
This is an electronic reprint of the original article.
This reprint may differ from the original in pagination and typographic detail.

Jung, Alexander; Sarcheshmehpour, Yasmin
Local Graph Clustering with Network Lasso

Published in:
IEEE Signal Processing Letters

DOI:
[10.1109/LSP.2020.3045832](https://doi.org/10.1109/LSP.2020.3045832)

Published: 01/01/2021

Document Version
Publisher's PDF, also known as Version of record

Published under the following license:
CC BY

Please cite the original version:
Jung, A., & Sarcheshmehpour, Y. (2021). Local Graph Clustering with Network Lasso. *IEEE Signal Processing Letters*, 28, 106-110. [9298875]. <https://doi.org/10.1109/LSP.2020.3045832>

This material is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

Local Graph Clustering With Network Lasso

Alexander Jung , *Member, IEEE*, and Yasmin SarcheshmehPour

Abstract—We study the statistical and computational properties of a network Lasso method for local graph clustering. The clusters delivered by nLasso can be characterized elegantly via network flows between cluster boundaries and seed nodes. While spectral clustering methods are guided by a minimization of the graph Laplacian quadratic form, nLasso minimizes the total variation of cluster indicator signals. As demonstrated theoretically and numerically, nLasso methods can handle very sparse clusters (chain-like) which are difficult for spectral clustering. We also verify that a primal-dual method for non-smooth optimization allows to approximate nLasso solutions with optimal worst-case convergence rate.

Index Terms—Clustering algorithms, distributed algorithms, flow graphs, machine learning, network theory (graphs), semisupervised learning.

I. INTRODUCTION

MANY application domains generate network structured data. Networked data arises in the study of self-organizing systems constituted by individual agents who can interact [24], [25]. Networked data also arises in computer vision where nodes represent individual pixels that are connected if they are close-by. Metrological observations collected by spatially distributed stations form a network of time series with edges connecting close-by stations. We represent networked data conveniently using an “empirical” or “similarity” graph \mathcal{G} [5], [33].

The analysis of networked data is often facilitated by grouping or clustering the data points into coherent subsets of data points. Clustering methods aim at finding subsets (clusters) of data points that are more similar to each other than to the remaining data points. Most existing clustering methods are unsupervised as they do not require the true cluster assignments for any data point [20]–[23], [26], [33].

Local graph clustering starts from “seed nodes” and explores their neighbourhoods to find clusters around them [30], [32]. With a runtime depending only on the resulting clusters, these methods are attractive for massive graphs [9], [30].

Spectral clustering methods use the eigenvectors of the graph Laplacian matrix to approximate the indicator function of each cluster [1], [6], [26], [29], [33]. These methods are computationally attractive as they can be implemented as scalable message

passing [8], [35]. Our approach also results in a message passing implementation but uses a different approximation of the cluster indicators.

To approximate cluster indicators, we use the solutions of a particular instance of the network Lasso (nLasso) optimization problem [11]. We solve this nLasso clustering problem using an efficient primal-dual method. This primal-dual method has attractive convergence guarantees and can be implemented as a scalable message passing (see Section IV).

Building on our recent work on the duality between TV minimization and network flow optimization [17], [18], we show that the proposed nLasso clustering method can be interpreted (in a precise sense) as a flow-based clustering method [19], [31], [32], [34]. As detailed in Section III, our nLasso problem (and its dual flow optimization problem) is similar but different from the TV minimization problems (and its dual flow optimization problems) studied in [17], [18].

Compared with spectral methods, flow-based methods (including our approach) better handle sparsely connected (chain-like) clusters (see Section VI) and are more robust to “structural heterogeneities” [13], [34]. In contrast to existing flow-based local clustering methods, our approach is based on efficient convex optimization methods instead of computationally expensive combinatorial algorithms.

This paper makes the following contributions:

- Section II formulates local graph clustering as a particular instance of the nLasso problem.
- In Section III we derive the dual problem of the nLasso. We provide an interpretation of this dual problem as an instance of network flow optimization.
- Section IV presents a local clustering method by applying a primal-dual method to the nLasso problem. This method is appealing for big data applications as it can be implemented as a scalable message-passing method.
- Section V characterizes the clusters delivered by nLasso in terms of the amount of flow that can be routed from cluster boundaries to the seed nodes within that cluster. This offers a novel link between flow-based clustering and convex optimization.

