q \in S_\infty^n} \sum_{i=1}^n P(\|u_i\|) + \sum_{i=1}^n \sum_{x \in V} u_i(x) (C_i(x) + (\operatorname{div}_w q_i)(x)), \tag{31}$$

where S_∞^n is defined in (30). It will be shown that the size constraints (7) or penalty term (8) can be implicitly incorporated by introducing the dual variables $\rho_i^1, \rho_i^2 \in \mathbb{R}_+, i = 1, \dots, n$ as

$$\min_{u \in \mathcal{B}'} \sup_{q \in S_\infty^n, \rho^1, \rho^2 \in [0, \gamma]^n} E(u; q, \rho^1, \rho^2) = \sum_{i=1}^n \sum_{x \in V} u_i(x) \left\{ C_i(x) + (\operatorname{div}_w q_i)(x) + \rho_i^2 - \rho_i^1 \right\} + \left(\rho_i^1 S_i^\ell - \rho_i^2 S_i^u \right), \tag{32}$$

The primal–dual problem (32) satisfies all the conditions of the mini-max theorem (see, e.g., Chapter 6, Proposition 2.4 of [32]). The constraint sets for q, ρ^1, ρ^2 and u are compact and convex, and the energy function $E(u, q)$ is convex l.s.c. for fixed q and concave u.s.c. for fixed u . This implies the existence of at least one primal–dual solution (saddle point) of finite energy value.

For a given u , the terms involving ρ^1 and ρ^2 can be rearranged as

$$\sup_{0 \leq \rho_i^1 \leq \gamma} \rho_i^1 \left(S_i^\ell - \sum_{x \in V} u_i(x) \right) = \begin{cases} 0 & \text{if } \sum_{x \in V} u_i(x) \geq S_i^\ell \\ \gamma \left(S_i^\ell - \sum_{x \in V} u_i(x) \right) & \text{if } \sum_{x \in V} u_i(x) < S_i^\ell \end{cases} \tag{33}$$

$$\sup_{0 \leq \rho_i^2 \leq \gamma} \rho_i^2 \left(\sum_{x \in V} u_i(x) - S_i^u \right) = \begin{cases} 0 & \text{if } \sum_{x \in V} u_i(x) \leq S_i^u \\ \gamma \left(\sum_{x \in V} u_i(x) - S_i^u \right) & \text{if } \sum_{x \in V} u_i(x) > S_i^u \end{cases} \tag{34}$$

Consider the above three choices for γ . In case $\gamma = 0$, the class sizes do not contribute to the energy. In case $0 < \gamma < \infty$, the two above terms summed together is exactly equal to the size penalty term $P(\|u_i\|)$. In case $\gamma = \infty$, the constraint set on ρ^1, ρ^2 is no longer compact, but we can apply Sion’s generalization of the mini-max theorem [79], which allows either the primal or dual constraint set to be non-compact. It follows that if the size constraints (7) are not satisfied, the energy would be infinite, contradicting existence of a primal–dual solution.

From the mini-max theorems, it also follows that the inf and sup operators can be interchanged as follows

$$\min_{u \in \mathcal{B}'} \sup_{q \in S_{\infty}^n, \rho^1, \rho^2 \in [0, \gamma]^n} E(u; q, \rho^1, \rho^2) = \sup_{q \in S_{\infty}^n, \rho^1, \rho^2 \in [0, \gamma]^n} \min_{u \in \mathcal{B}'} E(u; q, \rho^1, \rho^2). \tag{35}$$

For notational convenience, we denote the unit simplex point-wise as

$$\Delta_+^n = \{(u_1, \dots, u_n) \in [0, 1]^n : \sum_{i=1}^n u_i = 1\} \tag{36}$$

For an arbitrary vector $F = (F_1, \dots, F_n) \in \mathbb{R}^n$, observe that

$$\min_{(u_1, \dots, u_n) \in \Delta_+} \sum_{i=1}^n u_i F_i = \min(F_1, \dots, F_n). \tag{37}$$

Therefore, the inner minimization of (35) can be solved analytically at each position $x \in V$, and we obtain the dual problem

$$\begin{aligned} & \sup_{q \in S_{\infty}^n, \rho^1, \rho^2 \in [0, \gamma]^n} (\rho_1^1 S_i^\ell - \rho_1^2 S_i^u) \\ & + \sum_{x \in V} \min_{u(x) \in \Delta_+} \sum_{i=1}^n u_i \{C_i + \text{div}_w q_i + \rho_i^2 - \rho_i^1\}(x) \\ = & \sup_{q \in S_{\infty}^n, \rho^1, \rho^2 \in [0, \gamma]^n} (\rho_1^1 S_i^\ell - \rho_1^2 S_i^u) \\ & + \sum_{x \in V} \min_{i \in I} \{C_i(x) + (\text{div}_w q_i)(x) + \rho_i^2 - \rho_i^1\}. \end{aligned} \tag{38}$$

Assuming a solution of the dual problem $q^*, \rho^{1*}, \rho^{2*}$ has been obtained, the following theorem characterizes the corresponding primal variable u^*

Theorem 2 *There exists a maximizer $q^*, \rho^{1*}, \rho^{2*}$ to the dual problem (27). At the point $x \in V$, let $I_m(x) = \{i_1, \dots, i_k\}$ be the set of indices such that*

$$I_m(x) = \arg \min_{i \in I} (C_i(x) + (\text{div}_w q_i^*)(x) + \rho_i^{2*} - \rho_i^{1*}). \tag{38}$$

There exists a solution u^ to the primal problem (25) such that $(u^*; q^*, \rho^{1*}, \rho^{2*})$ is a primal–dual pair. At the point x , $u^*(x)$ must satisfy*

$$\sum_{i \in I_m(x)} u_i^*(x) = 1 \text{ and } u_j^*(x) = 0, j \notin I_m(x). \tag{39}$$

If the minimizer (38) is unique at the point $x \in V$, then the corresponding primal solution u^ at the point x must be valued*

$$u_i^*(x) = \begin{cases} 1, & \text{if } i = I_m(x) \\ 0, & \text{if } i \neq I_m(x) \end{cases}, \quad i = 1, \dots, n. \tag{40}$$

If the minimizer (38) is unique at every point $x \in V$, then the corresponding primal solution u^ , given by the formula (40), is an exact global binary minimizer of the original non-convex problem (20).*

Proof Since all conditions of the mini-max theorem [32,79] are satisfied (c.f. proof of Theorem 1), there must exist a maximizer $q^*, \rho^{1*}, \rho^{2*}$ of the dual problem (27) and a minimizer u^* of the primal problem (25) such that (u^*, q^*) is a solution of the primal–dual problem (31) (see, e.g., [32]). For arbitrary vectors $u \in \Delta_+^n$ and $F \in \mathbb{R}^n$, it must hold that $\sum_{i \in I} u_i F_i \geq \min_{i \in I} F_i$. Therefore, at the point x , u^* must satisfy

$$\begin{aligned} & \sum_{i \in I} u_i^*(x) ((C_i + \text{div}_w q_i^*)(x) + \rho_i^{2*} - \rho_i^{1*}) \\ & = \min_{i \in I} ((C_i + \text{div}_w q_i)(x) + \rho_i^{2*} - \rho_i^{1*}); \end{aligned}$$

otherwise, the primal–dual energy would exceed the dual energy, contradicting strong duality. The above expression can be further decomposed as follows

$$\begin{aligned} & = \sum_{i \in I_m(x)} u_i^*(x) ((C_i + \text{div}_w q_i^*)(x) + \rho_i^{2*} - \rho_i^{1*}) \\ & + \sum_{i \notin I_m(x)} u_i^*(x) ((C_i + \text{div}_w q_i^*)(x) + \rho_i^{2*} - \rho_i^{1*}) \\ & = \left(\sum_{i \in I_m(x)} u_i^*(x) \right) \min_{i \in I} ((C_i + \text{div}_w q_i^*)(x) + \rho_i^{2*} - \rho_i^{1*}) \\ & + \sum_{i \notin I_m(x)} u_i^*(x) ((C_i + \text{div}_w q_i^*)(x) + \rho_i^{2*} - \rho_i^{1*}) \end{aligned}$$

Since $((C_j + \text{div}_w q_j^*)(x) + \rho_j^{2*} - \rho_j^{1*})(x) > \min_{i \in I} ((C_i + \text{div}_w q_i^*)(x) + \rho_i^{2*} - \rho_i^{1*})(x)$ for all $j \notin I_m(x)$, the above can only be true provided $\sum_{i \in I_m(x)} u_i^* = 1$ and $u_i^*(x) = 0$ for $i \notin I_m(x)$.

If the minimizer $I_m(x)$ is unique, it follows directly from (39) that $u_i^*(x)$ must be the indicator vector (40).

If the minimizer $I_m(x)$ is unique at every point $x \in V$, then the corresponding primal solution u^* given by (40) is contained in the binary set \mathcal{B} . By Proposition 1, u^* is a global minimizer of (20). \square

It can also be shown that an exact binary primal solution exists if there are two non-unique minimal components to the vector

$$(C(x) + \operatorname{div}_w q^*(x) + \rho^{2*} - \rho^{1*})$$

but this result only holds in case there are no constraints acting on the class sizes.

Theorem 3 *Assume that q^* is a maximizer of the dual problem (27) with $\gamma = 0$, i.e., no class size constraints. If (38) has at most two minimal components for all $x \in V$, then there exists a corresponding binary primal solution to the convex relaxed primal problem (25), which is a global minimizer of the original non-convex problem (20).*

A constructive proof of Theorem 3 is given in ‘‘Appendix.’’

If the vector $(C(x) + \operatorname{div}_w q^*(x) + \rho^{2*} - \rho^{1*})$ has three or more minimal components, it cannot in general be expected that a corresponding binary primal solution exists, reflecting that one can probably not obtain an exact solution to the NP-hard problem (20) in general by a convex relaxation. Experiments indicate that this very rarely, if ever, happens in practice for the classification problem (20).

As an alternative thresholding scheme, u^T can be selected based on the formula (40) after a dual solution to the convex relaxation has been obtained. If there are multiple minimal components to the vector $(C + \operatorname{div} q^*)(x)$, one can select $u^T(x)$ to be one for an arbitrary one of those indices, just as for the ordinary thresholding scheme (26). Experiments will demonstrate and compare both schemes in Sect. 5.

4 ‘Max-Flow’ Formulation of Dual Problem and Algorithm

A drawback of the dual model (27) is the non-smoothness of the objective function, which is also a drawback of the original primal formulation of the convex relaxation. This section reformulates the dual model in a structurally similar way to a max-flow problem, which is smooth and facilitates the development of a very efficient algorithm based on the augmented Lagrangian theory.

The resulting dual problem can be seen as a multiclass variant of the max-flow model proposed in our work [68] for two classes, and a graph analogue of the max-flow model given for image domains in [95]. Note that our derivations differ conceptually from [68, 95], because we directly utilize the dual problem derived in the last section. Furthermore, the new flexible size constraint (7) and penalty term (8) are incorporated naturally in the max-flow problem by a modified flow conservation condition, which indicates that there should be a constant flow excess at each node. The amount of flow excess is expressed with a few additional optimization variables in the algorithm, and they can be optimized over with very little additional computational cost.

4.1 ‘Max-Flow’ Reformulation Dual Problem

We now derive alternative dual and primal–dual formulations of the convex relaxed problem that are more beneficial for computations. The algorithm will be presented in the next section.

