Lagrangian Constrained Community Detection

Mohadeseh Ganji **James Bailey** **Peter J. Stuckey**
School of Computing and Information Systems, University of Melbourne, Australia
sghasempour@student.unimelb.edu.au, {baileyj,pstuckey}@unimelb.edu.au

Abstract

Semi-supervised or constrained community detection incorporates side information to find communities of interest in complex networks. The supervision is often represented as constraints such as known labels and pairwise constraints. Existing constrained community detection approaches often fail to fully benefit from the available side information. This results in poor performance for scenarios such as: when the constraints are required to be fully satisfied, when there is a high confidence about the correctness of the supervision information, and in situations where the side information is expensive or hard to achieve and is only available in a limited amount. In this paper, we propose a new constrained community detection algorithm based on Lagrangian multipliers to incorporate and fully satisfy the instance level supervision constraints. Our proposed algorithm can more fully utilise available side information and find better quality solutions. Our experiments on real and synthetic data sets show our proposed LagCCD algorithm outperforms existing algorithms in terms of solution quality, ability to satisfy the constraints and noise resistance.

Introduction

Community detection is an important task in knowledge discovery which aims to identify densely connected subgraphs or communities in networks. Many physical, social and biological networks exhibit community structure which is fundamental for analyzing such complex systems (Lancichinetti and Fortunato 2009). A community can be defined as a group of vertices where there are more edges inside the community than edges linking vertices of the community with the rest of the network (Fortunato 2010). As community detection is mainly deployed as an unsupervised task in knowledge discovery, communities are found by analyzing just the topological structure of the network. However, in some applications of community detection, there exists other information available about the true communities or desired assignments of vertices, which can help the community detection process to achieve better quality results and noise resistance (Eaton and Mansbach 2012). The side information may be obtained as a result of complex experiments or from expert knowledge in the domain. Some constraints may also be imposed on the system due to known natural or resource limitations.

Incorporating side information has been studied in constrained clustering schemes such as constrained k-means adaptations (Ganji, Bailey, and Stuckey 2016; Wagstaff et al. 2001; Pelleg and Baras 2007). However, one *cannot* directly apply the k-means type constrained clustering algorithms for a constrained community detection problem without proper embedding of the network into Euclidean space. However, choice of a proper embedding technique for semi-supervised applications without compromising on the network topological information is a challenge. To the best of our knowledge, this approach is not used in constrained community detection literature.

In recent years, some semi-supervised approaches have been devised for community detection to incorporate known labels and pairwise instance level supervision constraints. Some directly modify the network adjacency matrix based on the supervision information (Zhang 2013; Ma et al. 2010). In another scheme (Eaton and Mansbach 2012), a variation of so called modularity measure (Newman and Girvan 2004) is proposed to incorporate supervision constraints. Some approaches also have been proposed based on NMF, non-negative matrix factorization, (Ma et al. 2010; Li et al. 2017) which consider a fixed penalty term for violating the supervision constraints.

However, these approaches have some drawbacks and limitations: First, they often have little control over the constraint satisfaction rate and may fail to satisfy many of the constraints. This is mainly because the existing approaches don't have any mechanism to differentiate between easy and difficult constraints. Some constraints are easy to satisfy even using an unsupervised scheme, because the constraint aligns with the similarity measure or objective of the corresponding algorithm. However, some constraints are difficult (eg. those which cannot be satisfied by an unsupervised scheme). The existing approaches assume the same level of difficulty for all of the supervision constraints and they don't have any adaptation mechanism to deal with more difficult constraints. Consequently they often leave such constraints unsatisfied. Second, existing semi-supervised community detection algorithms often require a large amount of side information to be able to achieve a significant im-

provement in the solution quality in comparison to the unsupervised case. This is not always feasible for real world problems. Third, although some of the semi-supervised approaches are shown to be more noise resistant than the unsupervised approaches, they still can be quite sensitive to noisy links in the data.

In this paper, we propose a constrained community detection approach which is targeted to satisfy all of the constraints to fully utilise available side information and achieve high quality solutions. We next describe three example scenarios where failing to fully satisfy the constraints can lead to poor performance.

First, scenarios where the constraints are imposed on the problem by natural and physical barriers or resource limitations and the constraints are consequently strictly required to be satisfied. E.g., in a road network clustering problem where some construction makes a road unusable, a solution violating this unavailability constraint is not a feasible solution. Second, scenarios where a high degree of confidence is available about the correctness of the side information. Achieving correct supervision information often requires time, cost and effort. For example, expensive biological experiments or medical tests like biopsies. In these cases one expects the community detection algorithms to fully encode all the supervision information into the best possible solution as a return on the investment in providing the supervision. Third, for vague and difficult-to-detect communities and noisy data, supervision can make a tangible difference on the solution quality and this is only achievable by approaches which are able to fully benefit from the supervision. Hence, algorithms which cannot fully utilize the side information, often provide poor quality solutions on difficult problems.

In this paper, we propose a Lagrange multipliers based constrained community detection framework to find the maximum modularity network partition satisfying user specified constraints. We show that in contrast to existing approaches, our proposed LagCCD algorithm is able to satisfy almost all the constraints and achieve better solutions even when there is a limited amount of side information available. We also show the LagCCD outperforms existing semi-supervised community detection techniques for a number of real networks and for vague communities and noisy data.

Related Work

There has been an increasing interest in the semi-supervised community detection problem in the last decade. Some approaches such as the label propagation proposed by Silva et al. (Silva and Zhao 2012) focus on situations when the labels of a fraction of vertices are known in advance. Starting from each vertex as a community, the label propagation algorithm (Silva and Zhao 2012) merges non-labeled communities with the labeled ones with highest gain in modularity score and continue until all communities are labeled. This ordered way of labelling vertices however, doesn't necessarily results in maximum modularity. Allahverdyan et al. (Allahverdyan, Ver Steeg, and Galstyan 2010) studied the effect of known labels supervision type on the detectability threshold of communities in sparse graphs as it is known that when inter-cluster connections are dense, communities are not detectable properly.