II. LOCAL GRAPH CLUSTERING

We consider networked data which is represented by a simple undirected weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$. The nodes $\mathcal{V} = \{1, \dots, n\}$ represent individual data points. Undirected edges $e = \{i, j\} \in \mathcal{E}$ connect similar data points $i, j \in \mathcal{V}$ and are assigned a positive weight $W_{i,j} > 0$. Absence of an edge between nodes $i, j \in \mathcal{V}$ implies $W_{i,j} = 0$. The neighbourhood of a node $i \in \mathcal{V}$ is $\mathcal{N}_i := \{j \in \mathcal{V} : \{i, j\} \in \mathcal{E}\}$.

It will be convenient to define a directed version of the graph \mathcal{G} by replacing each undirected edge $\{i, j\}$ with the directed edge $(\min\{i, j\}, \max\{i, j\})$. We overload notation and use \mathcal{G} to denote the undirected and directed version of the empirical

Manuscript received September 30, 2020; revised December 4, 2020; accepted December 15, 2020. Date of publication December 18, 2020; date of current version January 18, 2021. This work was supported in part by the Academy of Finland (AoF) under Grant 331197. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Z. Jane Wang. (*Corresponding author: Alexander Jung.*)

Alexander Jung is with the Department of Computer Science, Aalto University, 02150 Espoo, Finland (e-mail: alexander.jung@aalto.fi).

Yasmin SarcheshmehPour is with the Department of Mathematical Sciences, Sharif University of Technology, Tehran 11365-11155, Iran (e-mail: sarcheshmehpours@gmail.com).

This letter has supplementary downloadable material available at <https://doi.org/10.1109/LSP.2020.3045832>, provided by the authors.

Digital Object Identifier 10.1109/LSP.2020.3045832

graph. The directed neighbourhoods of a node $i \in \mathcal{V}$ are $\mathcal{N}_i^+ := \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$, and $\mathcal{N}_i^- := \{j \in \mathcal{V} : (j, i) \in \mathcal{E}\}$. (1)

Local graph clustering starts from a given set of seed nodes $\mathcal{S} = \{i_1, \dots, i_{|\mathcal{S}|}\} \subset \mathcal{V}$. (2)

The seed nodes might be obtained by exploiting domain knowledge and are grouped into batches \mathcal{S}_k ,

$$\mathcal{S} = \mathcal{S}_1 \cup \dots \cup \mathcal{S}_F. \quad (3)$$

Each batch contains L_k seed nodes of the same cluster \mathcal{C}_k .

We allow the number of seed nodes to be a vanishing fraction of the entire graph. This is an extreme case of semi-supervised learning where the labeling ratio (viewing seed nodes as labeled data points) goes to zero.

The proposed local graph clustering method (see Section 3) operates by exploring the neighbourhoods of the seed nodes \mathcal{S} . It constructs clusters \mathcal{C}_k around the seed nodes \mathcal{S}_k such that only few edges leave the cluster \mathcal{C}_k .

We characterize a cluster \mathcal{C}_k via its boundary

$$\partial\mathcal{C}_k := \{(i, j) \in \mathcal{E} : i \in \mathcal{C}_k, j \notin \mathcal{C}_k\}. \quad (4)$$

A good cluster \mathcal{C}_k is such that the total weight of the edges in its boundary $\partial\mathcal{C}_k$ is small. We make this characterization more precise in Section V using network flows to quantify the connectivity between cluster boundary and seed nodes.

III. NETWORK LASSO AND ITS DUAL

Local graph clustering methods learn graph signals $\hat{\mathbf{x}} \in \mathbb{R}^{\mathcal{V}}$ that are good approximations to the indicator signals $\mathbf{x}^{(k)} = (x_1^{(k)}, \dots, x_n^{(k)})^T \in \mathbb{R}^{\mathcal{V}}$. These indicator signals represent the clusters $\mathcal{C}_k \subseteq \mathcal{V}$ around the seed nodes \mathcal{S}_k (see (3)) via

$$x_i^{(k)} = 1 \text{ for nodes } i \in \mathcal{C}_k \text{ and } x_i^{(k)} = 0 \text{ otherwise.} \quad (5)$$

Spectral graph clustering uses eigenvectors of the graph Laplacian matrix to approximate cluster indicator signals. In contrast, we use TV minimization to learn approximations $\hat{\mathbf{x}}$ to the cluster indicators $\mathbf{x}^{(k)}$, for $k = 1, \dots, F$.

