AN EFFECTIVE REGION FORCE FOR SOME VARIATIONAL MODELS FOR LEARNING AND CLUSTERING

KE YIN, XUE-CHENG TAI, AND STANLEY OSHER

Abstract. In this paper we propose several algorithms for some variational models for semi-supervised clustering of high-dimensional data. The new models produces substantial improvements of the classification accuracy in comparison with the corresponding models without the regional force in cases that the sample rate is relatively low. For the proposed models, the data points are modeled as vertices of a weighted graph, and the labeling function defined on each vertex takes values from the unit simplex, which can be interpreted as the probability of belonging to each class. The algorithm is proposed as a minimization of a convex functional of the labeling function. There are two versions of the models. The first one combines the Rayleigh quotient for the graph Laplacian and a region-force term, and the second one only replaces the Rayleigh quotient with the total variation of the labeling function. The region-force term is calculated by the affinity between each vertex and the training samples, characterizing the conditional probability of each vertex belonging to each class. The numerical methods for solving these two versions of the proposed algorithm are presented, and both are tested on several benchmark data sets such as handwritten digits (MNIST) and moons data. Experiments indicate that the correction rates and the computational speed are competitive with the state-of-the-art in multi-class semi-supervised clustering algorithms. Numerical experiments also confirm that the total variation model out performs the Laplacian counter part in most of the tests.

Keywords. semi-supervised clustering, graphical model, multi-class segmentation, region force penalty, Chan-Vese model

1. Introduction

The problem of partitioning a large dataset into a prescribed number of sensible groups is a fundamental task in machine learning and imaging. Many unsupervised and semi-supervised learning algorithm fall into this category. Also the contour detection and image segmentation can be viewed as data clustering [2, 17]. Many clustering algorithms are based on the

(Ke Yin, Stanley Osher) Department of Mathematics, University of California Los Angeles, 405 Hilgard Ave, Los Angeles CA 90095, USA.

(Xue-Cheng Tai) Department of Mathematics, University of Bergen, Postboks 7800, 5020 Bergen, Norway.

E-mail address: {kyin, sjo}@math.ucla.edu, tai@math.uib.no.

graphical model [11, 26, 32], where the data points are vertices of a weighted graph, and the edge weights models the affinity between pairs of data points. The popular spectral graph partitioning finds a minimal K-way graph cut or normalized graph cut. It is a NP-hard combinatorial problem. Some influential solvers using the lowest K eigenvectors of the graph Laplacian are discussed in [25]. Recently it is discovered that the minimization of graph cut has a continuous relaxation based on the min-cut-max-flow equivalence, see [40–42], and a primal-dual hybrid gradient method is proposed in [35]. There are some other algorithms based on Cheeger cut [4]. These problems are also shown to be NP-hard, and various continuous relaxation, though non-convex, are proposed and their algorithms are presented in [4, 15, 17, 39]. Some of them are based on solving the eigenvalue problem associated with the normalized graph Laplacian [17, 39], while others using non-linear optimization or partial differential equation solvers are discussed in [4, 15]. Due to the non-convexity of the proposed minimization problems, there could be difficulty finding the global minimizers.

In this paper we focus on multi-class semi-supervised clustering algorithms. It is assumed that the number of clusters are prescribed, and in each cluster there are a few samples that are already labeled. Our goal is to infer the labels for the rest of the data points from these labeled ones. The basic assumption is that vertices in the graph that are connected by edges of large weight should belong to the same cluser. The idea of geometric diffusion is proposed in the seminal paper by Coifman et. al. [10]. In there the diffusion map built on the eigenvectors of the graph Laplacian embeds the graph into the feature space with the diffusion distance as the new metric. Then the propagation of the labels is driven by a diffusion kernel defined in the feature space. Some other ideas inspired by the variational methods in image segmentation are also proposed, such as [4], where the Mumford-Shah-Potts Model on the graph is demonstrated to be useful. There the Cheeger cut is interpreted as the perimeter for the clusters, which we call the edge force. However a region force which penalizes the inhomogeneity inside the same cluster is absent. The celebrated Chan-Vese model [6] which combines an edge force and a region force can also be applied on the graph segmentation problems. Some early attempts in this spirit are presented in [16, 19]. Apart from these, there are some emerging interest in the application of the partial differential equation techniques, such the diffuse interface approach [13, 22]. The diffusion interface approach uses the phase field representation for the clusters, and a graph-based Merriman-Bence-Osher scheme [23] is established for solving the diffusion equation with double-well potential.

We propose a novel approach that combines the graph cut in the spectral clustering method and a region force inspired by the Chan-Vese model in image segmentation. In particular, we minimize the functional of the labeling function that is the sum of the K-way graph cut and the region force

for each cluster. The region force can be interpreted as a data fitting constraint. However, this constraint is not only imposed on the already labeled vertices on the graph, but on all vertices. The strength of the constraint is proportional to the conditional probability of each vertex belonging to each cluster, which depends on the affinity between that vertex and the given labeled vertices. Various ways of calculating the conditional probability are also discussed. The labeling function assumes values in the unit simplex. It is therefore an optimization problem of a convex objective function over the domain of the unit simplex. There are two versions of expressing the K-way graph cut in our approach. One is using the original definition, which is equivalent to the total variation on the graph, and the other is the quadratic relaxation of it or the Rayleigh quotient of the graph Laplacian. We also present efficient algorithms for solving this convex optimization problems. For the graph-cut version we use the primal-dual hybrid gradient method on the graph which is first described in [44]. For the Rayleigh quotient version we apply the projected gradient method with Barzilai–Borwein step sizes [12].

The rest of the paper is organized as follows. In Section 2, we discuss prior related work, as well as motivation for the methods proposed here. We then describe our two new versions of the proposed approach in Section 3. In Section 4, we present experimental results on benchmark data sets, demonstrating the effectiveness of our methods. Finally, in Section 5, we conclude and discuss ideas for future work.

2. Prior related work

2.1. **Graphical model for the data set.** The data points can be modeled as a weighted undirected graph $G = (V, E, w)$, where V and E are vertex set and edge set respectively, $w : E \to \mathbb{R}_+$ is the weight function defined on the edges. The coordinates $\{x_i\}$ for the vertices can be either the data vectors of the same dimension, or feature vectors after some transformation or filtering. For $x_i, x_j \in V$, $w_{ij} = w(x_i, x_j)$ measures the similarity between the two vertices. $W = (w_{ij})$ is called the affinity matrix, and usually assumed to be a symmetric matrix with non-negative entries [9].

Even though the weight function can be defined on all pairs of vertices, only a small portion of them are used in practice, depending on the topology of underlying structure of the graph. Therefore the affinity matrix W is sparse. For instance, under the assumption that the data points are uniformly distributed on a manifold X, the graph G is constructed by connecting k-nearest neighbors of each vertex with edges for a small k. The choice of k may also depend on the co-dimension of the manifold X. Under this assumption, some popular choices for the weight function are the Radial Basis Functions (RBF) [27]

$$w(x_i, x_j) = \exp(-d(x_i, x_j)^2/(2\epsilon)), \tag{1}$$

for the distance metric d, and the Zelnik-Manor and Perona (ZMP) [43] weight function

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

where σ is the local variance. Another popular choice in natural language processing for the weight function is the cosine similarity [29]

$$w(x_i, x_j) = \cos(x_i, x_j) = \frac{\langle x_i, x_j\rangle}{\|x_i\|\|x_j\|}. \tag{3}$$

If the manifold assumption is taken, then the graph G can be constructed by k-Nearest-Neighbor (k-NN). The number of neighborhood points k is practical, which may depend on the dimension of the underlying manifold for the graph. For some other types of graph topology such as the scale-free networks, for example the popular Barabasi-Albert (BA) model and Erdös-Rényi (ER) model, the tree structure may be more appropriate. We also note that there are some successful attempt to approximate a given *dense* graph G by a sparse graph on the same set of vertices [30].

