Multi-way Constrained Spectral Clustering by Nonnegative Restriction

Han Hu, Jiahuan Zhou, Jianjiang Feng and Jie Zhou
State Key Laboratory on Intelligent Technology and Systems, TNList
Department of Automation, Tsinghua University, Beijing, China, 100084
huh04@mails.thu.edu.cn, zhoujh09@gmail.com, {jfeng, jzhou}@tsinghua.edu.cn

Abstract

Clustering often benefits from side information. In this paper, we consider the problem of multi-way constrained spectral clustering with pairwise constraints which encode whether two nodes belong to the same cluster or not. Due to the nontransitive property of cannot-link constraints, it is hard to incorporate cannot-link constraints into the framework. We settle this difficulty by restricting the spectral vectors with nonnegative elements. An iterative method is proposed to optimize the objective. Experiments on several publicly available datasets demonstrate the effectiveness of our algorithm.

1. Introduction

Clustering is one of the most widely used techniques for data analysis. Typically, it works in an unsupervised manner, with performance highly depending on the designed distance (or similarity) metric. A major difficulty for clustering lies in the large semantic gap between clustering results and feature based distance. Recent research efforts have shown that the semantic gap can be reduced by incorporating high-level information [12, 7, 13, 9], referred as constrained clustering.

Wagstaff and Cardie [12] are the first to consider constrained clustering problem. They incorporate pairwise constraints, which specifies whether two nodes belong to the same cluster or not, into k-means and achieve much better performance. Since then, a lot of studies have been made (see [1] for a overview).

We focus on the problem of integrating pairwise constraints into spectral clustering [10, 11]. A major difficulty for constrained spectral clustering lies in the nontransitive property of cannot-link constraints[14, 6]. As a result, the cannot-link constraints are usually either discarded [14] or limited to two-class problem [13, 9]. To utilize cannot-link constraints in multi-way spectral clustering, a few algorithms were proposed. Kamvar et al. [3] and Kulis et al. [4] modified the similarities according to the constraints, and used standard spectral clustering algorithms or kernel k-means on the modified similarities to achieve multi-way clustering. Li et al. [6] calculated the first k eigenvectors of an unconstrained normalized cut problem, and adapted them to both must-link and cannot-link constraints by a semi-definite programming routine. Although these methods gain certain success in clustering accuracy, the cannot-link constraints are far from being gracefully and fully exploited.

In this paper, we propose a novel method for multi-way constrained spectral clustering, namely Nonnegative Constrained Spectral Clustering (NCSC). It is achieved by adding nonnegativity constraints to the spectral clustering problem, and so that the cannot-link constraints could be gracefully incorporated. We also present an iterative algorithm to optimize the problem. Experiments on different datasets show that our algorithm performs much better than the state-of-the-art algorithms.

2. Normalized Cut and Spectral Clustering

We first give a brief review to the normalized cut and spectral clustering problem [10, 11]. Denote $G(V, E, W)$ as an undirected graph G with vertex set V and edge set E, together with edge weights $W : V \times V \to \mathbb{R}_+^{n\times n}$, where $n = |V|$ is the cardinality of V. The task of clustering is to partition vertex set V into c clusters $\{C_i\}_{i=1}^c$, with $|C_i| = n_i$. We define the cut and the volume as, $cut(C_1, C_2) = \sum_{i\in C_1, j\in C_2} W_{ij}, vol(C) = \sum_{i\in C} \mathbf{d}_i$, with $\mathbf{d}_i = \sum_{j\in V} W_{ij}$. Then the normalized cut achieves clustering by minimizing the total cut balanced with the cluster volume, as [11],

$$J_{\text{ncut}} = \sum_{i=1}^{c} \frac{cut(C_i, \overline{C_i})}{vol(C_i)}, \tag{1}$$

where $\overline{C_i} = V\backslash C_i$. Denote $\mathbf{y}_i \in \{0, 1\}^{n\times 1}$ as the indicator matrix for cluster C_i, and let

$$Y = [\mathbf{y}_1/\sqrt{vol(C_1)}, ..., \mathbf{y}_c/\sqrt{vol(C_c)}], \tag{2}$$

Then the minimization of eq. (1) becomes [11],

$$\min_{Y^TDY=I,\ Y\ as\ eq.(2)} tr(Y^TLY), \tag{3}$$

where $D = diag(\mathbf{d})$ is the degree matrix, and $L = D - W$ is the Laplacian matrix. The well known spectral clustering algorithm relaxes the binary constraints for Y in eq. (3) to real value and solves the problem by eigenvalue decomposition. However, the eigenvector solutions are with mixed signs which makes incorporating the cannot-link constraints difficult. In the next section, we will show how this difficulty can be settled by adding nonnegativity constraints.