We have recently explored the relation between network flow problems and TV minimization [17], [18]. Loosely speaking, the solution of TV minimization is piece-wise constant over clusters whose boundaries have a small total weight. This property motivates us to learn the indicator function for the cluster \mathcal{C}_k around the seed nodes \mathcal{S}_k by solving

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x} \in \mathbb{R}^{\mathcal{V}}} \sum_{i \in \mathcal{S}_k} (x_i - 1)^2 / 2 + \sum_{i \notin \mathcal{S}_k} \alpha x_i^2 / 2 + \lambda \|\mathbf{x}\|_{\text{TV}}. \quad (6)$$

Here, we used the total variation (TV)

$$\|\mathbf{x}\|_{\text{TV}} = \sum_{\{i, j\} \in \mathcal{E}} W_{i, j} |x_i - x_j|. \quad (7)$$

The convex optimization problem (6) is a special case of the nLasso problem [11]. The solution to (6) is unique for any $\alpha > 0$ since then the objective function is strictly convex [2].

We solve a separate nLasso problem (6) for each batch \mathcal{S}_k , for $k = 1, \dots, F$, of seed nodes in the same cluster. The nodes $i \notin \mathcal{S}_k$ which are not seed nodes for \mathcal{C}_k belong to one of two groups. One group $\mathcal{C}_k \setminus \mathcal{S}_k$ of nodes which belong to \mathcal{C}_k and the other group of nodes $i \notin \mathcal{C}_k$ outside the cluster.

The special case of (6) when $\alpha = 0$ is studied in [17]. We can also interpret (6) as TV minimization using soft constraints instead of hard constraints [18]. While [18] enforces \hat{x}_i for each

seed node $i \in \mathcal{S}_k$, (6) uses soft constraints such that typically $\hat{x}_i < 1$ at seed nodes $i \in \mathcal{S}_k$. Spectral methods use optimization problems similar to (6) but with the Laplacian quadratic form $\sum_{\{i, j\} \in \mathcal{E}} W_{i, j} (x_i - x_j)^2$ instead of TV (7).

We hope that any solution to (6) is a good approximation to the indicator function $x^{(k)}$ of a well-connected subset around the seed nodes \mathcal{S}_k . We use the graph signal $\hat{x} : i \mapsto \hat{x}_i$ obtained from solving (6) to determine a reasonable cluster $\mathcal{C}_k \supseteq \mathcal{S}_k$.

The idea of determining clusters via learning graph signals as (approximations of) indicator functions of good clusters is also underlying spectral clustering [33]. Instead of TV minimization underlying nLasso (6), spectral clustering uses the matrix Laplacian to score candidates for cluster indicator functions. Moreover, spectral clustering methods do not require any seed nodes with known cluster assignment.

The choice of the tuning parameters α and λ in (6) crucially influence the behaviour of the clustering method and the properties of clusters delivered by (6). Their choice can be based on the intuition provided by a minimum cost flow problem that is dual (equivalent) to nLasso (6). This minimum cost flow problem is not defined directly on the empirical graph \mathcal{G} but the augmented graph $\tilde{\mathcal{G}} = (\tilde{\mathcal{V}}, \tilde{\mathcal{E}})$. This augmented graph is obtained by augmenting the graph \mathcal{G} with an additional node “ \star ” and edges (i, \star) for each node $i \in \mathcal{V}$.

As detailed in the supplementary material, the nLasso (6) is equivalent (dual) to the minimum cost flow problem [17], [18]

$$\min_{y \in \mathbb{R}^{\mathcal{E}}} \sum_{i \in \mathcal{S}_k} (y_{(i, \star)} - 1)^2 + (1/\alpha) \sum_{i \notin \mathcal{S}_k} y_{(i, \star)}^2 \quad (8)$$

$$\text{s.t. } \sum_{j \in \mathcal{N}_i^+} y_{(i, j)} = \sum_{j \in \mathcal{N}_i^-} y_{(j, i)} \text{ for all nodes } i \in \tilde{\mathcal{V}} \quad (9)$$

$$|y_e| \leq \lambda W_e \text{ for all } e \in \mathcal{E}. \quad (10)$$

The constraints (9) enforce conservation of the flow y_e at every node $i \in \tilde{\mathcal{V}}$. The constraints (10) enforce the flow y_e not exceeding the edge capacity λW_e . There are no capacity constraints for augmented edges (i, \star) with $i \in \mathcal{V}$.

