A Total Variation-based Graph Clustering Algorithm for Cheeger Ratio Cuts

Arthur Szlam Xavier Bresson*

August 23, 2009

Abstract

In this work, inspired by [3] and [13], we give a continuous relaxation of the Cheeger cut problem on a weighted graph. We show that the relaxation is actually equivalent to the original problem, and based on [8, 16], we give an algorithm which experimentally is very efficient on some clustering benchmarks. We also give a heuristic variant of the algorithm which is faster but often gives just as accurate clustering results.

1 Introduction

Over the past several years, spectral clustering methods have become very popular; see [12] and [14] for an excellent introduction. These methods start with a (nonnegative, symmetric) matrix W which collects the relative similarities between a set of points V to be clustered, and then makes the assumption that in some sense, the cluster indicators should be smooth with respect to W. A simple such notion is that the length of the boundary of the clusters should be small relative to their area. This motivates the definition of the Cheeger cut value of a partition $P = \{V_1, V_2\}$ of V into two pieces given by

$$\mathcal{C}(V_1, V_2) = \frac{\mathrm{Cut}(V_1, V_2)}{\min(|V_1|, |V_2|)},$$

where

$$\mathrm{Cut}(V_1, V_2) = \sum_{i \in V_1, j \in V_2} W_{ij},$$

and $|V|$ is just the cardinality of V. Since finding the optimal Cheeger cut is NP-hard, the Cheeger cut is usually approximated by the second eigenvalue of the combinatorial Laplacian $D - W$, where $D_{ii} = \sum_j W_{ij}$, such that:

$$\frac{1}{2 \max_i D_{ii}} \mathcal{C}^2 \leq \lambda_2 \leq 2\mathcal{C}.$$

*Authors are with the Department of Mathematics, University of California, Los Angeles, USA. Email: {aszlam,xbresson}@math.ucla.edu. This work was supported in part by ONR N00014-03-1-0071 and ONR MURI subcontract from Stanford University.

See [4] for the continuous version, and for the discrete version, see [5]. Note also that the parametric max flow-min cut (a.k.a. graph-cut) was used in [9] to minimize the biased ratio cut $\frac{\mathrm{Cut}(V_1,V_2)}{|V_1|}$, but cannot be used to solve the unbiased ratio cut defined as $Rcut(V_1,V_2) = \frac{\mathrm{Cut}(V_1,V_2)}{|V_1|} + \frac{\mathrm{Cut}(V_1,V_2)}{|V_2|}$, which is NP-hard.
Using the Raleigh quotient formula for the eigenvalue gives

$$\lambda_2 = arg \min_{f \in \mathcal{L}^2(V)} \mathcal{H}_2(f)$$

$$= arg \min_{f \in \mathcal{L}^2(V)} \frac{\sum ||\nabla f||^2}{||f - M(f)||_2^2},$$

where for $p \geq 1$, $||\nabla f||^p$ at i is given by

$$||\nabla f||^p(i) = \sum_j W_{ij} |f(i) - f(j)|^p,$$

and where $M(f)$ is the mean of f. The functional $\mathcal{H}_2$ measures smoothness. It has long been known that L^2 measures of smoothness are not as well suited for dealing with functions with jumps as L^1 measures of smoothness; in image processing, see for example [11]. Very recently in [3] (also see [1]), it was shown that

$$\lim_{p \mapsto 1} \min_f \frac{\sum_i ||\nabla f||^p(i)}{\min_c ||f - c||_p^p} = \min_P \mathcal{C}(P).$$