2.2. **Some useful graph operators.** Let the matrix $W = (w_{ij})$ be constructed as outlined in the last section and $D = (d_{ii})$ be the diagonal matrix with the diagonal entries are equal to the sum of the entries on the same row in W.

The following are some graph operators that are useful in the subsequent discussions [9]:

$$\begin{cases} L = D - W, \text{ the graph Laplacian} \\ W_{\mathrm{s}} = D^{-1/2} W D^{-1/2}, \text{ the normalized affinity matrix} \\ L_{\mathrm{s}} = I - W_{\mathrm{s}}, \text{the normalized graph Laplacian.} \end{cases} \tag{4}$$

Assume $u \in L^2(V)$ is a function defined on the vertex set of graph G. The gradient operator [14]

$$\nabla : L^2(V) \to L^2(V, L^2(V)) \tag{5}$$

is defined by

$$\nabla u(x_i)(x_j) = w_{ij}(u(x_j) - u(x_i)). \tag{6}$$

So $\nabla u(x_i)$ is a function in $L^2(V)$ for each $x_i \in V$. It is assumed that G is a sparse graph and each x_i has at most d neighbors. Therefore, $\nabla u(x_i)$ can be understood as a sparse vector

$$\nabla u(x_i) = (w_{ij}(u(x_j) - u(x_i)))_{x_j \in \mathcal{N}(x_i)} \tag{7}$$

with at most d non-zeros which are in the neighborhood of x_j ($\mathcal{N}(x_i)$).

The adjoint operator of the gradient is called divergence, which is formally written as

$$\mathrm{div} : L^2(V, L^2(V)) \to L^2(V). \tag{8}$$

For $f \in L^2(V, L^2(V))$,

$$\operatorname{div}(f)(x_i) = \sum_{x_j \in \mathcal{N}(x_i)} w_{ij}(f(x_j)(x_i) - f(x_i)(x_j)). \tag{9}$$

Evaluating gradient and divergence both have $O(d|V|)$ complexity, since there are at most d non-zero terms in both (7) and (9) for each $x_i \in V$.

3. Graph partitioning by minimizing graph cuts

3.1. **Variational models for graph partitioning.** Suppose the graph of n vertices is partitioned into K clusters $V_1, \ldots, V_k$. We denote the $n \times K$ partition matrix by $\Psi = (\psi_{ik})$, where

$$\psi_{ik} = \begin{cases} 1 & \text{if } x_i \in V_k, \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

Therefore, Ψ is a binary matrix. Under the assumption that each vertex belongs to one and only one cluster, we have

$$\Psi \mathbf{1} = \mathbf{1}. \tag{11}$$

The K-way graph cut uses the following energy functional [17, 25] :

$$C = \sum_{k=1}^{K} \sum_{(x_i, x_j) \in E} w_{ij}(\psi_{ik} - \psi_{jk})^2, \tag{12}$$

which can be written as

$$C = \sum_{k=1}^{K} \Psi_k^T L \Psi_k, \tag{13}$$

with the matrix L defined as in (4). The K-way normalized graph cut is trying to minimize the following functional [17]:

$$C_{\mathrm{n}} = \sum_{k=1}^{K} \frac{\Psi_k^T L \Psi_k}{\Psi_k^T D \Psi_k^T} + (K-1) \frac{\Psi_k^T L \Psi_k}{\operatorname{Tr}(D) - \Psi_k^T D \Psi_k} \tag{14}$$

The Cheeger cut is also related [3, 9], which is trying to find a minimizer for:

$$C_{\mathrm{c}} = \sum_{k=1}^{K} \frac{\Psi_k^T L \Psi_k}{\min((K-1)\Psi_k^T D \Psi_k, \operatorname{Tr}(D) - \Psi_k^T D \Psi_k)}. \tag{15}$$

We note that for partition matrix Ψ taking binary values, the graph cut (12) can also be written as

$$C = \sum_{k=1}^{K} \sum_{(x_i, x_j) \in E} w_{ij} |\psi_{ik} - \psi_{jk}|, \tag{16}$$

As is often done in the literature, the above functional of Ψ is often defined as the (weighted) total variation of the labeling function on the graph, or written as

$$C = \sum_{k=1}^{K} \|\nabla \Psi_k\|_1. \tag{17}$$

In (14)–(15), $d_s = \mathrm{Tr}(D)$ is the sum of all the degrees of the graph, and $\|\Psi_k\|_2^2 = \Psi_k^T D \Psi_k$ is the square of the weighted L^2 norm of the labeling function for kth cluster. In the ideal case where the partition matrix taking values from $\{0, 1\}$, $\|\Psi_k\|_1 = \mathrm{sum}(D\Psi_k) = \Psi_k^T D \Psi_k$, that is the weighted L^1 norm of the labeling function for kth cluster is equal to that of the weighted L^2 norm squared. Using these notations, the minimization functional for the K-way normalized graph cut can be rewritten as

$$C_{\mathrm{n}} = \sum_{k=1}^{K} \frac{\|\nabla \Psi_k\|_1}{\|\Psi_k\|_1} + (K-1)\frac{\|\nabla \Psi_k\|_1}{d_s - \|\Psi_k\|_1}, \tag{18}$$

and the minimization functional for the Cheeger cut is

$$C_{\mathrm{c}} = \sum_{k=1}^{K} \frac{\|\nabla \Psi_k\|_1}{\min((K-1)\|\Psi_k\|_1,\, d_s - \|\Psi_k\|_1)}. \tag{19}$$

The variational models for unsupervised graph partitioning can be proposed as finding the binary partition matrix Ψ that minimizes the chosen graph cut from the above. These minimization problems are combinatorial in nature, and proven to be NP-hard [17]. Several relaxations are proposed, such as defining the new partition matrix $\Phi = (\phi_{ik}) \in \mathbb{R}^{n\times K}$ satisfying $\phi_{ik} \in [0, 1]$ and the relaxed minimization problem for graph cut is proposed as

$$\begin{aligned} \Phi = \underset{\phi_{ik}\in[0,1]}{\mathrm{argmin}} & \sum_{k=1}^{K} \Phi_k^T L \Phi_k \\ \text{s.t. } & \Phi \mathbf{1} = \mathbf{1}, \end{aligned} \tag{20}$$

or in the line of (17)

$$\begin{aligned} \Phi = \underset{\phi_{ik}\in[0,1]}{\mathrm{argmin}} & \sum_{k=1}^{K} \|\nabla \Phi_k\|_1 \\ \text{s.t. } & \Phi \mathbf{1} = \mathbf{1}. \end{aligned} \tag{21}$$

We note that if the ordinary graph cut (13) is chosen, then the trivial minimizer for the above minimization problems that all vertices belong to the same cluster should be avoided. Several algorithms for minimizing graph-cut (12) or normalized graph cut (14) are proposed in [17, 25, 39], which are based on lowest eigenfunctions of the graph Laplacian L or the normalized graph Laplacian L_{s}. Several non-linear optimization algorithms for minimizing Cheeger cut (19) and its relaxations are described in [4, 5, 15].

In [21, 40, 42], (21) is interpreted as the dual of a max-flow problem and showed that the non-relaxed min-cut problem is equivalent to the continuous max-flow problem and thus to (21). We want to emphasis that such an equivalence is not valid for the Laplacian counter part (20). The proof of [40,42] were given for continuous setting and (21) is the discettized version of the corresponding continuous problems.

