CLUSTERING ON DYNAMIC GRAPHS BASED ON TOTAL VARIATION

Peter Berger, Thomas Dittrich, and Gerald Matz

Institute of Telecommunications, TU Wien (Vienna, Austria)
Email: `firstname.lastname@nt.tuwien.ac.at`

ABSTRACT

We consider the problem of multiclass clustering on dynamic graphs. At each time instant, the proposed algorithm performs local updates of the clusters in regions of nodes whose cluster affiliation is uncertain and may change. These local cluster updates are carried out through semi-supervised multiclass total variation (TV) based clustering. The resulting optimization problem is shown to be directly connected to a minimum cut and thus very well suited to capture local changes in the cluster structure. We propose an ADMM based algorithm for solving the TV minimization problem. Its per iteration complexity scales linearly with the number of edges present in the local areas under change and linearly with the number of clusters. We demonstrate the usefulness of our approach by tracking several objects in a video with static background.

1. INTRODUCTION

Graph-based clustering is a powerful tool to identify communities in many real-world networks. Graphs are capable of capturing network structures in a straightforward manner. For example, in social networks the users can be represented as nodes and friendship relations can be modelled via the edges of a graph. Many networks evolve over time as users leave/join the network and relations between individual users may change. In such scenarios, the objects to be clustered are observed at different moments in time and the aim is to obtain accurate clustering results at each time instant [1, 2]. Clustering evolving networks raises new challenges and opens new possibilities. For example, previous clustering results can be used to reduce the computational complexity required for the current state [3, 4]. Furthermore, a good clustering should not only fit the current data well, but also not deviate too much from recent clustering results. This observation motivated the incorporation of temporal smoothness into clustering algorithms [5–7]. The clustering algorithm, presented in this paper, is based on the smoothness assumption that most nodes preserve their cluster membership between two consecutive time instants, cf., the temporal smoothness definition PCM in [6]. However, in contrast to [6] we do not incorporate temporal smoothness into our cost function, but rather update only the cluster labels of nodes with uncertain cluster affiliation. This leads to a significant reduction of the complexity of the clustering algorithm. The local cluster updates are performed by semi-supervised TV based clustering. The corresponding optimization problem is shown to be closely related to a minimum cut problem and therefore an excellent match for tracking local changes in the cluster structure. Furthermore, TV based clustering leads in general to much more accurate clustering results than algorithms using the Laplacian quadratic form (such as spectral clustering) [8]. Furthermore, as shown in [9] our TV based clustering algorithm can be extended to signed graphs without additional cost. Signed graphs can capture besides similarity relations also dissimilarity relations like being blocked in a social network. The inclusion of only a few dissimilarity edges can significantly improve the clustering performance [9].

Funding by WWTF Grant ICT15-119.

2. TV BASED DYNAMICAL CLUSTERING

Problem formulation

Consider a time-varying graph $\mathcal{G}_t = (\mathcal{V}, \mathbf{W}^t)$, where $\mathcal{V} = \{1, \ldots, N\}$ is the node set and $\mathbf{W}^t$ is the weighted adjacency matrix at time instant $t = 1, \ldots, T$. The non-negative entry $W_{ij}^t \geq 0$ captures the amount of similarity between node i and j. We assume that the graph is undirected, i.e., $\mathbf{W} = \mathbf{W}^T$. At each time instant t we aim for partitioning the node set $\mathcal{V}$ into K clusters $\mathcal{V}_1^t, \ldots, \mathcal{V}_K^t$ ($\bigcup_{k=1}^K \mathcal{V}_k^t = \mathcal{V}$, $\mathcal{V}_i^t \cap \mathcal{V}_j^t = \emptyset$) such that nodes are more similar within clusters than across clusters. We assume that between two consecutive time instances the graph undergoes significant changes in only a limited area and remains largely unchanged otherwise. In such scenarios it is sufficient to update the clusters only locally. As will be seen in Section 2, TV based clustering is directly connected to a minimum cut and therefore predestined for carrying out those local updates.