With this in mind we can relax the problem

$$\min_P \mathcal{C}(P)$$

as follows: for any binary valued function $f = \chi_{V_1}$, $V_1 \subsetneq V$,

$$||f - \mathrm{m}(f)||_1 = \begin{cases} |V_2| & |V_1| > |V_2| \\ |V_1| & |V_1| \leq |V_2|, \end{cases}$$

where $\mathrm{m}(f)$ is the median of f, and V_2 is the compliment in V of V_1. Then

$$\frac{\sum_i ||\nabla f_P||(i)}{||f_P - \mathrm{m}(f_P)||_1} = 2 \frac{\sum_{v_i \in V_1} \sum_{v_j \in V_2} W_{ij}}{\min(|V_1|, |V_2|)}$$

$$= 2\mathcal{C}(V_1, V_2).$$

Thus

$$\min_f \frac{\sum_i ||\nabla f||(i)}{||f - \mathrm{m}(f)||_1} \tag{1}$$

is a relaxation of the Cheeger cut problem, and

$$\min_f \frac{\sum_i ||\nabla f||(i)}{||f - \mathrm{m}(f)||_1} \leq \min_P \mathcal{C}(P).$$

In this work we will show that the inequality (1) is actually an equality, and for any solution f of the relaxed minimization that there is a threshold γ so that the binary function

$$f_\gamma = \begin{cases} 1 & f > \gamma \\ 0 & f \leq \gamma, \end{cases}$$

has the same energy as the minimum cut. A similar approach has been studied in the continuous setting by Strang in [13]. We will then give an algorithm for minimizing the ratio energy which is experimentally efficient, and then another algorithm for minimizing a similiar but easier energy. Finally, we will provide some experiments on the quality of the clusterings given by the algorithms we have presented.

2 Equivalence of the TV problem and the Ratio Cut problem

In this section we fix a set of points V and a similarity matrix W between these points. For a function $f : V \mapsto \mathbb{R}$

$$|f|_{\mathrm{TV}} = \sum_i ||\nabla f||.$$

Note that this is a norm on the set of function modulo constants. As before, denote the median of f by $\mathrm{m}(f)$.

Lemma 2.1. *A function $f : V \mapsto R$ is an extreme point of the TV unit ball if and only if there is a number α such that $f = \alpha\chi_S$, where $S \subsetneq V$.*

Proof. Denote by $\{a_1, ..., a_n\}$ the distinct values of f arranged in increasing order. Let $S_r = \{v : f(v) = a_r\}$, $S_r^+ = \{v : f(v) > a_r\}$, and $S_r^- = \{v : f(v) < a_r\}$ pick indices t and s with $t \neq s$, and let $g = f + \epsilon_s\chi_{S_s} + \epsilon_t\chi_{S_t}$; and $h = f - \epsilon_s\chi_{S_s} - \epsilon_t\chi_{S_t}$, where ϵ_s and ϵ_t will be chosen in a moment. Note that adding $\epsilon_s\chi_{S_s}$ to f changes its total variation by

$$\epsilon_s \left(\sum_{i\in S_s, j\in S_s^-} W_{ij} \; - \sum_{i\in S_s, j\in S_s^+} W_{ij} \right),$$

and adding $\epsilon_t\chi_{S_t}$ to the resulting function changes the total variation by the corresponding expression with S_t, and and as long as ϵ_s and ϵ_t are chosen small enough to not upset the order of the values of f, the changes are independent. Thus by keeping

$$\epsilon_s = -\frac{\sum_{i\in S_s, j\in S_s^-} W_{ij} \; - \sum_{i\in S_s, j\in S_s^+} W_{ij}}{\sum_{i\in S_t, j\in S_t^-} W_{ij} \; - \sum_{i\in S_t, j\in S_t^+} W_{ij}}\epsilon_t,$$

and picking both small enough so that the order of the values does not change, we get that $|g|_{\mathrm{TV}} = |h|_{\mathrm{TV}} = 1$. Then $f = g/2 + h/2$, and so f is not an extreme point of the ball. To prove the converse, let $f = \alpha\chi_S$, and suppose that $\beta g + (1-\beta)h = f$, for some g and h in the TV unit ball on W. Let W^* be the weighted subgraph of W given by

$$W^*_{ij} := \begin{cases} W_{ij} & i \in S \text{ and } j \in S^c \\ 0 & \text{otherwise,} \end{cases}$$