3.2. **Variational models for semi-supervised graph partitioning.** If the vertices of the graph are partially labeled, then the graph partitioning problem becomes semi-supervised. It is desirable to infer the labels for the rest of the vertices from the labeled ones. In practice, the number of labeled vertices (called training samples) is only a small fraction of the total number of vertices. We denote those labeled samples in V_k by S_k, and $S = \bigcup_{k=1}^{K} S_k$.

One way to utilizing the training samples is to incorporate it as a data fidelity term in the minimization of the energy functional

$$E(\Phi) = R(\Phi) + \mu(\Phi, \hat{\Phi})\|\Phi - \hat{\Phi}\|_1, \tag{22}$$

where $\hat{\Phi} = (\hat{\phi}_{ik})$ is the $n \times K$ labeling matrix for the training samples. $\hat{\Phi}$ can be written as

$$\hat{\phi}_{ik} = \begin{cases} 1 & \text{if } x_i \in S_k, \\ 0 & \text{otherwise.} \end{cases} \tag{23}$$

$R(\Phi)$ is a regularization functional that can be taken as some graph cut described in the previous sub-section [13]. The data fidelity $\mu(\Phi, \hat{\Phi})$ is taken as a large constant defined on the training samples and zero elsewhere. Some other improvement of the data fidelity term includes updating μ iteratively based on the current solution of Φ [16].

Another way to incorporate this training sample data is to fix the values of Ψ and only update the values of Ψ in the rest of the vertices, i.e. we ask

$$\phi_{ik} = \begin{cases} 1 & \text{if } x_i \in S_k, \\ 0 & \text{if } x_i \in S \backslash S_k. \end{cases} \tag{24}$$

and only compute the values of ϕ_{ik} in $V \backslash S$.

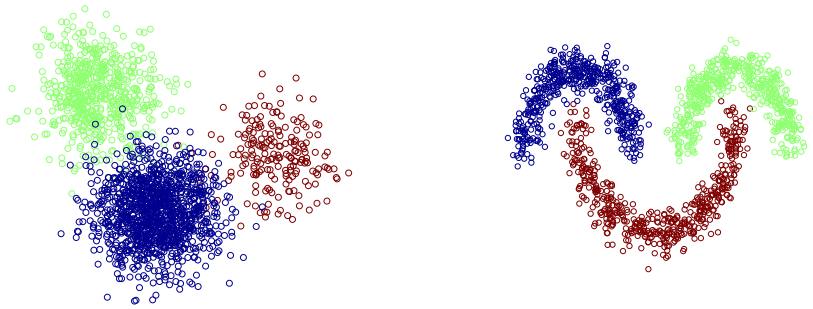
3.3. **An effective region force for graph partitioning.** We would like to replace the data fidelity in (22) by a region force term. It is inspired by the celebrated Chan-Vese model [6]. As in [16], Chan-Vese model for multi-class graph partitioning can be written as

$$\begin{aligned} \Phi = \underset{\phi_{ik} \in [0,1]}{\operatorname{argmin}} & \sum_{k=1}^{K} \sum_{x_i \in V} \left(g_k(x_i) \|(\nabla \Phi_k)_i\| + \mu \phi_{ik} \|x_i - c_k\|^2 \right), \\ & \text{s.t. } \Phi \mathbf{1} = \mathbf{1}, \end{aligned} \tag{25}$$

where Φ_k is the kth column of Φ, c_k is the centroid of each cluster in the feature space, and g_k is the edge detector function for k-th cluster. The centroids are updated iteratively based on the current labeling. In practice,

the centroids are calculated by the (weighted) average of the feature vectors in each cluster. See Figure 1a for an illustration of high dimensional data where the points of the same cluster are distributed around the centroids. The first term in (25) is called the edge force since it penalizes the (weighted) length of the boundary of each cluster, and the second term is the region force because it penalizes the inhomogeneity of the feature vectors inside each cluster.

This choice of region force is successful in image segmentation tasks, where the feature vectors are pixel values or some transformation of local patches. The image features are quite homogeneous in visually smooth regions, and the centroids are characteristic of the regions. However, in some machine learning and data clustering tasks, the feature vectors of the data points in the same class can be varying, and the centroids may not be inside the respective regions. A classical example is the three-moon synthetic data set, where the centroids of the all three classes coincide and is positioned outside of respective classes, see Figure 1b. For such cases, the region force given in (25) will not be appropriate.



(A) Three mixed Gaussian data (B) Three-Moon data

FIGURE 1. (1a) The centroids are the centers of each underlying Gaussian distribution, which are inside each cluster. (1b) The centroids are the centers of each arc, which are outside of the arcs.

We use a novel region force term without defining centroids, but using the affinity matrix W. It can be formally written as

$$p_k(W, x_i)(1 - \phi_{ik}) + (1 - p_k(W, x_i))\phi_{ik}, \tag{26}$$

where p_k is the probability measure on V characterizing the probability of each vertex belonging to cluster V_k. It can be considered as a conditional probability given labeled vertices.

There are several choices for choosing p_k for (26). They are based on the idea that the points "close" to the given labeled vertices should have high

probability of having similar labels. Similar ideas can be found in [36, 37], where a novel Markov chain based learning algorithm is proposed to infer the unknown labels of objects from known labeled ones. In our formulation, the conditional probability of a vertex x_i's membership in cluster S_k is calculated as the average of the "influence" of all labeled vertices in S_k. For instance, we assume that the "influence" of a labeled vertex $x_j \in S_k$ to the given vertex x_i is inversely proportional to the diffusion distance (proposed in [10]) between them. The m-th diffusion distance between two vertices x_i, x_j, denote by $d^{(m)}(x_i, x_j)$, characterizes the rate of connectivity between two points x_i, x_j through m edges. In particular when $m = 1$, it is the rate of connectivity between two neighbors, and when $m = 2$, it is the rate of connectivity between two second neighbors (vertices connected by two edges). It is small if there are a large number of weighted paths of length m connecting x_i and x_j. In this case, we can calculate p_k as

$$p_k(W, x_i) = \frac{\frac{1}{|S_k|}\sum_{j\in S_k}(d^{(m)}(x_i, x_j))^{-1}}{\sum_{r=1}^{K}\frac{1}{|S_r|}\sum_{j\in S_r}(d^{(m)}(x_i, x_j))^{-1}}. \tag{27}$$

Let $\hat{W} = D^{-1/2}WD^{-1/2}$ be the normalized affinity matrix and $\hat{W}^m = (\hat{w}_{ij}^{(m)})$ be the m-th power of it. The m-th diffusion distance can be calculated by [10]

$$d^{(m)}(x_i, x_j) = \hat{w}_{ii}^{(m)} + \hat{w}_{jj}^{(m)} - 2\hat{w}_{ij}^{(m)}. \tag{28}$$

Another way to calculate p_k can be

$$p_k(W, x_i) = \frac{\frac{1}{|S_k|}\sum_{j\in S_k} q_{ij}}{\sum_{r=1}^{K}\frac{1}{|S_r|}\sum_{j\in S_r} q_{ij}}, \tag{29}$$

where

$$q_{ij} = \frac{(\hat{w}_{ij}^{(m)})^2}{\hat{w}_{ii}^{(m)}\hat{w}_{jj}^{(m)}}. \tag{30}$$

In case when the denominator is zero, we set $p_k(W, x_i) = \frac{1}{K}$. In this paper, we use $m = 1$ or 2, which are found to have good performance in practice. For $m = 1$, the complexity of computing p_k is $O(Kd|S|)$, where d is the average degree of a vertex and $|S|$ is the total number of labeled points. We note that for $m = 2$, it is unnecessary to compute $\hat{W}^2$ in order to obtain $\hat{w}_{ij}^{(2)}$ for selected $j \in S_k$. It is readily checked that

$$\hat{w}_{jj}^{(2)} = \|\hat{W}_j\|_2 \tag{31}$$

where $\hat{W}_j$ is the j-th column of $\hat{W}$; and

$$\hat{W}_j^2 = \hat{W}\hat{W}_j \tag{32}$$

for j-th column of $\hat{W}^2$. Only $j \in S_k (k = 1, \ldots, K)$ are needed, usually a small fraction of all columns, i.e., $|S| \ll |V|$. In summary the complexity of computing p_k for $m = 2$ is $O(Kd|S||V|)$.