Semi-supervised clustering

For most nodes the cluster affiliation is maintained from the previous time instant to the next. This leads to a semi-supervised clustering problem, i.e., clustering where groups of nodes $\mathcal{L}_k \subset \mathcal{V}_k$, $k = 1, \ldots, K$, are known a priori to belong to cluster $\mathcal{V}_k$. Most semi-supervised clustering approaches determine the clusters $\mathcal{V}_1, \ldots, \mathcal{V}_K$ by solving a relaxed (and regularized) version of the minimum cut problem

$$\min_{\mathcal{V}_1,\dots,\mathcal{V}_K} \sum_{k=1}^{K} \Big(\sum_{i\in\mathcal{V}_k} \sum_{j\in\mathcal{V}\setminus\mathcal{V}_k} W_{ij} \Big) \qquad \text{s.t.} \quad \mathcal{L}_k \subset \mathcal{V}_k. \quad (1)$$

For the case of two clusters we can conveniently express the cluster affiliation in terms of a label vector $\mathbf{x} \in \{1,2\}^N$ with $x_i = 1$ if node i belongs to cluster $\mathcal{V}_1$ and $x_i = 2$ if node i belongs to cluster $\mathcal{V}_2$. Let $\mathbf{L} = \mathbf{D} - \mathbf{W}$ with $\mathbf{D} = \mathrm{diag}\{d_1,\dots,d_N\}$, $d_i = \sum_j W_{ij}$, denote the graph Laplacian. Widely used relaxations (for the case of two clusters) are to replace the graph cut in (1) with the Laplacian quadratic form $\mathbf{x}^T\mathbf{L}\mathbf{x} = \frac{1}{2}\sum_i\sum_j (x_i - x_j)^2 W_{ij}$ and the hard constraints $\mathcal{L}_k \subset \mathcal{V}_k$ with penalization terms of the form $\sum_k \sum_{i\in\mathcal{L}_k} (x_i - k)^2$ [10–14]. However, a much tighter relaxation is obtained by the use of the TV $\|\mathbf{x}\|_{\mathrm{TV}} \triangleq \sum_i\sum_j |x_i - x_j|\, W_{ij}$ instead of the Laplacian quadratic form [8].

TV based semi-supervised clustering

For more than two classes, a label vector $\mathbf{x} \in \{1,\dots,K\}^N$, together with the above definition of the TV or the Laplacian quadratic form, is no longer suitable for the approximation of the minimum cut. The reason for this is that the magnitude of $|x_i - x_j|$ and $(x_i - x_j)^2$ depends on the cluster indices the nodes belong to. This problem can be circumvented by representing the cluster affiliation by the binary indicator matrix $\mathbf{X} \in \{0,1\}^{N\times K}$ with $X_{ik} = 1$ if node i belongs to cluster k and $X_{ik} = 0$ otherwise. We denote by $\mathbf{x}_i$ the rows of $\mathbf{X}$, i.e., $\mathbf{X} = (\mathbf{x}_1^T,\dots,\mathbf{x}_N^T)^T$. The appropriate definition of the TV of a matrix $\mathbf{X}$ is then given by $\|\mathbf{X}\|_{\mathrm{TV}} = \sum_i\sum_j \|\mathbf{x}_i - \mathbf{x}_j\|_1 W_{ij}$ [8]. We cluster the node set based on the optimization problem [15, 16]

$$\min_{\mathbf{X}} \|\mathbf{X}\|_{\mathrm{TV}} \qquad \text{s.t.} \quad \mathbf{X} \in \mathcal{Q}, \qquad (2)$$

with the constraint set

$$\mathcal{Q} = \Big\{ \mathbf{X} \in [0,1]^{N\times K} : \mathbf{x}_i = \mathbf{e}_k \text{ for } i \in \mathcal{L}_k, \sum_k X_{ik} = 1 \text{ for } i = 1,\dots,N \Big\}.$$