Note that on W^*, still $\beta g + (1-\beta)h = f$; and $|f|_{\mathrm{TV}(W^*)} = 1$. By the sublinearity of the $\mathrm{TV}(W^*)$ norm

$$1 \leq \beta|g|_{\mathrm{TV}(W^*)} + (1-\beta)|h|_{\mathrm{TV}(W^*)},$$

but the choice of g and h show that

$$\beta|g|_{\mathrm{TV}(W^*)} + (1-\beta)|h|_{\mathrm{TV}(W^*)} \leq \beta + (1-\beta) = 1,$$

and so

$$1 = |g|_{\mathrm{TV}(W^*)}$$

and

$$|h|_{\mathrm{TV}(W^*)} = 1.$$

Therefore both h and g are constant on S and S^c, and were thus, up to a constant, multiples of f. □

We will need a slightly sharper version of of Lemma 2.1 below; however, the proof is essentially the same.

Lemma 2.2. *Let W be a weighted graph, and let I_+ and I_- be a partition of V. Let Q be the set of vectors in $\mathbb{R}^n$ such that f is nonnegative in the coordinates I_+, and nonpositive in the coordinates I_-; let B be the TV norm unit ball on Q. The vector f is an extreme points of B if and only if there exists a number α with $f = \alpha\chi_S$, where $S \subsetneq V$.*

Proof. The "if" direction is as above. The proof of the "only if" direction proceeds exactly as in Lemma 2.1 if f takes positive and negative values. If f takes nonnegative values, then the proof above works as long as we choose $a_t > a_s > 0$; and similarly for the nonpositive case. □

Theorem 2.3. *Consider the problem*

$$\lambda = \min_f \frac{|f|_{TV}}{||f - m(f)||_1}.$$

There is a binary valued minimizer, and

$$\lambda = \min_S \frac{Cut(S)}{\min\left(|S|, |S^c|\right)}.$$

Furthermore, for any minimizer f, there is a number γ so that the function

$$f_\gamma = \begin{cases} 1 & f > \gamma \\ 0 & f \leq \gamma, \end{cases}$$

is also a minimizer.

Proof. Suppose f is a minimizer. If $|f|_{\mathrm{TV}} = 0$, the characteristic function of the support of f is binary and also has TV norm zero. If not, because the functional has homogeneity 0, we can rescale f to fix the numerator of the energy as $|f|_{\mathrm{TV}} = 1$; f is thus a maximizer for the denominator, constrained to the TV ball. Because both numerator and denominator are unchanged by the addition of a constant to f, we may restrict attention to f with $\mathrm{m}(f) = 0$. Let I_1 be the indices where $f \leq 0$, and let I_2 be the indices where $f > 0$. Note that any function nonpositive on I_1 and nonnegative on I_2 also has median 0; denote this set of functions by Q. Denote by B the TV norm unit ball on Q. By definition, f is a solution to $\max_B ||f||_1$. The set B is convex, and $||\cdot||_1$ is a convex function on B, so it takes its maximum at an extreme point; by Lemma 2.2, there is a

binary valued maximizer $g = \alpha\chi_S$ for some set S, and the energy of g is exactly $\frac{\mathrm{Cut}(S)}{\min(|S|,|S^c|)}$.

To see the last part of the statement, note that f can be written as $f = \sum \beta_i g_i$ where the g_i are extreme points (and therefore characteristic functions), $\beta_i > 0$, and $\sum \beta_i = 1$. Because the L^1 norm is linear on the quadrant that f lies in, all the g_i are also minimizers. It thus suffices to pick γ to be the value of the greatest valued g_i. □

3 A Split-Bregman algorithm for ratio minimization

In this section, we show how to solve two minimization problems. The first minimzation problem is the following:

$$\min_{f\in\mathbb{R}} \frac{|f|_{\mathrm{TV}}}{||f - \mathrm{m}(f)||_1}$$

which is equivalent to solve the constrained minimization problem [6]:

$$\min_u \max_\lambda \; |f|_{\mathrm{TV}} - \lambda||f||_1 \;\; \text{s.t. } \mathrm{m}(f) = 0.$$