Proposition 2 *The dual problem (27) can equivalently be formulated as the dual ‘max-flow’ problem:*

$$\sup_{p_s, p, q, \rho^1, \rho^2} \sum_{x \in V} p_s(x) + \sum_{i=1}^n (\rho_i^1 S_i^\ell - \rho_i^2 S_i^u) \tag{41}$$

subject to, for all $i \in I$,

$$\|q_i(x, y)\| \leq 1, \quad \forall (x, y) \in E, \tag{42}$$

$$p_i(x) \leq C_i(x), \quad \forall x \in V, \tag{43}$$

$$(\operatorname{div}_w q_i - p_s + p_i)(x) = \rho_i^1 - \rho_i^2, \quad \forall x \in V, \tag{44}$$

$$0 \leq \rho_i^1, \rho_i^2 \leq \gamma. \tag{45}$$

Proof By introducing the auxiliary variable $p_s : V \mapsto \mathbb{R}$, the dual problem (27) can be reformulated as follows

$$\begin{aligned} & \sup_{q, \rho^1, \rho^2} \sum_{x \in V} \min_{i \in I} (C_i(x) + \operatorname{div}_w q_i(x) + \rho_i^2 - \rho_i^1) \\ & + \sum_{i=1}^n (\rho_i^1 S_i^\ell - \rho_i^2 S_i^u) \end{aligned}$$

subject to, for all $i \in I$,

$$\|q_i\|_{\mathcal{E}, \infty} \leq 1,$$

$$0 \leq \rho_i^1, \rho_i^2 \leq \gamma.$$

$$= \sup_{p_s, q, \rho^1, \rho^2} \sum_{x \in V} p_s(x) + \sum_{i=1}^n (\rho_i^1 S_i^\ell - \rho_i^2 S_i^u)$$

subject to, for all $i \in I$,

$$p_s(x) \leq (C_i + \operatorname{div}_w q_i)(x) + \rho_i^2 - \rho_i^1 \quad \forall x \in V,$$

$$\|q_i\|_{\mathcal{E}, \infty} \leq 1,$$

$$0 \leq \rho_i^1, \rho_i^2 \leq \gamma. \tag{46}$$

By adding another set of auxiliary variables $p_i : V \mapsto \mathbb{R}$, $i = 1, \dots, n$, the constraints (46) can be formulated as

$$\begin{aligned} p_s(x) &= p_i(x) + \operatorname{div}_w q_i(x) + \rho_i^2 - \rho_i^1, \\ p_i(x) &\leq C_i(x), \end{aligned} \tag{47}$$

for all $x \in V$ and all $i \in I$. Rearranging the terms in constraint (47), and using the definition of the infinity norm (10), leads to the ‘max-flow’ model (41) subject to (42)–(45). \square

Problem (41) with constraints (42)–(45) is structurally similar to a max-flow problem over n copies of the graph G ,

$(V_1, E_1) \times \dots \times (V_n, E_n)$, where $(V_i, E_i) = G$ for $i \in I$. The aim of the max-flow problem is to maximize the flow from a source vertex to a sink vertex under flow capacity at each edge and flow conservation at each node. The variable $p_s(x)$ can be regarded as the flow on the edges from the source to the vertex x in each of the subgraphs $(V_1, E_1), \dots, (V_n, E_n)$, which have unbounded capacities. The variables $p_i(x)$ and $C_i(x)$ can be regarded as the flow and capacity on the edge from vertex x in the subgraph (V_i, E_i) to the sink. Constraint (47) is the flow conservation condition. Observe that in case of size constraints/terms, instead of being conserved, there should be a constant excess flow $\rho_i^1 - \rho_i^2$ for each node in the subgraph (V_i, E_i) . The objective function (41) is a measure of the total amount of flow in the graph.

Utilizing results from Sect. 3.4, we now show that the convex relaxation (25) is the equivalent dual problem to the max-flow problem (41).

Theorem 4 *The following problems are equivalent to each other:*

- (1) *The max-flow problem (41), subject to (42)–(45);*
- (2) *The primal–dual problem:*

$$\begin{aligned} & \min_u \sup_{p_s, p, q, \rho^1, \rho^2} \left\{ E(p_s, p, q, \rho^1, \rho^2; u) \right. \\ & = \sum_{x \in V} p_s(x) + \sum_{i=1}^n \left(\rho_i^1 S_i^\ell - \rho_i^2 S_i^u \right) \\ & \left. + \sum_{i=1}^n \sum_{x \in V} u_i \left(\operatorname{div}_w q_i - p_s + p_i + \rho_i^2 - \rho_i^1 \right) (x) \right\} \end{aligned} \tag{48}$$

subject to (42), (43) and (45), where u is the relaxed region indicator function.

- (3) *The convex relaxed problem (25) with size constraint (7) if $\gamma = \infty$, size penalty term (8) if $0 < \gamma < \infty$ and no size constraints if $\gamma = 0$.*

Proof The equivalence between the primal–dual problem (48) and the max-flow problem (41) follows directly as u_i is an unconstrained Lagrange multiplier for the flow conservation constraint (47). Existence of the Lagrange multipliers follows as: (1) (41) is upper bounded, since it is equivalent to (27), which by Theorem 2 admits a solution, and (2) the constraints (44) are linear and hence differentiable.

The equivalence between the primal–dual problem (48), the max-flow problem (41) and the convex relaxed problem (25) now follows: By Proposition 2 the ‘max-flow’ problem (41) is equivalent to the dual problem (27). By Theorem 1, the dual problem (27) is equivalent to the convex relaxed problem (25) with size constraints (7) if $\gamma = \infty$, size penalty term (8) if $0 < \gamma < \infty$ and no size constraints if $\gamma = 0$. \square

Note an important distinction between the primal–dual problem (48) and the primal–dual problem (35) derived in the last section: The primal variable u is unconstrained in (48). The simplex constraint \mathcal{B}' is handled implicitly. It may not seem obvious from the proof how the constraints on u are encoded in the primal–dual problem; therefore, we give some further insights: For a given primal variable u , the maximization with respect to p_s of the primal–dual problem (48) at the point x can be rearranged as

$$\begin{aligned} & \sup_{p_s(x)} \left(\left(1 - \sum_{i=1}^n u_i \right) p_s \right) (x) \\ & = \begin{cases} 0 & \text{if } \sum_{i=1}^n u_i(x) = 1 \\ \infty & \text{if } \sum_{i=1}^n u_i(x) \neq 1 \end{cases} \end{aligned} \tag{49}$$

If u does not satisfy the sum to one constraint at x , then the primal–dual energy would be infinite, contradicting boundedness from above. In a similar manner, the optimization with respect to p_i can be expressed as

$$\sup_{p_i(x) \leq C_i(x)} u_i(x) p_i(x) = \begin{cases} (u_i C_i)(x) & \text{if } u_i(x) \geq 0 \\ \infty & \text{if } u_i(x) < 0. \end{cases} \tag{50}$$

which would be infinite if u does not satisfy the non-negativity constraints. If $u(x) = e_i$, the indicator function of class i , the value would be $C_i(x)$, which is indeed the pointwise cost of assigning x to class i .

4.2 Augmented Lagrangian Max-flow Algorithm

This section derives an efficient algorithm, which exploits the fact all constraints on u are handled implicitly in the primal–dual problem (48). The algorithm is based on the augmented Lagrangian theory, where u is updated as a Lagrange multiplier by a gradient descent step each iteration. Since no subsequent projection of u is necessary, the algorithm tolerates a wide range of step sizes and converges with high accuracy. The advantages of related ‘max-flow’ algorithms for ordinary 2D imaging problems over, e.g., Arrow–Hurwicz type primal–dual algorithms have been demonstrated in [5, 97].

From the primal–dual problem (48), we first construct the augmented Lagrangian functional:

$$\begin{aligned} L_c & = \sum_{x \in V} p_s + \sum_{i=1}^n \left(\rho_i^1 S_i^\ell - \rho_i^2 S_i^u \right) \\ & + \sum_{x \in V} u_i(x) \left(\operatorname{div}_w q_i - p_s + p_i + \rho_i^2 - \rho_i^1 \right) (x) \\ & - \frac{c}{2} \sum_{i=1}^n \left\| \operatorname{div}_w q_i - p_s + p_i + \rho_i^2 - \rho_i^1 \right\|_2^2. \end{aligned} \tag{51}$$

An augmented Lagrangian algorithm for minimizing the above functional is given below, which involves alternatively maximizing L_c for the dual variables and then updating the Lagrange multiplier u .

Note that if there are no constraints on the class sizes, $\gamma = 0$, then $\rho^{1k} = \rho^{2k} \equiv 0$ for every iteration k . The algorithm can in this case be simplified by setting $\rho^{1k} = \rho^{2k} \equiv 0$ for all k and ignoring all steps involving ρ^1 and ρ^2 .

Algorithm 1

Initialize $p_s^1, p^1, q^1, \rho^1, \rho^2$ and u^1 . For $k = 1, \dots$ until convergence:

- Optimize q flow, for $i \in I$

$$q_i^{k+1} = \arg \max_{|q(e)| \leq 1 \forall e \in E} -\frac{c}{2} \left\| \operatorname{div}_w q - F^k \right\|_2^2, \quad (52)$$

where $F^k = p_s^k - p_i^k + \frac{u_i^k}{c} - \rho_i^{2k} + \rho_i^{1k}$ is fixed.

- Optimize source flow p_s

$$p_s^{k+1} = \arg \max_{p_s(x)} \sum_{x \in V} p_s - \frac{c}{2} \left\| p_s - G^k \right\|_2^2, \quad (53)$$

where $G^k = p_i^k + \operatorname{div}_w q_i^{k+1} - \frac{u_i^k}{c} + \rho_i^{2k} - \rho_i^{1k+1}$ is fixed.

- Optimize sink flow p_i , for $i \in I$,

$$p_i^{k+1} := \arg \max_{p_i(x) \leq C_i(x) \forall x \in V} -\frac{c}{2} \left\| p_i - H^k \right\|_2^2, \quad (54)$$

where $H^k = p_s^{k+1} - \operatorname{div}_w q_i^{k+1} + \frac{u_i^k}{c} - \rho_i^{2k} + \rho_i^{1k}$ is fixed.

- Optimize ρ_i^1 , for $i \in I$,

$$\rho_i^{1k+1} = \arg \max_{0 \leq \rho_i^1 \leq \gamma} \sum_{x \in V} \rho_i^1 S_i^\ell - \frac{c}{2} \left\| \rho_i^1 - J^k \right\|_2^2, \quad (55)$$

where $J^k = -p_i^{k+1} - \operatorname{div}_w q_i^{k+1} + \frac{u_i^k}{c} + p_s^{k+1} - \rho_i^{2k}$ is fixed.

- Optimize ρ_i^2 , for $i \in I$,

$$\rho_i^{2k+1} = \arg \max_{0 \leq \rho_i^2 \leq \gamma} \sum_{x \in V} -\rho_i^2 S_i^u - \frac{c}{2} \left\| \rho_i^2 - M^k \right\|_2^2, \quad (56)$$

where $M^k = p_i^{k+1} + \operatorname{div}_w q_i^{k+1} - \frac{u_i^k}{c} - p_s^{k+1} - \rho_i^{1k+1}$ is fixed.

- Update u_i , for $i \in I$

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

The optimization problem (52) can be solved by a few steps of the projected gradient method as follows:

$$q_i^{k+1} = \Pi_W(q_i + c \nabla_w (\operatorname{div}_w q_i^k - F^k)), \quad (57)$$

Above, Π_w is a projection operator which is defined as

$$\Pi_W(s(x, y)) = \begin{cases} s(x, y) & \text{if } |s(x, y)| \leq 1, \\ \operatorname{sgn}(s(x, y)) & \text{if } |s(x, y)| > 1, \end{cases} \quad (58)$$

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., [35, 40]. In our experience, one gradient ascent iteration leads to the fastest overall speed of convergence.

The subproblems (53) and (54) can be solved by

$$p_s(x) = G^k(x) + \frac{1}{c}, \quad (59)$$

$$p_i(x) = \min \left(H^k(x), C_i(x) \right). \quad (60)$$

Consider now the subproblems (55) and (56). 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

$$\operatorname{mean} \left(-J^k + \frac{S_i^\ell}{c \|V\|} \right), \quad \operatorname{mean} \left(-M^k - \frac{S_i^u}{c \|V\|} \right),$$

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 problem can now be obtained by a projection onto that constraint as follows

$$\rho_i^{1k+1} = \min \left(\max \left(\operatorname{mean} \left(-J^k + \frac{S_i^\ell}{c \|V\|} \right), 0 \right), \gamma \right), \quad (61)$$

$$\rho_i^{2k+1} = \min \left(\max \left(\operatorname{mean} \left(-M^k - \frac{S_i^u}{c \|V\|} \right), 0 \right), \gamma \right). \quad (62)$$

Algorithm 1 is suitable for parallel implementation on GPU, since the subproblems at each substep can be solved point-wise independently of each other using simple floating point arithmetics. The update formula (57) for q only requires access to the values of neighboring nodes at the previous iterate. As discussed in Sect. 2, the number of neighbors may vary for nodes across the graph; therefore, special considerations should be taken when declaring memory. We have implemented the algorithm on CPU for experimental evaluation for simplicity.

5 Applications and Experiments

We now focus on specific applications of the convex framework. Experimental results on semi-supervised classification of high-dimensional data are presented in Sect. 5.1. Section 5.2 proposes specific terms in the general model (9) for segmentation of unstructured 3D point clouds and presents experimental results on LaDAR data acquired in outdoor scenes. In both cases, we give a thorough examination of accuracy of the results, tightness of the convex relaxations and convergence properties of the algorithms.

A useful quality measure of the convex relaxation is to what extent the computed solution is binary. Proposition 1 indicates that if the computed solution is completely binary, it is also an exact global minimizer to the original non-convex problem. Let u^{kT} be a thresholding of $u^k(x)$ in the sense that each row of u^k is modified to be the closest vertex in the unit simplex according to the scheme (26). As a quality measure of the solution u^k at each iteration k of Algorithm 1, we calculate the average difference between u^k and its thresholded version u^{kT} as follows:

$$b(u^k) = \frac{1}{2n||V||} \left(\sum_{i=1}^n \sum_{x \in V} |u_i^{kT}(x) - u_i^k(x)| \right), \tag{63}$$

where $||V||$ is the number of nodes and n is the number of classes. We call $b(u^k)$ the ‘binary difference’ of u^k at iteration k . Naturally, we want $b(u^k)$ to become as low as possible as the algorithm converges.