Here, $\mathbf{e}_k$ denotes the kth standard unit vector. One can either use (preconditioned) primal dual methods (e.g., [17, 18]) or the closely related augmented ADMM [19] to calculate a minimizer $\hat{\mathbf{X}}$ of (2) [9]. Node i is then assigned to the cluster for which $\hat{X}_{ik}$ is maximal. For the case of very few labeled nodes, optimization problem (2) has to be augmented with regularization terms to prevent label sets $\mathcal{L}_k$ to be separated out as clusters [9, 15, 16]. It is well known that the total variation leads to a tighter relaxation of minimum cut problems than the Laplacian quadratic form [8]. In Theorem 1 we prove that for the case that a minimizer of (2) assigns each node to exactly one cluster (each row of $\hat{\mathbf{X}}$ has an entry greater $\frac{1}{2}$), the induced partition is a solution of the minimum cut problem (1). Related results have been recently derived in [20, 21]. A proof of Theorem 1 can be found in the Appendix.

Theorem 1. *Let* $\mathbf{W}$ *be the weighted adjacency matrix of a graph and let* $\hat{\mathbf{X}}$ *be a minimizer of the TV minimization problem* (2). *If each row of* $\hat{\mathbf{X}}$ *has an entry greater* $\frac{1}{2}$, *then the partition* $(\mathcal{V}_k = \{i : \hat{X}_{ik} > \hat{X}_{il} \text{ for all } l \neq k\})_{k=1}^K$ *of* $\mathcal{V}$ *is a solution of the minimum cut problem* (1).

Local cluster updates

We are now in the position to formulate Algorithm 1, updating the clusters locally based on information of the current and previous time instant. Step 1 of Algorithm 1 determines all edges which have changed significantly from time instant $t-1$ to t and step 2 determines the set $\mathcal{V}_c$ of nodes adjacent to those edges. The number of operations required for step 1 scales linearly with $|\mathcal{E}^t \cup \mathcal{E}^{t-1}|$, where $\mathcal{E}^t$ denotes the edge set of the graph at time instant t (for $(i,j) \notin \mathcal{E}^t \cup \mathcal{E}^{t-1}$ we have $W_{ij}^{t-1} - W_{ij}^t = 0$). Step 3 determines the sets $\mathcal{A}_k$ of nodes whose cluster assignment is possibly incorrect. Here we used the sets of nodes whose attraction to its assigned cluster $\mathcal{V}_k^{(t-1)}$ is only slightly higher than the attraction to one of the other clusters. E.g., in the case $\delta = 0$, this yields the nodes that are in the topological boundary of one of the clusters. However, any other selection of the set of uncertain nodes is possible. Step 4 defines the set $\mathcal{U}$ of unlabelled points as the union of the sets $\mathcal{A}_k$ and $\mathcal{V}_c$, and step 5 defines the sets $\mathcal{L}_k$ of the labelled points of cluster $\mathcal{V}_k^t$. In the final step 6, the new clusters are determined by updating the unlabeled points $\mathcal{U}$. Since all labelled points are kept constant, the per-iteration complexity of the augmented ADMM solving (2) scales linearly with the number of edges adjacent to an unlabelled point $|\cup_{i\in\mathcal{U}} \cup_{j\in\{m:W_{im}>0\}} \{(i,j)\}|$ and the number of clusters K, cf. [9]. We point to the fact that both, the removal and the insertion of graph nodes can be handled by our algorithm without additional effort. Inserted nodes just have to be treated as unlabelled points. The disappearance of a cluster also does not effect our algorithm. However, the appearance of an additional cluster can not be handled with the current version of our algorithm. For the time instant a new cluster appears another (unsupervised) clustering algorithm has to be used. Thereafter, one can continue updating according to Algorithm 1.