3. Constrained Spectral Clustering

Denote $Q^m,\ Q^c$ as the constraint matrices, where the element $q^m_{ij},\ q^c_{ij} \in \{0, 1\}$ encodes the must-link and cannot-link constraint between node i and j. Denote $\mathbf{f}_i^T$ as the i^{th} row of Y, which represents the indicators for node i. For a must-link constraint between i and j, the indicators $\mathbf{f}_i$ and $\mathbf{f}_j$ should be the same. Thus we can have an objective function as,

$$\begin{aligned} J_{m\text{-}link} &= \sum_{i,j\in V} q^m_{ij}||\mathbf{f}_i - \mathbf{f}_j||^2 \\ &= 2tr(Y^T(D^m - Q^m)Y), \end{aligned} \tag{4}$$

where $D^m = diag(\mathbf{d}^m_i)$, with $\mathbf{d}^m_i = \sum_{j\in V} q^m_{ij}$.

However, since the solutions are with mixed signs, it is hard to formulate cannot-link constraints into the optimization. We settle this difficulty by restricting Y with nonnegative values. Under nonnegativity constraints of Y, for any two nodes i and j, $\mathbf{f}_i^T \cdot \mathbf{f}_j \geq 0$ holds. If a cannot-link constraint exists between i and j, $\mathbf{f}_i^T \cdot \mathbf{f}_j = 0$. Thus we can encode the cannot-link constraints by minimizing,

$$\begin{aligned} J_{c\text{-}link} &= \sum_{i,j\in V} q^c_{ij}(\mathbf{f}_i^T \cdot \mathbf{f}_j) \\ &= tr(Y^TQ^cY). \end{aligned} \tag{5}$$

Based on the above analysis, we propose the following optimization problem (NCSC),

$$\begin{aligned} \min \quad & tr(Y^TLY) + \gamma_m tr(Y^T(D^m - Q^m)Y) \\ & + \gamma_c tr(Y^TQ^cY) \\ s.t. \quad & Y^TDY = I,\ Y \geq 0. \end{aligned} \tag{6}$$

Besides encoding cannot-link constraints into the optimization framework, the nonnegativity restriction also helps assigning the clusters, which is usually done by k-means or spectral rotation in previous researches.

3.1. Optimization

In this subsection, we develop an algorithm to solve the optimization problem shown in eq. (6). Formally, the optimization in eq. (6) is equivalent to,

$$\begin{aligned} \min \quad & tr(Y^T(G - \tfrac{\sigma}{\mathbf{d}_{min}}D)Y) \\ s.t. \quad & Y^TDY = I,\ Y \geq 0, \end{aligned} \tag{7}$$

where $G = L + \gamma_m(D^m - Q^m) + \gamma_c Q^c$, as $tr(Y^T(\frac{\sigma}{\mathbf{d}_{min}}D)Y) = \frac{\sigma}{\mathbf{d}_{min}}tr(I) = \frac{n\sigma}{\mathbf{d}_{min}}$ is a constant. We set $\sigma = \lambda_m$ to be the largest eigenvalue of G, and thus $G - \frac{\sigma}{\mathbf{d}_{min}}D$ becomes non-positive definite. This step makes the optimization as a well-behaved problem [8].

Since $Y^TDY = I$, we introduce Lagrangian multiplier $\Lambda \in \mathbb{R}^{c\times c}$, and thus the Lagrangian function is,

$$L(Y) = tr(Y^THY) + tr(\Lambda(Y^TDY - I)), \tag{8}$$

where $H = G - \frac{\sigma}{d_{\min}}D$.