Pairwise instance level constraints are another supervision type which indicate pairs of vertices which must be assigned to same community (*must-link* constraint) and pairs that should be assigned to different communities (*cannot-link* constraint). Yang et al. (Yang et al. 2015) incorporate must-link constraints in a semi-supervised community detection scheme which is a unified representation and generalized interpretation for spectral clustering and NMF based methods. The must-link information is encoded by adding a graph regularization term to penalize the latent space dissimilarity of corresponding pairs.

Zhang et al. (Zhang 2013) directly modify the adjacency matrix based on the available supervision constraints. This is equivalent to connecting and disconnecting edges between must-link and cannot-link pairs in the original graph. Then the modified adjacency matrix can be used in a spectral clustering, NMF or other schemes to community detection. Xiaoke Ma et al. (Ma et al. 2010) proposed a NMF based algorithm called SNMF-SS to incorporate pairwise constraints. They define violation cost matrices for must-link and cannot-link pairs and modify the similarity matrix K to $\overline{K} = K - \alpha W_{ML} + \beta W_{CL}$ where parameters α and β, the relevant importance of the two constraint types, are set in a way that the $\overline{K}$ remains non-negative.

Newman (Newman and Girvan 2004) proposed a global criterion called "modularity" for unsupervised community detection problem which soon after its introduction, became one of the most popular community detection measures (Fortunato 2010). Modularity quantifies deviations of the network from a randomized network with the assumption that there exist no community structure in random graphs. Many unsupervised community detection algorithms has been proposed based on modularity maximization so far (Newman 2006; Fortunato 2010; Inderjit S. Jutla and Mucha 2011). Modularity quantifies deviations of the network from randomness according to a null model. However, there are few approaches which apply the modularity measure in semi-supervised community detection.

Among the existing approaches, Eaton et al. (Eaton and Mansbach 2012) proposed a modularity-like measure based on a variation of Potts spin-glass model from statistical mechanics which incorporates pairwise constraints. A constraint violation cost is added to penalize both must-link and cannot-link violations. The spin-glass model can be seen as a modified modularity matrix which can be optimized using some optimization schemes.

However, none of the above semi-supervised community detection approaches is committed to fully satisfy the constraints and benefit from each and every single constraint.

In our previous work (Ganji, Bailey, and Stuckey 2017), we proposed a constraint programming framework for incorporating variety of constraint types and objective functions for constrained community detection. To the best of our knowledge, this work is the only method in the literature of constrained community detection which aims (and has the ability) to satisfy all of the supervision constraints. Although the constraint programming framework has the flexibility to

incorporate variety of constraint types including size and number of communities, pairwise and ad hoc constraints, it doesn't scale to larger graphs. When the only supervision available is limited number of pairwise must-link and cannot-link constraints, the constraint programming framework may not find a solution in a reasonable time. This paper addresses this gap to take advantage of any limited number of supervision constraints in a very short time.

Incorporating the side-information has also been studied in constrained clustering schemes such as the popular k-means type algorithms (Ganji, Bailey, and Stuckey 2016; Wagstaff et al. 2001; Pelleg and Baras 2007). However, application of such algorithms on constrained community detection is not straightforward. In order to use the k-means type clustering algorithms for community detection, one should first transform the network into Euclidean space preserving the topological information of the graph and then apply k-means type algorithms. Although in the context of constrained clustering, using graph based approaches has been investigated (e.g. for distance learning (Anand and Reddy 2011) and spectral clustering (Wang and Davidson 2010; Ng, Jordan, and Weiss 2002)), to the best of our knowledge converting the graph to Euclidean space to apply constrained clustering techniques is not used for constrained community detection in the literature.

In our previous work on constrained clustering (Ganji, Bailey, and Stuckey 2016) we proposed a Lagrangian approach for a k-means type constrained clustering in order to fully benefit from the supervision constraints. However, this current paper is different from (Ganji, Bailey, and Stuckey 2016) because i) this paper has a different objective function (e.g. maximizing the partition's modularity value) from sum-of-squares clustering ii) in clustering problems, the calculation of distances and cluster centers in Euclidean space makes it appropriate for continuous Lagrange multiplier methods. However, constrained community detection in this paper is a discrete optimization problem in which the continuous method is no longer applicable. More details on our discrete Lagrangian framework are provided next.

Proposed framework

Problem statement: Given a network $G(V, E)$ with vertices V and edges E, and a set of pairwise must-link (ML) and cannot-link (CL) constraints, it is desired to find a partition which satisfies the ML and CL constraints and has the highest possible modularity score.

To tackle this problem, the proposed LagCCD algorithm uses discrete Lagrange multipliers method to convert the constrained problem to an unconstrained one by adding penalties to the objective function for any constraint violations. The LagCCD then systematically increases the penalties to force the solution towards satisfying all of the constraints.

A discrete Lagrange multiplier method

The Lagrange multiplier technique is a well established and efficient method for solving constrained optimization problems (Bertsekas 2014) which is able to maintain numerical stability and solution accuracy (Choi, Lee, and Stuckey 2000). Application of Lagrange multiplier methods to real variable problems is more straightforward and well understood (Bertsekas 2014). One can relax the discrete variable problem to real variable problem by introducing some extra constraints to restrict the real variable to only accept integer values and then apply the continuous Lagrange multiplier method on the relaxed problem. However, handling the additional constraints usually makes the computation expensive and impractical. A better approach is the discrete Lagrange multipliers method proposed by Shang and Wah (Shang and Wah 1998; Wah and Wu 1999).

Suppose a minimization problem in the form of model (1) with discrete variable $\vec{x}$ where $\vec{x} = (x_1, x_2, ...)$ is a vector.

$$min \quad f(\vec{x}) \quad \text{subject to} \quad g_i(\vec{x}) = 0 \quad \forall i \in 1..m \qquad (1)$$

The Lagrangian objective function then can be defined similar to the continuous type as follows:

$$L(\vec{x}, \vec{\lambda}) = f(\vec{x}) + \sum_{i=1}^{m} \lambda_i g_i(\vec{x}) \qquad (2)$$

Then one can find the optimum of the original constrained minimization problem (1) by finding a saddle point to the Lagrangian function $L(\vec{x}, \vec{\lambda})$. This relationship is based on discrete saddle point theorem restated as follows.