5.1 Semi-supervised Classification Results

In this section, we describe the supervised classification results, using the algorithm with and without the size constraints (5), (7) or penalty term (8).

We compare the accuracy of the results with respect to the ground truth. The results are also compared against other local minimization approaches in terms of the final total variation energies:

$$E(u) = \frac{1}{2} \sum_{i=1}^n \sum_{x,y \in V} w(x,y) |u_i(x) - u_i(y)|,$$

where n is the number of classes. A lower value of E is better. The energy contribution from the fidelity term is ignored because the solution satisfies the supervised constraints by construction, thus giving zero contribution from those terms.

To compute the weights for the data sets, we use the Zelnik-Manor and Perona weight function [74]. The function is defined as:

$$w(x,y) = \exp \left(- \frac{d(x,y)^2}{\sqrt{\tau(x)\tau(y)}} \right), \tag{64}$$

where $d(x,y)$ is a distance measure between vertices x and y , and $\sqrt{\tau(x)}$ is the distance between vertex x and its M th closest neighbor. If y is not among the M -nearest neighbors of x , then $w(x,y)$ is set to 0. After the graph is computed, we symmetrize it by setting

$$w(x,y) = \max(w(x,y), w(y,x)).$$

Here, M is a parameter to be chosen. The weight function will be defined more specifically for each application.

We run the minimization procedure until the following stopping criterion is satisfied:

$$\frac{1}{||V||} \left(\sum_i \sum_{x \in V} |u_i(x) - u_i^{old}(x)| \right) < \delta,$$

where u^{old} is the u from the previous iteration, and the value of δ varies depending on the data set (anywhere from 10^{-12} to 10^{-10}).

All experiments were performed on a 2.4 GHz Intel Core i2 Quad CPU. We initialize $C_i(x) = \text{constant}$ (in our case, the constant is set to 500) if x is a supervised point of any class but class i , and 0 otherwise, for all $i \in I$. The variables $u, q_i, \rho_i^1, \rho_i^2$ are initialized to zero for all $i \in I$. The variable p_s is initialized to C_n , where n is the number of classes. We set $p_i = p_s \quad \forall i \in I$.

In the following, we give details about the setup and results for each data set, before we draw some general conclusions in the end.

5.1.1 MNIST

The MNIST data set [56], affiliated with the Courant Institute of New York University, consists of 70,000 28×28 images of handwritten digits 0 through 9. Some of the images in the database are shown in Fig. 3. The objective is, of course, to assign the correct digit to each image; thus, this is a 10-class segmentation problem.



Fig. 3 Examples of digits from the MNIST data base

We construct the graph as follows; each image is a node on a graph, described by the feature vector of 784 pixel intensity values in the image. These feature vectors are used to compute the weights for pairs of nodes. The weight matrix is computed using the Zelnik–Manor and Perona weight function (64) with local scaling using the 8th closest neighbor. We note that preprocessing of the data is not needed to obtain an accurate classification; we do not perform any preprocessing. The parameter c used was 0.05.

The average accuracy results over 100 different runs with randomly chosen supervised points are shown in Table 1 in case of no size constraints. We note that the new approaches reach consistently higher accuracies and lower energies than related local minimization approaches, and that incorporation of size information can improve the accuracies further. The computation times are highly efficient, but not quite as fast as MBO, which only uses 10 iteration to solve the problem in an approximate manner. The Log_{10} plots of the binary difference versus iteration, depicted in Fig. 7, show

that the binary difference converges to an extremely small number.

The results of the data set are visualized in Fig. 4. For the visualization procedure, we use the first and the sixth eigenvector of the graph Laplacian. The dimension of each of the eigenvectors is $N \times 1$, and each node of the data set is associated with a value of each of the vectors. One way to visualize a classification of a data set such as MNIST, which consists of a collection of images is to plot the values of one eigenvector of the graph Laplacian versus another and use colors to differentiate classes in a given segmentation. In this case, the plots in Fig. 4 graph the values of the first versus the sixth eigenvector (of the graph Laplacian) relating to the nodes of class 4 and 9 only. The blue and red region represents nodes of class 4 and 9, respectively. The green region represents misclassified points.

Moreover, we compare our results to those of other methods in Table 1, where our method's name is written in bold. Note that algorithms such as linear and nonlinear classifiers, boosted stumps, support vector machines and both neural and convolution nets are all supervised learning approaches, which use around 60,000 of the images as a training set (86% of the data) and 10,000 images for testing. However, we use only 3.57% (or less) of our data as supervised training points, and obtain classification results that are either competitive or better than those of some of the best methods. Moreover, note that no preprocessing was performed on the data, as was needed for some of the methods we compare with; we worked with the raw data directly.

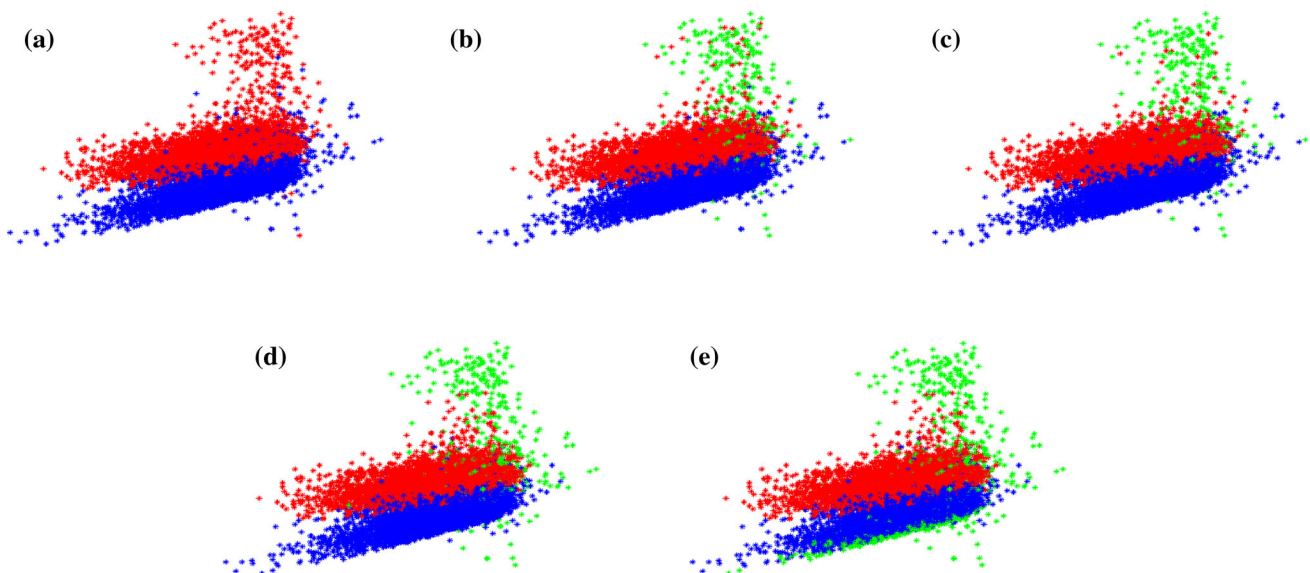


Fig. 4 MNIST results. These graphs visualize the values of the first versus the sixth eigenvector (of the graph Laplacian) relating to the nodes of class 4 and 9 only. The *blue* and *red* region represents nodes of class 4 and 9, respectively. The *green* region represents misclassified

points. **a** Ground truth; **b** proposed result (randomly selected supervised points); **c** proposed result (non-randomly selected supervised points); **d** MBO result (randomly selected supervised points); **e** MBO result (non-randomly selected supervised points) (Color figure online)

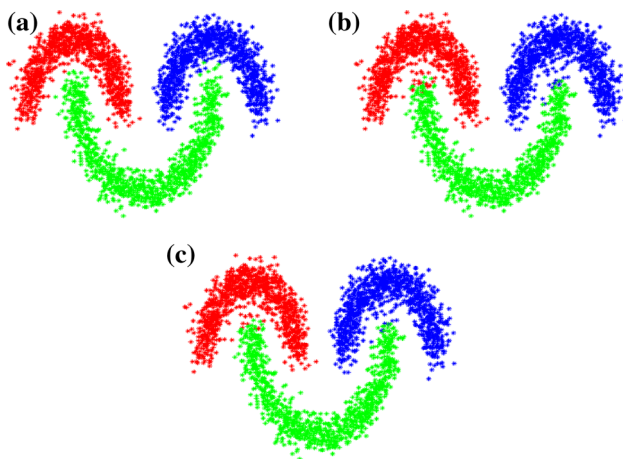


Fig. 5 Three moons results. **a** Ground truth; **b** proposed result **c** MBO result

5.1.2 Three Moons Data Set

We created a synthetic data set, called the three moons data set, to test our method. The set is constructed as follows. First, consider three half circles in \mathbb{R}^2 . The first two half top circles are unit circles with centers at $(0, 0)$ and $(3, 0)$. The third half circle is a bottom half circle with radius of 1.5 and center at $(1.5, 0.4)$. A thousand points from each of the three half circles are sampled and embedded in \mathbb{R}^{100} by adding Gaussian noise with standard deviation of 0.14 to each of the 100 components of each embedded point. The goal is to segment the circles, using a small number of supervised points from each class. Thus, this is a 3-class segmentation problem. The noise and the fact that the points are embedded in high-dimensional space make this difficult.

We construct the graph as follows; each point is a node on a graph, described by the feature vector consisting of the 100 dimensions of the point. To compute the distance component of the weight function for pairs of nodes, we use these feature vectors. The weight matrix is computed using the Zelnik–Manor and Perona weight function (64) with local scaling using the 10th nearest neighbor. The parameter c was 0.1.

The results of the data set are visualized in Fig. 5, and the accuracies are shown in Table 1. This is the only data set where the proposed approach got lower accuracy than MBO. For this particular example, the global minimizer does not seem the best in terms of accuracy, which is a fault of the model rather than an optimization procedure.

5.1.3 COIL

We evaluated our performance on the benchmark COIL data set [25, 73] from the Columbia University Image Library. This is a set of color 128×128 images of 100 objects, taken at different angles. The red channel of each image was down-

sampled to 16×16 pixels by averaging over blocks of 8×8 pixels. Then, 24 of the objects were randomly selected and then partitioned into six classes. Discarding 38 images from each class leaves 250 per class, giving a data set of 1500 data points and 6 classes.

We construct the graph as follows; each image is a node on a graph. We apply PCA to project each image onto 241 principal components; these components form the feature vectors. The vectors are used to calculate the distance component of the weight function. The weight matrix is computed using the Zelnik–Manor and Perona weight function (64) with local scaling using the 4th nearest neighbor. The parameter c used was 0.03.

Resulting accuracies are shown in Table 1, indicating that our method outperforms local minimization approaches and is comparable to or better than some of the other best existing methods. The results of the data set are visualized in Fig. 6; the procedure used is similar to that of the MNIST data set visualization procedure. The plots in the figure graph the values of the first versus the third eigenvector of the graph Laplacian. The results of the classification are labeled by different colors.

5.1.4 Landsat Satellite Data Set

We also evaluated our performance on the Landsat Satellite data set, obtained from the UCI Machine Learning Repository [4]. This is a hyperspectral data set which is composed of sets of multispectral values of pixels in 3×3 neighborhoods in a satellite image; the portions of the electromagnetic spectrum covered include near-infrared. The goal is to predict the classification of the central pixel in each element of the data set. The six classes are red soil, cotton crop, gray soil, damp gray soil, soil with vegetation stubble and very damp gray soil. There are 6435 nodes in the data set.

We construct the graph as follows. The UCI Web site provides a 36-dimensional feature vector for each node. The feature vectors are used to calculate the distance component of the weight function. The weight matrix is computed using the Zelnik–Manor and Perona weight function (64) with local scaling using the 4th nearest neighbor. The parameter c used was 0.3.

Table 1 includes comparison of our method to some of the best methods (most cited in [70]). One can see that our results are of higher accuracy. We now note that, except the GL and MBO algorithms, all other algorithms we compare the Landsat satellite data to are supervised learning methods, which use 80% of data for training and 20% for testing. Our method was able to outperform these algorithms while using a very small percentage of the data set (10%) as supervised points. Even with 5.6% supervised points, it outperforms all but one of the aforementioned methods.

