Multiclass Clustering using a Semidefinite Relaxation

Jason Lee
Institute of Computational and Mathematical Engineering
Stanford University
email: jdl17@stanford.edu

Abstract

We propose a semidefinite relaxation for graph clustering known as Max-cut clustering. The clustering problem is formulated in terms of a discrete optimization problem and then relaxed to a SDP. The SDP is solved using a low-rank factorization trick that reduces the number of variables, and then using a simple projected gradient method. This is joint work with Nathan Srebro at the Toyota Technology Institute-Chicago and part of research was performed at the Toyota Technology Institute-Chicago.

1 Introduction

Graph clustering is often formulated as a discrete optimization problem with the goal of balancing two criteria: cluster coherence and cluster size balance. We first discuss the specific case of binary clustering to highlight the tradeoffs between cluster coherence and cluster size. We then formulate this tradeoff as a discrete quadratic integer program and discuss the connection with the max-cut problem. As with max-cut, the discrete problem can be relaxed to a semidefinite program (SDP) that can be solved with standard solvers. To make the algorithm scalable, a low rank factorization approach similar to [1] is used to solve the SDP.

1.1 Graph Clustering

There are many possible clustering objectives that have been proposed in the literature and it is simple to construct new objectives that achieve the two desired properties of cluster size and quality. The two most common formulations are known as the Ratio Cut [2] and Normalized Cut [3, 4]. Both objectives are discrete and known to be NP-hard, however there is a continuous eigenvalue relaxation of the problems which leads to spectral clustering on a graph Laplacian. Due to the ease of computing eigenvectors and the intricate connection with spectral graph theory, graph Laplacian relaxation methods are commonly used. See [5] for further discussion and comparison of graph Laplacian-based clustering methods. A problem with the eigenvalue approach is the eigenvalue relaxation is a very loose approximation to the original discrete optimization problem; several authors have proposed tighter relaxations using semidefinite programming [6, 7], but these methods only scale to small-size problems. Instead of trying to design better clustering objectives, which has been extensively studied, we propose a simple discrete objective, that admits a solvable, yet tight relaxation through the max-cut SDP.

2 Binary Clustering

2.1 Max-Cut Problem

The well-studied max-cut problem (partition the vertices into C and $\overline{C}$ such that the number of edges between C and $\overline{C}$ is maximized) can be formulated as the following binary quadratic integer

program:

$$\text{maximize} \frac{1}{4} \sum_i \sum_j Q_{ij}(1 - x_i x_j) \text{ subject to } x_i^2 = 1, \text{ for all } i. \tag{1}$$

The max-cut problem attempts to maximize the number of edges cut, so a first attempt at using it for clustering is to let $Q = -W$. However, this leads to the trivial solution of $C = V$. A simple fix is to define $Q = \delta J - W = \delta e e^T - W$ for some $\delta > 0$, where J is the matrix of all ones and e is the vector of all ones. This choice of Q forces balanced clusters while minimizing the inter-cluster edges. Thus we solve the following problem, which is equivalent to max-cut with $Q = \delta J - W$:

$$\text{maximize} \sum_{i,j} (W_{ij} - \delta) x_i x_j \text{ subject to } x_i^2 = 1. \tag{2}$$

The objective function can be rewritten as $\sum_{i,j} W_{ij} x_i x_j - \delta(\sum_l x_l)^2$.The term $\delta(\sum_l x_l)^2$ can be viewed as a penalty function for unbalanced clusters since perfectly balanced clusters satisfy $\sum_l x_l = 0$. Using this observation, Equation 2 is equivalent to

$$\sum_{i,j} W_{ij} x_i x_j \quad \text{subject to} \quad x_i^2 = 1 \quad \text{and} \quad (\sum_i x_i)^2 \leq B \tag{3}$$

Thus Equation 2 is an example of a bi-criterion objective; maximizing the objective leads to maximizing $W(C, C) + W(\overline{C}, \overline{C})$ while minimizing $|C| - |\overline{C}|$. We can easily modify the penalty function to penalize the difference in volume by replacing $(\sum_i x_i)^2 \leq B$ with $(\sum_i d_i x_i)^2 \leq B$.