Replacing the region force term in (25) by (26) we have a new model

(33)
$$\Phi = \underset{\phi_{ik}\in[0,1]}{\operatorname{argmin}} (1-\tau) \sum_{k=1}^{K} \sum_{x_i \in V} g_k(x_i) \|(\nabla \Phi_k)_i\| + \tau (p_k(W, x_i)(1-\phi_{ik}) + (1-p_k(W, x_i))\phi_{ik}),$$
$$\text{s.t. } \Phi \mathbf{1} = \mathbf{1},$$

where $\tau \in (0, 1)$. It is a convex optimization problem. For all the application condidered in this work, the edge detector g_k can be set to be constant 1. Under this relaxation, we rewrite the above model as

(34)
$$\Phi = \underset{\phi_{ik}\in[0,1]}{\operatorname{argmin}} (1-\tau) \sum_{k=1}^{K} \sum_{x_i \in V} \|(\nabla \Phi_k)_i\| + \tau (p_k(W, x_i)(1-\phi_{ik}) + (1-p_k(W, x_i))\phi_{ik}),$$
$$\text{s.t. } \Phi \mathbf{1} = \mathbf{1}.$$

As a quadratic relaxation of (34), we have

(35)
$$\Phi = \underset{\phi_{ik}\in[0,1]}{\operatorname{argmin}} \frac{1-\tau}{2} \sum_{k=1}^{K} \sum_{x_i \in V} \|(\nabla \Phi_k)_i\|^2 + \tau (p_k(W, x_i)(1-\phi_{ik}) + (1-p_k(W, x_i))\phi_{ik}),$$
$$\text{s.t. } \Phi \mathbf{1} = \mathbf{1}.$$

Comparing this model with the model in (20), a region force has been added. The objective functional of this convex optimization problem is differentiable.

As an extreme case, if no vertices are labeled, then for each cluster V_k, p_k is defined as the uniform distribution

$$p_k(W, x_i) = \frac{1}{K}, \tag{36}$$

and the proposed models (34)–(35) has trivial solutions.

We could also add the region force of (22) to our model or fix the label values at the sample points as in (24). We find that this is not increasing the accuracy in most of the cases, especially when there are sampled points near the boundary of the classes. Thus, we have chosen not to do so in our tests given later.

Given a partition matrix Φ taking values on $[0, 1]$, the label for each vertex x_i is calculated by $\underset{1\le k\le K}{\operatorname{argmax}} \phi_{ik}$.

3.4. **Algorithms for solving graph partitioning with a region force.** In this section, we describe iterative algorithms for solving (34)–(35). In the following, we define $P = (p_{ik})$ as an $n \times K$ matrix with $p_{ik} = p_k(W, x_i)$, and

Φ_k is the kth column of Φ. (35) can be written as

$$\Phi = \underset{\phi_{ik}\in[0,1]}{\operatorname{argmin}} \frac{1-\tau}{2}\sum_{k=1}^{K}\langle \Phi_k, L\Phi_k\rangle + \tau(\langle p_k, \mathbf{1}-\Phi_k\rangle + \langle \mathbf{1}-p_k, \Phi_k\rangle), \quad \text{s.t. } \Phi\mathbf{1}=\mathbf{1}. \tag{37}$$

It is a quadratic minimization problem with convex constraint. It can be solved by projected gradient method with Barzilai-Borwein step sizes [12]. The feasible domain for (37) is

$$\Delta = \{\Phi \in \mathbb{R}^{n\times K} : \Phi\mathbf{1}=\mathbf{1}\}. \tag{38}$$

The objective functional in (37) is denoted by $J(\Phi)$. Then the projected gradient method for solving (37) in one iteration can be written as

$$\Phi^{(j+1)} = \Pi_\Delta(\Phi^{(j)} - \alpha^{(j)}\partial J(\Phi^{(j)})\Lambda^{(j)}), \tag{39}$$

where $\alpha^{(j)} = 1$ by default, $\Lambda^{(j)}$ is an $K\times K$ diagonal matrix with diagonal entries $(\lambda_1^{(j)},\ldots,\lambda_K^{(j)})$ equal to the step sizes for the gradient descent directions. The step sizes $\lambda_k^{(j)}$ alternate between the two choices

$$\lambda_k^{(j)} = \frac{\|\boldsymbol{s}_k^{(j-1)}\|^2}{\langle \boldsymbol{s}_k^{(j-1)}, \boldsymbol{y}_k^{(j-1)}\rangle}, \tag{40}$$

and

$$\lambda_k^{(j)} = \frac{\langle \boldsymbol{s}_k^{(j-1)}, \boldsymbol{y}_k^{(j-1)}\rangle}{\|\boldsymbol{y}_k^{(j-1)}\|^2}, \tag{41}$$

where $\boldsymbol{s}_k^{(j-1)}$ and $\boldsymbol{y}_k^{(j-1)}$ are kth columns of $\boldsymbol{s}^{(j-1)} = \Phi^{(j)} - \Phi^{(j-1)}$ and $\boldsymbol{y}^{(j-1)} = \partial J(\Phi^{(j)}) - \partial J(\Phi^{(j-1)})$ respectively. The projection operator onto the feasible domain Π_Δ is implemented by the fast algorithm proposed in [8], the complexity of which is $O(|V|K\log K)$. The projected gradient algorithm does not guarantee the decreasing of the objective functional. To ensure sufficient decreasing, a non-monotone line search is taken by decreasing the parameter $\alpha^{(j)}$ in (39), based on Armijo-type acceptability test (see [12] and the references therein)

$$J(\Phi^{(j+1)}) \le J(\Phi^{(j)}) + \theta \mathrm{Tr}(\partial J(\Phi^{(j)})^T \boldsymbol{s}^{(j)}), \tag{42}$$

for some small constant $\theta > 0$. It is summarized as Algorithm 1 (named as LapRF). The stopping criterion is chosen as

$$|J(\Phi^{(j+1)}) - J(\Phi^{(j)})| \le \epsilon J(\Phi^{(j)}) \tag{43}$$

for some small $\epsilon > 0$. The complexity of each outer iteration is dominated by computing projected gradient as in (39), which is $O(Kd|V| + |V|K\log K)$.