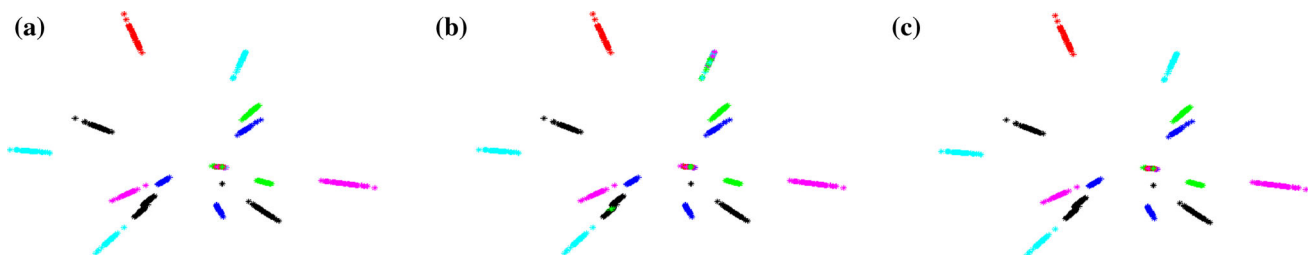


Fig. 6 COIL Results. These graphs visualize the values of the first versus the third eigenvector of the graph Laplacian. The results of the classification are labeled by *different colors*: **a** Ground truth; **b** MBO

result (non-randomly selected supervised points); **c** proposed result (non-randomly selected supervised points) (Color figure online)

5.1.5 Non-uniform Distribution of Supervised Points

In all previous experiments, the supervised points have been sampled randomly from all the data points. To test the algorithms in more challenging scenarios, we introduce some bias in the sampling of the supervised points, which is also a more realistic situation in practice. We used two different data sets for this test: the MNIST data set and the COIL data set.

In the case of the MNIST data set, we chose the supervised points non-randomly for digits 4 and 9 only. To obtain the non-randomness, we allowed a point to be chosen as supervised only if it had a particular range of values for the second eigenvector. This resulted in a biased distribution of the supervised points. The results for this experiment were the following: For the max-flow algorithm, the overall accuracy was 97.734%, while for digits 4 and 9, it was 96.85%. For comparison, the non-convex MBO algorithm [38] gave an accuracy of 95.60% overall, but 89.71% for digits 4 and 9. The MBO method was also a bit more unstable in its accuracy with respect to different distributions of the supervised points. The max-flow algorithm was very stable, with a very small standard deviation for a set of accuracies for different supervised point distributions.

In the case of the COIL data set, we chose the supervised points non-randomly for classes 2 and 6. The non-randomness was achieved in the same way as for the MNIST data set. The results were the following: The overall accuracy of the max-flow was 92.69%, while for classes 2 and 6, it was 90.89%. The MBO algorithm [38] gave an accuracy of 83.90% overall, but 77.24% for classes 2 and 6.

These results are summarized in Table 2 and are visualized in Figs. 4 and 6 for MNIST and COIL data sets, respectively.

5.1.6 Experiments with Size Constraints and Penalty Term

The exact size constraints (5) could improve the accuracies if knowledge of the exact class sizes are available. However, it is not realistic to obtain the exact knowledge of the class sizes in practice, and this was the motivation behind developing the flexible constraints (7) or the penalty term (8). In

order to simulate the case that only an estimate of the class sizes are known, we perturb the exact class sizes by a random number ranging between 1 and 20 % of $\|V\|/n$. The lower and upper bounds in (7) and (8) are centered around the perturbed class size, and the difference between them is chosen based on the uncertainty of the estimation, which we assume to be known. More specifically, denoting the exact class size c_i , the perturbed class size \tilde{c}_i is chosen as a random number in the interval $[c_i - p, c_i + p]$. In experiments, we select p as 1, 10 and 20 % of $\|V\|/n$. The lower and upper bounds in the flexible size constraint (7) and the penalty term (8) are chosen as $S_i^l = \tilde{c}_i - p$ and $S_i^u = \tilde{c}_i + p$. The parameter γ in the penalty term is set to 10 for all data sets.

We run the algorithm for each choice of p several times with different random selections of the perturbed class size \tilde{c}_i each time. The average accuracies over all the runs for each choice of p are shown in Table 3. The flexible size constraints or penalty term improve the accuracy compared to the case when no size information was given, as shown in Table 1. Note that the accuracy improves also in cases of great uncertainty in the class size estimates ($p = 20\%$). The exact size constraints can be seen to not be suitable in case knowledge of the exact class sizes are not known, as imposing them significantly reduces the accuracies in those cases (Table 4).

5.1.7 Summary of Experimental Results

Experimental results on the benchmark data sets, shown in Table 1, indicate a consistently higher accuracy of the proposed convex algorithm than related local minimization approaches based on the MBO or Ginzburg–Landau scheme. The improvements are especially significant when the supervised points are not uniformly distributed among the data set as shown in Table 2. On one synthetic data set, ‘three moons,’ the accuracy of the new algorithm was slightly worse, indicating that the global minimizer was not the best in terms of accuracy for this particular toy example. Table 5 shows that the new algorithm reaches the lowest energy on all of the experiments, further indicating that MBO and Ginzburg–

Table 1 Accuracy compared to ground truth of the proposed algorithm versus other algorithms

Method	Accuracy (%)
<i>MNIST</i> (10 classes) ^a	
p-Laplacian [19]	87.1
Multicut normalized 1-cut [43]	87.64
Linear classifiers [56,57]	88
Cheeger cuts [82]	88.2
Boosted stumps ^a [49,56]	92.3–98.74
Transductive classification [83]	92.6
Tree GL [37]	93.0
<i>k</i> -nearest neighbors ^a [56,57]	95.0–97.17
Neural/conv. nets ^a [27,56,57]	95.3–99.65
Nonlinear classifiers* [56,57]	96.4–96.7
SVM ^a [29,57]	98.6–99.32
GL [38] (3.57% supervised pts.)	96.8
MBO [38] (3.57% supervised pts.)	96.91
Proposed (3.57% supervised pts.)	97.709
<i>Three moons</i> (5% supervised points)	
GL [38]	98.4
MBO [38]	99.12
Proposed	98.714
<i>COIL</i> (10% supervised points)	
<i>k</i> -nearest neighbors [81]	83.5
LapRLS [8,81]	87.8
sGT [47,81]	89.9
SQ-Loss-I [81]	90.9
MP [81]	91.1
GL [38]	91.2
MBO [38]	91.46
Proposed	93.302
<i>Landsat satellite data set</i> ^b	
SC-SVM ^b [70]	65.15
SH- SVM ^b [70]	75.43
S-LS ^b [70]	65.88
Simplex boosting ^b [70]	86.65
S-LS rbf. ^b [70]	90.15
GL [38] (10% supervised pts.)	87.62
GL [38] (5.6% supervised pts.)	87.05
MBO [38] (10% supervised pts.)	87.76
MBO [38] (5.6% supervised pts.)	87.25
Proposed (10% supervised pts.)	90.267
Proposed (5.6% supervised pts.)	88.621

Bold values indicate the results of the new algorithm

^a Note that some of the comparable algorithms, marked by ^a, use substantially more data for training (85.7% at most and 21.4% at smallest) than the proposed algorithm; see the main text for more information.

^b Marked use 80% of the data set for training; see main text for more information

Table 2 Accuracies in case of non-uniformly distributed supervised points

	Overall (%)	Classes 4 and 9 (%)
Proposed, MNIST	97.734	96.85
MBO, MNIST	95.60	89.72
	Overall (%)	Classes 2 and 6 (%)
Proposed, COIL	92.69	90.89
MBO, COIL	83.90	77.24

Bold values indicate the results of the new algorithm

Table 3 Accuracies for experiments with class size incorporation.

Max size perturbation (<i>p</i>)	1%	10%	20%
<i>MNIST</i> , 3.57% supervised points			
Flexible size constraints (7)	97.761	97.725	97.716
Penalty term (8)	97.755	97.739	97.722
Exact size constraints (5)	96.139	70.820	63.660
<i>Three moons</i>			
5% supervised points			
Flexible size constraints (7)	99.374	98.829	98.750
Penalty term (8)	99.368	98.789	98.718
Exact size constraints (5)	99.108	72.685	66.627
0.6% supervised points			
Flexible size constraints (7)	97.833	97.738	97.160
Penalty term (8)	97.848	97.793	97.406
Exact size constraints (5)	97.706	68.956	66.872
<i>COIL</i>			
10% supervised points			
Flexible size constraints (7)	93.403	93.535	93.527
Penalty term (8)	93.360	93.418	93.325
Exact size constraints (5)	92.990	59.936	55.624
5% supervised points			
Flexible size constraints (7)	90.428	90.892	90.730
Penalty term (8)	89.957	90.967	90.712
Exact size constraints (5)	89.931	55.152	54.674
<i>Landsat satellite data set</i>			
10% supervised points			
Flexible size constraints (7)	90.504	90.397	90.344
Penalty term (8)	90.479	90.371	90.347
Exact size constraints (5)	87.773	67.687	65.757
5% supervised points			
Flexible size constraints (7)	89.024	89.022	88.848
Penalty term (8)	89.025	89.018	88.987
Exact size constraints (5)	86.327	60.904	51.276

The exact class sizes are perturbed by a random number within *p* % of the size and the accuracies are computed by averaging over multiple runs. See Sect. 5.1.6 for details

Table 4 Timing results (in seconds)

	MBO [38]	GL [38]	Proposed
MNIST	15.4	153.1	42.5
3 moons	3.7	3.9	2.7
COIL	1.18	1.19	1.4
Satellite	16.4	23	16.5

Bold values indicate the results of the new algorithm

Table 5 Initial and final energy

	Initial energy	Final energy (MBO) [38]	Final energy proposed
MNIST	225,654	15,196	12,324
3 moons	5982.79	433.19	420.24
COIL	1774.3	24.61	24.18
Satellite	5116.9	221.87	214.95

Bold values indicate the results of the new algorithm

Landau are not able to converge to the global minimum. Table 1 shows that the accuracies of the proposed algorithm are also highly competitive against a wide range of other established algorithms, even when substantially less training data than those algorithms are being used. Table 3 shows that that the flexible size constraints (7) and penalty term (8) can improve the accuracy, if a rough estimate of the approximate class sizes are given.

The binary difference (63), plotted on log-scale against the iteration count, is depicted for each experiment in Fig. 7. For experiments without any size information, the average binary difference tends to less than 10^{-16} , which is vanishingly low and more or less indicates that an exact global minimizer has been obtained. For experiments with size constraints or penalty terms, the binary difference also gets very low, although not as low. This indicates convergence to at least a very close approximation of the global minimizer. These observations agree well with the theoretical results in Sect. 3.4, where the strongest results were also obtained in case of no size information.

Note that a lot more iterations than necessary have been used in the binary difference plots. In practice, the algorithm reaches sufficient stability in 100–300 iterations. The CPU times, summarized in Table 4, indicate a fast convergence of the new algorithm, much faster than GL, although not quite as fast as the MBO scheme. It must be noted that MBO is an extremely fast front propagation algorithm that only uses a few (e.g., 10) iterations, but its accuracy is limited due to the large step sizes. A deeper discussion on the number of iterations needed to reach the exact solution after thresholding will be given at the end of the next section on point cloud segmentation.

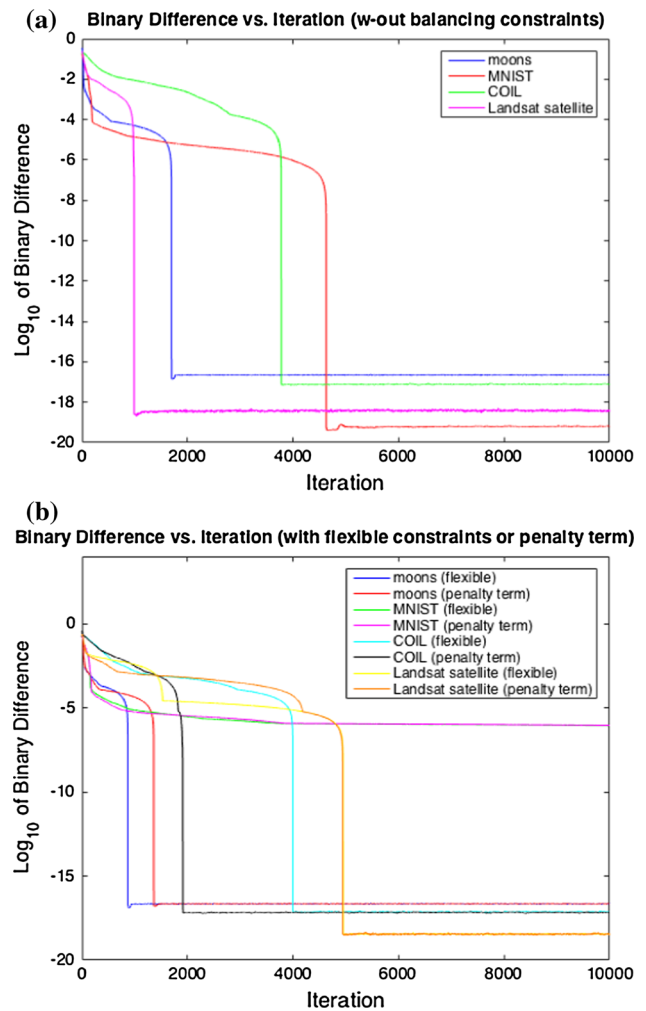


Fig. 7 \log_{10} plot of binary difference $b(u^k)$ versus iteration count. **a** Algorithm without size constraints; **b** algorithm with flexible constraints (7) and penalty term (8) acting on class sizes

5.2 Segmentation of 3D Point Clouds

The energy function (9) that combines region homogeneity terms and dissimilarity across region boundaries will be demonstrated for segmentation of unstructured 3D point clouds, where each point is a vertex in V . Point clouds arise for instance through laser-based range imaging or multiple view scene reconstruction. The results of point cloud segmentation are easy to visualize and the choice of each term in the energy function will have a clear intuitive meaning that may be translated to other graph-based classification problems in the future. We focus especially on point clouds acquired through the concept of laser detection and ranging (LaDAR) in outdoor scenarios. A fundamental computer vision task is to segment such scenes into classes of similar objects. Roughly, some of the most common object classes in an outdoor scene are the ground plane, veg-

etation and human-made ‘objects’ with a certain regular structure.