We will use the following iterative process:
Algorithm 1:
While not converged do:
1. $f^{n+1/2} = \arg\min_f \; |f|_{\mathrm{TV}} - \lambda^n||f||_1$ (*)
2. $f^{n+1} = f^{n+1/2} - m(f^{n+1/2})$
3. $\lambda^{n+1} = |f^{n+1}|_{\mathrm{TV}}/||f^{n+1}||_1$
End while.
The minimization problem (*) belongs to the class of $L1$-regularized problems. Several schemes have been introduced in the literature to solve this class of problems. In this work, we will use the efficient split-Bregman method originally introduced in [8] to solve the TVL2 problem and the compressed sensing problem and extended in [16] for the non-local/graph framework. However, the minimization (*) is slightly different from [8, 16] for two reasons. Firstly, it uses a negative $L1$ term, i.e. $-||f||_1$, which requires a new minimizing operator different from wavelet shrinkage. Secondly, the graph-based TV norm is anisotropic unlike the anisotropic graph-based TV defined in [16]. We now develop the numerical scheme. Energy (*) can be written as:

$$\min_f \; \sum_i \sum_{j\sim i} w_{ij}|f_j - f_i| - \lambda|f_i|.$$

We introduce two splitting variables to deal with the two $L1$ norms:

$$\min_{f,d,e} \; \sum_i \sum_{j\sim i} w_{ij}|d_{ij}| - \lambda|e_i| \;\; \text{s.t. } d_{ij} = f_j - f_i, \; e_i = f_i.$$

Then, the Bregman iteration method [2, 10] is used to solve the previous constrained minimization problem as follows:

$$\begin{cases} (f^{k+1}, d^{k+1}, e^{k+1}) & = & \arg\min_{f,d,e} \sum_i \sum_{j\sim i} w_{ij}|d_{ij}| - \lambda|e_i| + \\ & & \frac{\lambda_1}{2}(d_{ij} - (f_j - f_i) - b_{ij}^k)^2 + \frac{\lambda_2}{2}(e_i - f_i - c_i^k)^2 \\ b_{ij}^{k+1} & = & b_{ij}^k + f_j^{k+1} - f_i^{k+1} - d_{ij}^{k+1} \\ e_i^{k+1} & = & e_i^k + f_i^{k+1} - e_i^{k+1} \end{cases} \quad (2)$$

The first line is solved by alternate minimization since the total energy is convex. First, the Euler-Lagrange equation of the minimization problem w.r.t. f defined as:

$$\min_f \sum_i \sum_{j\sim i} \frac{\lambda_1}{2}(d_{ij} - (f_j - f_i) - b_{ij}^k)^2 + \frac{\lambda_2}{2}(e_i - f_i - c_i^k)^2$$

is given by:

$$\lambda_1 \sum_{j\sim i}(d_{ij} - d_{ji} - 2(f_j - f_i) - b_{ij}^k + b_{ji}^k) + \lambda_2(f_i - e_i + c_i^k) = 0$$

whose solution, we call f^{k+1}, is given by a few Gauss-Seidel iterations as follows:

$$f_i^{k+1,m+1} = \frac{1}{2\lambda_1 \sum_{j\sim i} + \lambda_2}\Big(- \lambda_1 \sum_{j\sim i}(d_{ij} - d_{ji} - b_{ij}^k - 2f_j^m + b_{ji}^k) + \lambda_2(e_i - c_i^k)\Big) \quad (3)$$

starting from $f^{k+1,m=0} = f^k$.
The minimization process (2) w.r.t. d is defined as:

$$\min_d \sum_i \sum_{j\sim i} w_{ij}|d_{ij}| + \frac{\lambda_1}{2}(d_{ij} - (f_j^{k+1} - f_i^{k+1}) - b_{ij}^k)^2$$

whose solution is known as the shrinkage operator [7]:

$$d_{ij}^{k+1} = \frac{f_j^{k+1} - f_i^{k+1} + b_{ij}^k}{|f_j^{k+1} - f_i^{k+1} + b_{ij}^k|} \max\Big(|f_j^{k+1} - f_i^{k+1} + b_{ij}^k| - \frac{w_{ij}}{\lambda_1}, 0\Big) \quad (4)$$