2.2 Semidefinite Relaxation

The discrete optimization problem proposed in Equation 2 is known to be NP-hard to solve, so we reforumlate it as a continuous problem:

$$\text{maximize} \sum_{ij} (W_{ij} - \delta) X_{ij} \quad \text{subject to} \quad X = xx^T \quad \text{and} \quad X_{ii}^2 = 1 \tag{4}$$

This reformulation is equivalent to the discrete problem. However, it is non-convex due to the rank 1 constraint on X. By dropping this constraint, we arrive at the semidefinite relaxation of max-cut.

$$\text{maximize} \sum_{ij} (W_{ij} - \delta) X_{ij} \quad \text{subject to} \quad X_{ii}^2 = 1 \quad \text{and} \quad X \succeq 0 \tag{5}$$

The semidefinite relaxation is a convex problem that can be efficiently solved using standard interior point solvers. Unfortunately, these solvers do not scale to problems with more than a few hundred variables. To motivate our solution method, we first study an equivalent vector formulation of the semidefinite relaxation.

$$\text{maximize} \sum_{ij} (W_{ij} - \delta) < v_i, v_j > \quad \text{subject to} \quad ||v_i||^2 = 1 \quad \text{and} \quad v_i \in \mathbb{R}^n \tag{6}$$

In the vector formulation, each binary variable x_i is replaced with a vector $v_i \in \mathbb{R}^n$. The variables in the vector formulation is $V \in \mathbb{R}^{nxn}$ where v_i are the rows of V. This semidefinite program has n^2 variables; a key idea from Burer and Monteiro [1] is that the number of variables can be reduced if we constrain each $v_i \in \mathbb{R}^r$ for $r < n$. For $r = 1$, the rank-constrained formulation is recovered. Burer and Monteiro show that if r is large enough the solution to the non-convex problem with $v_i \in \mathbb{R}^r$ is equivalent to the global optimum of the sdp (Equation 6). The final reformulation that we solve is the rank r constrained relaxation to the discrete problem.

$$\text{maximize } \text{Tr}((W - \delta J) V V^T) \quad \text{subject to} \quad ||v_i||^2 \leq 1 \quad \text{and} \quad v_i \in \mathbb{R}^r \tag{7}$$

2.3 Projected Gradient Method

To solve Equation 7, we use a simple projected gradient algorithm. The projected gradient algorithm updates with the rule

$$V \leftarrow \mathcal{P}(V + \tau(W - \delta J)V) \tag{8}$$

where $\mathcal{P}$ can be computed by normalizing the rows of V. This algorithm is extremely simple and efficient; each iteration involves matrix-multiplication and normalizing the rows of V. The storage required is nr variables and in the experiments we choose $r \leq 20$. Furthermore, we are guaranteed the global optimum of the sdp formulation if $rk(V^*) < r$ [1].

2.4 Max-Cut Clustering

We first formulate the Max-Cut clustering as a discrete problem of the form 2 and then employ the same relaxation as described in the binary case. Let k denote the number of clusters and $x_{ia} = -1, 1$ be cluster indicator variables where $x_{ia} = 1$ means node i belongs to cluster a. The x_{ia} satisfy $\sum_a x_{ia} = 2 - k$ to ensure each node belongs only to one cluster. A binary quadratic program for multiclass clustering can be posed as:

$$\text{maximize} \sum_{ij,ab} W_{ij,ab} x_{ia} x_{jb} \tag{9}$$

$$\text{subject to} \tag{10}$$

$$\sum_a x_{ia} = 2 - k, \quad \sum_i x_{ia} = (2-k)\frac{n}{k} \quad \text{and} \quad x_{ia}^2 = 1 \tag{11}$$

$W_{ij,ab} = 1[a = b]W_{ij}$ where W_{ij} is the weighted adjacency matrix. The second constraint is a cluster size constraint that forces each cluster to be of similar size. Using a quadratic penalty function on the constraints $\sum_a x_{ia} = 2 - k$ and $\sum_i x_{ia} = (2-k)\frac{n}{k}$, this can be converted to a max-cut type problem similar to Equation 2.[1] Similarly, a low-rank factorization type relaxation can be used to solve this. The final problem in vector form is:

$$\text{maximize} \sum_{ij,ab} (W_{ij,ab} - \delta 1[a=b] - \lambda 1[i=j]) < v_{ia}, v_{jb} > \quad \text{subject to} \quad ||v_i||^2 = 1 \quad \text{and} \quad v_i \in \mathbb{R}^r \tag{12}$$

This equation can be rewritten in the form of Equation 7 by defining $D_{ij,aa} = 1$ and $Q_{ii,ab} = 1$. D is rank k and Q is sparse. The objective is $\text{Tr}\,(W - \delta D - \lambda Q)VV^T$.

2.5 Computational Considerations

In the binary case, the algorithm requires a matrix-matrix multiplication at each iteration. The required work per iteration is $O(n^2 r)$ and the storage required is the matrix V which is $O(nr)$, where $n = |V|$.