3. EXPERIMENTS

To verify this approach to be valid we use it to track the position of two balls in a video in front of a static background. For that, each frame of the video is transformed into a graph by treating each pixel as a node and connecting it with its neighbours. The weight of the edges is defined by the Gaussian kernel

$$W_{i,j} = \begin{cases} \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma_x^2} - \frac{\|I_i - I_j\|_2^2}{2\sigma_I^2}\right), & \|\mathbf{x}_i - \mathbf{x}_j\|_2 \leq r, \\ 0, & \text{otherwise,} \end{cases}$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ are the positions of the i-th and j-th pixel and I_1 and I_2 are the grayscale intensities. The parameters

Algorithm 1 TV based cluster update

Input: $\mathbf{W}^{t-1}$, $\mathbf{W}^{t}$, $\mathcal{V}_1^{t-1}, \dots, \mathcal{V}_K^{t-1}$, $\varepsilon \geq 0$, $0 \leq \delta < 1$

1: $\mathcal{E}_c = \{(i,j) : |W_{ij}^{t-1} - W_{ij}^{t}| > \varepsilon\}$
2: $\mathcal{V}_c = \bigcup_{(i,j)\in\mathcal{E}_c} \{i,j\}$
3: $\mathcal{A}_k = \big\{ i \in \mathcal{V}_k^{t-1} : \exists l \text{ with } \delta \sum_{j\in\mathcal{V}_k} W_{i,j}^{t-1} < \sum_{j\in\mathcal{V}_l} W_{i,j}^{t-1} \big\}$ for $k = 1,\dots,K$
4: $\mathcal{U} = \bigcup_k \mathcal{A}_k \cup \mathcal{V}_c$
5: $\mathcal{L}_k = (\mathcal{V}\backslash\mathcal{U}) \cap \mathcal{V}_k^{t-1}$ for $k = 1,\dots,K$
6: Determine the new clusters $\mathcal{V}_1^t, \dots, \mathcal{V}_K^t$ via (2) using $\mathcal{L}_1, \dots, \mathcal{L}_K$ as label sets and $\mathbf{W}^t$ for the TV

Output: $\mathcal{V}_1^t, \dots, \mathcal{V}_K^t$

σ_x, σ_I and r can be used to set the influence of the distance, the grayscale intensities, and the maximum distance of pixels that are connected, respectively. In our setup of the algorithm we used $\sigma_x = 5$, $\sigma_I = 35$, and $r = \sqrt{2}$ and for the parameters in step 2 and step 3 of Algorithm 1 we set $\varepsilon = 0.5$ and $\delta = 0$. $\delta = 0$ gives a selection of possibly incorrect pixels which is precisely the union of topological boundaries of all clusters. As initial value for the labels, we manually assigned a label to each pixel of the first frame.

Figure 1 shows the results of clustering for selected frames of a video in which two balls collide. The first column of images shows the starting position (top) of the balls and the manually generated cluster assignment (bottom), the second image in this column is completely black as there were no pixels for which a cluster assignment was calculated. In the second column it can be seen that during a collision the clusters do not join to a single cluster. The reason for this can be seen in the image of updated pixels: For both balls there are black spots in its interior, which means that there are also samples of this cluster. In this frame there also appears a tail at one of the balls which is a false result that appears because the individual frames are blurry and so there are no sharp edges that can be detected. Due to the selection of the pixels that are updated in every step, this tail is corrected in the following steps. In frame 12, directly after the collision, it disconnects from the ball until it vanishes completely in frame 22.

In all images in the second row there are several small white spots which mean that the graph changed in this area. Indeed the intensitiy of each pixel is disturbed by noise. Together with $\varepsilon = 0.5$ it is sufficient that the distance of the intensities of two neighboring pixels changes from 0 to 7 to cause a significant change in the graph. But as those spots do not have interior points, they converge immediately to the surrounding cluster and do not yield false results.

4. CONCLUSIONS

In this paper, we considered the problem of clustering on dynamic graphs. We presented a low complexity ADMM based algorithm performing only local cluster updates via semi-supervised TV minimization. A direct connection of the optimization problem to a minimum cut was derived. In our numerical experiments, we demonstrated the usefulness of our approach via tracking objects in a video with static background.