The node signal $\hat{\mathbf{x}}$ solves (6) and the edge signal $\hat{\mathbf{y}}$ solves (8), respectively, if and only if [28, Ch. 31]

$$-\sum_{j \in \mathcal{N}_i^+} \hat{y}_{(i, j)} + \sum_{j \in \mathcal{N}_i^-} \hat{y}_{(j, i)} = \hat{x}_i - 1 \text{ for } i \in \mathcal{S}_k \quad (11)$$

$$-\sum_{j \in \mathcal{N}_i^+} \hat{y}_{(i, j)} + \sum_{j \in \mathcal{N}_i^-} \hat{y}_{(j, i)} = \alpha \hat{x}_i \text{ for } i \notin \mathcal{S}_k \quad (12)$$

$$|\hat{y}_e| \leq \lambda W_e \text{ for all edges } e \in \mathcal{E} \quad (13)$$

$$\hat{x}_i - \hat{x}_j = 0 \text{ for } e = (i, j) \in \mathcal{E} \text{ with } |\hat{y}_{(i, j)}| < \lambda W_e. \quad (14)$$

We can interpret conditions (11), (12) as conservation laws satisfied by any flow \hat{y}_e that solves the nLasso dual (8). We can think of injecting (extracting) a flow of value $\hat{x}_i - 1$ at seed nodes $i \in \mathcal{S}_k$. The nodes $i \notin \mathcal{S}_k$ are leaking a flow of value $\alpha \hat{x}_i$. The optimal flow \hat{y}_e has to provide these demands while respecting the capacity constraints (13).

We illustrate the conditions (11)–(14) in Fig. 1 for a simple chain graph. According to (14), the nLasso solution $\hat{\mathbf{x}}$ can only change across edges $e = (i, j)$ which are saturated $|\hat{y}_e| = \lambda W_e$. For a chain graph, using a suitable choice for α and λ in (6),

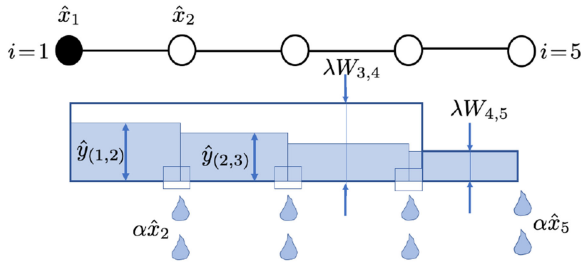


Fig. 1. The dual (8) of nLasso optimizes the flow through a leaky network obtained from the empirical graph \mathcal{G} .

nLasso recovers the cluster structure as soon as the weights of boundary edges exceed the weights of intra-cluster edges.

We use the optimality conditions (11)–(14) to characterize the solutions of nLasso (6) in Section V. Combining this characterization with generative models for the clusters \mathcal{C}_k , such as stochastic block models, allows to derive sufficient conditions on the parameters of the generative model such that solutions of (6) allow to recover the true underlying local clusters [15].

IV. COMPUTATIONAL ASPECTS

The necessary and sufficient conditions (11)–(14) characterize any pair of solutions for (6) and its dual (8). We can find solutions to the conditions (11)–(14), which provides a solution to nLasso in turn, by reformulating those coupled conditions as a fixed point equation.

There are many different fixed-point equations that are equivalent to the optimality conditions (11)–(14). We will use a particular construction which results in a method that is guaranteed to converge to a solution of (6) and (8) and can be implemented as a scalable message passing on the empirical graph \mathcal{G} . This construction is discussed in great detail in [3] and has been applied to the special case of nLasso (6) for $\alpha = 0$ in our recent work [17]. For local graph clustering, we need $\alpha > 0$ to force the solutions of (6) to decay towards zero outside the local cluster around the seed nodes \mathcal{S}_k .

Applying tools from [17], we obtain the following updates generating two sequences $\hat{x}_i^{(r)}$ and $\hat{y}_e^{(r)}$, for $r = 0, 1, \dots$, converging to solutions of (6) and (8), respectively.

$$\tilde{x}_i := 2\hat{x}_i^{(r)} - \hat{x}_i^{(r-1)} \text{ for } i \in \mathcal{V} \quad (15)$$

$$\hat{y}_e^{(r+1)} := \hat{y}_e^{(r)} + (1/2)(\tilde{x}_i - \tilde{x}_j) \text{ for } e = (i, j) \in \mathcal{E} \quad (16)$$

$$\hat{y}_e^{(r+1)} := \hat{y}_e^{(k+1)} / \max\{1, |\hat{y}_e^{(r+1)}| / (\lambda W_e)\} \text{ for } (i, j) \in \mathcal{E} \quad (17)$$

$$\hat{x}_i^{(r+1)} := \hat{x}_i^{(r)} - \gamma_i \left[\sum_{j \in \mathcal{N}_i^+} \hat{y}_{(i,j)}^{(r+1)} - \sum_{j \in \mathcal{N}_i^-} \hat{y}_{(j,i)}^{(r+1)} \right] \text{ for } i \in \mathcal{V} \quad (18)$$

$$\hat{x}_i^{(r+1)} := \left(\gamma_i + \hat{x}_i^{(r+1)} \right) / (\gamma_i + 1) \text{ for every } i \in \mathcal{S}_k \quad (19)$$

$$\hat{x}_i^{(r+1)} := \hat{x}_i^{(r+1)} / (\alpha \gamma_i + 1) \text{ for every } i \in \mathcal{V} \setminus \mathcal{S}_k. \quad (20)$$