Theorem 1: Discrete saddle point theorem: (Choi, Lee, and Stuckey 2000; Shang and Wah 1998; Wah and Wu 1999)

A vector of integer variable $\vec{x}$ is the minimum of the constrained minimization problem (1) where for all $i \in 1..m$, $g(\vec{x})$ is non-negative for all possible values of $\vec{x}$ if and only if there exist Lagrange multipliers $\vec{\lambda}^*$ such that $(\vec{x}^*, \vec{\lambda}^*)$ is the saddle point of the Lagrangian function (2)

Similar to the continues space the saddle point of the discrete Lagrange function can be defined as follows.

Definition 1: $(\vec{x}^*, \vec{\lambda}^*)$is a discrete saddle point of the Lagrange function (2) if for all $\vec{x}$ in neighborhood of $\vec{x}^*$, and all possible $\vec{\lambda}$:

$$L(\vec{x}^*, \vec{\lambda}) \leqslant L(\vec{x}^*, \vec{\lambda}^*) \leqslant L(\vec{x}, \vec{\lambda}^*) \qquad (3)$$

According to Definition 1, the saddle point $(\vec{x}^*, \vec{\lambda}^*)$ of the Lagrangian function $L(\vec{x}, \vec{\lambda})$ is the minimum of $L(\vec{x}, \vec{\lambda})$ in x–space and a maximum of $L(\vec{x}, \vec{\lambda})$ in λ–space. However, because there are no differentiation in discrete space, none of the calculus in continuous space is applicable. Hence a local search is performed based on discrete gradient descent to find a saddle point of the Lagrangian function. Based on the understanding of gradients in continuous space, they actually define directions in a small neighborhood in which function values decreases. Let $N_i(\vec{x})$ be the neighborhood of a point $\vec{x}$ along the ith direction. Then, the ith component of discrete gradient Δ is defined as $L(\vec{x}, \vec{\lambda}) - L(\vec{x'}, \vec{\lambda})$ where $\vec{x'} \in N_i(\vec{x})$ and $L(\vec{x'}, \vec{\lambda}) \leqslant L(\vec{x''}, \vec{\lambda})$ for all $\vec{x''} \in N_i(\vec{x})$. In other words, the ith component of discrete gradient returns the greatest drop in the Lagrangian function along the ith direction. The gradient descent function GD then returns a vector for updating $\vec{x}$ according to the discrete gradient $\Delta_{\vec{x}}$. There is no unique GD function and it can also depend on the current position $(\vec{x}, \vec{\lambda})$ and the current iteration t.

Lagrangian constrained community detection (LagCCD)

Modeling (f, L): We first explain how to model the semi-supervised community detection's objective function using Lagrange multipliers method. Suppose the network is represented by adjacency matrix A where $A_{ij} = 1$ if there is an edge between i and j and $A_{ij} = 0$ otherwise. Let k_i be the degree of vertex i. The modularity of a partition then can be calculated according to Equation (4) where n is the number of vertices and W is the modularity matrix (Newman and Girvan 2004) which quantifies the deviations of the network from randomness. Elements of the modularity matrix W are equal to $A_{ij} - \frac{k_i k_j}{2|E|}$. Modularity of a partition is calculated by summation over modularity values between pairs of the same community using the 0,1 variable x. x_{ij} is equal to 1 when i and j are assigned to same community and 0 otherwise.

$$Q(\vec{x}) = \frac{1}{2|E|} \sum_{i=1}^{n} \sum_{j=1}^{n} W_{ij} x_{ij} \tag{4}$$

In order to be consistent to the previous section, we turn the modularity maximization to a minimization problem by changing the objective from $Q(\vec{x})$ to $f(\vec{x}) = -Q(\vec{x})$. We also ignore the constant normalization factor $1/2|E|$ for simplicity. The modularity-based semi-supervised community detection in presence of must-link and cannot-link constraints can be represented by (5).

$$\begin{aligned} min \quad & f(\vec{x}) = -\sum_{i=1}^{n} \sum_{j=1}^{n} W_{ij} x_{ij} \\ & \text{subject to} \\ & 1 - x_{ij} = 0 \qquad \forall (i,j) \in ML \\ & x_{ij} = 0 \qquad \forall (i,j) \in CL \\ & x_{ij} \in \{0,1\} \end{aligned} \tag{5}$$

The Lagrangian function (6) then adds the penalty terms correspond to violated constraints to the original objective function.

$$\begin{aligned} & L(\vec{x}, \vec{\lambda}, \vec{\mu}) = \\ & -\sum_{i=1}^{n} \sum_{j=1}^{n} W_{ij} x_{ij} + \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{(i,j) \in ML} \lambda_{(i,j)} P_{ij}^{ML} (1 - x_{ij}) \\ & + \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{(i,j) \in CL} \mu_{(i,j)} P_{ij}^{CL} x_{ij} \end{aligned} \tag{6}$$

In the Lagrangian function (6), P^{ML} and P^{CL} are the penalty matrices whose elements are zero except for pairs (i, j) corresponding to must-link and cannot-link constraints which have a positive penalty. The penalty value can be set according to the importance of each constraint or based on the degree of confidence on correctness of them. It can also be set as a proportion of the modularity value or any other similarity score between pairs which implies violating a constraint has a higher penalty when the supervision is align with the similarity score of the pair. The penalties will affect the objective value just when a constraint is violated. Note that an ML constraint is violated when the corresponding pair (i, j) are not in same community which is encoded by $1 - x_{ij}$. In the Lagrangian function (6), λ, μ are the Lagrange multipliers correspond to the must-link and cannot-link constraints. These multipliers lead the algorithm toward satisfying all the constraints by systematically increasing the penalty over violated constraints. Note that if the constraints are all satisfied then $L(\vec{x}, \vec{\lambda}, \vec{\mu}) = f(\vec{x})$.