5.2.1 Construction of the Energy Function

We construct the graph by connecting each node to its k -nearest neighbors (kNN) based on the Euclidian distance as described at the beginning of Sect. 2. In experiments, we set $k = 20$. We construct region terms that favor homogeneity of geometrical features based on a combination of point coordinates, normal vectors and variation of normal vectors. The construction is a concrete realization of the general region terms introduced in [59,60,84]. We also propose to use a contour term that favors alignment of the boundaries of the regions at ‘edges,’ indicated by sharp discontinuities of the normal vectors. Our model can be seen as a point cloud analogue of variational models for traditional image segmentation that combine region- and edge-based features in a single energy functional [15,39,48]. In contrast to the work [2,72,87] our model does not rely on training data.

Normal vectors in a point cloud can be estimated from principal component analysis locally around each point, as in, e.g., [31,55,60]. For each point $x \in V$, let y^1, \dots, y^m denote the set of neighboring points and define for notational convenience $y^0 = x$. Define the normalized vectors $\bar{y}^i = y^i - \text{mean}(y^0, y^1, \dots, y^m)$ for $i = 0, 1, \dots, m$ and construct the matrix

$$Y = [\bar{y}^0 \bar{y}^1 \bar{y}^2 \dots \bar{y}^m]. \tag{65}$$

Let $\mathbf{v}^1(x), \mathbf{v}^2(x), \mathbf{v}^3(x)$ be the eigenvectors and $\lambda^1(x), \lambda^2(x), \lambda^3(x)$ be the eigenvalues of the correlation matrix YY^T . The first eigenvector $\mathbf{v}^1(x)$ points in the direction of least variation between the points $\bar{y}^1, \dots, \bar{y}^m$ and the first eigenvalue $\lambda^1(x)$ indicates the variation along the direction of $\mathbf{v}^1(x)$.

The variable $\mathbf{v}^1(x)$ is consequently a discrete estimation of the normal vector at x and the first eigenvalue $\lambda^1(x)$ indicates to which extent the normal vectors vary locally around the point x . If all the points were laying on a plane, then $\lambda^1(x)$ would be zero and $\mathbf{v}^1(x)$ would be the normal vector of the plane.

The region term for region V_i can be constructed to be small at the point x if the value of $\lambda^1(x)$ is close to the expected value λ_i of region i , and be large otherwise. This can be achieved by requiring the following term to be small

$$|\lambda^1(x) - \lambda_i|^2, \quad \forall x \in V, \quad i = \{v, h, g\}. \tag{66}$$

For instance, λ_v^1, λ_h^1 and λ_g^1 for vegetation, human-made objects and the ground plane can be estimated from measurements. Note that their particular values depend on characteristics of the LaDAR, such as the angular scanning

resolution and depth resolution. If an estimate of λ_i is not known, λ_i could be part of the minimization problem in a similar way to the mean intensity values in the Chan–Vese model [23].

Furthermore, the region terms can be constructed for discriminating regions where the normal vector are oriented either parallel with or perpendicular to a specific direction \mathbf{n}^i by requiring the following terms to be small, respectively,

$$-|\mathbf{v}^1(x) \cdot \mathbf{n}^i|, \quad |\mathbf{v}^1(x) \cdot \mathbf{n}^i|.$$

For instance, the normal vectors of the ground plane are expected to point predominantly in the upward direction. The ground plane can also be characterized by its height, defined by the x_3 -coordinate of the points, which is generally lower than the heights of other objects in the nearby surroundings. Assuming a rough estimate of the local height of the ground plane $h^*(x)$ at the point x is known, the fidelity term (9) can be modified to take into account both normals vectors and height by requiring the following term to be small

$$-|\mathbf{v}^1(x) \cdot \mathbf{n}^i(x)| + H(x_3, h^*(x)), \tag{67}$$

where H is an increasing function in x_3 and, in addition, $H(h^*(x), h^*(x)) = 0$. We have used the term $H(x_3, h^*(x)) = \theta(x_3 - h^*(x))$ and simply estimated $h^*(x)$ as the average x_3 coordinate of the points in the neighborhood of x .

The weight function w is constructed to encourage spatial grouping of the points, and so that it is favorable to align the border between regions at locations where the normal vectors change from pointing upwards to pointing outwards, i.e., where the scene is convex-shaped. On the contrary, locations where the scene is concave, such as the transition from the side of buildings to the roof, should be unfavorable for the region boundaries. Such assumptions can be incorporated by modifying the Gaussian weight function (1) as follows:

$$w(x, y) = e^{-\frac{d(x,y)^2}{\sigma^2} + \gamma \frac{v_1^1(y) - v_3^1(x)}{d(x,y)} \text{SIGN}(y_1 - x_1)} \tag{68}$$

Here $v_1^1(x)$ and $v_3^1(x)$ are the first and third components of the vector $\mathbf{v}^1(x)$, and a coordinate system has been assumed where the positive x_1 axis points outwards from the view direction and the positive x_3 axis points upwards. An illustration is given in Fig. 8, where edges at convex parts of the scene are given a low energy value, marked by the color code of light blue.

Taking the above information into account, an example of how the different region terms can be constructed for the ground plane, human-made structures and vegetation, respectively, is

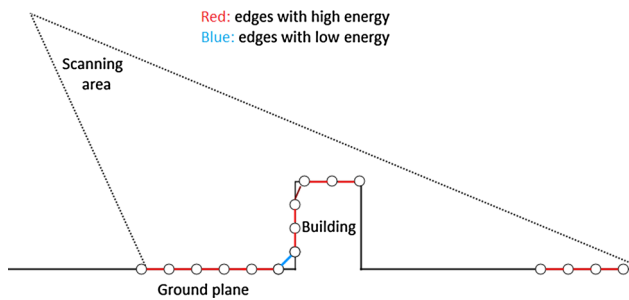


Fig. 8 Illustration of construction of a graph to separate the ground plane from human-made structures, view point from the side. The edges are assigned a low energy at convex parts of the scene, marked in *light blue*, making it favorable to place the boundary between the regions at such locations (Color figure online)

$$f_g(x) = (1 - C)|\lambda^1(x) - \lambda_g|^2 + C\{-|\mathbf{v}^1(x) \cdot \mathbf{n}^g(x)| + H(x_2, h^*(x))\}. \quad (69)$$

$$f_h(x) = (1 - C)|\lambda^1(x) - \lambda_h|^2 + C|\mathbf{v}^1(x) \cdot \mathbf{n}^g(x)|, \quad (70)$$

$$f_v(x) = C|\lambda^1(x) - \lambda_v|^2. \quad (71)$$

Here, $C \in (0, 1)$ is a parameter that balances considerations between variation and direction/height. In experiments, we set $\lambda_g = \lambda_h$ and set C to a low value so that only vegetation is distinguished from other regions by the value of λ_1 . In some experiments, we also use two regions for vegetation with two different values of λ_i . This makes it possible to distinguish different kinds of vegetation, those with leaves or needles tend to have a lower mean value λ_i than those without them. Smoke can also be characterized by its irregular surface, and its region term constructed as (66) with an intermediate value of λ_i .

5.2.2 Experiments

Some illustrative experiments are shown in Figs. 9 and 10. Ordinary photographs of the scenes are shown on the top and the red rectangles indicate the areas that have been scanned by the LaDAR. The point clouds have been segmented into three regions as described above and the results are visualized by brown color for points assigned to the ground plane region, green color for points assigned to the vegetation region and blue color for points assigned to the region of human-made objects. In Fig. 11, vegetation with and without leaves is indicated by dark and light green, respectively.

It can be observed that the algorithm leads to consistent results even though these scenes are particularly challenging because the tilt and height of the ground plane vary highly over the scene due to the hilly landscape, and some of the trees and bushes are completely aligned with and touches the

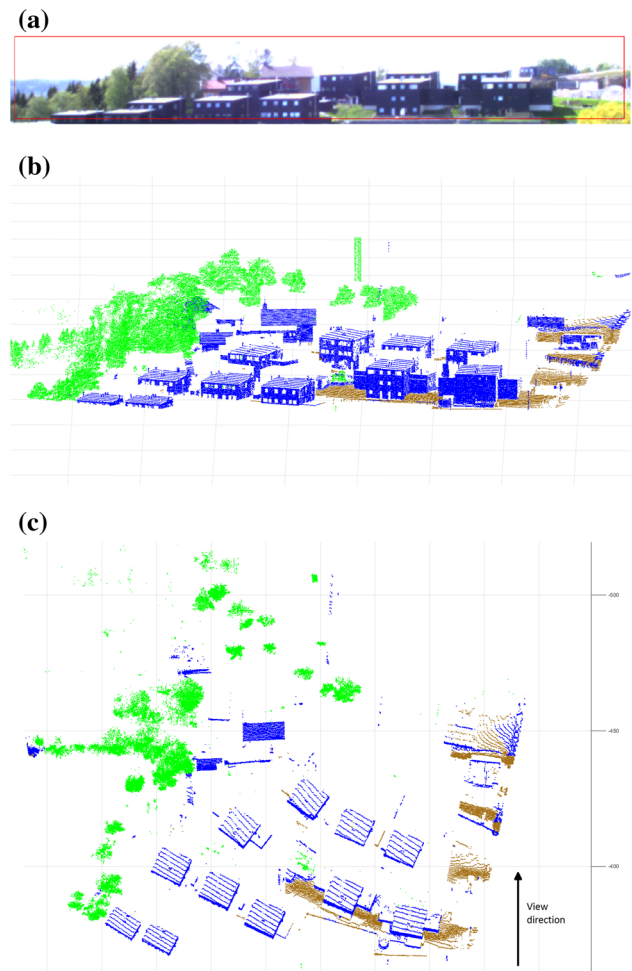


Fig. 9 **a** Scanning area of LaDAR. **b, c** Segmentation of acquired point cloud, consisting of 93,641 points, into three regions: ground plane (*brown*), vegetation (*green*) and human-made objects (*blue*). **a** Scanning area; **b** segmentation, view from front; **c** segmentation, view from top (Color figure online)

buildings. Note that buildings hidden behind vegetation get detected since the laser pulses are able to partially penetrate through the leaves. A misassignment can be observed in the middle of Fig. 10, where only the roof of one of the buildings is visible due to occlusions. Since no points are observed from the wall of the building, the roof gets assigned to the ground plane region. Some large rocks on Fig. 10 also get assigned to the blue region due to their steep and smooth surfaces (Fig. 12).