Finally, the minimization process (2) w.r.t. e is:

$$\min_e \sum_i -\lambda|e_i| + \frac{\lambda_2}{2}(e_i - u_i - c_i^k)^2 \quad (5)$$

whose solution is different from the minimizing solution given by the shrinkage operator as in (4) because of the minus sign in front of the $L1$ norm, i.e. $-|.|$. However, the minimizing solution of (5) has also a nice closed-form solution defined as:

$$e_i^{k+1} = f_i^{k+1} + c_i^k + \frac{f_i^{k+1} + c_i^k}{|f_i^{k+1} + c_i^k|}\frac{\lambda}{\lambda_2} \quad (6)$$

Alternating (3), (4) and (6) until convergence provides the solution $f^{n+1/2} = f^{k\to\infty}$.

The second minimization problem that we want to solve is the following (this iterative scheme is faster than the previous one):
Algorithm 2:
While not converged do:
1. Minimize a few steps $|f|_{\mathrm{TV}}$, call $f^{n+1/3}$ the solution after a few steps (**)
2. $f^{n+1/2} = f^{n+1/3} - mean(f^{n+1/3})$
3. Compute f^{n+1} s.t. $\sum_i \ f_i^{n+1/2} = ct$
End while.
The minimization step (**) can be done using a similar approach as the first model. Indeed, a few steps of minimization of (**) can be done with this iterative scheme:

$$\begin{cases} (f^{k+1}, d^{k+1}) & = & \arg\min_{f,d,e} \ \sum_i \sum_{j\sim i} w_{ij}|d_{ij}| + \frac{\lambda_1}{2}(d_{ij} - (f_j - f_i) - b_{ij}^k)^2 \\ b_{ij}^{k+1} & = & b_{ij}^k + f_j^{k+1} - f_i^{k+1} - d_{ij}^{k+1} \end{cases}$$

The minimization w.r.t. f is done by a few Gauss-Seidel iterations as follows:

$$f_i^{k+1,m+1} = -\frac{1}{2\sum_{j\sim i}} \sum_{j\sim i} (d_{ij} - d_{ji} - 2f_j^m - b_{ij}^k + b_{ji}^k)$$

starting from $f^{k+1,m=0} = f^k$.
Finally, the minimizing solution w.r.t. d is:

$$d_{ij}^{k+1} = \frac{f_j^{k+1} - f_i^{k+1} + b_{ij}^k}{|f_j^{k+1} - f_i^{k+1} + b_{ij}^k|} \max\left(|f_j^{k+1} - f_i^{k+1} + b_{ij}^k| - \frac{w_{ij}}{\lambda_1}, 0\right)$$

4 Experiments

In all experiments we use a 10-NN graph with the self-tuning weights as in [15], and the neighbor parameter set to 10. The optimization parameters for *algorithm 1* for all expriments are fixed as follows: $\lambda_1 = 1$, $\lambda_2 = 1.5$, the number of n iterations is 15, the number of k iterations is 30 and the number of m is 5. The optimization parameters for *algorithm 2* for all expriments are fixed as follows: $\lambda_1 = 0.3$, the number of n iterations is 50, the number of k iterations is 1 and the number of m is 15. Both methods are intialized by multiplying the indicator of a random point by the averaging-normalized weight matrix $D^{-1}W$ (where $D_{ii} = \sum_j W_{ij}$) 1000 times.