Here, $\gamma_i = 1/d_i$ is the inverse of the node degree $d_i = |\mathcal{N}_i|$. Starting from an arbitrary initialization $\hat{x}_i^{(0)}$ and $\hat{y}_e^{(0)}$, the iterates $\hat{x}_i^{(r)}$ and $\hat{y}_e^{(r)}$ converge to a solution of nLasso (6) and its dual (8), respectively [12].

The updates (15)–(20) define a message-passing on the empirical graph \mathcal{G} to jointly solve nLasso (6) and its dual (8). The computational complexity of one full iteration of (15)–(20) is proportional to the number of edges in the empirical graph. The overall complexity also depends on the number of iterations required to ensure the iterate $\hat{x}_i^{(r)}$ being sufficiently close to the nLasso (6) solution.

Basic analysis of proximal methods shows that the number of required iterations scales inversely with the desired sub-optimality of $\hat{x}_i^{(r)}$ (see [4], [7]). This convergence rate cannot be improved for chain graphs [14]. For a fixed number of iterations and empirical graphs with bounded maximum node degree, the computational complexity of our method scales linearly with the number of nodes (data points). This ensures scalability of our method for applications involving massive networked data.

We now develop an interpretation of the updates (15)–(20) as an iterative method for network flow optimization. The update (17) enforces the capacity constraints (10) to be satisfied for the flow iterates $\hat{y}_e^{(r)}$. The update (18) amounts to adjusting the current nLasso estimate $\hat{x}_i^{(r)}$, for each node $i \in \mathcal{V}$ by the demand induced by the current flow approximation $\hat{y}_e^{(r)}$.

Together with the updates (19) and (20), the update (18) enforces the flow $\hat{y}_e^{(r)}$ to satisfy the conservation laws (11) and (12). The update (16) aims at enforcing (14) by adjusting the cumulated demands $\hat{x}_i^{(r)}$ via the flow $\hat{y}_{(i,j)}^{(r)}$ through an edge $e = (i, j) \in \mathcal{E}$ according to the difference $(\tilde{x}_i - \tilde{x}_j)$.

The above interpretation helps to guide the choice for the parameters α and λ in (6). The edge capacities λW_e limit the rate by which the values $\hat{x}_i^{(r)}$ can be “build up”. Choosing λ too small would, therefore, slow down the convergence of $\hat{x}_i^{(r)}$. On the other hand, using nLasso (6) with too large λ does not allow to detect small local clusters \mathcal{C}_k (see Section V).

V. CLUSTER CHARACTERIZATION

We use the solution \hat{x}_i of nLasso (6) to approximate the indicator of a local cluster around the seed nodes \mathcal{S}_k . The cluster delivered by our method is obtained by thresholding,

$$\mathcal{C}_k := \{i \in \mathcal{V} : \hat{x}_i > 1/2\}. \quad (21)$$

In practice, we replace the exact nLasso solution \hat{x}_i in (21) with the iterate $\hat{x}_i^{(r)}$ obtained after a sufficient number r of primal-dual updates (15)–(20) (see Section VI). The threshold $1/2$ in (21) is somewhat arbitrary. Our theoretical results can be easily adapted for other choices for the threshold. The question if there exists an optimal choice for the threshold and what this actually means precisely is beyond the scope of this paper.

Our main theoretical result is a necessary condition on the cluster (21) and the nLasso parameters α and λ (see (6)).

Proposition 1: Consider the cluster (21) obtained from the nLasso solution. Then, if $\mathcal{S}_k \subseteq \mathcal{C}_k$,

$$\lambda \sum_{e \in \partial \mathcal{C}_k} W_e \leq 1 - (\alpha/2) \sum_{i \in \mathcal{C}_k \setminus \mathcal{S}_k} \hat{x}_i^{(r)}. \quad (22)$$