In the multiclass case, the algorithm also requires a matrix-matrix multiplication at each iteration. The required work per iteration is $O(n^2 k^2 r)$ and the storage required is $O(nkr)$. The adjacency matrix W is frequently sparse and the two penalty terms are sparse and rank k, respectively.

2.6 Recovering the discrete solution

2.6.1 Rounding Scheme

After solving the optimization problem given in Equation 7, we have a vector v_i for each vertex. The Goemans-Williamson randomized hyperplane rounding assigns $x_i = sgn(< r, v_i >)$ where $r \sim \mathcal{N}(0, I)$. We repeat randomized hyperplane rounding R times and choose the assignment with largest objective as the final clustering. In the multiclass setting, the cluster labels are chosen as $c_i = argmax_a < r, v_{ia} >$. This is repeated R times and choose the assignment with largest objective. We also post-process the best label selected by hyperplane rounding using the relaxation labeling method developed for inference in markov random fields. See [9] for details.

3 Experimental Comparison

For all the experiments, we first build a K-nearest neighbor graph with $K = 10$ and weights W_{ij} defined as $W_{ij} = max(s_i(j), s_j(i))$, with $s_i(j) = exp(-\frac{||x_i - x_j||^2}{2\sigma_i^2})$ and σ_i equal to the distance from x_i to its K nearest neighbor.

[1] In the conversion to a max-cut problem, we omit terms of the type $\sum_{ia} \beta_{ia} x_{ia}$ i.e. linear terms in x_{ia}. The linear terms can be handled within our low-rank factorization framework by introducing the dummy variable x_0 and replacing all linear terms with $\sum_{ia} \beta_{ia} x_{ia} x_0$. This observation allows us to do MAP estimation in Markov Random Fields using the same low-rank relaxation technique. In the multi-class clustering case, the linear terms correspond to uniform priors over the label set and thus do not change the solution. See [8] for details. Experimental results for the MRF are not reported here due to lack of space.

Description			Misclassification Rate				
Dataset	N_c	k	Max-Cut	NJW(k)	SM(k)	NJW (k+1)	SM(k+1)
Pdigit (1,7)	100	2	0.08	**0.075**	**0.075**	**0.075**	**0.075**
Pdigit (0,1)	100	2	**0**	**0**	**0**	**0**	**0**
Pdigit(0,1,2,3,4)	100	5	**0.062**	0.298	0.298	0.074	0.08
Pdigit(1,3,5,7,9)	100	5	0.15	0.174	0.16	**0.138**	0.19
Pdigit(0,2,4,6,8)	100	5	**0.018**	0.056	0.252	0.056	0.184
Pdigit(0,1,2,4,5,6,8)	100	7	**0.123**	0.151	0.237	0.126	0.236
Pdigits(0,1,2,3,4,5,6,7)	100	8	**0.11875**	0.185	0.2	**0.1188**	0.1225
MNIST(0,1,2,3,4)	100	5	**0.09**	0.216	0.302	0.122	0.27
MNIST(0,1,2,3,4,6,7)	100	7	0.18	0.3571	0.3671	**0.12**	0.13
MNIST(1,3,5,7,9)	100	5	**0.186**	0.356	0.342	0.35	0.476
MNIST(0,2,4,6,8)	100	5	**0.148**	0.404	0.452	0.172	0.46
MNIST(0,1,2,3,4,5,6,7,8,9)	100	10	**0.421**	0.479	0.543	0.463	0.515
MNIST(1,3,4,6,8)	100	5	**0.13**	0.302	0.31	0.28	0.38
MNIST(0,1,2,4,5,7,8)	100	7	**0.4257**	0.5229	0.6329	0.52	0.5214
MNIST(5,6,7,8,9)	100	5	**0.354**	0.426	0.43	0.37	0.484
MNIST(0,1,2,3,4,5,6,7,8)	800	9	0.3058	0.2415	0.2569	**0.1106**	0.2564
MNIST(2,3,4)	300	3	**0.0222**	0.0456	0.0544	0.0244	0.0367
MNIST(6,7,8)	300	3	**0.0067**	0.011	0.012	0.0133	0.0133
MNIST(6,7,8,9)	300	4	0.2652	0.1975	0.2067	**0.1567**	0.1667
NewsgroupsA(7,16)	100	2	0.17	0.25	0.275	0.195	**0.04**
NewsgroupsA(7,16)	200	2	0.1475	0.245	0.34	**0.145**	0.4025
NewsGroups(11,12)	300	2	**0.08**	0.0967	0.1083	0.0883	0.1067
NewsGroups(7,15,17)	200	3	**0.2567**	0.375	0.425	0.33	0.43
NewsGroups(7,10,15,17)	200	4	**0.445**	0.4888	0.5525	0.545	0.6062
NewGroups(2,10,15,18)	200	4	0.4513	0.4513	0.53	**0.44**	0.5325