Corollary 1: A matrix of integer variable $\vec{x}$ is the optimum of the semi-supervised modularity optimization problem (5) if and only if there exist Lagrange multipliers $\vec{\lambda}^*$ and $\vec{\mu}^*$ such that $(\vec{x}^*, \vec{\lambda}^*, \vec{\mu}^*)$ is the saddle point of the Lagrangian function (6).

Proof: Since both $g^{ML}(\vec{x}) = 1 - x_{ij}$ and $g^{CL}(\vec{x}) = x_{ij}$ in problem (5) are non-negative for all possible values of $\vec{x}$ and the penalty matrices P_{ij}^{ML} and P_{ij}^{CL} are also non-negative, Theorem 1 is applicable. Hence, the optimum of the problem (5) is equal to the saddle point of its corresponding Lagrange function (6), if one exists.

Next we present a theorem for finding a saddle point of the Lagrangian function also showing this function can be represented as a modularity-like objective.

Theorem 2: A saddle point of the Lagrangian function $L(\vec{x}, \vec{\lambda}, \vec{\mu})$ is a matrix of integer variable $\vec{x}$ and Lagrange multipliers $\vec{\lambda}$ and $\vec{\mu}$ that satisfies all the below three conditions:

$$\Delta_x \widetilde{L}(\vec{x}, \vec{\lambda}, \vec{\mu}) = \Delta_x (-\sum_{i=1}^{n} \sum_{j=1}^{n} W_{ij}^{c} x_{ij}) = 0 \tag{7}$$

$$g^{ML}(x) = 0 \tag{8}$$

$$g^{CL}(x) = 0 \tag{9}$$

Where given $\oplus$ and $\ominus$ as element-wise matrix operations, W^c is defined as:

$$W^c = W \oplus \sum_{(i,j) \in ML} \lambda_{(i,j)} P^{ML} \ominus \sum_{(i,j) \in CL} \mu_{(i,j)} P^{CL} \tag{10}$$

Proof: The proof contains two parts:

Firs we show the $\Delta_x \widetilde{L}(\vec{x}, \vec{\lambda}, \vec{\mu})$ is equal to $\Delta_x L(\vec{x}, \vec{\lambda}, \vec{\mu})$ where Δ is a discrete space operator. By expanding the Lagrangian function of equation (6) and rewriting based on the terms including $\vec{x}$ we reach to equation:

$$\begin{aligned} & L(\vec{x}, \vec{\lambda}, \vec{\mu}) = \\ & -\sum_{i=1}^{n} \sum_{j=1}^{n} \left(W_{ij} + \sum_{(i,j) \in ML} \lambda_{(i,j)} P_{ij}^{ML} - \sum_{(i,j) \in CL} \mu_{(i,j)} P_{ij}^{CL} \right) x_{ij} \\ & + \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{(i,j) \in ML} \lambda_{(i,j)} P_{ij}^{ML} \end{aligned} \tag{11}$$

Given a Lagrange multiplier λ, the second part of the Lagrangian function (11) is a constant term with zero effect on $\Delta_x L(\vec{x}, \vec{\lambda}, \vec{\mu})$ and can be ignored. The remaining part of

(11) is equal to $\widetilde{L}(\vec{x}, \vec{\lambda}, \vec{\mu})$ given the W^c described in equation (10).

We prove the two sides of the theorem as follows:

" $\Rightarrow$ " side: Given a saddle point $(\vec{x}^*, \vec{\lambda}^*, \vec{\mu}^*)$ we prove it satisfies the three conditions. (7) is true from the definition of saddle point which L cannot be improved in $N(\vec{x}^*)$. Hence $\Delta_x L(\vec{x}, \vec{\lambda}, \vec{\mu})$ is equal to zero which means $\Delta_x \widetilde{L}(\vec{x}, \vec{\lambda}, \vec{\mu})$ is also equal to zero. Conditions (8) and (9) must be true as the constraints must be satisfied at any solution point.

" $\Leftarrow$ " side: Given a solution $(\vec{x}^*, \vec{\lambda}^*, \vec{\mu}^*)$ to equations (7), (8) and (9), we prove it to be a discrete saddle point of the Lagrange function (6). Because condition (7) holds, and $\Delta_x \widetilde{L}(\vec{x}, \vec{\lambda}, \vec{\mu}) = \Delta_x L(\vec{x}, \vec{\lambda}, \vec{\mu})$ then $\Delta_x L(\vec{x}^*, \vec{\lambda}^*, \vec{\mu}^*) = 0$ which means no drop in L can be found in neighbourhood of $\vec{x}^*$. Hence, $L(\vec{x}^*, \vec{\lambda}^*, \vec{\mu}^*) \leqslant L(\vec{x}, \vec{\lambda}^*, \vec{\mu}^*)$. Because $g^{ML}(x^*) = 0$ and $g^{CL}(x^*) = 0$ according to conditions (8) and (9), then $L(\vec{x}^*, \vec{\lambda}^*, \vec{\mu}^*) \geqslant L(\vec{x}^*, \vec{\lambda}, \vec{\mu})$. So $(\vec{x}^*, \vec{\lambda}^*, \vec{\mu}^*)$ is a saddle point to the Lagrange function.

According to Theorem 2, a gradient descent local search is devised to find a saddle point of the Lagrangian function (6).

Gradiend descent and updating conditions (GD,$U_{\vec{\lambda}}$): According to condition (7) in Theorem 2, the function $\widetilde{L}(\vec{x}, \vec{\lambda}, \vec{\mu})$ can be considered in GD function for updating variable $\vec{x}$. Recall that discrete gradient Δ is a local search mechanism to find the direction with largest drop in the function. Starting from the initial partition of each vertex as a community, the discrete gradient function evaluates all possible merges of a vertex with other vertices and records the best merge with largest drop in $\widetilde{L}(\vec{x}, \vec{\lambda}, \vec{\mu})$ to input to the GD function. GD function then according to the best merges found by Δ_x, merges the corresponding vertices iteratively until no more merges can improve the value of the Lagrange function. The solution found at this stage is an update of variable $\vec{x}$. The Lagrange multipliers are then updated based on the constraint violations. If a constraint is violated, the corresponding Lagrange multiplier increases by a factor α.