As was the case for experiments involving semi-supervised classification, the approximation errors of the convex relaxation practically vanish. Figure 13c depicts the binary difference (63) as a function of the iteration count in the experiment shown in Fig. 10. As can be seen, the solution of the convex relaxation converges to a binary function; after 10,000 iterations, the average binary difference (63) was 5.74×10^{-10} . Note, however, that a lot less iterations are necessary before

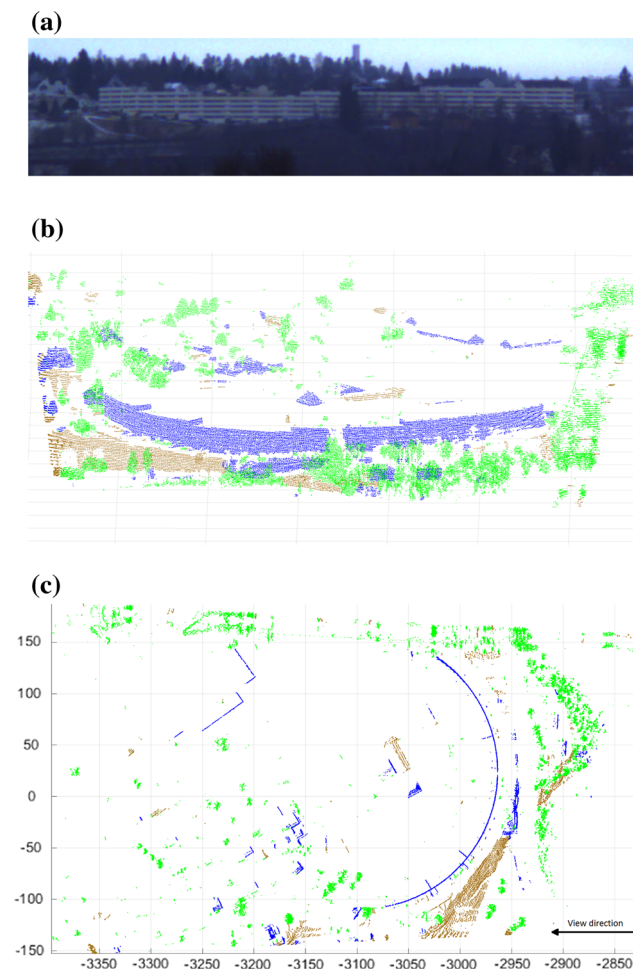


Fig. 10 **a** Scanning area of LaDAR. **b, c** Segmentation of acquired point cloud, consisting of 80,937 points, into three regions: ground plane (*brown*), vegetation (*green*) and human-made objects (*blue*). **a** Scanning area; **b** segmentation, view from front; **c** segmentation, view from top (Color figure online)

the thresholded function stabilizes at the global minimum. Figure 13 (left) depicts the energy evolution as a function of the iteration count for the relaxed solution (blue), thresholded solution with scheme (26) (red) and with scheme (40) (green). Figure 13 (right) depicts a log plot of the absolute energy precision $\frac{\|E^i - E_{relaxed}^*\|}{E_{relaxed}^*}$, where $E_{relaxed}^*$ is the global minimum of the relaxed problem, estimated by 10,000 iterations of the algorithm. E^i is the energy at iteration i of the relaxed solution (blue), thresholded solution with scheme (26) (red) and thresholded solution with scheme (40) (green). This plot demonstrates that the binary solution obtained by the thresholding scheme (26) stabilizes after about 300 iterations, after which the energy is within 10^{-16} of the energy of the ground truth solution of the relaxed problem estimated at iteration 10,000. The threshold scheme (40) takes more iterations before stabilizing, but also eventually converges to the correct solution after about 3500 iterations. The CPU

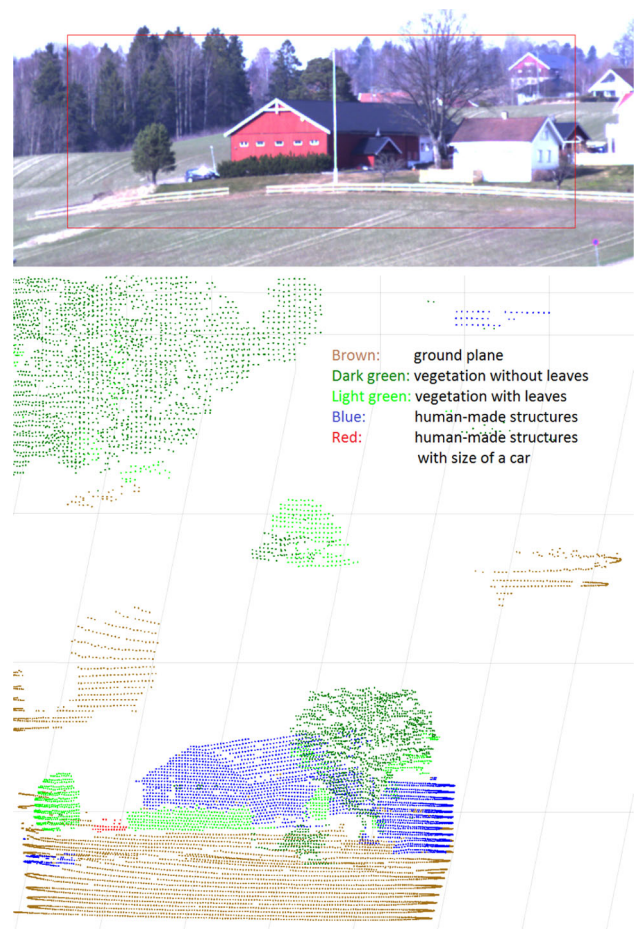


Fig. 11 *Top*: Scanning area of LaDAR. *Bottom*: Segmentation of acquired point cloud, consisting of 14,002 points, into four regions: ground plane (*brown*), and human-made objects (*blue*), vegetation with (*light green*) and without (*dark green*) leaves/needles (Color figure online)

times of the experiments were in the range 5–15 s on an Intel i5-4570 3.2 Ghz CPU for point clouds with around 80,000 points. For comparison, the inference step of the related MRF approaches [2, 72, 87] took around 9 minutes for a scan with around 30,000 points, as reported in [72], but of course on older hardware. The proposed algorithm is also suitable for parallel implementation on GPU as discussed at the end of Sect. 4.2.

6 Conclusions

Variational models on graphs have shown to be highly competitive for various data classification problems, but are inherently difficult to handle from an optimization perspective, due to NP-hardness except in some restricted special cases. This work has developed an efficient convex algorithmic framework for a set of classification problems

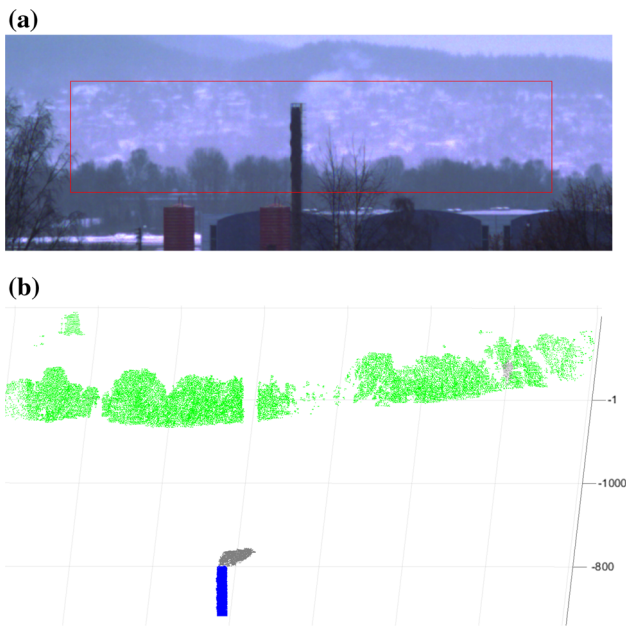


Fig. 12 *Top*: Scanning area of LaDAR. *Bottom*: Segmentation of a point cloud (81,551 points) into smoke (gray), vegetation (green) and human-made structures (blue). **a** Scanning area; **b** segmentation result (Color figure online)

with multiple classes involving graph total variation, region homogeneity terms, supervised information and certain constraints or penalty terms acting on the class sizes. Particular problems that could be handled as special cases included semi-supervised classification of high-dimensional data and unsupervised segmentation of unstructured 3D point clouds. The latter involved minimization of a novel energy function enforcing homogeneity of point coordinate-based features within each region, together with a term aligning the region boundaries along edges. Theoretical and experimental analysis revealed that the convex algorithms were able to produce vanishingly close approximations to the global minimizers of the original problems in practice.

Experiments on benchmark data sets for semi-supervised classification resulted in higher accuracies of the new algorithm compared to related local minimization approaches. The accuracies were also highly competitive against a wide range of other established algorithms. The advantages of the proposed algorithm were particularly prominent in case of sparse or non-uniformly distributed training data. The accuracies could be improved further if an estimate of the approximate class sizes were given in advance. Experiments also demonstrated that 3D point clouds acquired by a LaDAR in outdoor scenes could be segmented into object classes with a high degree of accuracy, purely based on the geometry of the points and without relying on training data. The computational efficiency was at least an order of magnitude faster than related work reported in the literature.

In the future, it would be interesting to investigate region homogeneity terms for general unsupervised classification problems. In addition to avoiding the problem of trivial global minimizers, the region terms may improve the accuracy compared to models based primarily on boundary terms. Region homogeneity may, for instance, be defined in terms of the eigendecomposition of the covariance matrix or graph Laplacian.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

Appendix: Proof of Theorem 3

To aid the proof of Theorem 3, we first give the following lemma, which is a graph extension of Proposition 4 given in [6] for image domains.

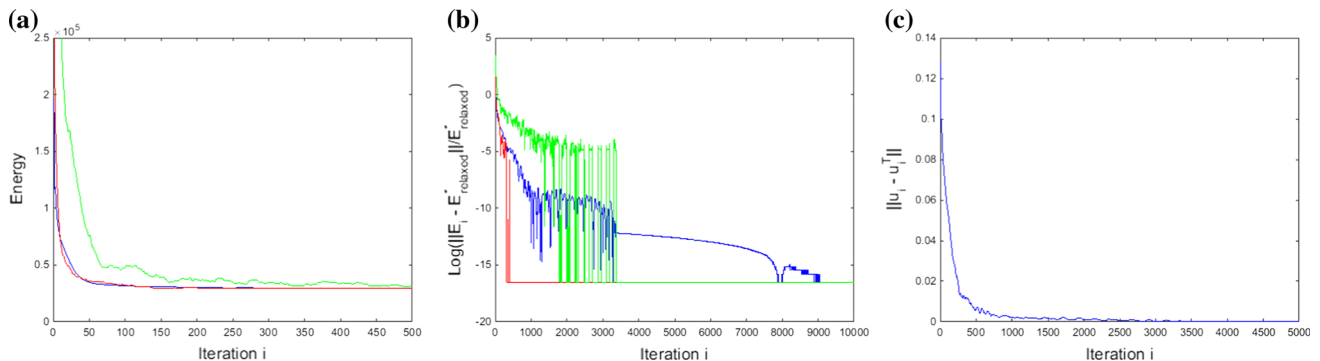


Fig. 13 **a**, **b** Energy evolution of u (blue), u^T with threshold scheme (26) (red), and u^T with threshold scheme (40) (green) for the experiment in Fig. 10. **a** Primal energies; **b** Log of $\frac{\|E^i - E^*_{relaxed}\|}{E^*_{relaxed}}$; **c** Binary difference (63) (Color figure online)

Lemma 1 Assume that for a function $u : V \mapsto [0, 1]$, q^* maximizes

$$q^* = \arg \max_{\|q\|_{\mathcal{E}, \infty} \leq 1} \sum_{x \in V} u(x)(\operatorname{div}_w q)(x)$$

Define the thresholded function

$$u^\alpha(x) = \begin{cases} 1 & \text{if } u(x) \geq \alpha \\ 0 & \text{otherwise} \end{cases} \quad (72)$$

For almost any threshold level $\alpha \in (0, 1]$, q^* also maximizes

$$q^* = \arg \max_{\|q\|_{\mathcal{E}, \infty} \leq 1} \sum_{x \in V} u^\alpha(x)(\operatorname{div}_w q)(x)$$

Proof The coarea formula on graphs says that

$$\sum_{x \in V} |\nabla_w u(x)| = \int_0^1 \sum_{x \in V} |\nabla_w u^\alpha(x)| \, d\alpha;$$

see, for instance, appendix B of [90] for a proof. Together with the fact that $u(x) = \int_0^{u(x)} d\alpha = \int_0^1 u^\alpha(x) d\alpha$, we can deduce that

$$\begin{aligned} & \int_0^1 \sum_{x \in V} u^\alpha(x)(\operatorname{div}_w q^*)(x) \, d\alpha \\ &= \sum_{x \in V} \left(\int_0^1 u^\alpha(x) \, d\alpha \right) (\operatorname{div}_w q^*)(x) \\ &= \sum_{x \in V} u(x)(\operatorname{div}_w q^*)(x) \\ &= \sum_{x \in V} |\nabla_w u(x)| = \int_0^1 \sum_{x \in V} |\nabla_w u^\alpha(x)| \, d\alpha \\ &= \int_0^1 \left(\sup_{\|q\|_{\mathcal{E}, \infty} \leq 1} \sum_{x \in V} u^\alpha(x)(\operatorname{div}_w q)(x) \, d\alpha \right). \end{aligned}$$

Since in general

$$\begin{aligned} & \sup_{\|q\|_{\mathcal{E}, \infty} \leq 1} \sum_{x \in V} u^\alpha(x)(\operatorname{div}_w q)(x) \, d\alpha \\ & \geq \sum_{x \in V} u^\alpha(x)(\operatorname{div}_w q^*)(x) \, d\alpha, \end{aligned}$$

the above equality can only be true provided that

$$\begin{aligned} & \sup_{\|q\|_{\mathcal{E}, \infty} \leq 1} \sum_{x \in V} u^\alpha(x)(\operatorname{div}_w q)(x) \\ & = \sum_{x \in V} u^\alpha(x)(\operatorname{div}_w q^*)(x), \end{aligned}$$

for almost every $\alpha \in (0, 1]$. □

Utilizing Lemma 1, we will now prove Theorem 3:

Proof By the assumptions of the theorem, for a finite number of connected components in the graph, the minimizer $I_{\min}(x)$ contains two indices. Assume without loss of generality that $V_{k,j} \subset V$ is one such connected component where $I_m(x) = k, j$ for all $x \in V_{k,j}$. That is, for any two nodes x and y in $V_{k,j}$, there is a path of edges $(x, z_1), (z_1, z_2), \dots, (z_n, y) \subset E$ such that $z_1, \dots, z_n \in V_{k,j}$.