Algorithm 1 Laplacian-based multi-class graph partitioning with a region force (LapRF)

Require: L, $P = [p_1 \dots p_K]$, τ
Ensure: Φ
1: **function** LapRF
2: **while** "not converged" **do**
3: $\alpha = 1$;
4: calculate Φ by (39);
5: **while** (42) is not satisfied **do**
6: $\alpha = 0.8\alpha$;
7: re-calculate Φ by (39);
8: **end while**
9: **end while**
10: **return** Φ.
11: **end function**

Next, we describe the algorithm for solving (34). First, we rewrite it using simpler notations for the variables as

$$\Phi = \underset{\phi_{ik}\in[0,1]}{\operatorname{argmin}} \sum_{k=1}^{K} (1-\tau)\|\nabla\Phi_k\|_1 + \tau(\langle p_k, \mathbf{1}-\Phi_k\rangle + \langle \mathbf{1}-p_k, \Phi_k\rangle), \quad \text{s.t. } \Phi\mathbf{1} = \mathbf{1}. \tag{44}$$

We first note that the variational formulation for total variation

$$\|\nabla\Phi_k\|_1 = \max_{q_k\in\mathbb{R}^{n\times n},\, \|q_k\|_\infty\le 1} \langle \Phi_k, \operatorname{div} q_k\rangle, \tag{45}$$

then the minimization problem (44) can be proposed as a saddle-point problem

$$\min_{\Phi\in\Delta} \max_{\|q_k\|_\infty\le 1} \sum_{k=1}^{K} (1-\tau)\langle \Phi_k, \operatorname{div} q_k\rangle + \tau(\langle p_k, \mathbf{1}-\Phi_k\rangle + \langle \mathbf{1}-p_k, \Phi_k\rangle). \tag{46}$$

This saddle point problem can be solved by primal-dual hybrid gradient method, which alternates between two steps: gradient ascent for dual variables q_k and gradient descent for primal variable Φ_k [44]. More specifically, they are

$$q_k^{(j)} = \Pi_{\|q_k\|_\infty\le 1}(q_k^{(j-1)} - \beta^{(j-1)}\nabla\Phi_k^{(j-1)}) \text{ for } j = 1, \dots, K, \tag{47a}$$

$$\Phi^{(j)} = \Pi_\Delta(\Phi^{(j-1)} - \gamma^{(j-1)}((1-\tau)\operatorname{div} Q^{(j)} + \tau(\mathbf{1}-2P))), \tag{47b}$$

where $Q = [q_1 \dots q_K]$. The choice for the step sizes β, γ follows the theoretical analysis in [1]. It is summarized as Algorithm 2 (named as TVRF). Line 5 of this algorithm is solved by hard-thresholding and line 7 is calculated by the projection onto simplex algorithm described in [8]. The stopping criterion is similar to (43). The complexity of each iteration is dominated by

Algorithm 2 TV-based multi-class graph partitioning with a region force (TVRF)

Require: $P = [p_1 \dots p_K]$, W, $\{\beta_l = 0.2l\}$, $\{\gamma_l = 0.1/(1 + 0.1l)\}$, τ
Ensure: Φ
function TVRF
  $l = 0$,
  while "not converged" **do**
    for $k = 1 \dots K$ **do**
      calculate $\nabla\Phi_k$,
      $q_k = \Pi_{\|q_k\|_\infty \le 1}(q_k - \beta_l \nabla\Phi_k)$,
    end for
    calculate div$Q$,
    $\Phi = \Pi_\Delta(\Phi - \gamma_l((1-\tau)\text{div}Q + \tau(\mathbf{1} - 2P)))$,
    $l = l + 1$,
  end while
  return Φ.
end function

computing (47), in which $O(Kd|V|)$ for (47a) and $O(Kd|V| + |V|K \log K)$ for (47b).

4. Numerical experiments

In this section, we evaluate the performance of our proposed algorithms on several benchmark semi-supervised learning data sets, including `Text`, `COIL`, `Opt-Digits` and `MNIST`. `Text` and `COIL` data sets are from the supplementary material of [7] (`http://olivier.chapelle.cc/ssl-book/benchmarks.html`). `Opt-Digits` comes from "UCI machine learning repository" (`http://archive.ics.uci.edu/ml/datasets.html`). `MNIST` is from "The MNIST Database of Handwritten Digits" (`http://yann.lecun.com/exdb/mnist/`). The number of classes of these data sets are known. Among them Text is binary class, while the rest are multi-class. Also we test against a synthetic data set – the three-moon model. The basic properties of the benchmark data sets are shown in Table 1.

Table 1. Basic properties of the benchmark data sets.

Data set	Classes	Dimension	Points
`Three Moon`	3	100	1500
`COIL`	6	241	1500
`Text`	2	11,960	1500
`Opt-Digits`	10	64	5620
`MNIST`	10	784	60,000

As the formulation for the graphical model, k-NN graphs are constructed for the data sets. Here we make use of an implementation of the randomized

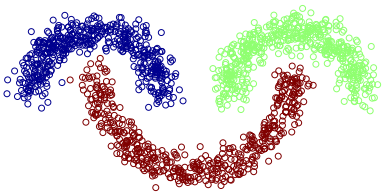
kd-tree [24, 28], called `VLFeat` [34], for finding k-nearest neighbors. Except for `Text` data set, the Zelnik-Manor and Perona weight function (2) are used, with the standard deviations estimated by the distance from the k-th nearest neighbor to the current point. `Text` data set uses cosine similarity weight function.

We apply Algorithm 1 (LapRF) and Algorithm 2 (TVRF) respectively with different number of randomly selected training samples for these data sets. Unless specified otherwise, in Algorithm 1 we choose $\tau = 0.4$ and in Algorithm 2 $\tau = 0.8$. p_k is calculated by (29) for $m = 1$ or $m = 2$. As for the region force term (26), the conditional probability p_k is calculated by (29). The stopping criteria for LapRF and TVRF are (43) for $\epsilon = 10^{-6}$, with maximum number of outer iterations is set to 100. The correction rate, or accuracy is defined as the percentage of correctly labeled data points. The accuracy for the classification results of `COIL`, `Text` and `Opt-Digits` are compared with five existing methods: k-NN, SGT, LapRLS, SQ-Loss-I and MP where the results are taken from [31]. The `Three Moon` and `MNIST` are compared against recently proposed methods called multiclass Ginsburg-Landau MBO scheme (multiclass-MBO) [13], since these two data sets are not listed in the previous reference [31].

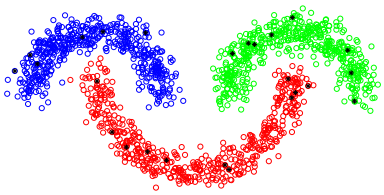
In the tables in the testing examples given in the following, l is the number of labeled sample points. The number in the brackets after l is the percentage of the samples compared with the total number of input data. The tables show the accuracy of the corrected computed labels.

4.1. **Three Moon data set.** The three-moon synthetic data is constructed by three one-dimensional half circles with added gaussian noise. Here we choose the three circles are centered at (0,0), (3,0), (1.5, 0.4) respectively with radius 1, 1, and 1.5. 500 points are uniformly sampled from each half circle. They are embedded into $\mathbb{R}^{100}$ by appending zeros to the coordinates and i.i.d Gaussian noise of standard deviation equal to 0.14 are added to each entry of the coordinates. See Figure 2a for an illustration in the first two dimensions.

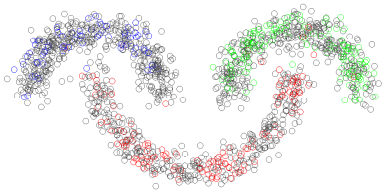
A k-NN graph with $k = 10$ is built for the data set. The distance metric is set to be the Euclidean distance between two points in $\mathbb{R}^{100}$. Total number of $l = 25, 50, 75$ labeled samples are used for testing the accuracy of the partitioning, where each l is tested with 10 different sets of randomly selected labeled samples. For comparison, we calculate p_k by (29) for both $m = 1$ and 2. Some illustrations of p_k and the final partition results are in Figures 2c-2f. The average rate of correctly labeled points for each l is listed in Table 2, comparing also with multiclass-MBO method, the parameters of which are the same as in [13]. From Table 2, we see that all the methods perform rather well with high sample rate (5%). With low sample rate, i.e. $l = 25, 50$, the proposed method TVRF outperform the multiclass MBO by a huge margin.