Variable initialization ($I_{\vec{\lambda}}$,$I_{\vec{\mu}}$,$I_{\vec{x}}$): Because the update of Lagrange multipliers is non-decreasing, generally, any non-negative value can be used to initialize the Lagrange multipliers. In this work, we initialize all the Lagrange multipliers to zero. We also initialize the variable $\vec{x}$ using an unconstrained community detection solution.

The input to LagCCD are the modularity matrix, ML and CL constraint sets, and an additional stopping criterion. The pseudo-code of the LagCCD algorithm is shown in Figure 1. After initializing the variables and setting the penalty terms (lines 1-5), the main loop (lines 6-22) continues updating the variable $\vec{x}$ and the Lagrange multipliers $\vec{\lambda}$ and $\vec{\mu}$ and keep tracking of the best solution found until all constraints are satisfied or a maximum iteration threshold is reached. In the while loop, first the variable $\vec{x}^t$ is updated according to the specified GD function (line 8) which basically finds a new partition of the network according to $\widetilde{L}(x^t, \vec{\lambda}^t, \vec{\mu}^t)$ which is based on W^{c^t}. Then the Lagrange multipliers $\vec{\lambda}^t$ and $\vec{\mu}^t$ and the violation set $Viol$ are updated according to the $\vec{x}^{t+1}$ (lines 9-18). Note that the Lagrange multipliers correspond to satisfied constraints remain the same as previous iterations (line 13 and 18). Then the objective value of the partition $\vec{x}^{t+1}$ is calculated and the algorithm keeps track of the best solution found using $bestX$. The best solution is defined as the lexicographic minimum in the number of violations and the objective value of the partition (line 20).

Procedure *LagCCD*(W,ML, CL,max_iter)
1. $t \leftarrow 0$
2. Initialize the value of $\vec{x}^t$ with an unconstrained solution
3. Initialize the value of $\vec{\lambda}^t$ and $\vec{\mu}^t$ to 0
4. Build P^{ML}, P^{CL}
5. $bestV \leftarrow |ML| + |CL|$; $bestObj \leftarrow 1$; $bestX \leftarrow \vec{x}$
6. **while** $L(\vec{x}^t, \vec{\lambda}^t, \vec{\mu}^t)$-$f(\vec{x}^t) > 0$ and $t < max\_iter$ **do:**
7. $\quad Viol \leftarrow \{\}$
8. $\quad$ update the variable $\vec{x}^{t+1}$ based on $GD_x(\widetilde{L}(\vec{x}^t, \vec{\lambda}^t, \vec{\mu}^t))$
9. $\quad$ **for** $(i,j) \in ML$
10. $\quad\quad$ **if** $\vec{x}_{ij}^{t+1}$==0
11. $\quad\quad\quad \lambda_{(i,j)}^{t+1} \leftarrow \alpha \times max(1, \lambda_{(i,j)}^t)$
12. $\quad\quad\quad Viol \leftarrow Viol \cup \{(i,j)\}$
13. $\quad\quad$ **else** $\lambda_{(i,j)}^{t+1} \leftarrow \lambda_{(i,j)}^t$
14. $\quad$ **for** $(i,j) \in CL$
15. $\quad\quad$ **if** $\vec{x}_{ij}^{t+1}$==1
16. $\quad\quad\quad \mu_{(i,j)}^{t+1} \leftarrow \alpha \times max(1, \mu_{(i,j)}^t)$
17. $\quad\quad\quad Viol \leftarrow Viol \cup \{(i,j)\}$
18. $\quad\quad$ **else** $\mu_{(i,j)}^{t+1} \leftarrow \mu_{(i,j)}^t$
19. $\quad Obj \leftarrow f(\vec{x}^{t+1})$
20. $\quad$ **if** $(|Viol|, Obj) < (bestV, bestObj)$
21. $\quad\quad bestObj \leftarrow Obj$; $bestV \leftarrow |Viol|$; $bestX \leftarrow \vec{x}^{t+1}$
22. $\quad t \leftarrow t + 1$
23. **return** $bestX$

Figure 1: The LagCCD procedure

Note that any modularity-like scoring matrix such as the generalized modularity (Ganji et al. 2015) and VSP (Li and Pang 2014) can be used as W in the objective function. In addition, any supervision in the form of known labels can be decoded as must-link (between vertices with the same label) and cannot-link (between vertices with different labels) constraints and then be incorporated by the LagCCD algorithm.

Experiments

In our first experiment setup, we generated different number of constraints based on the ground truth of the real data sets and compared the performance of different algorithms in terms of their ability to satisfy the constraints and produce good solutions. The information about the real data sets including their number of vertices (n) and edges (|E|) and number of ground truth communities (k) are shown in Table 1. We use an information theoretic measure called Normalized Mutual Information (NMI) (Leon Danon and Arenas 2006) to evaluate quality of the solutions against the ground truth.

For each real data set of size n we generated $n/2$, n and $2n$ constraints equally divided into must-link and cannot-link pairs. For generating must-link (cannot-link) constraints, we randomly pick an instance and then pick another random in-

Table 1: Information of the real data sets

Data set	n	|E|	k
Sampson T4 (Sampson 1968)	18	15	4
Sampson T1T5 (Sampson 1968)	25	69	2
Strike (Michael 1997)	24	38	3
Zakhary's Karate club (Zachary 1977)	34	78	2
Mexican (Gil-Mendieta and Schmidt 1996)	35	117	2
Dolphin (Lusseau et al. 2003)	62	159	2
Political Books (Krebs unpublished)	105	441	3
Word adjacencies (Newman 2006)	112	425	2
Football (Girvan and Newman 2002)	115	441	12
Political blogs (Adamic and Glance 2005)	1490	9545	2

stance from the same (different) community based on the ground truth labels. For each size of the constraint sets we generated 5 different sets of random constraints and reported the results of 50 independent executions in Table 2.