Appendix: Proof of Theorem 1

Let $\hat{\mathbf{X}}$ be a minimizer of (2) with each row having an entry greater than $\frac{1}{2}$. For matrices $\mathbf{X} \in \{0,1\}^{N\times K}$ the induced clusters $\mathcal{V}_k \triangleq \{i : X_{ik} = 1\}$ fulfil

$$\|\mathbf{X}\|_{\mathrm{TV}} = \sum_{k=1}^{K} \Big(\sum_{i\in\mathcal{V}_k} \sum_{j\in\mathcal{V}\backslash\mathcal{V}_k} W_{ij} \Big). \tag{3}$$

Consequently the minimum cut problem (1) is equivalent to the TV minimization problem (2) restricted to matrices $\mathbf{X} \in \{0,1\}^{N\times K}$. We now show that the matrix $\mathbf{Y}$ given by

$$Y_{ik} = \begin{cases} 1, & \text{if } \hat{X}_{ik} > \hat{X}_{il} \text{ for all } l \neq k, \\ 0, & \text{else,} \end{cases} \tag{4}$$

has the same TV as $\hat{\mathbf{X}}$, and is therefore also a minimizer of (2). Since (1) is equivalent to (2) for matrices $\mathbf{X} \in \{0,1\}^{N\times K}$, the clusters induced by $\mathbf{Y}$ minimize (1) and the theorem holds true.

Let $\mathbf{S} \in \mathbb{R}^{N\times K}$ be given by $\mathbf{S} = \mathbf{Y} - \hat{\mathbf{X}}$. Then, for $\varepsilon > 0$,

$$\|\hat{\mathbf{X}} \pm \varepsilon\mathbf{S}\|_{\mathrm{TV}} = \sum_i \sum_j \sum_k |\hat{X}_{ik} - \hat{X}_{jk} \pm \varepsilon(S_{ik} - S_{jk})| W_{ij}. \tag{5}$$

We set $A_{ij}^k = \hat{X}_{ik} - \hat{X}_{jk}$ and $B_{ij}^k = S_{ik} - S_{jk}$. If $|A_{ij}^k| \geq \varepsilon|B_{ij}^k|$, then

$$|A_{ij}^k \pm \varepsilon B_{ij}^k| = \begin{cases} |A_{ij}^k| \pm \varepsilon|B_{ij}^k|, & A_{ij}^k B_{ij}^k > 0, \\ |A_{ij}^k| \mp \varepsilon|B_{ij}^k|, & A_{ij}^k B_{ij}^k < 0. \end{cases}$$

This in combination with (5) implies that

$$\|\hat{\mathbf{X}} \pm \varepsilon\mathbf{S}\|_{\mathrm{TV}} = \|\hat{\mathbf{X}}\|_{\mathrm{TV}} \pm \varepsilon C \tag{6}$$

for some constant $C \in \mathbb{R}$. If $\hat{X}_{ik} > 0$ and $\hat{X}_{jk} > 0$ (or $\hat{X}_{ik} < 0$ and $\hat{X}_{jk} < 0$), then $S_{ik} = 1 - \hat{X}_{ik}$ and $S_{jk} = 1 - \hat{X}_{jk}$ (or $S_{ik} = -1 - \hat{X}_{ik}$ and $S_{jk} = -1 - \hat{X}_{jk}$). Consequently $|S_{ik} - S_{jk}| = |\hat{X}_{ik} - \hat{X}_{jk}|$ and $|A_{ij}^k| \geq \varepsilon|B_{ij}^k|$ holds true for $\varepsilon = 1$. If $\hat{X}_{ik} < 0$ and $\hat{X}_{jk} > 0$ (or vice versa), then $|\hat{X}_{ik} - \hat{X}_{jk}| > 0$ and $|A_{ij}^k| \geq \varepsilon|B_{ij}^k|$ holds true for $\varepsilon \leq \frac{|A_{ij}^k|}{|B_{ij}^k|}$. We further note that $\hat{\mathbf{X}} + \varepsilon\mathbf{S} \in \mathcal{Q}$ for $\varepsilon \in [-1,1]$. Since $\hat{\mathbf{X}}$ has minimum TV the constant C in (6) has to be zero (otherwise