Let u^* be any primal solution for which $(u^*; q^*)$ is a primal–dual pair. By Theorem 2, u^* must in $V_{k,j}$ satisfy

$$u_k^*(x) + u_j^*(x) = 1, \quad u_i^*(x) = 0, \quad \text{for } i \neq k, j, \quad (73)$$

For an arbitrary threshold level $\alpha \in (0, 1)$ construct the binary function

$$u_k^\alpha(x) = \begin{cases} 1 & \text{if } u_k^*(x) \geq \alpha \\ 0 & \text{otherwise} \end{cases}. \quad (74)$$

From (73), we can write $u_j^*(x) = 1 - u_k^*(x)$ in $V_{k,j}$, and together with (74) it follows that $1 - u_k^\alpha(x) = u_j^{1-\alpha}(x)$ in $V_{k,j}$.

Construct now the function $u^t : V \mapsto \mathbb{R}^n$ as follows:

$$u^t(x) = u^*(x) \text{ for } x \in V \setminus V_{k,j} \quad (75)$$

$$u_i^t(x) = \begin{cases} u_k^\alpha(x) & \text{if } i = k \\ u_j^{1-\alpha}(x) & \text{if } i = j \\ 0 & \text{if } i \neq k, j \end{cases} \text{ for } x \in V_{k,j} \quad (76)$$

For the given q^* , we have that

$$\begin{aligned} E(u^*, q^*) &= \sum_{i \in I} \sum_{x \in V} u_i^*(x) \{C_i(x) + (\operatorname{div}_w q_i^*)(x)\} \\ &= \sum_{i \in I \setminus \{k, j\}} \sum_{x \in V \setminus V_{k,j}} u_i^*(x) \{C_i(x) + (\operatorname{div}_w q_i^*)(x)\} \\ &\quad + \sum_{x \in V_{k,j}} u_k^*(x) \{C_k(x) + (\operatorname{div}_w q_k^*)(x)\} \\ &\quad + \sum_{x \in V_{k,j}} u_j^*(x) \{C_j(x) + (\operatorname{div}_w q_j^*)(x)\} \\ &= \sum_{i \in I \setminus \{k, j\}} \sum_{x \in V \setminus V_{k,j}} u_i^*(x) \{C_i(x) + (\operatorname{div}_w q_i^*)(x)\} \\ &\quad + \sum_{x \in V_{k,j}} (u_k^*(x) + (1 - u_k^*(x))) \{C_k(x) + (\operatorname{div}_w q_k^*)(x)\} \\ &= \sum_{i \in I \setminus \{k, j\}} \sum_{x \in V \setminus V_{k,j}} u_i^*(x) \{C_i(x) + (\operatorname{div}_w q_i^*)(x)\} \\ &\quad + \sum_{x \in V_{k,j}} (u_k^\alpha(x) + (1 - u_k^\alpha(x))) \{C_k(x) + (\operatorname{div}_w q_k^*)(x)\} \end{aligned}$$

$$\begin{aligned}
 &= \sum_{i \in I \setminus \{k, j\}} \sum_{x \in V \setminus V_{k, j}} u_i^*(x) \{C_i(x) + (\operatorname{div}_w q_i^*)(x)\} \\
 &+ \sum_{x \in V_{k, j}} u_k^\alpha(x) \{C_k(x) + (\operatorname{div}_w q_k^*)(x)\}, \\
 &+ \sum_{x \in V_{k, j}} u_j^{1-\alpha}(x) \{C_j(x) + (\operatorname{div}_w q_j^*)(x)\}, \\
 &= E(u^t, q^*) \tag{77}
 \end{aligned}$$

where we have used that $C_k(x) + (\operatorname{div}_w q_k^*)(x) = C_j(x) + (\operatorname{div}_w q_j^*)(x)$ in $V_{k, j}$. By applying Lemma 1 on the last two terms with threshold level α and $1 - \alpha$, respectively, it can be deduced that

$$\sup_{q \in S_\infty^n} E(u^t, q) = \sup_{q \in S_\infty^n} E(u^*, q) = E(u^*, q^*) = E(u^t, q^*)$$

Consequently (u^t, q^*) is an optimal primal–dual pair.

Assume now there is another connected component $V_{k_2, j_2}^2 \subset V$ where $I_{\min} = \{k_2, j_2\}$. By setting $u^* = u^t$ and repeating all arguments above, it follows that u^* can be thresholded in V_{k_2, j_2}^2 to yield a binary minimizer in V_{k_2, j_2}^2 . The same process can be repeated for all connected components until a binary minimizer is obtained over the whole domain V . By Proposition 1, such a binary function is a global minimizer of the original non-convex problem. \square

References

1. Altman, N.S.: An introduction to kernel and nearest-neighbor non-parametric regression. *Am. Stat.* **46**, 175–185 (1992)
2. Anguelov, D., Taskar, B., Chatalbashev, V., Koller, D., Gupta, D., Heitz, G., Ng, A. Y.: Discriminative learning of Markov random fields for segmentation of 3d scan data. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Diego, CA, USA, pp. 169–176 (2005)
3. Aujol, J.-F., Gilboa, G., Papadakis, N.: Fundamentals of non-local total variation spectral theory. In: *Proceedings Scale Space and Variational Methods in Computer Vision*, pp. 66–77 (2015)
4. Bache, K., Lichman, M.: UCI machine learning repository (2013)
5. Bae, E., Tai, X.-C., Yuan, J.: Maximizing flows with message-passing: Computing spatially continuous min-cuts. In: *Energy Minimization Methods in Computer Vision and Pattern Recognition—10th International Conference*, pp. 15–28 (2014)
6. Bae, E., Yuan, J., Tai, X.-C.: Global minimization for continuous multiphase partitioning problems using a dual approach. *Int. J. Comput. Vis.* **92**(1), 112–129 (2011)
7. Bae, E., Tai, X.-C.: Efficient global minimization methods for image segmentation models with four regions. *J. Math. Imaging Vis.* **51**(1), 71–97 (2015)
8. Belkin, M., Niyogi, P., Sindhwani, V.: Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. *J. Mach. Learn. Res.* **7**, 2399–2434 (2006)
9. Bentley, J.L.: Multidimensional binary search trees used for associative searching. *Commun. ACM* **18**, 509–517 (1975)
10. Bertozzi, A.L., Flenner, A.: Diffuse interface models on graphs for classification of high dimensional data. *Multiscale Model. Simul.* **10**(3), 1090–1118 (2012)

11. Boykov, Y., Veksler, O., Zabih, R.: Markov random fields with efficient approximations. In: *1998 Conference on Computer Vision and Pattern Recognition (CVPR '98)*, 23–25 June 1998, Santa Barbara, pp. 648–655 (1998)
12. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.* **23**(11), 1222–1239 (2001)
13. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Mach. Intell.* **26**, 359–374 (2001)
14. Bresson, X., Laurent, T., Uminsky, D., von Brecht, J., Multiclass total variation clustering. In: *Advances in Neural Information Processing Systems*, pp. 1421–1429 (2013)
15. Bresson, X., Esedoglu, S., Vandergheynst, P., Thiran, J.P., Osher, S.: Fast global minimization of the active contour/snake model. *J. Math. Imaging Vis.* **28**(2), 151–167 (2007)
16. Bresson, X., Laurent, T., Uminsky, D., von Brecht, J.H.: Convergence and energy landscape for Cheeger cut clustering. *Adv. Neural Inf. Process. Syst.* **25**, 1394–1402 (2012)
17. Bresson, X., Tai, X.-C., Chan, T.F., Szlam, A.: Multi-class transductive learning based on ℓ_1 relaxations of cheeger cut and mumford-shah-potts model. *J. Math. Imaging Vis.* **49**(1), 191–201 (2014)
18. Brown, E.S., Chan, T.F., Bresson, X.: Completely convex formulation of the Chan-Vese image segmentation model. *Int. J. Comput. Vis.* **98**, 103–121 (2011). doi:10.1007/s11263-011-0499-y
19. Bühler, T., Hein, M.: Spectral clustering based on the graph p-Laplacian. In: *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 81–88. ACM (2009)
20. Chambolle, A., Cremers, D., Pock, T.: A convex approach to minimal partitions. *SIAM J. Imaging Sci.* **5**(4), 1113–1158 (2012)
21. Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vis.* **40**(1), 120–145 (2011)
22. Chan, T.F., Esedoglu, S., Nikolova, M.: Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM J. Appl. Math.* **66**(5), 1632–1648 (2006)
23. Chan, T., Vese, L.A.: Active contours without edges. *IEEE Imag. Proc.* **10**, 266–277 (2001)
24. Chan, T.F., Zhang, X.: Wavelet inpainting by nonlocal total variation. *Inverse Probl. Imaging* **4**(1), 191–210 (2010)
25. Chapelle, O., Schölkopf, B., Zien, A.: *Semi-supervised Learning*, vol. 2. MIT Press, Cambridge (2006)
26. Cheeger, J.: *A Lower Bound for the Smallest Eigenvalue of the Laplacian*. Princeton University Press, Princeton (1970)
27. Cireşan, D.C., Meier, U., Masci, J., Gambardella, L.M., Schmidhuber, J.: Flexible, high performance convolutional neural networks for image classification. In: *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pp. 1237–1242 (2011)
28. Dahlhaus, E., Johnson, D. S., Papadimitriou, C. H., Seymour, P. D., Yannakakis, M.: The complexity of multiway cuts (extended abstract). In *STOC '92: Proceedings of the 24th annual ACM symposium on Theory of computing*, pp. 241–251, New York, NY, USA, 1992. ACM (1992)
29. Decoste, D., Schölkopf, B.: Training invariant support vector machines. *Mach. Learn.* **46**(1), 161–190 (2002)
30. Desquesnes, X., Elmoataz, A., Lezoray, O.: Eikonal equation adaptation on weighted graphs: fast geometric diffusion process for local and non-local image and data processing. *J. Math. Imaging Vis.* **46**, 238–257 (2013)
31. Digne, J.: Similarity based filtering of point clouds. In: *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, Providence, RI, USA, June 16–21, 2012, pp. 73–79 (2012)