The gradient of $L(Y)$ with respect to Y is,

$$\frac{\partial L(Y)}{\partial Y} = 2HY + 2DY\Lambda. \tag{9}$$

Using the Karush-Kuhn-Tucker complementarity condition [2] $(\frac{\partial L(Y)}{\partial Y})_{ij}Y_{ij} = 0$, we get

$$(HY + DY\Lambda)_{ij}Y_{ij} = 0. \tag{10}$$

Since H and Λ may take mixed signs, we introduce $H = H^+ - H^-$ and $\Lambda = \Lambda^+ - \Lambda^-$, where $^+$ and $^-$ indicate respectively the positive and negative part of a matrix. Then we get the following updating rule:

$$Y_{ij} \leftarrow Y_{ij}\sqrt{\frac{[H^-Y + DY\Lambda^-]_{ij}}{[H^+Y + DY\Lambda^+]_{ij}}}. \tag{11}$$

It remains to determine the Lagrangian multiplier Λ. Following the similar deduction in [8], we obtain $\Lambda = -Y^THY$.

Next, we show that the updating algorithm as eq. (11) converges.

Definition 1. [5] $Z(h, h')$ is an auxiliary function for $F(h)$ if the conditions $Z(h, h') \geq F(h)$, $Z(h, h) = F(h)$ are satisfied.

Lemma 1. [5] If Z is is an auxiliary function for F, then F is non-increasing under the following updating rule,

$$h^{(t+1)} = arg\min_h Z(h, h^{(t)}). \tag{12}$$

Theorem 1. Let $J(Y) = tr(Y^THY) + tr(\Lambda(Y^TDY))$ by ignoring $-tr(\Lambda)$ of eq. (8). Then the following function

$$\begin{aligned} Z(Y, Y') &= \sum_{ij} \frac{(Y'H^+)_{ij}Y^2_{ij}}{Y'_{ij}} + \sum_{ij} \frac{(DY'\Lambda^+)_{ij}Y^2_{ij}}{Y'_{ij}} \\ &- \sum_{ijk} (H^-)_{jk}Y'_{ji}Y'_{ki}(1 + \log \frac{Y_{ji}Y_{ki}}{Y'_{ji}Y'_{ki}}) \\ &- \sum_{ijkl} (\Lambda^-)_{kj}D_{jl}Y'_{ji}Y'_{lk}(1 + \log \frac{Y_{ji}Y_{lk}}{Y'_{ji}Y'_{lk}}) \end{aligned}$$

is an auxiliary function for $J(Y)$. Furthermore, it is a convex function in Y and its global minimum is,

$$Y_{ij} = Y'_{ij}\sqrt{\frac{[H^-Y + DY\Lambda^-]_{ij}}{[H^+Y + DY\Lambda^+]_{ij}}}. \quad (13)$$

Proof. For space limits, we omit it. It will be presented in the longer version of this paper. □

Theorem 2. Under the updating rule of eq. (11), the Lagrangian function $L(Y)$ in eq. (8) decreases monotonically.

Proof. By Lemma 1 and Theorem 1, we have $J(Y^{(t)}) = Z(Y^{(t)}, Y^{(t)}) \geq Z(Y^{(t+1)}, Y^{(t)}) \geq J(Y^{(t+1)})$. Thus $J(Y^{(t)})$ (and $L(Y^{(t)})$) is monotonically decreasing. □

4. Discussion

4.1. Relationship with LCSC Algorithm

Linear Constrained Spectral Clustering (LCSC) algorithm [14] only considers must-link constraints. Given a must-link constraint between node i and j ($Q^m_{ij} = 1$), LCSC encodes it by $U_k^T Y = 0$, where U_k is an $n \times 1$ vector with only two non-zero elements: $U_k(i) = 1, U_k(j) = -1$. For all must-link constraints, the linear constraint is $U^T Y = 0$, where $U = [U_1, U_2, ..., U_{n_m}]$, with n_m denoting the number of must-link constraints. We have the following proposition:

Proposition 1. NCSC leads to LCSC when $\gamma_m \to +\infty$, $\gamma_c = 0$ and the nonnegativity constraints are discarded.