The first two columns of Table 2 are the data sets and their number of vertices and the third column is the total number of constraints equally divided to must-link and cannot-link constraints. Note that as a preprocessing step one can augment the set of must-link and cannot-links by inferring new constraints from them. However, in order to have total control over the ultimate number of constraints provided, we don't augment the constraint sets in our experiments.

Our proposed algorithm is denoted by LagCCD. The parameter α is 1.2 and maximum iteration is set to 30. Violation penalty in P^{ML} and P^{CL} are set to 1. We used GenLouvain (Inderjit S. Jutla and Mucha 2011) as the optimization step in GD where we used W^c as the similarity matrix. Given a similarity scoring matrix, first, GenLouvain finds the communities of highest gain and greedily merges the communities and reconstructs the network in the second phase to finally find the partition with the maximum score.

The next column section in Table 2 reports the results of the spin-glass model (Eaton and Mansbach 2012). We derived the spin-glass matrix based on the parameters and formulations of the paper (Eaton and Mansbach 2012). To be consistent with execution of our algorithm, we used GenLouvain (Inderjit S. Jutla and Mucha 2011)[1] as the optimization algorithm.

Considering the adjacency matrix as a similarity matrix, we also compared with the spectral clustering algorithm described in (Zhang 2013), denoted by SC in Table 2 which systematically modifies the adjacency matrix based on the side information.

The next algorithm in Table 2 is called SNMF-SS which incorporates pairwise constraints in a non-negative matrix factorization model (Ma et al. 2010). We used an adjacency based similarity matrix and set violation cost to 1 for both must-link and cannot link constraints (entries of W_{ML} and W_{CL}). The parameters α and β should be set in a way that the similarity matrix $\overline{K}$ in $\overline{K} = K - \alpha W_{ML} + \beta W_{CL}$ remains non-negative. We set α and β to 0 and 0.03 respectively according to (Ma et al. 2010) to ensure nonnegativity. We modified the implementation of (Kuang, Ding, and Park 2012) to perform SNMF-SS. Note that we don't compare with K-means type constrained clustering algorithms (Ganji, Bailey, and Stuckey 2016; Wagstaff et al. 2001; Pelleg and Baras 2007) because their calculations are in Euclidean space and one cannot directly apply them to a network data without proper transformations.

[1] A variation of Louvain (Blondel et al. 2008)

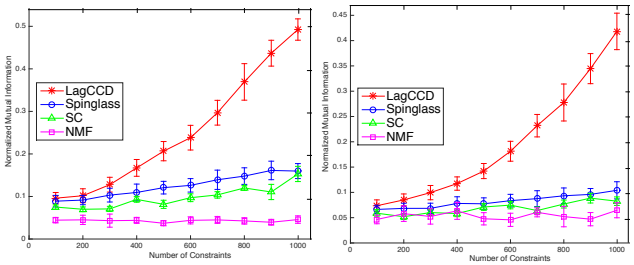

Figure 2: Sensitivity to number of constraints on LFR - 0.75 (left) and LFR - 0.8 (right)

As the results in Table 2 show, our LagCCD algorithm satisfies all of the constraints in most of the data sets. Moreover, it is considerably better than the baseline algorithms, which produce many more constraint violations. The solution quality (NMI) achieved by LagCCD has the highest NMI score across all of the test cases but one. A Friedman statistical test with 95% confidence level is performed for average number of violations and NMI over each constraint size (e.g $n/2, n, 2n$). The very low P-values for both violations and NMI, show the LagCCD outperforms the other methods statistically significantly.

Sensitivity to the number of constraints

In this experiment we evaluate the abilities of the algorithms to take advantage of provided side information by increasing the number of supervision constraints gradually and recording the NMI achieved by each algorithm. We use LFR data sets proposed in (Lancichinetti, Fortunato, and Radicchi 2008). In LFR data sets, vertex degrees and community size follow a power-law distribution with parameter α and β respectively. A mixing parameter, μ is the proportion of external degree for each vertex. According to (Lancichinetti, Fortunato, and Radicchi 2008) we fixed the parameters α and β to 1 and 2 respectively. We generated data sets of size N=1000 with communities of size 50 to 100 and mixing parameter 0.75 and 0.8 where the communities are a bit vague and difficult to detect. We increased the number of constraints from 100 to 1000. Note that this number of pairwise constraints is still a small percentage of the total number of pairs in the data. We keep the number of constraints low to evaluate which algorithm can most benefit from the limited amount of side information. For each constraint size, the average and standard deviation of 50 independent executions (10 repetitions on 5 constraint sets) are shown in Figure 2 . As it is demonstrated in the figures, the LagCCD algorithm takes the most advantage of the provided supervision and as opposed to the other algorithms, its performance is improved by even a small increase in the number of supervision constraints.

Table 2: Mean and standard deviation of number of violations and NMI score on different data sets