4.1 MNIST

We test on the combined training and test samples from the MNIST dataset, available at `http://yann.lecun.com/exdb/mnist/`. This data set consists of 70000 28×28 images of handwritten digits, 0 through 9. The data was preprocessed by projecting onto 50 principal components.

mode/true	0	1	2	3	4	5	6	7	8	9
0	6857	2	30	3	4	26	23	3	9	12
1	4	3642	8	3	3	1	9	13	26	1
1	1	4013	3	4	5	0	5	10	20	5
2	5	109	6855	48	4	1	1	36	8	5
3	3	26	15	6931	4	6170	14	5	257	170
4	1	18	4	1	6612	4	6	20	10	56
6	18	4	5	4	15	72	6815	0	19	3
7	3	12	34	24	10	3	0	7122	10	62
8	5	14	30	102	6	14	3	11	6448	33
9	6	37	6	21	161	22	0	73	18	6611

Figure 1: Confusion matrix for the clustering of MNIST using the relaxed Cheeger model (Algorithm 1). Each row is a cluster; the number in the leftmost column of each row is the dominant label of that cluster. The 3's and 5's are merged, but otherwise the clustering is very accurate. The total computation time (not including constructing the weights) was 2710 seconds ($\sim$ 45 minutes).

The goal in this data set is to discover the 10 digit classes. The methods described above give a binary clustering, so in order to obtain 10 clusters, we iteratively subdivide in the standard way. That is, we tentatively divide each of the l current clusters in two, and keep the division minimizing the sum of the Cheeger cut values between each cluster and the union of all the others; now we have $l + 1$ clusters, and we repeat till we have 10.

The confusion matrices for the results of the clustering using the relaxed Cheeger algorithm and the constrained TV algorithm are presented in Figures 1 and 4.1

4.2 Two moons

We construct the two moons data set as in [3]. We take the half of a circle of radius one in $\mathbb{R}^2$ with positive second coordinate sampled with a thousand points, and the half with negative second coordinate also sampled at a thousand points, but shifted 1 in the positive first coordinate direction, and .5 in the positive second coordinate direction. The data set is embedded in $\mathbb{R}^{100}$, and Gaussian noise with $\sigma = .02$ is added.

We calculate the clustering with the relaxed Cheeger cut model and the constrained TV model over 100 instantations of the data set. The average errors and run times are in Figure 4.2

mode/true	0	1	2	3	4	5	6	7	8	9
0	6846	4	28	2	3	17	15	7	4	16
1	0	3983	1	0	3	0	2	10	11	4
1	3	3628	5	2	6	1	7	17	16	1
2	6	82	6834	46	7	1	2	30	9	5
3	6	58	20	6948	2	6194	48	5	348	165
4	0	18	5	2	6598	4	3	23	12	67
6	23	5	6	1	49	45	6786	0	12	9
7	1	13	54	30	9	3	0	7107	11	52
8	13	57	30	83	5	24	13	6	6379	20
9	5	29	7	27	142	24	0	88	23	6619

Figure 2: Confusion matrix for the clustering of MNIST using the constrained TV model (Algorithm 2). Each row is a cluster; the number in the leftmost column of each row is the dominant label of that cluster. The 3's and 5's are merged, but otherwise the clustering is very accurate; the accuracy is only slightly less than the clustering using Algorithm 1. The total computation time (not including constructing the weights) was 428 seconds ($\sim$ 7 minutes).

References

[1] S. Amghibech. Eigenvalues of the discrete p-laplacian for graphs. *Ars Comb.*, 67, 2003.

[2] L. Bregman. The Relaxation Method of Finding the Common Points of Convex Sets and Its Application to the Solution of Problems in Convex Programming. *USSR Computational Mathematics and Mathematical Physics*, 7:200–217, 1967.

[3] T. Buhler and M. Hein. Spectral Clustering based on the graph p-Laplacian. In *International Conference on Machine Learning (ICML)*, 2009.