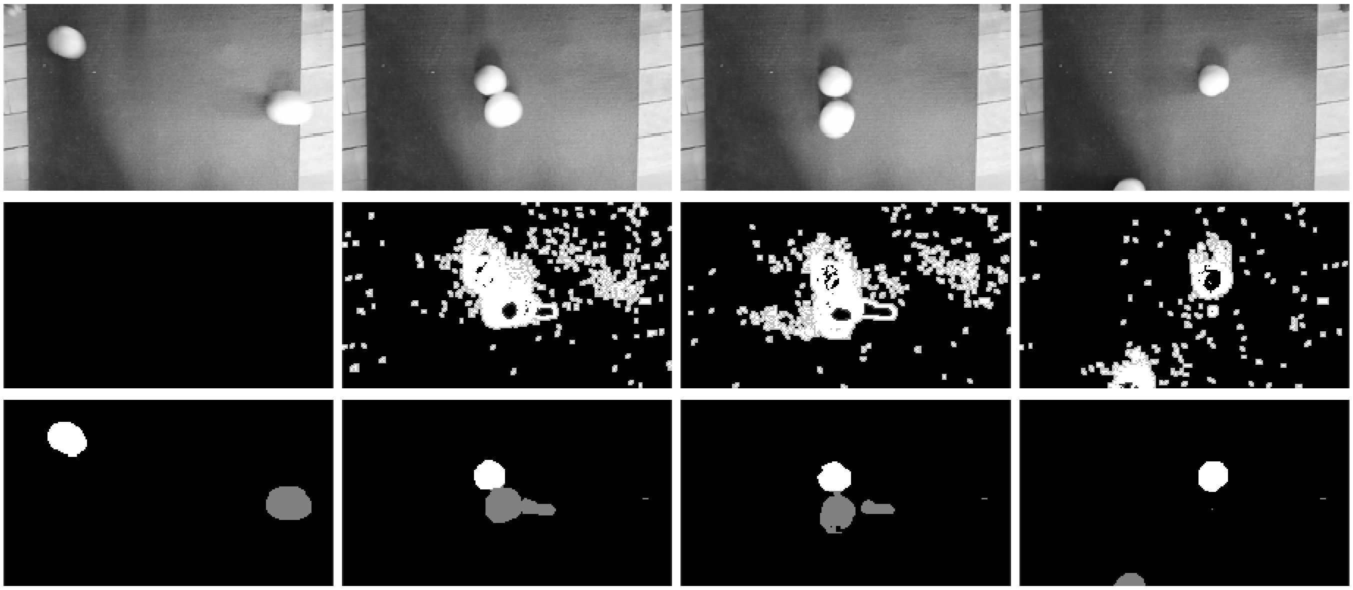

Fig. 1: Comparison of frames of the original video and the results of clustering. This comparison shows frame number 1, 11, 12 and 22 from left to right, respectively. The rows of images show the original frames, the pixels that have been updated in white and the constant pixels in black, and the three resulting clusters in white, grey, and black from top to bottom, respectively.

either $\|\hat{\mathbf{X}}+\varepsilon\mathbf{S}\|_{\mathrm{TV}} < \|\hat{\mathbf{X}}\|_{\mathrm{TV}}$ or $\|\hat{\mathbf{X}}-\varepsilon\mathbf{S}\|_{\mathrm{TV}} < \|\hat{\mathbf{X}}\|_{\mathrm{TV}}$) and we can add the matrix $\varepsilon\mathbf{S}$ to the matrix $\hat{\mathbf{X}}$ without increasing the TV. By successive addition of multiples of $\mathbf{S}$ to $\hat{\mathbf{X}}$ we derive that $\|\mathbf{Y}\|_{\mathrm{TV}} = \|\hat{\mathbf{X}}\|_{\mathrm{TV}}$.

REFERENCES

[1] T. Hartmann, A. Kappes, and D. Wagner. *Clustering Evolving Networks*, pages 280–329. Springer International Publishing, Cham, 2016.

[2] S. Fortunato and D. Hric. Community Detection in Networks: A User Guide. *Physics Reports*, 659:1 – 44, Nov. 2016.

[3] H. Ning, W. Xu, Y. Chi, Y. Gong, and T. S. Huang. Incremental Spectral Clustering by Efficiently Updating the Eigen-System. *Pattern Recogn.*, 43(1):113–127, Jan. 2010.