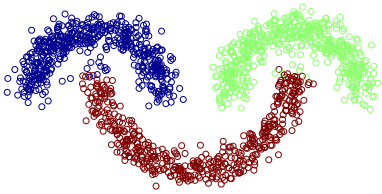
(A) Groud truth labels



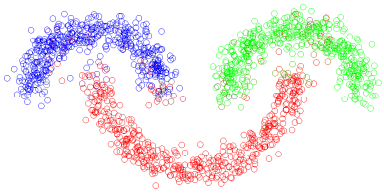
(B) 25 randomly sampled labels in black



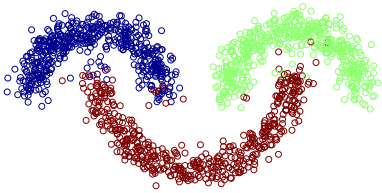
(C) Calculated p_k by (29) for $m = 1$



(D) Partition result by TVRF ($m = 1$)



(E) Calculated p_k by (29) for $m = 2$



(F) Partition result by TVRF ($m = 2$)

FIGURE 2. Three-moon synthetic data. Each point is colored by RGB vector (p_1, p_2, p_3) up to a proper scaling. The RGB vectors $(1, 0, 0)$, $(0, 1, 0)$ and $(0, 0, 1)$ represent blue, green and red respectively. For black points $p_1 = p_2 = p_3 = 1/3$.

TABLE 2. Comparison of accuracy (%) for different numbers (l) of labeled samples using various algorithms for three-moon.

l	25 (1.25%)	50 (2.5%)	75 (5%)
multiclass MBO	68.3	84.1	94.3
LapRF ($m = 1$)	95.1	96.4	98.1
TVRF ($m = 1$)	96.4	98.2	98.4
LapRF ($m = 2$)	96.4	97.9	98.5
TVRF ($m = 2$)	**96.4**	**98.2**	**98.6**

Next, we compare the result of TVRF (Algorithm 2) with that of minimizing the functional (22) that uses data fitting defined only on the points that are labeled, i.e we compare our model with the model in (22). More exactly, we solve the minimization problem

$$\Phi = \underset{\phi_{ik}\in[0,1]}{\operatorname{argmin}} \sum_{k=1}^{K}(1-\tau)\|\nabla\Phi_k\|_1 + \sum_{x_i\in V}\tau_k(x_i)(1-\Phi_k(x_i)), \quad \text{s.t. } \Phi\mathbf{1} = \mathbf{1}, \tag{48}$$

where

$$\tau_k(x_i) = \begin{cases} \tau & \text{if } x_i \in S_k, \\ 0 & \text{otherwise.} \end{cases} \tag{49}$$

We compare the result obtained from solving (48) (TV with point-wise data fidelity, abbreviated as TVP) and that of region force penalty (TVRF) using the same parameters (in particular $\tau = 0.8$). The average rate of correctly labeled points for each l is listed in Table 3. We can see that the region force has a significant contribution to the partition accuracy. Compared with the data fidelity that defined only at the already labeled points, using region force gives much better accuracy especially in the low labeling rate cases.

TABLE 3. Comparison of accuracy (%) for different numbers (l) of labeled samples using TVRF and TVP (48) for `Three-Moon`.

l	25 (1.25%)	50 (2.5%)	75 (5%)
multiclass MBO	68.3	84.1	94.3
TVP	60.0	73.5	93.4
TVRF ($m = 1$)	**96.4**	**98.2**	**98.4**

These comparisons show that the combination of total variation (TV) and the region force can be competitive against existing methods, especially when the sample rate of the labeled data is low. We also note the slight difference between TVRF ($m = 1$) and TVRF ($m = 2$). The difference lie

in the calculation of conditional probability $p_k(x_i)$ of an unknown label given existing labels. The former one ($m = 1$) considers the direct neighbors of the labeled points and latter ($m = 2$) uses the second neighbors. From Figures 2c 2e we can see that in $m = 2$ case, p_k is non-trivial for almost all points, which places strong assumption on the probability of the labels. It is useful when the noise in the data is large (see the Text data set below). However, for $m = 2$ the computational complexity of p_k is bigger ($O(Kd|S||V|)$) than $m = 1$'s $O(Kd|S|)$.

4.2. **MNIST data set.** The MNIST data set consists of 70,000 size-normalized and centered 28×28 images of handwritten digits 0-9. So there are 10 classes of data, and they are roughly balanced. We use the training set of 60,000 points. The task is to partition the data set into 10 classes, given relative few labeled samples.

A k-NN graph with $k = 10$ is built for the data set. The distance metric is set to be the Euclidean distance between two points as 784 dimensional vectors. In Algorithm 1 (LapRF) we choose $\tau = 0.5$ for the case $m = 1$, and $\tau = 0.4$ for the case $m = 2$. In Algorithm 2 (TVRF) $\tau = 0.9$ for the case $m = 1$ and $\tau = 0.6$ for $m = 2$. For TVP, $\tau = 0.99$ is used for (48). Total number of $l = 150, 300, 600$ labeled samples are used for testing the accuracy of the partitioning, where each l is tested 10 times. The average correction rate for each l is listed in Table 4. We can see that the correction rate is very high even for very low sample rate of labeled data.

We have also done other tests as were done for the three-moon data set. The conclusions are essentially similar, i.e. our model gives substantial improvement of labelling accuracy when the sample rate is relatively low. TVRF with $m = 1$ (and some times $m = 2$) produces the best accuracy in comparison with mutlclass-MBO, LapRF and TVP. The results with our proposed region force gives clearly better results than without it.

Next, we simply threshold p_k calculated by (29) for $m = 2$ to partition the data set. More exactly, the label for a point x_i is calculated by $\max_k p_k(W, x_i)$. The result is recorded also in Table 4. We can see that the region force in the case of $m = 2$ alone cannot do a good job partitioning the data set with low rate of labeled data, but the graph cut minimization, or equivalently total variation (TV) minimization significantly improves the partition accuracy.

As for the time comparison, we perform all testing on MATLAB on a Linux machine with octa-core Intel Core i7-4770S CPU at 3.10GHz and 7.7GB memory. The construction of k-NN graph with $k = 10$ using `VLFeat` [34] takes 26.7 seconds. The rest of the computation time are shown also in Table 4. The MBO method takes into account of the time spent on calculating the eigenvectors. We can see that TVRF is at least 10 times faster than multiclass-MBO, and the correction rate is better.

We also note that in our experiment, we make use of no image features other than the pixel values that form the data vector. Of course, if the

TABLE 4. Comparison of accuracy (%) for different numbers (l) of labeled samples using various algorithms for `MNIST` data set. multiclass MBO includes the time for computing eigenvectors of W.

l	0.25%	0.5%	1%	ave. time (s)
$p_k(m=2)$	35.5	52.3	71.5	0.4
TVP	83.7	86.3	90.8	66
multiclass MBO	73.0	90.1	94.9	845
LapRF ($m=1$)	84.2	90.9	95.1	18
TVRF ($m=1$)	**93.4**	**96.4**	**96.8**	66
LapRF ($m=2$)	91.0	94.2	95.6	14
TVRF ($m=2$)	**94.6**	**96.6**	**96.7**	61

images are preprocessed by some advanced filters before constructing the affinity matrix, the accuracy of clustering result can be improved. Here we experiment with the adjacency matrix A provided in [38] and also used in [3]. There A is a binary matrix characterizing the k-NN graph structure for $k = 10$. The distance is not defined as the Euclidean distance between two vectors of image pixel values, but is calculated by extracting scattering features in the images first before using the new feature vectors to define the distance. The scattering features is proposed in [20] for signal processing. Using this A we compute W as a weighted affinity matrix, where the indices of nonzeros are the same as A, but the values are recalculated by (2). We then use this W for TVRF ($m = 2$) and compare the results with the Multiclass Total Variation (MTV) method [3], which is based on some relaxation of Cheeger cut. MTV can be used for both unsupervised and semi-supervised clustering, and here we can compare with the semi-supervised version. The comparison is shown in Table 5. We can see that our method compares well with MTV in speed and accuracy.