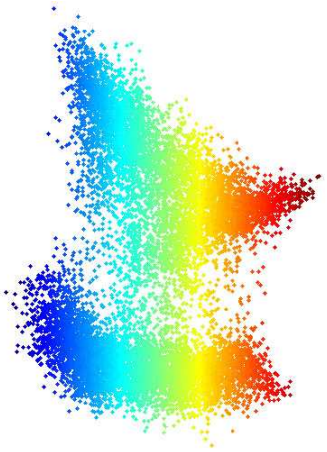
[4] J Cheeger. A lower bound for the smallest eigenvalue of the laplacian. In RC Gunning, editor, *Problems in Analysis*, pages 195–199. Princeton Univ. Press.

[5] F. R. K. Chung. *Spectral graph theory*, volume 92 of *CBMS Regional Conference Series in Mathematics*. Published for the Conference Board of the Mathematical Sciences, Washington, DC, 1997.

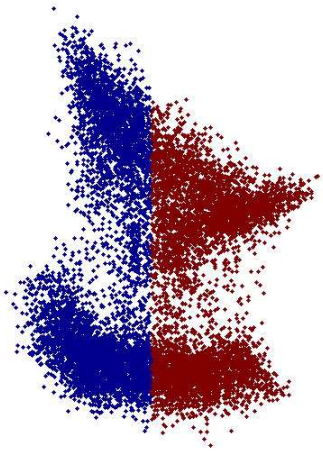
[6] W. Dinkelbach. On Nonlinear Fractional Programming. *Management Science*, 13:492–498, 1967.

[7] D. Donoho. De-Noising by Soft-Thresholding. *IEEE Transactions on Information Theory*, 41(33):613–627, 1995.

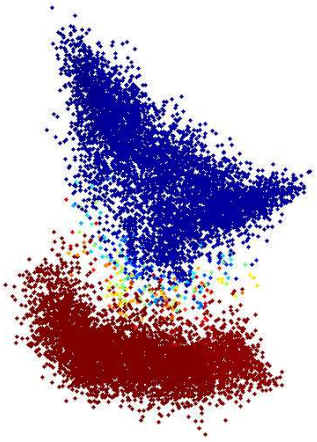
[8] T. Goldstein and S. Osher. The Split Bregman Method for L1-Regularized Problems. *SIAM Journal on Imaging Sciences*, 2(2):323–343, 2009.



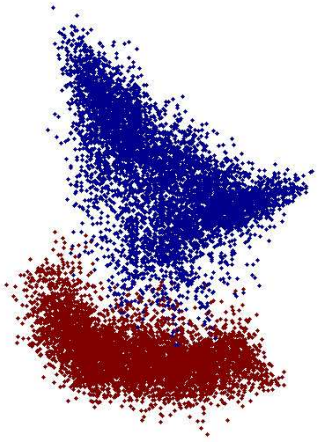
(a) Second eigenvector of the Laplacian.



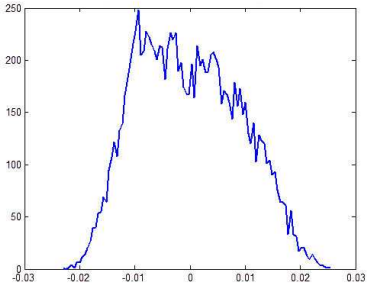
(b) Optimal Cheeger cut obtained by thresholding the second eigenvector of the Laplacian, value is 0.28.



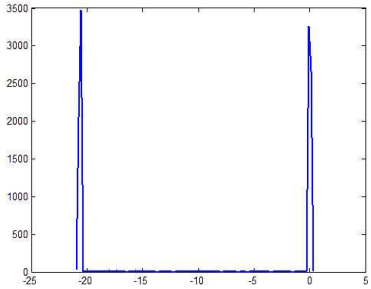
(c) Output of algorithm 1.



(d) Optimal Cheeger cut obtained by thresholding the output of algorithm 1, value is 0.1.



(e) Histogram of the second eigenvector of the Laplacian.



(f) Histogram of the solution of the output of algorithm 1.

Figure 3: Results for the 4's and 9's in the MNIST dataset, 11791 points in $28^2 (= 784)$ dimensions.