32. Ekeland, I., Témam, R.: *Convex analysis and variational problems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (1999)
33. Elmoataz, A., Lezoray, O., Bougleux, S.: Nonlocal discrete regularization on weighted graphs: a framework for image and manifold processing. *IEEE Trans. Image Process.* **17**, 1047–1060 (2008)
34. Elmoataz, A., Touttatin, M., Tenbrinck, D.: On the p-laplacian and infinity-laplacian on graphs with application in image and data processing. *SIAM J. Imaging Sci.* **8**, 2412–2451 (2015)
35. Esser, J. E.: *Primal dual algorithms for convex models and applications to image restoration, registration and nonlocal inpainting*. (Ph.D. thesis, UCLA CAM-report 10-31), April (2010)
36. Fowlkes, C., Belongie, S., Chung, F., Malik, J.: Spectral grouping using the Nyström method. *IEEE Trans. Pattern Anal. Mach. Intell.* **26**(2), 214–225 (2004)
37. Garcia-Cardona, C., Flenner, A., Percus, A.G.: Multiclass diffuse interface models for semi-supervised learning on graphs. In: *Proceedings of the 2th International Conference on Pattern Recognition Applications and Methods*. SciTePress (2013)
38. Garcia-Cardona, C., Merkurjev, E., Bertozzi, A.L., Flenner, A., Percus, A.G.: Multiclass data segmentation using diffuse interface methods on graphs. *IEEE Trans. Pattern Anal. Mach. Intell.* **36**(8), 1600–1613 (2014)
39. Gilboa, G., Osher, S.: Nonlocal linear image regularization and supervised segmentation. *SIAM Multiscale Model. Simul. (MMS)* **6**(2), 595–630 (2007)
40. Goldstein, T., Bresson, X., Osher, S.: Global minimization of markov random fields with applications to optical flow. *UCLA cam-report 09-77* (2009)
41. Golovinskiy, A., Kim, V. G., Funkhouser, T.: Shape-based recognition of 3D point clouds in urban environments. In: *International Conference on Computer Vision (ICCV)*, pp. 2154–2161 (2009)
42. Hein, M., Audibert, J., Von Luxburg, U.: From graphs to manifolds—weak and strong pointwise consistency of graph laplacians. In: *Proceedings of the 18th Conference on Learning Theory (COLT)*, pp. 470–485. Springer, New York (2005)
43. Hein, M., Setzer, S.: Beyond spectral clustering—tight relaxations of balanced graph cuts. In: J. Shawe-Taylor, R.S. Zemel, P. Bartlett, F.C.N. Pereira, and K.Q. Weinberger, (eds.) *Advances in Neural Information Processing Systems 24*, pp. 2366–2374 (2011)
44. Hein, M., Bühler, T.: An inverse power method for nonlinear eigenproblems with applications in 1-spectral clustering and sparse PCA. *Adv. Neural Inf. Process. Syst.* **23**, 847–855 (2010)
45. Hidane, M., Lezoray, O., Elmoataz, A.: Nonlinear multilayered representation of graph-signals. *J. Math. Imaging Vis.* **45**, 114–137 (2013)
46. Indyk, P.: Chapter 39 : Nearest neighbours in high-dimensional spaces. In: *Handbook of Discrete and Computational Geometry (2nd ed.)*, pp. 1–16. CRC Press, Boca Raton (2004)
47. Joachims et al, T.: Transductive learning via spectral graph partitioning. In: *International Conference on Machine Learning*, 20, pp. 290 (2003)
48. Jung, M., Peyré, G., Cohen, L.D.: Nonlocal active contours. *SIAM J. Imaging Sci.* **5**(3), 1022–1054 (2012)
49. Kégl, B., Busa-Fekete, R.: Boosting products of base classifiers. In: *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 497–504 (2009)
50. Kimmel, R., Caselles, V., Sapiro, G.: Geodesic active contours. *Int. J. Comput. Vision* **22**, 61–79 (1997)
51. Kleinberg, J.M., Tardos, É.: Approximation algorithms for classification problems with pairwise relationships: metric labeling and markov random fields. *J. ACM* **49**(5), 616–639 (2002)
52. Klodt, M., Cremers, D.: A convex framework for image segmentation with moment constraints. In: *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6–13, 2011*, pp. 2236–2243 (2011)
53. Kolmogorov, V., Zabih, R.: What energy functions can be minimized via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.* **26**, 65–81 (2004)
54. Komodakis, N., Tziritis, G.: Approximate labeling via graph-cuts based on linear programming. *Pattern Anal. Mach. Intell.* **29**(8), 1436–1453 (2007)
55. Lai, R., Liang, J., Zhao, H.K.: A local mesh method for solving pdes on point clouds. *Inverse Probl. Imaging* **7**, 737–755 (2013)
56. LeCun, Y., Cortes, C.: The MNIST database of handwritten digits. <http://yann.lecun.com/exdb/mnist/>
57. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. *Proc. IEEE* **86**(11), 2278–2324 (1998)
58. Lellmann, J., Kappes, J., Yuan, J., Becker, F., Schnörr, C., Convex multi-class image labeling by simplex-constrained total variation. In: X.-C. Tai., K. Mörken., M. Lysaker., K.-A. Lie., (eds) *Scale Space and Variational Methods in Computer Vision (SSVM 2009)*, vol. 5567 of LNCS, pp. 150–162. Springer, New York (2009)
59. Lezoray, O., Elmoataz, A., Ta, V.-T.: Nonlocal pdes on graphs for active contours models with applications to image segmentation and data clustering. In: *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 873–876. IEEE (2012)
60. Lezoray, O., Lozes, F., Elmoataz, A.: Partial difference operators on weighted graphs for image processing on surfaces and point clouds. *IEEE Trans. Imag. Process.* **23**, 3896–3909 (2014)
61. Li, F., Ng, M.K., Zeng, T., Shen, C.: A multiphase image segmentation method based on fuzzy region competition. *SIAM J. Imaging Sci.* **3**(3), 277–299 (2010)
62. Lozes, F., Elmoataz, A., Lezoray, O.: Nonlocal processing of 3d colored point clouds. In: *Proceedings of the 21st International Conference on Pattern Recognition, ICPR 2012, Tsukuba, Japan, November 11–15, 2012*, pp. 1968–1971 (2012)
63. Macdonald, C.B., Merriman, B., Ruuth, S.J.: Simple computation of reaction-diffusion processes on point clouds. *Proc. Natl. Acad. Sci.* **110**, 3009–3012 (2013)
64. Martello, S., Toth, P.: *Knapsack Problems: Algorithms and Computer Implementations*. Wiley, New York (1990)
65. Merkurjev, E., Sunu, J., Bertozzi, A.L.: Graph MBO method for multiclass segmentation of hyperspectral stand-off detection video. In: *2014 IEEE International Conference on Image Processing (ICIP)*, pp. 689–693. IEEE (2014)
66. Merkurjev, E., Kostic, T., Bertozzi, A.L.: An MBO scheme on graphs for classification and image processing. *SIAM J. Imaging Sci.* **6**(4), 1903–1930 (2013)
67. Merkurjev, E., Garcia-Cardona, C., Bertozzi, A.L., Flenner, A., Percus, A.G.: Diffuse interface methods for multiclass segmentation of high-dimensional data. *Appl. Math. Lett.* **33**, 29–34 (2014)
68. Merkurjev, E., Bae, E., Bertozzi, A.L., Tai, X.-C.: Global binary optimization on graphs for classification of high-dimensional data. *J. Math. Imaging Vis.* **52**(3), 414–435 (2015)
69. Miller, R. E., Thatcher, J. W.: editors. In: *Proceedings of a symposium on the Complexity of Computer Computations, held March 20–22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, The IBM Research Symposia Series*. Plenum Press, New York. (1972)
70. Mroueh, Y., Poggio, T., Rosasco, L., Slotine, J.-J.: Multiclass learning with simplex coding. In: *Advances in Neural Information Processing Systems*, pp. 2789–2797 (2012)
71. Mumford, D., Shah, J.: Optimal approximation by piecewise smooth functions and associated variational problems. *Comm. Pure Appl. Math.* **42**(42), 577–685 (1989)

72. Munoz, D., Vandapel, N., Hebert, M.: Directional associative markov network for 3-d point cloud classification. In: International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT) (2008)
73. Nene, S.A., Nayar, S.K., Murase, H.: Columbia Object Image Library (COIL-100). Technical Report CUCS-006-96, (1996)
74. Perona, P., Zelnik-Manor, L.: Self-tuning spectral clustering. *Adv. Neural Inf. Process. Syst.* **17**, 1601–1608 (2004)
75. Pock, T., Cremers, D., Bischof, H., Chambolle, A.: An algorithm for minimizing the piecewise smooth Mumford-Shah functional. In: IEEE International Conference on Computer Vision (ICCV), Kyoto, Japan, (2009)
76. Ranftl, R., Bredies, K., Pock, T.: Non-local total generalized variation for optical flow estimation. In: Computer Vision - ECCV 2014—13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part I, pp. 439–454 (2014)
77. Sawatzky, A., Tenbrinck, D., Jiang, X., Burger, M.: A variational framework for region-based segmentation incorporating physical noise models. *J. Math. Imaging Vis.* **47**(3), 179–209 (2013)
78. Shi, J., Malik, J.: Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(8), 888–905 (2000)
79. Sion, M.: On general minimax theorems. *Pacific J. Math.* **8**, 171–176 (1958)
80. Strang, G.: Maximum flows and minimum cuts in the plane. *Adv. Mech. Math.* **III**, 1–11 (2008)
81. Subramanya, A., Bilmes, J.: Semi-supervised learning with measure propagation. *J. Mach. Learn. Res.* **12**, 3311–3370 (2011)
82. Szlam, A., Bresson, X.: A total variation-based graph clustering algorithm for Cheeger ratio cuts. In: Proceedings of the 27th International Conference on Machine Learning, pp. 1039–1046 (2010)
83. Szlam, A.D., Maggioni, M., Coifman, R.R.: Regularization on graphs with function-adapted diffusion processes. *J. Mach. Learn. Res.* **9**, 1711–1739 (2008)
84. Tenbrinck, D., Lozes, F., Elmoataz, A.: Solving minimal surface problems on surfaces and point clouds. In: Scale Space and Variational Methods in Computer Vision—5th International Conference, SSVN 2015, Lège-Cap Ferret, France, May 31–June 4, 2015, Proceedings, pp. 601–612 (2015)
85. Tian, L., Macdonald, C. B., Ruuth, S. J.: Segmentation on surfaces with the closest point method. In: Proceedings ICIP09, 16th IEEE International Conference on Image Processing, pp. 3009–3012 (2009)
86. Toutain, M., Elmoataz, A., Lzoray, O.: Geometric pdes on weighted graphs for semi-supervised classification. In: 13th International Conference on Machine Learning and Applications (ICMLA), pp. 231–236 (2014)
87. Triebel, R., Kersting, K., Burgard, W.: Robust 3D scan point classification using associative markov networks. In: Proceedings of the International Conference on Robotics and Automation (ICRA), pp. 2603–2608 (2006)
88. Trillos, N.G., Slepcev, D.: Continuum limit of total variation on point clouds. *Arch. Ration. Mech. Anal.* **220**, 193–241 (2016)
89. Trillos, N. G., Slepcev, D., von Brecht, J., Laurent, T., Bresson, X.: Consistency of cheeger and ratio graph cuts. Technical report, [arXiv:1411.6590](https://arxiv.org/abs/1411.6590), (2014)
90. van Gennip, Y., Guillen, N., Osting, B., Bertozzi, A.L.: Mean curvature, threshold dynamics, and phase field theory on finite graphs. *Milan J. Math.* **82**(1), 3–65 (2014)
91. van Gennip, Y., Bertozzi, A.L.: Gamma-convergence of graph Ginzburg–Landau functionals. *Adv. Differ. Equ.* **17**(11–12), 1115–1180 (2012)
92. Wang, J., Jebara, T., Chang, S.F.: Graph transduction via alternating minimization. In: Proceedings of the 25th International Conference on Machine Learning, pp. 1144–1151 (2008)
93. Witkin, A., Kass, M., Terzopoulos, D.: Snakes Active contour models. *Int. J. Comput. Vis.* **1**, 321–331 (1988)
94. Yin, K., Tai, X.-C., Osher, S.: An effective region force for some variational models for learning and clustering. *UCLA CAM Report 16-18* (2016)
95. Yuan, J., Bae, E., Tai, X.-C., Boykov, Y.: A continuous max-flow approach to potts model. In: European Conference on Computer Vision, vol. 6316 of LNCS, pp. 379–392 (2010)
96. Yuan, J., Bae, E., Tai, X.-C.: A study on continuous max-flow and min-cut approaches. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 2217–2224 (2010)
97. Yuan, J., Bae, E., Tai, X.-C., Boykov, Y.: A spatially continuous max-flow and min-cut framework for binary labeling problems. *Numer. Math.* **126**(3), 559–587 (2013)
98. Zach, C., Gallup, D., Frahm, J.-M., Niethammer, M.: Fast global labeling for real-time stereo using multiple plane sweeps. In: Vision, Modeling and Visualization Workshop (VMV) (2008)
99. Zhang, X., Burger, M., Bresson, X., Osher, S.: Bregmanized non-local regularization for deconvolution and sparse reconstruction. *SIAM J. Imaging Sci.* **3**(3), 253–276 (2010)
100. Zhou, D., Schölkopf, B.: A regularization framework for learning from graph data. In: Workshop on Statistical Relational Learning. International Conference on Machine Learning (2004)
101. Zhou, D., Bousquet, O., Lal, T.N., Weston, J., Schölkopf, B.: Learning with local and global consistency. *Adv. Neural Inf. Process. Syst.* **16**, 321–328 (2004)
102. Zhu, X.: Semi-supervised learning literature survey. Computer Sciences Technical Report 1530, University of Wisconsin-Madison (2005)



Egil Bae is a research scientist at the Norwegian Defence Research Establishment (FFI). He received the MS and PhD degrees in applied mathematics from the University of Bergen in Norway in 2007 and 2011, respectively. From 2012 to 2014 he was as a postdoctoral scholar at the Department Mathematics, University of California, Los Angeles. His research interests include optimization, variational, and PDE-based methods for image processing, computer vision, machine learning and computer graphics.



Ekaterina Merkurjev is an Assistant Professor at Michigan State University with a joint appointment at the Department of Mathematics and the Department of Computational Mathematics, Science and Engineering. She obtained her Bachelors, Masters and PhD degrees in Applied Mathematics from University of California, Los Angeles. Her advisor was Professor Andrea Bertozzi. Following graduate school, she was a UC President's Postdoctoral Fellow at University of California, San Diego, from 2015 to 2016. Her research interests include optimization methods for semi-supervised learning, unsupervised learning, and image processing.