Proof. By moving the linear constraint of LCSC to the objective function, we have,

$$\begin{array}{ll} \min & tr(Y^T LY) + \gamma tr(Y^T UU^T Y) \\ s.t. & Y^T DY = I. \end{array} \quad (14)$$

where γ should $\to +\infty$ to ensure the linear constraint satisfied. Since U is formed by the nonzero elements of Q^m, it is easy to check that $U^T U \equiv 2(D^m - Q^m)$. Thus proposition 1 holds. □

Only regarding must-link constraints, besides imposing nonnegativity constraints, our NCSC algorithm has an advantage over LCSC in at least two aspects: 1) the proposed NCSC can encode soft constraints into the optimization, which is especially useful when the constraints are noisy, inconsistent or in continuous form; 2) the proposed NCSC does not need to compute the inverse of an $n_m \times n_m$ matrix or the SVD of an $n \times n_m$ matrix, which makes LCSC impossible to work when n_m is very large.

In the experiments, it turned out that typically the bigger the γ_m was (more must-link constraints were satisfied), the better the performance was. So we fixed γ_m of NCSC as 10^4 in our experiments, by which we only need to tune one parameter γ_c.

4.2. Relationship with Similarity Modification based Algorithms

Similarity modification based algorithms [3, 4] modify the similarities according to the constraints, and achieve multi-way clustering by standard spectral clustering algorithms or kernel k-means. In [3], the similarities are modified to 1's and 0's for must-link and cannot-link nodes, respectively. In [4], the similarities are shifted by $\pm n/(cn_{mc})$, with n, c and n_{mc} being the numbers of nodes, clusters and pairwise constraints, respectively.

In NCSC algorithm, however, the constraints are not formulated into the similarities, but contribute as independent penalty items (although they together form a quadratic function). In this way, we will not change the structure of the original similarities.

5. Experiments

We compare the proposed NCSC algorithm with LCSC [14], Spectral Learning (SL)[3][1], and Constrained Clustering with Spectral Regularization (CCSR) [6]. LCSC only utilizes must-link constraints. SL and CCSR and our NCSC incorporate both must-link and cannot-link constraints. The results of Normalized Cut (NC) [10] are also shown for reference.

All the algorithms are graph based, and to make fair comparisons, we use the same graphs for all algorithms. We use the weighted k-nearest-neighbor graph with $k = 20$ and σ determined following the self-tuning algorithm[15]. For NCSC, we use the results from the algorithm without nonnegativity constraints as initialization. We fix $\gamma_m = 10^4$, and tune γ_c from $linspace(0.1, 1, 10) \bigcup linspace(1, 10, 10)$ and report the best results.

We have collected four public datasets, including two UCI datasets Iris, Sonar[2], one hand written digital image dataset USPS[3] and one face image dataset Extended Yale Face B (EYaleB)[4]. Detailed information of the four datasets is summarized in Table 1.

For each dataset, 10 different numbers of pairwise constraints are randomly generated using ground truth

[1] For similarity modification based algorithms, we only shown the results of [3] as it performs better than [4] on most of the datasets.

[2] *http://archive.ics.uci.edu/ml/*

[3] *http://www-i6.informatik.rwth-aachen.de/~keysers/usps.html*

[4] *http://vision.ucsd.edu/~leekc/ExtYaleDatabase/ExtYaleB.html*. We resize the images with 30×40 pixels, and choose the last 10 subjects to form our dataset

Dataset	Size	Dimension	# of Clusters
Iris	150	4	3
Sonar	208	60	2
USPS	9298	256	10
EYaleB	5760	1200	10

Table 1. Datasets descriptions.

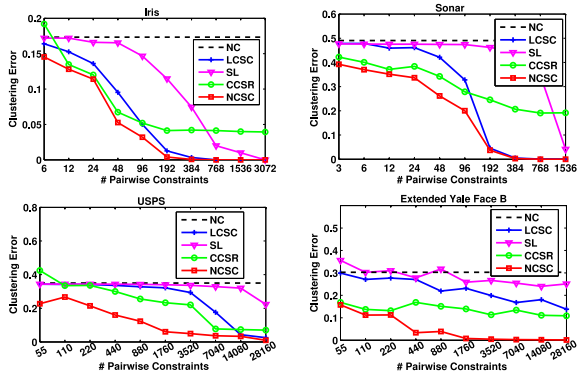


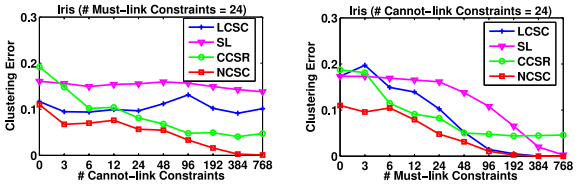
Figure 1. Clustering error vs. # pairwise constraints for all algorithms.

labels. For a fixed number of pairwise constraints, we report the results averaged over 10 trials.