[4] R. Görke, T. Hartmann, and Do. Wagner. Dynamic Graph Clustering using Minimum-Cut Trees. In *Algorithms and Data Structures*, pages 339–350, Banff, Alberta, Canada, Aug. 2009.

[5] D. Chakrabarti, R. Kumar, and A. Tomkins. Evolutionary Clustering. In *Proc. 12th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pages 554–560, Philadelphia, PA, USA, Aug. 2006.

[6] Y. Chi, X. Song, D. Zhou, K. Hino, and B. L. Tseng. Evolutionary Spectral Clustering by Incorporating Temporal Smoothness. In *Proc. 13th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, pages 153–162, San Jose, California, USA, 2007.

[7] Y. R. Lin, Y. Chi, S. Zhu, H. Sundaram, and B. L. Tseng. FacetNet: A Framework for Analyzing Communities and Their Evolutions in Dynamic Networks. In *Proc. 17th Int. Conf. on World Wide Web 2008*, pages 685–694, Beijing, China, April 2008.

[8] X. Bresson, T. Laurent, D. Uminsky, and J. von Brecht. Multiclass Total Variation Clustering. In *Advances in Neural Information Processing Systems 26*, pages 1421–1429, Lake Tahoe (NV), Dec. 2013.

[9] P. Berger, T. Dittrich, G. Hannak, and G. Matz. Semi-Supervised Multiclass Clustering Based on Signed Total Variation. In *ICASSP*, pages 4953–4957, May 2019.

[10] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-Supervised Learning using Gaussian Fields and Harmonic Functions. In *Proc. Int. Conf. Machine Learning*, pages 912–919, Washington, DC, USA, Aug. 2003.

[11] X. Wang and I. Davidson. Flexible Constrained Spectral Clustering. In *Proc. 16th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, pages 563–572, Washington, DC, USA, July 2010.

[12] V. Sindhwani, P. Niyogi, and M. Belkin. Beyond the Point Cloud: From Transductive to Semi-Supervised Learning. In *Proc. Int. Conf. Machine Learning*, pages 824–831, Bonn, Germany, Aug. 2005.

[13] U. von Luxburg. A Tutorial on Spectral Clustering. *Statistics and Computing*, 17(4):395–416, Dec. 2007.

[14] W. Liu, J. Wang, and S. Chang. Robust and Scalable Graph-Based Semisupervised Learning. *Proc. IEEE*, 100(9):2624–2638, Sept. 2012.

[15] K. Yin, Tai XC., and S. J. Osher. An Effective Region Force for Some Variational Models for Learning and Clustering. *Technical report, UCLA*, 2016.

[16] K. Yin and XC. Tai. An Effective Region Force for Some Variational Models for Learning and Clustering. *J. Scientific Computing*, 74(1):175–196, Jan. 2018.

[17] T. Pock and A. Chambolle. Diagonal Preconditioning for First Order Primal-Dual Algorithms in Convex Optimization. In *Proc. Int. Conf. Computer Vision*, pages 1762–1769, Barcelona (Spain), Nov. 2011.

[18] P. Berger, G. Hannak, and G. Matz. Graph Signal Recovery via Primal-Dual Algorithms for Total Variation Minimization. *IEEE J. Sel. Topics in Signal Processing*, 11(6):842–855, Sept. 2017.

[19] Y. Zhu. An Augmented ADMM Algorithm with Application to the Generalized Lasso Problem. *J. Comput. Graph. Statist.*, 26(1):195–204, Feb. 2017.

[20] E. Merkurjev, E. Bae, A. L. Bertozzi, and X.-C. Tai. Global Binary Optimization on Graphs for Classification of High-Dimensional Data. *J. Math. Imaging and Vision*, 52(3):414–435, July 2015.

[21] E. Bae and E. Merkurjev. Convex Variational Methods on Graphs for Multiclass Segmentation of High-Dimensional Data and Point Clouds. *J. Math. Imaging and Vision*, 58(3):468–493, July 2017.