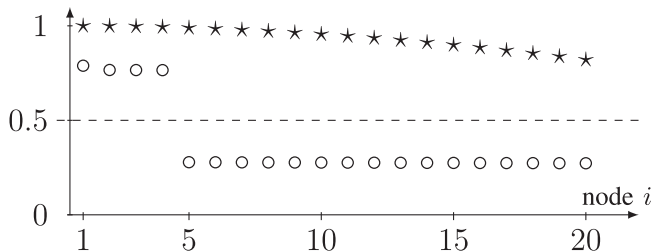


Fig. 2. Signal (“o”) obtained after 1000 iterations of (15)–(20) for the chain graph \mathcal{G}_c . The resulting local cluster is $\mathcal{C}_1 = \{1, 2, 3, 4\}$ (see (21)). We also depict the graph signal (“*”) obtained by the eigenvector of the normalized graph Laplacian corresponding to the smallest non-zero eigenvalue.

and

$$\lambda \sum_{e \in \partial \mathcal{C}_k} W_e \leq \alpha \sum_{i \notin \mathcal{C}_k} \hat{x}_i^{(r)}. \quad (23)$$

Proof: Follows from the optimality conditions (11)–(14). ■

The necessary conditions (22) and (23) can guide the choice of the parameters α and λ in (6). We can enforce nLasso to deliver clusters with small boundary $\partial \mathcal{C}_k$ by using a large λ in (6). Since the left side of (22) must not exceed the right hand side, using a large λ enforces a cluster (21) such that $\sum_{e \in \partial \mathcal{C}_k} W_e$ is small. In the extreme case of very large λ , this leads to $\partial \mathcal{C}_k$ being empty. There is a critical value for λ in (6) beyond which the cluster \mathcal{C}_k (21) contains all connected components with seed nodes \mathcal{S}_k .

We can combine (23) with an upper bound U on the number of nodes $i \notin \mathcal{C}_k$ reached by message-passing updates (15)–(20). Inserting this bound U on the number of “relevant” nodes $i \notin \mathcal{C}_k$ into (23), yields the necessary condition

$$\lambda \sum_{e \in \partial \mathcal{C}_k} W_e \leq U \alpha / 2. \quad (24)$$

VI. NUMERICAL EXPERIMENTS

We verify Proposition 1 numerically on a chain graph \mathcal{G}_c with nodes $\mathcal{V} = \{1, \dots, 100\}$. Consecutive nodes i and $i + 1$ are connected by edges of weight $W_e = 5/4$ with the exception of edge $e' = \{4, 5\}$ with the weight $W_{e'} = 1$.

We determine a cluster \mathcal{C}_1 around seed node $i = 1$ using (21). The updates (15)–(20) are iterated for a fixed number of $K = 1000$ iterations. The nLasso parameters were set to $\lambda = 2/10$ and $\alpha = 1/200$ (see (6)). These parameter values ensure conditions (22) and (24) (with $U = 80$) are satisfied for the resulting cluster is $\mathcal{C}_1 = \{1, 2, 3, 4\}$.

We depict the resulting graph signal $\hat{x}_i^{(K)}$ (“o”) for the first 20 nodes of \mathcal{G}_c in Fig. 2. We also show the (scaled) eigenvector (“*”) of the graph Laplacian corresponding to the smallest non-zero eigenvalue. This eigenvector is known as the Fiedler vector and used by spectral graph clustering methods to approximate the cluster indicators [27]. According to Fig. 2, the (approximate) nLasso solution allows to recover the true cluster $\mathcal{C}_1 = \{1, 2, 3, 4\}$ via thresholding the signal values at $1/2$. In contrast, spectral graph clustering fails here as the Fiedler vector is not well localized on the true cluster.

A second experiment compares our method with state-of-the-art local clustering methods for an image segmentation task [10]. We represent a greyscale image as a grid graph whose nodes are individual pixels. Vertically and horizontally adjacent pixels are

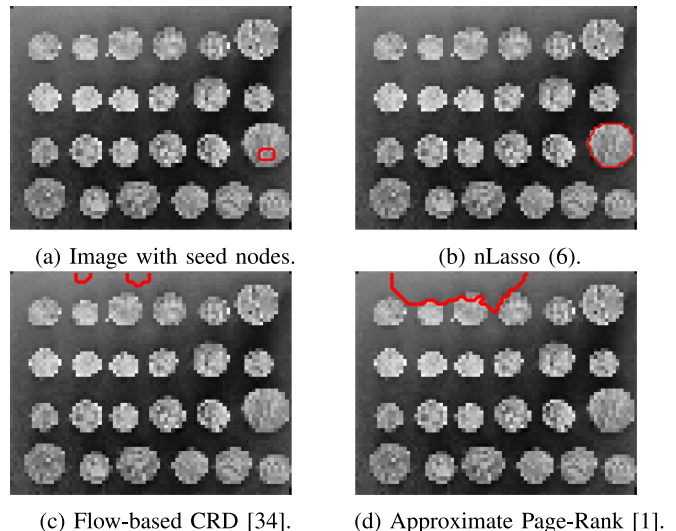


Fig. 3. Image segmentation results.

connected by edges with weight $W_{i,j} = \exp(-(g_i - g_j)^2/20^2)$ with the greyscale value $g_i \in \{0, \dots, 255\}$ of the i -th pixel.