We use clustering error (ERR) as the evaluation metric. Denote q_i as the clustering result from the clustering algorithm and p_i as the ground truth label of $\mathbf{x}_i$. ERR is defined as: $ERR = 1 - \frac{1}{n}\sum_{i=1}^{n}\delta(p_i, map(q_i))$, where $\delta(x, y) = 1$ if $x = y$; $\delta(x, y) = 0$ otherwise, and $map(q_i)$ is the best mapping function that permutes clustering labels to match the ground truth labels using the Kuhn-Munkres algorithm.

Figure 1 shows *ERR vs. # pairwise constraints* on the four datasets for all algorithms. We can see that the proposed NCSC outperforms all the other algorithms in nearly all of the cases.

To illustrate the performances of utilizing must-link and cannot-link constraints respectively, we respectively vary the number of cannot-link and must-link constraints while fixing the number of the other ones. The *ERR vs. # cannot-link constraints* and *ERR vs. # must-link constraints* on Iris dataset are shown in Figure 2. From Figure 2, we have the following conclusions: 1) when there are no cannot-link constraints, LCSC performs better than CCSR. This is because LCSC exploits the must-link constraints fully by guaranteeing all the must-link constraints satisfied. From Proposition 1, LCSC can be regarded as a special case of NCSC, and thus NCSC has similar performance as LCSC for must-link constraints; 2) CCSR achieves certain success in exploiting cannot-link constraints. But NCSC performs



(a) (b)

Figure 2. Clustering error with # cannot-link constraints varying (a) and # must-link constraints varying (b).

better especially for the numbers near two ends. When the number of cannot-link constraints rises high, NCSC can even reach 100% accuracy.

To sum up, NCSC encodes both must-link and cannot-link constraints best among all the algorithms.

6. Acknowledgement

This work was supported by the National Natural Science Foundation of China under Grants 61020106004, 61021063 and 61005023, and partly supported by Ministry of Transport of China.

References

[1] S. Basu, I. Davidson, and K. Wagstaff. *Constrained Clustering: Advances in Algorithms, Theory, and Applications*. Chapman & Hall/CRC, 1st edition, 2008.

[2] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge Univ Pr, 2004.

[3] S. D. Kamvar, D. Klein, and C. D. Manning. Spectral learning. In *IJCAI*, pages 561–566, 2003.

[4] B. Kulis, S. Basu, I. S. Dhillon, and R. J. Mooney. Semi-supervised graph clustering: a kernel approach. In *ICML*, pages 457–464, 2005.

[5] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *NIPS*, pages 556–562, 2000.

[6] Z. Li, J. Liu, and X. Tang. Constrained clustering via spectral regularization. In *CVPR*, pages 421–428, 2009.

[7] Z. Lu and M. Á. Carreira-Perpiñán. Constrained spectral clustering through affinity propagation. In *CVPR*, 2008.

[8] D. Luo, C. H. Q. Ding, H. Huang, and T. Li. Non-negative laplacian embedding. In *ICDM*, pages 337–346, 2009.

[9] S. Maji, N. K. Vishnoi, and J. Malik. Biased normalized cuts. In *CVPR*, pages 2057–2064, 2011.

[10] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE TPAMI*, 22(8):888–905, 2000.

[11] U. von Luxburg. A tutorial on spectral clustering. *CoRR*, abs/0711.0189, 2007.

[12] K. Wagstaff and C. Cardie. Clustering with instance-level constraints. In *ICML*, pages 1103–1110, 2000.

[13] X. Wang and I. Davidson. Flexible constrained spectral clustering. In *KDD*, pages 563–572, 2010.

[14] S. X. Yu and J. Shi. Segmentation given partial grouping constraints. *IEEE TPAMI*, 26(2):173–183, 2004.

[15] L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. In *NIPS*, 2004.