Table 1: N_c is number of points per cluster and k is the number of clusters. 5 different methods are compared: our low rank method, Shi-Malik spectral clustering with k eigenvectors, Ng-Jordan-Weiss spectral clustering with k eigenvectors, Shi-Malik spectral clustering with $k+1$ eigenvectors, and Ng-Jordan-Weiss spectral clustering with $k+1$ eigenvectors.

We test on the following datasets:

1. PenDigits (0-9). Handwritten digit dataset. Each data is 8 x-y plane measurements of the pen position.

2. MNIST (0-9). Handwritten digit dataset. Each data point is a 28 x 28 image of a single handwritten digit.

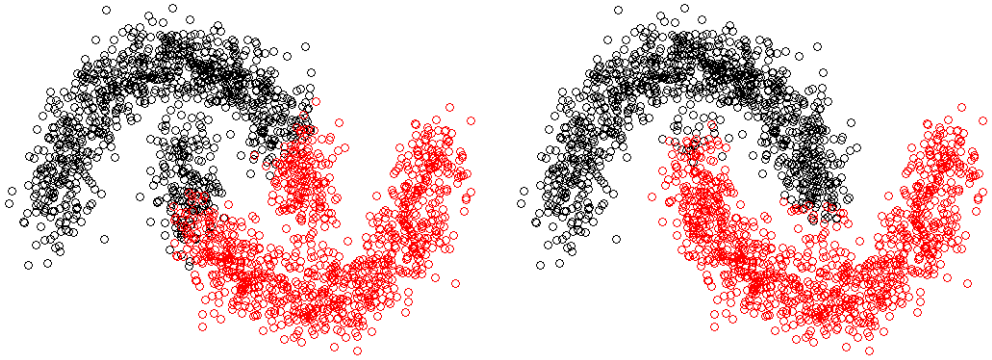
3. 20 Newsgroups. Each data point is a vector of term frequency. Data is collected from 20 Newsgroups on different topics.

4. Synthetic Two-Moons dataset. This experiment is reproduced from [10].

For all the experiments, $\lambda = .05$ and $\delta = \frac{W_{tot}}{4|V|^2}$. The algorithm is run for 4000 iterations with $r = 30$.

We compare against 2 different variants of the spectral clustering algorithm by Shi-Malik and Ng-Jordan-Weiss with 50 replications of k-means. The spectral clustering is run with k and $k+1$ eigenvectors[2]. The clustering accuracy is evaluated by the number of mis-clustered points each algorithm makes (the lowest among all $k!$ permutations is reported).

[2]Spectral clustering is generally done with k eigenvectors where k is the number of classes. We also test with $k+1$ because many of the errors in spectral clustering are due to the k and $k+1$ eigenvalue being very close. In fact, spectral clustering with $k+1$ eigenvectors does better in our experiments.



(a) Spectral Clustering (b) Max-cut clustering

Figure 1: *Comparison of spectral clustering (left) with* MAX-CUT *clustering (right).*

References

[1] Samuel Burer and R. D. C. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming (Series B)*, 95:329–357, 2003.

[2] L. Hagen and A.B. Kahng. New spectral methods for ratio cut partitioning and clustering. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 11(9):1074 –1085, September 1992.

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

[4] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS*, pages 849–856. MIT Press, 2001.

[5] Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17:395–416, 2007. 10.1007/s11222-007-9033-z.

[6] Eric Xing, Eric P. Xing, Michael Jordan, and Michael I. Jordan. On semidefinite relaxations for normalized k-cut and connections to spectral clustering, 2003.

[7] Tijl De Bie and Nello Cristianini. Fast sdp relaxations of graph cut clustering, transduction, and other combinatorial problems. *J. Mach. Learn. Res.*, 7:1409–1436, December 2006.

[8] M. Pawan, Kumar V. Kolmogorov, and P. H. S. Torr. An analysis of convex relaxations for map estimation, 2008.

[9] Timothee Cour and Jianbo Shi. Solving markov random fields with spectral relaxation. In *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, volume 11, 2007.

[10] Jason D. Lee, Benjamin Recht, Ruslan Salakhutdinov, Nathan Srebro, and Joel A. Tropp. Practical large-scale optimization for max-norm regularization. In *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS*. MIT Press, 2010.