		LagCCD				Spinglass				SC				SNMF-SS			
		Violations		NMI		Violations		NMI		Violations		NMI		Viol		NMI	
data	#const	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std	mean	std
Sampson T4	8	0.00	0.00	**0.63**	0.06	0.00	0.00	**0.63**	0.06	0.70	1.07	0.62	0.05	3.28	0.50	0.59	0.03
	18	0.00	0.00	**0.76**	0.04	0.20	0.40	**0.76**	0.05	0.40	0.49	0.75	0.05	5.40	1.37	0.59	0.02
	36	0.00	0.00	**0.82**	0.08	0.80	0.76	0.79	0.07	2.26	1.44	0.76	0.10	10.48	2.04	0.59	0.03
Sampson T1T5	12	0.00	0.00	**0.61**	0.15	0.20	0.40	0.59	0.14	0.80	1.09	0.56	0.11	2.34	1.39	0.53	0.06
	24	0.08	0.27	**0.91**	0.12	0.60	0.81	0.83	0.15	1.58	0.93	0.75	0.09	6.42	1.58	0.51	0.06
	50	0.00	0.00	**1.00**	0.00	0.98	1.20	0.90	0.13	2.58	1.21	0.76	0.09	11.08	3.12	0.51	0.07
Strike	12	0.00	0.00	**0.73**	0.04	0.40	0.49	0.72	0.05	0.60	0.49	0.73	0.05	2.02	1.56	0.65	0.05
	24	0.00	0.00	**0.75**	0.12	1.40	1.03	0.72	0.05	1.40	1.03	0.73	0.04	3.64	2.25	0.65	0.05
	48	0.00	0.00	**0.90**	0.09	1.60	1.03	0.81	0.06	2.20	1.74	0.76	0.06	9.08	2.97	0.63	0.04
Karate club	16	0.00	0.00	0.72	0.09	0.00	0.00	**0.73**	0.10	0.24	0.51	0.70	0.08	3.14	0.69	0.69	0.02
	34	0.00	0.00	**0.88**	0.06	0.18	0.39	0.85	0.10	0.36	0.60	0.84	0.11	9.38	1.61	0.68	0.01
	68	0.00	0.00	**0.97**	0.07	0.32	0.74	0.94	0.08	0.48	0.79	0.94	0.09	15.24	2.81	0.69	0.02
Mexican	16	0.00	0.00	**0.31**	0.07	2.22	1.25	0.29	0.05	3.70	1.81	0.29	0.05	5.48	2.64	0.27	0.04
	34	0.00	0.00	**0.43**	0.11	3.90	2.26	0.30	0.06	6.08	2.42	0.31	0.05	12.00	2.10	0.27	0.04
	70	0.00	0.00	**0.79**	0.10	4.24	2.65	0.60	0.14	14.48	3.52	0.37	0.10	26.12	4.08	0.28	0.03
Dolphin	30	0.00	0.00	**0.65**	0.07	3.66	1.27	0.60	0.04	4.60	1.43	0.57	0.05	8.88	1.89	0.49	0.02
	62	0.00	0.00	**0.71**	0.10	6.20	1.63	0.65	0.05	7.62	1.34	0.61	0.04	18.40	2.56	0.48	0.02
	124	0.00	0.00	**0.96**	0.05	4.76	3.11	0.81	0.11	13.26	2.90	0.65	0.05	31.92	2.28	0.49	0.01
Political Books	52	0.00	0.00	**0.66**	0.03	7.20	2.03	0.56	0.02	7.64	1.78	0.58	0.02	18.42	2.70	0.43	0.02
	104	0.20	0.45	**0.76**	0.06	13.46	1.70	0.60	0.01	14.28	1.62	0.59	0.02	34.02	2.71	0.43	0.02
	210	0.18	0.48	**0.93**	0.04	18.70	4.89	0.68	0.06	27.06	3.01	0.63	0.02	70.18	7.14	0.43	0.02
Word adjacency	56	0.04	0.20	**0.02**	0.01	10.62	2.08	0.01	0.01	13.20	2.40	0.01	0.01	28.40	3.72	0.01	0.01
	112	1.20	1.58	**0.06**	0.02	21.98	2.76	0.01	0.01	28.06	2.44	0.01	0.01	56.44	2.79	0.01	0.01
	224	8.88	2.84	**0.32**	0.05	51.92	2.05	0.03	0.01	64.78	3.52	0.02	0.01	112.22	5.40	0.01	0.01
Football	56	0.00	0.00	**0.27**	0.02	20.34	3.15	0.21	0.01	21.16	3.01	0.22	0.01	24.78	3.89	0.19	0.01
	114	0.08	0.27	**0.39**	0.03	41.24	2.86	0.23	0.02	43.68	4.32	0.23	0.02	52.48	3.47	0.19	0.01
	230	0.10	0.46	**0.68**	0.05	85.32	3.68	0.23	0.01	92.44	2.87	0.24	0.02	106.82	5.71	0.19	0.01
Political blogs	744	0.74	1.08	**0.36**	0.01	38.28	3.39	0.36	0.00	56.88	4.40	0.33	0.01	371.56	1.50	0.05	0.00
	1490	5.46	2.22	**0.47**	0.01	81.62	6.80	0.43	0.01	135.54	10.39	0.36	0.01	743.44	2.35	0.05	0.00
	2980	10.36	4.77	**0.75**	0.02	145.58	8.13	0.59	0.01	313.54	13.01	0.43	0.01	1489.22	3.04	0.05	0.00
P-value	size 1					0.005		0.020		0.002		0.011		0.002		0.002	
	size 2	base		base		0.002		0.011		0.002		0.002		0.002		0.002	
	size 3					0.002		0.002		0.002		0.002		0.002		0.002	

Sensitivity to noise

Real world data is usually perturbed by noise and it is crucial for community detection algorithms to maintain their solution quality at acceptable level in noisy situations. In this experiment we evaluate the noise resistance of the proposed LagCCD algorithm and compare it with other existing approaches. To generate noisy data we perturbed a fraction of the network by adding or deleting the edge between a randomly picked pair of vertices. The noise ratio in the figure is based on the fraction of original number of edges in the data which has been modified. Figures 3 show the mean and standard deviation of the results on Dolphin data set (Lusseau et al. 2003) and the data set about books on US politics(Krebs unpublished). Each point in these figures is the result of 125 independent executions over five independently generated noisy data and 5 supervision constraint sets of the same size (equal to the network's number of vertices). As it is shown in both instances of Figures 3, similar to other methods, the performance of the LagCCD algorithm decreases in noisy situations. However, the LagCCD algorithm can maintain its higher performance and demonstrate more noise resistance than the other semi-supervised algorithms when the data gets noisier.

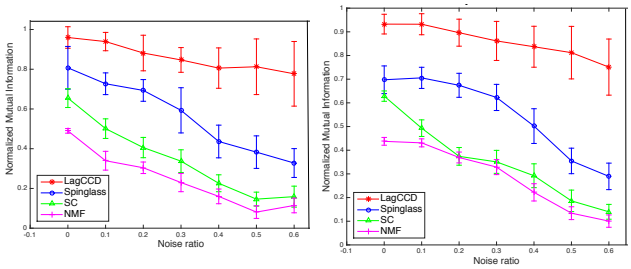

Figure 3: Sensitivity to noise on Dolphin (left) and Political Book (right) data sets

Conclusion

In this paper, we proposed a constrained community detection algorithm based on Lagrange multipliers which incorporates known labels and pairwise supervision types in a modularity maximization scheme. Its philosophy is to exploit available side information as fully as possible. Our experiments on real and synthetic data sets demonstrated improved performance of LagCCD in comparison to existing state of the art algorithms, in both noisy situations and scenarios with limited amounts of information. An interesting future direction is to encode other constraint types for community detection using our framework.