$\frac{\|f\|_{\mathrm{TV}}}{\|\|f - m(f)\|\|_1}$	8.2
$\|f\|_{\mathrm{TV}}$, $\sum f = 0$, $\sum \|f\|=1$	13.8
2nd Eigenvector method	16.4

Figure 4: Average error in percent on the two moons data set over 100 instances.

$\frac{\|f\|_{\mathrm{TV}}}{\|\|f - m(f)\|\|_1}$	5.6
$\|f\|_{\mathrm{TV}}$, $\sum f = 0$, $\sum \|f\|=1$	.6
2nd Eigenvector method	.05

Figure 5: Average runtime in seconds on the two moons data set over 100 instances, not counting time spent building the weights.

[9] V. Kolmogorov, Y. Boykov, and C. Rother. Applications of Parametric Maxflow in Computer Vision. In *International Conference on Computer Vision*, pages 1–8, 2007.

[10] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin. An Iterative Regularization Method for Total Variation-based Image Restoration. *SIAM Multiscale Modeling and Simulation*, 4:460–489, 2005.

[11] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Phys. D*, 60(1-4):259–268, 1992.

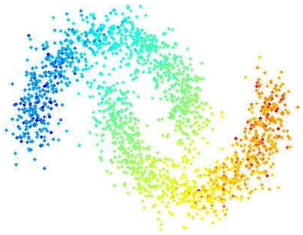
[12] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Tran PAMI*, 22(8):888–905, 2000.

[13] G. Strang. Maximal Flow Through A Domain. *Mathematical Programming*, 26:123–143, 1983.

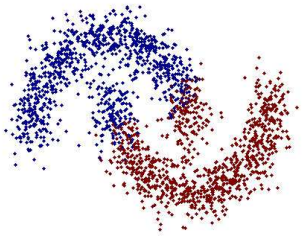
[14] U. von Luxburg. A tutorial on spectral clustering. Technical Report 149, 08 2006.

[15] L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. In *Advances in Neural Information Processing Systems 17 (NIPS 2004)*, 2004.

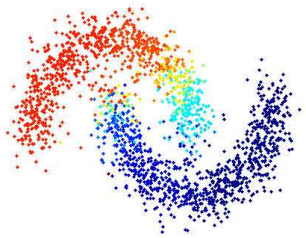
[16] X. Zhang, M. Burger, X. Bresson, and S. Osher. Bregmanized Nonlocal Regularization for Deconvolution and Sparse Reconstruction. *CAM Report 09-03*, 2009.



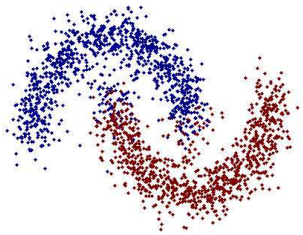
(a) Second eigenvector of the Laplacian.



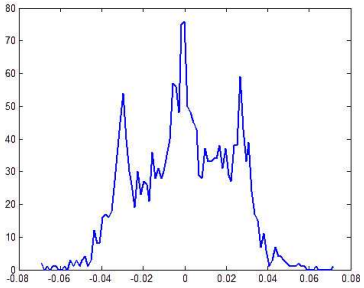
(b) Optimal Cheeger cut obtained by thresholding the second eigenvector of the Laplacian, value is 0.60.



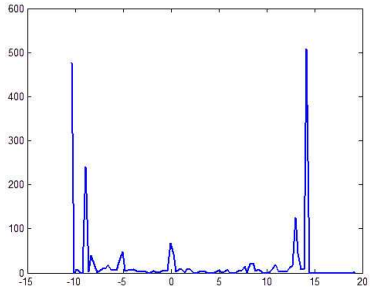
(c) Output of algorithm 1.



(d) Optimal Cheeger cut, value is 0.41.



(e) Histogram of the second eigenvector of the Laplacian.



(f) Histogram of the output of algorithm 1.

Figure 6: Results for the two moons dataset, 2000 points in 100 dimensions.