TABLE 5. Comparison of accuracy (%) using scattering features of images for constructing adjacency/affinity matrix for `MNIST` data set. The average running time in seconds are inside the parentheses.

l	0.25%	0.5%	1%
MTV	**97.56** (110)	97.66 (70)	97.64 (65)
TVRF ($m=2$)	97.52 (60)	**97.66** (64)	**97.70** (55)

4.3. **COIL data set.** The Columbia object image library (COIL-100) is a set of color images of 100 different objects taken from different angles (in steps of 5 degrees) at a resolution of 128×128 pixels. To create our data set, we first down-sampled the red channel of each image to 16×16 pixels

by averaging over blocks of 8×8 pixels. We then randomly selected 24 of the 100 objects (with $24 \times 360/5 = 1728$ images). The set of 24 objects was partitioned into six classes of four objects each. We then randomly discarded 38 images of each class, to leave 250 each.

A k-NN graph with $k = 5$ is built for the data set. The distance metric is set to be the Euclidean distance between two images as 241 dimensional vectors. In Algorithm 1 we choose $\tau = 0.1$ and in Algorithm 2 $\tau = 0.8$. Total number of $l = 50, 100, 150$ labeled samples are used for testing the accuracy of the partitioning, where each l is tested 10 times. The average rate of correctly labeled points for each l is listed in Table 6. In this case, TVRF ($m = 1$) is consistently better or at least comparable to other methods.

TABLE 6. Comparison of accuracy (%) for different numbers (l) of labeled samples using various algorithms for `COIL`.

l	50 (3.3%)	100 (6.7%)	150 (10%)
k-NN	66.9	79.2	83.5
SGT	78.0	89.0	89.9
LapRLS	78.4	84.5	87.8
SQ-Loss-I	**81.0**	89.0	90.9
MP	78.5	**90.2**	**91.1**
LapRF ($m = 1$)	71.7	87.0	91.0
TVRF ($m = 1$)	**80.3**	**90.0**	**91.7**

4.4. **Opt-Digits data set.** Preprocessing programs made available by NIST are used to extract normalized bitmaps of handwritten digits from a preprinted form. 32×32 bitmaps are divided into non-overlapping blocks of 4×4 and the number of on pixels are counted in each block. This generates an input matrix of 8×8 where each element is an integer in the range 0 to 16.

A k-NN graph with $k = 10$ is built for the data set. The distance metric is set to be the Euclidean distance between two images as 64 dimensional vectors. Total number of $l = 50, 100, 150$ labeled samples are used for testing the accuracy of the partitioning, where each l is tested 10 times. The average rate of correctly labeled points for each l is listed in Table 7. In this case, TVRF ($m = 1$) is consistently better or at least comparable to other methods.

4.5. **Text data set.** This is the 5 comp.* groups from the Newsgroups data set and the goal is to classify the ibm category versus the rest [33]. A tf-idf (term frequency – inverse document frequency) encoding resulted in a sparse representation with 11,960 dimensions. For the benchmark, 750 points of each class have been randomly selected and the features randomly permuted.

A k-NN graph with $k = 50$ is built for the data set. The distance metric is set to be the cosine similarity between two points as 11960 dimensional

TABLE 7. Comparison of accuracy (%) for different numbers (l) of labeled samples using various algorithms for `Opt-Digits`.

l	50(0.89%)	100(1.78%)	150(2.67%)
k-NN	85.5	92.0	93.8
SGT	91.4	97.4	97.4
LapRLS	92.3	**97.6**	97.3
SQ-Loss-I	**95.9**	97.3	**97.7**
MP	94.7	97.0	97.1
LapRF ($m = 1$)	79.0	95.2	96.8
TVRF ($m = 1$)	**95.9**	**97.2**	**98.3**

vectors. In Algorithm 1 (LapRF) we choose $\tau = 0.9$ and in Algorithm 2 (TVRF) $\tau = 0.9$. Total number of $l = 50, 100, 150$ labeled samples are used for testing the accuracy of the partitioning, where each l is tested 10 times. The average correction rate for each l is listed in Table 8. Again, our method produces better or nearly as good results as the state-of-art methods. For this examples, we observe that the choice $m = 2$ for calculating p_k performs better than $m = 1$ in the clustering result.

TABLE 8. Comparison of accuracy (%) for different numbers (l) of labeled samples using various algorithms for `Text` data set.

l	50 (3.3%)	100 (6.7%)	150 (10%)
k-NN	71.6	72.3	74.5
SGT	73.1	**77.0**	**78.1**
LapRLS	71.2	74.2	76.2
SQ-Loss-I	**74.1**	76.8	76.6
MP	73.0	75.4	77.9
LapRF ($m = 1$)	69.4	73.5	77.2
TVRF ($m = 1$)	71.3	75.2	77.9
LapRF ($m = 2$)	73.4	77.4	79.0
TVRF ($m = 2$)	**74.5**	**78.1**	**79.4**

These examples show that the combination of total variation (TV) and the region force compare favorably to existing methods. In the calculation of p_k, the choice of $m = 1$ has similar or better performance than that of $m = 2$, except for the `Text` data set.

5. Conclusions

In this paper we presented two graph based algorithms for clustering high-dimensional data given a few portion of training samples. The first algorithm

is based on minimizing a convex functional combining the Rayleigh quotient for the graph Laplacian and a region-force term, subject to a simplex feasible domain. The second algorithm is similar to the first one, except that the Rayleigh quotient for the graph Laplacian is replaced by the total variation of the labeling function defined on the graph [4, 13, 15]. These two algorithms are related to the spectral clustering algorithms in the use of the graph Laplacian and the concept of minimizing the graph cut, although without the need of computing the eigenfunctions of the graph Laplacian as in related methods such as [13,15]. They are also inspired by the celebrated Chan-Vese model in the use of the region force. However the region force proposed in this paper is novel, because there is no need of computing the centroids of the heuristic regions. The new region force characterized the probability of each unlabeled points belonging to each cluster, conditioned on the given labeled points. The conditional probability is calculated based on some average diffusion distance to the labeled points of the same cluster.

The k-nearest-neighbor (k-NN) graph structure is used throughout the discussion. It is based on the assumption that the sub-graph of each cluster can be embedded onto a low-dimensional smooth manifold. However, our algorithm can run without on this assumption. We also note that the sparsity of the graph contributes to the linear scaling of our algorithm.

The feasible domain is a unit K-simplex, characterizing the probability of each point belonging to each of the K clusters. In the end each point is assigned to the cluster with the largest probability. This avoids doing one-versus-all clustering for multiple times.

The numerical tests on several popular benchmark data sets demonstrate the promise of our algorithm: it is competitive with some state-of-the-art methods, especially for TVRF algorithm when the rate of labeled data is low. In all this tests, very little data-dependent features are explored, but only the simplest kernel functions based on the Euclidean distance or cosine similarity are used. We note that the need of parameter tuning is minimal in our algorithm. In the numerical examples presented in this paper, only the number of neighbors k (as in k-NN) and the parameter τ balancing the edge force and region force are slightly different from case to case.

Labelling techniques related to spectral clustering through graph Laplacian is still widely used in the industry. Tests given in this work show that replacing the Laplacian term by the corresponding TV energy improve the accuracy and in some cases rather substantial. The minimization problem related to the Laplacian is quadratic and thus easy to solve. However, through some proper primal-dual approaches, the costs for solving the nonlinear TV model is not much higher than solving the quadratic Laplacian models.