We determine a local cluster around a set of seed nodes (see Fig. 3(a)) using $K = 1000$ iterations of (15)–(20) to approximately solve nLasso (6) (see Fig. 3(b)). The local cluster obtained by the flow-based capacity releasing diffusion (CRD) method [34] is depicted in Fig. 3(c). The local clustering obtained by the spectral method presented in [1] is shown in Fig. 3(d). The seed nodes and resulting clusters obtained by the three methods are enclosed by a red contour line in Fig. 3. Only our method is able to accurately determine the pixels belonging to the foreground object (a coin) around the seed nodes.

A third experiment compares our method with existing clustering methods for an empirical graph being the realization of a partially labeled stochastic block model (SBM). We used a SBM with two blocks or clusters \mathcal{C}_1 and \mathcal{C}_2 . Each cluster consists of 10000 nodes. A randomly chosen pair of nodes is connected by an edge with probability $1/5$ ($1/100$) if they belong to the same block (different blocks). The cluster (21) delivered by nLasso (6), with $\alpha = 1/10$ and $\lambda = 1/1000$ and using 1000 randomly chosen seed nodes (10%), perfectly recovered the true clusters. The spectral method [1] achieved labeling accuracy (fraction of correctly labeled nodes) of $1/2$. The flow-based methods [19], [34] achieved a labeling accuracy of around $9/10$. The source code for the above experiments can be found at <https://github.com/alexjungaalto/>.

VII. CONCLUSION

We have studied the application of nLasso to local graph clustering. Our main technical result is a characterization of the nLasso solutions in terms of network flows between cluster boundaries and seed nodes. Conceptually, we provide an interesting link between flow-based clustering and non-smooth convex optimization. This work offers several avenues for follow-up research. We have recently proposed networked exponential families to model networks of local datasets [16]. It would be interesting to study how the statistical properties of local datasets could be leveraged for clustering methods applied to such networked data.