References

Adamic, L. A., and Glance, N. 2005. The political blogosphere and the 2004 us election: divided they blog. In *LinkKDD*, 36–43. ACM.

Allahverdyan, A. E.; Ver Steeg, G.; and Galstyan, A. 2010. Community detection with and without prior information. *Europhysics Letters* 90(1).

Anand, R., and Reddy, C. 2011. Graph-based clustering with constraints. *Advances in Knowledge Discovery and Data Mining* 51–62.

Bertsekas, D. P. 2014. *Constrained optimization and Lagrange multiplier methods*. Academic press.

Blondel, V. D.; Guillaume, J.-L.; Lambiotte, R.; and Lefebvre, E. 2008. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment* 2008(10):P10008.

Choi, K. M.; Lee, J. H.; and Stuckey, P. J. 2000. A lagrangian reconstruction of genet. *Artificial Intelligence* 123(1):1–39.

Eaton, E., and Mansbach, R. 2012. A spin-glass model for semi-supervised community detection. In *AAAI*.

Fortunato, S. 2010. Community detection in graphs. *Phys. Reports* 486(3).

Ganji, M.; Bailey, J.; and Stuckey, P. J. 2016. Lagrangian constrained clustering. In *Proceedings of the 2016 SIAM International Conference on Data Mining*, 288–296. SIAM.

Ganji, M.; Bailey, J.; and Stuckey, P. J. 2017. A declarative approach to constrained community detection. In *International Conference on Principles and Practice of Constraint Programming*, 477–494. Springer.

Ganji, M.; Seifi, A.; Alizadeh, H.; Bailey, J.; and Stuckey, P. J. 2015. Generalized modularity for community detection. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases* 655–670.

Gil-Mendieta, J., and Schmidt, S. 1996. The political network in mexico. *Social Networks* 18(4).

Girvan, M., and Newman, M. E. 2002. Community structure in social and biological networks. *Proceedings of the national academy of sciences* 99(12):7821–7826.

Inderjit S. Jutla, L. G. S. J., and Mucha, P. J. 2011.

Krebs, V. (unpublished). *http://www.orgnet.com/*.

Kuang, D.; Ding, C.; and Park, H. 2012. Symmetric nonnegative matrix factorization for graph clustering. In *Proceedings of the 2012 SIAM International Conference on Data Mining*, 106–117. SIAM.

Lancichinetti, A., and Fortunato, S. 2009. Community detection algorithms: a comparative analysis. *Physical review E* 80(5):056117.

Lancichinetti, A.; Fortunato, S.; and Radicchi, F. 2008. Benchmark graphs for testing community detection algorithms. *Physical Review E* 78(4).

Leon Danon, A. D.-G., and Arenas, A. 2006. The effect of size heterogeneity on community identification in complex networks. *Journal of Statistical Mechanics: Theory and Experiment* P11010.

Li, K., and Pang, Y. 2014. A unified community detection algorithm in complex network. *Neurocomputing* 130:36–43.

Li, Z.; Gong, Y.; Pan, Z.; and Hu, G. 2017. An efficient semi-supervised community detection framework in social networks. *PloS one* 12(5):e0178046.

Lusseau, D.; Schneider, K.; Boisseau, O. J.; Haase, P.; Slooten, E.; and Dawson, S. M. 2003. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology* 54(4):396–405.

Ma, X.; Gao, L.; Yong, X.; and Fu, L. 2010. Semi-supervised clustering algorithm for community structure detection in complex networks. *Physica A* 389(1):187–197.

Michael, J. H. 1997. Labor dispute reconciliation in a forest products manufacturing facility. *Forest products journal* 47(11/12):41.

Newman, M. E., and Girvan, M. 2004. Finding and evaluating community structure in networks. *Physical review E* 69(2):026113.

Newman, M. E. 2006. Finding community structure in networks using the eigenvectors of matrices. *Physical review E* 74(3):036104.

Ng, A. Y.; Jordan, M. I.; and Weiss, Y. 2002. On spectral clustering: Analysis and an algorithm. In *Advances in neural information processing systems*, 849–856.

Pelleg, D., and Baras, D. 2007. K-means with large and noisy constraint sets. In *ECML*, 674–682. Springer.

Sampson, S. F. 1968. A novitiate in a period of change: An experimental and case study of social relationships. *Cornell University*.

Shang, Y., and Wah, B. W. 1998. A discrete lagrangian-based global-search method for solving satisfiability problem. *Journal of global optimization* 12(1):61–99.

Silva, T. C., and Zhao, L. 2012. Semi-supervised learning guided by the modularity measure in complex networks. *Neurocomputing* 78(1):30–37.

Wagstaff, K.; Cardie, C.; Rogers, S.; Schrödl, S.; et al. 2001. Constrained k-means clustering with background knowledge. In *ICML*, volume 1, 577–584.

Wah, B. W., and Wu, Z. 1999. The theory of discrete lagrange multipliers for nonlinear discrete optimization. In *CP*, 28–42.

Wang, X., and Davidson, I. 2010. Flexible constrained spectral clustering. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 563–572.

Yang, L.; Cao, X.; Jin, D.; Wang, X.; and Meng, D. 2015. A unified semi-supervised community detection framework using latent space graph regularization. *IEEE trans. on cybernetics* 45(11):2585–2598.

Zachary, W. W. 1977. An information flow model for conflict and fission in small groups. *Journal of anthropological research* 452–473.

Zhang, Z.-Y. 2013. Community structure detection in complex networks with partial background information. *Europhysics Letters* 101(4):48005.