References

[1] Silvia Bonettini and Valeria Ruggiero. On the Convergence of Primal Dual Hybrid Gradient Algorithms for Total Variation Image Restoration. *Journal of Mathematical Imaging and Vision*, 44(3):236–253, January 2012.

[2] Yuri Boykov and Gareth Funka-Lea. Graph Cuts and Efficient N-D Image Segmentation. *International Journal of Computer Vision*, 70(2):109–131, November 2006.

[3] Xavier Bresson, Thomas Laurent, David Uminsky, and James von Brecht. Multiclass total variation clustering. In *Advances in Neural Information Processing Systems*, pages 1421–1429, 2013.

[4] Xavier Bresson, Xue-Cheng Tai, Tony F. Chan, and Arthur Szlam. Multi-class Transductive Learning Based on l1 Relaxations of Cheeger Cut and Mumford-Shah-Potts Model. *Journal of Mathematical Imaging and Vision*, 49(1):191–201, August 2013.

[5] Thomas Bühler and Matthias Hein. Spectral clustering based on the graph p-Laplacian. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 81–88. ACM, 2009.

[6] T.F. Chan and L.A. Vese. Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277, February 2001.

[7] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.

[8] Yunmei Chen and Xiaojing Ye. Projection onto a simplex. *arXiv preprint arXiv:1101.6081*, 2011.

[9] Fan Chung. *Spectral Graph Theory (CBMS Regional Conference Series in Mathematics, No. 92)*. American Mathematical Society, December 1996.

[10] R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *Proceedings of the National Academy of Sciences of the United States of America*, 102(21):7426–7431, May 2005.

[11] T. Cour, F. Benezit, and Jianbo Shi. Spectral segmentation with multiscale graph decomposition. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005*, volume 2, pages 1124–1131 vol. 2, June 2005.

[12] Yu-Hong Dai and Roger Fletcher. Projected Barzilai-Borwein methods for large-scale box-constrained quadratic programming. *Numerische Mathematik*, 100(1):21–47, March 2005.

[13] C. Garcia-Cardona, E. Merkurjev, A.L. Bertozzi, A. Flenner, and A.G. Percus. Multiclass Data Segmentation Using Diffuse Interface Methods on Graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1600–1613, August 2014.

[14] Guy Gilboa and Stanley Osher. Nonlocal operators with applications to image processing. *Multiscale Modeling & Simulation*, 7(3):1005–1028, 2008.

[15] Matthias Hein and Simon Setzer. Beyond spectral clustering - tight relaxations of balanced graph cuts. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 2366–2374. Curran Associates, Inc., 2011.

[16] Huiyi Hu, Justin Sunu, and Andrea L. Bertozzi. Multi-class graph Mumford-Shah model for plume detection using the MBO scheme. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 209–222. Springer, 2015.

[17] Jianbo Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, August 2000.

[18] Yan Nei Law, Hwee Kuan Lee, Michael K Ng, and Andy M Yip. A semisupervised segmentation model for collections of images. *Image Processing, IEEE Transactions on*, 21(6):2955–2968, 2012.

[19] O. Lézoray, A. Elmoataz, and V. T. Ta. 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), pages 873–876, March 2012.

[20] Stéphane Mallat. Group invariant scattering. *Communications on Pure and Applied Mathematics*, 65(10):1331–1398, 2012.

[21] Ekaterina Merkurjev, Egil Bae, Andrea L Bertozzi, and Xue-Cheng Tai. Global binary optimization on graphs for classification of high-dimensional data. *Journal of Mathematical Imaging and Vision*, 52(3):414–435, 2015.

[22] Ekaterina Merkurjev, Tijana Kostic, and Andrea L Bertozzi. An mbo scheme on graphs for classification and image processing. *SIAM Journal on Imaging Sciences*, 6(4):1903–1930, 2013.

[23] Barry Merriman, James Kenyard Bence, and Stanley Osher. *Diffusion generated motion by mean curvature*. Department of Mathematics, University of California, Los Angeles, 1992.

[24] Marius Muja and David G. Lowe. Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration. *VISAPP (1)*, 2:331–340, 2009.

[25] Andrew Y. Ng, Michael I. Jordan, Yair Weiss, and others. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 2:849–856, 2002.

[26] B. Osting, C. White, and E. Oudet. Minimal Dirichlet Energy Partitions for Graphs. *SIAM Journal on Scientific Computing*, 36(4):A1635–A1651, January 2014.

[27] Bernhard Schölkopf, Koji Tsuda, and Jean-Philippe Vert. *Kernel methods in computational biology*. MIT press, 2004.

[28] Chanop Silpa-Anan and Richard Hartley. Optimised KD-trees for fast image descriptor matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[29] Amit Singhal. Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.*, 24(4):35–43, 2001.

[30] D. Spielman and N. Srivastava. Graph Sparsification by Effective Resistances. *SIAM Journal on Computing*, 40(6):1913–1926, January 2011.

[31] Amarnag Subramanya and Jeff Bilmes. Semi-supervised learning with measure propagation. *The Journal of Machine Learning Research*, 12:3311–3370, 2011.

[32] Arthur D. Szlam, Mauro Maggioni, and Ronald R. Coifman. Regularization on graphs with function-adapted diffusion processes. *The Journal of Machine Learning Research*, 9:1711–1739, 2008.

[33] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *The Journal of Machine Learning Research*, 2:45–66, 2002.

[34] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. `http://www.vlfeat.org/`, 2008.

[35] Ke Wei, Xue-Cheng Tai, Tony F Chan, and Shingyu Leung. Primal-dual method for continuous max-flow approaches. In *Computational Vision and Medical Image Processing V: Proceedings of the 5th Eccomas Thematic Conference on Computational Vision and Medical Image Processing (VipIMAGE 2015, Tenerife, Spain, October 19-21, 2015)*, page 17. CRC Press, 2015.

[36] Qingyao Wu, Michael K Ng, and Yunming Ye. Markov-miml: A markov chain-based multi-instance multi-label learning algorithm. *Knowledge and information systems*, 37(1):83–104, 2013.

[37] Qingyao Wu, Michael K Ng, Yunming Ye, Xutao Li, Ruichao Shi, and Yan Li. Multi-label collective classification via markov chain based learning method. *Knowledge-Based Systems*, 63:1–14, 2014.

[38] Zhirong Yang, Tele Hao, Onur Dikmen, Xi Chen, and Erkki Oja. Clustering by nonnegative matrix factorization using graph random walk. In *Advances in Neural Information Processing Systems*, pages 1079–1087, 2012.

[39] S.X. Yu and J. Shi. Multiclass spectral clustering. In *Ninth IEEE International Conference on Computer Vision, 2003. Proceedings*, pages 313–319 vol.1, October 2003.

[40] Jing Yuan, Egil Bae, and Xue-Cheng Tai. A study on continuous max-flow and min-cut approaches. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2217–2224. IEEE, 2010.

[41] Jing Yuan, Egil Bae, Xue-Cheng Tai, and Yuri Boykov. A continuous max-flow approach to potts model. In *Computer Vision–ECCV 2010*, pages 379–392. Springer, 2010.

[42] Jing Yuan, Egil Bae, Xue-Cheng Tai, and Yuri Boykov. A spatially continuous max-flow and min-cut framework for binary labeling problems. *Numerische Mathematik*, 126(3):559–587, 2014.

[43] Lihi Zelnik-Manor and Pietro Perona. Self-tuning spectral clustering. In *Advances in neural information processing systems*, pages 1601–1608, 2004.

[44] Mingqiang Zhu and Tony Chan. An efficient primal-dual hybrid gradient algorithm for total variation image restoration. *UCLA CAM Report*, pages 08–34, 2008.