REFERENCES

- [1] R. Andersen, F. Chung, and K. Lang, "Local graph partitioning using pagerank vectors," in *Proc. 47th Annu. IEEE Symp. Found. Comput. Sci.*, Oct. 2006, pp. 475–486.
- [2] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
- [3] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *J. Math. Imag. Vis.*, vol. 40, pp. 120–145, 2011.
- [4] A. Chambolle and T. Pock, "An introduction to continuous optimization for imaging," *Acta Numerica*, vol. 25, pp. 161–319, 2016.
- [5] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-Supervised Learning*. Cambridge, MA, USA: MIT Press, 2006.
- [6] F. R. K. Chung, "Spectral graph theory," in *Regional Conference Series in Mathematics*, vol. 92, 1997, doi: [10.1090/cbms/092](https://doi.org/10.1090/cbms/092).
- [7] P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," in *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, vol. 49, H. Bauschke, R. Burachik, P. Combettes, V. Elser, D. Luke, and H. Wolkowicz, Eds., New York, NY, USA: Springer, 2011.
- [8] A. G. Dimakis, S. Kar, J. M. F. Moura, M. G. Rabbat, and A. Scaglione, "Gossip algorithms for distributed signal processing," *Proc. IEEE*, vol. 98, no. 11, pp. 1847–1864, Nov. 2010.
- [9] K. Fountoulakis, D. F. Gleich, and M. W. Mahoney, "A short introduction to local graph clustering methods and software," in *Proc. 7th Int. Conf. Compl. Netw. App.*, Dec. 2018, pp. 56–58.
- [10] K. Fountoulakis, M. Liu, D. Gleich, and Michael Mahoney, Local graph clustering. Github repository, 2020. [Online]. Available: <https://github.com/kfoynt/LocalGraphClustering>
- [11] D. Hallac, J. Leskovec, and S. Boyd, "Network lasso: Clustering and optimization in large graphs," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2015, pp. 387–396.
- [12] B. He, Y. You, and X. Yuan, "On the convergence of primal-dual hybrid gradient algorithm," *SIAM J. Imag. Sci.*, vol. 7, no. 4, pp. 2526–2537, 2014.
- [13] L. G. S. Jeub, P. Balachandran, M. A. Porter, P. J. Mucha, and M. W. Mahoney, "Think locally, act locally: Detection of small, medium-sized, and large communities in large networks," *Phys. Rev. E*, vol. 91, no. 1, Jan. 2015, Art. no. 012821.
- [14] A. Jung, "On the complexity of sparse label propagation," *Front. Appl. Math. Stat.*, vol. 4, p. 22, Jul. 2018.
- [15] A. Jung, "Clustering in partially labeled stochastic block models via total variation minimization," in *Proc. 54th Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, USA, Nov. 2020.
- [16] A. Jung, "Networked exponential families for big data over networks," *IEEE Access*, vol. 8, pp. 202897–202909, 2020.
- [17] A. Jung, "On the duality between network flows and network lasso," *IEEE Signal Process. Lett.*, vol. 27, pp. 940–944, 2020.
- [18] A. Jung, A. O. Hero, A. Mara, S. Jahromi, A. Heimowitz, and Y. C. Eldar, "Semi-supervised learning in network-structured data via total variation minimization," *IEEE Trans. Signal Process.*, vol. 67, no. 24, pp. 6256–6269, Dec. 2019.
- [19] K. Lang and S. Rao, "A flow-based method for improving the expansion or conductance of graph cuts," in *Integer Programming and Combinatorial Optimization*, D. Bienstock and G. Nemhauser, Eds., Berlin, Germany: Springer, 2004, pp. 325–337.
- [20] H. Li, Z. Bu, Z. Wang, and J. Cao, "Dynamical clustering in electronic commerce systems via optimization and leadership expansion," *IEEE Trans. Ind. Inform.*, vol. 16, no. 8, pp. 5327–5334, Aug. 2020.
- [21] H.-J. Li, Z. Bu, Z. Wang, J. Cao, and Y. Shi, "Enhance the performance of network computation by a tunable weighting strategy," *IEEE Trans. Emerg. Topics. Comput. Intell.*, vol. 2, no. 3, pp. 214–223, Jun. 2018.
- [22] H.-J. Li and J. J. Daniels, "Social significance of community structure: Statistical view," *Phys. Rev. E*, vol. 91, no. 1, Jan. 2015, Art. no. 012801.
- [23] H.-J. Li and L. Wang, "Multi-scale asynchronous belief percolation model on multiplex networks," *New J. Phys.*, vol. 21, Jan. 2019, Art. no. 015005.
- [24] H.-J. Li, Q. Wang, S. Liu, and J. Hu, "Exploring the trust management mechanism in self-organizing complex network based on game theory," *Physica A: Statist. Mech. App.*, vol. 542, Mar. 2020, Art. no. 123514.
- [25] M. E. J. Newman, *Networks: An Introduction*. London, U.K.: Oxford Univ. Press, 2010.
- [26] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. 14th Int. Conf. Neural Inf. Process. Syst.: Natural Synthetic*, Jan. 2001, pp. 849–856.
- [27] P. Orponen and S. E. Schaeffer, "Local clustering of large graphs by approximate fiedler vectors," in *Proc. 4th Int. Conf. Exp. Efficient Algorithms*, 2005, pp. 524–533.
- [28] R. T. Rockafellar, *Convex Analysis*. Princeton, NJ, USA: Princeton Univ. Press, 1970.
- [29] D. Spielman, "Spectral graph theory," in *Combinatorial Scientific Computing*, U. Naumann and O. Schenk, Eds., New York, NY, USA: Chapman & Hall/CRC Press, 2012.
- [30] D. A. Spielman and S.-H. Teng, "A local clustering algorithm for massive graphs and its application to nearly-linear time graph partitioning," *SIAM J. Comput.*, vol. 42, no. 1, pp. 1–26, Jan. 2013.
- [31] N. Veldt, D. F. Gleich, and M. W. Mahoney, "A simple and strongly-local flow-based method for cut improvement," in *Proc. 33rd Int. Conf. Mach. Learn.*, 2016, vol. 48, pp. 1938–1947.
- [32] N. Veldt, C. Klymko, and D. F. Gleich, "Flow-based local graph clustering with better seed set inclusion," in *Proc. SIAM Int. Conf. Data Mining*, May 2019, pp. 378–386.
- [33] U. von Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 17, no. 4, pp. 395–416, Dec. 2007.
- [34] D. Wang, K. Fountoulakis, M. Henzinger, M. W. Mahoney, and S. Rao, "Capacity releasing diffusion for speed and locality," in *Proc. ICML*, 2017, pp. 3598–3607.
- [35] L. Xiao, S. Boyd, and S.-J. Kim, "Distributed average consensus with least-mean-square deviation," *J. Parallel Distrib. Comput.*, vol. 67, no. 1, pp. 33–46, 2007.