Asymmetric Cheeger cut and application to multi-class unsupervised clustering

Xavier Bresson * Thomas Laurent †

April 8, 2012

Abstract

Cheeger cut has recently been shown to provide excellent classification results for two classes. Whereas the classical Cheeger cut favors a 50-50 partition of the graph, we present here an asymmetric variant of the Cheeger cut which favors, for example, a 10-90 partition. This asymmetric Cheeger cut provides a powerful tool for unsupervised multi-class partitioning. We use it in recursive bipartitioning to detach one after the other each of the classes. This asymmetric recursive algorithm handles equally well any number of classes, as opposed to symmetric recursive bipartitioning which is naturally better suited for 2^m classes. We obtain an error classification rate of 2.35% and 4.07% for MNIST and USPS benchmark datasets respectively, drastically improving the former 11.7% and 13% error rate obtained in the literature with symmetric Cheeger cut bipartitioning algorithms.

1 Introduction

Partitioning data points into sensible groups is a fundamental problem in machine learning and science in general. Given a set of data points $V = \{x_1, \ldots, x_n\}$ and similarity weights $\{w_{i,j}\}_{1 \leq i,j \leq n}$, an efficient approach is to find a balanced cut of the graph of the data. A popular balanced cut is the Cheeger cut [5] defined as a minimizer of

$$\mathcal{C}(S) = \frac{\sum_{x_i \in S} \sum_{x_j \in S^c} w_{i,j}}{\min\{|S|, |S^c|\}} \tag{1}$$

over all the subset $S \subset V$. Here $|S|$ is the number of points in S. The above balanced cut problem is NP-hard and approximate solutions are therefore needed. To this aim spectral clustering methods are widely used. They consist in minimizing the Rayleigh quotient

$$\mathcal{E}(f) = \frac{\frac{1}{2} \sum_{i,j} w_{i,j} |f_i - f_j|^2}{\sum_i |f_i - m_2(f)|^2} \tag{2}$$

over the function $f : \Omega \to \mathbb{R}$. Here $m_2(f)$ stands for the mean of f. The minimizer of (2) is the first nontrivial eigenvector of the Graph Laplacian matrix and therefore it can be computed very efficiently with standard linear algebra software for very large data set. Unfortunately the approximation provided by spectral clustering methods can be weaker than the solutions of (1) and therefore can fail to cluster somewhat benign problems; for example the two-moon example.

The authors of [3] introduce the ℓ^p, $1 < p < 2$, equivalent of (2)

$$\mathcal{E}(f) = \frac{\frac{1}{2} \sum_{i,j} w_{i,j} |f_i - f_j|^p}{\sum_i |f_i - m_p(f)|^p}, \tag{3}$$

where $m_p(f) = \text{argmin}_c \sum_i |f_i - c|^p$. For p close to 1, minimizing this energy gives a better approximation of the Cheeger cut than the one provided by (2). In [16] and subsequently in [10, 11, 2] it was proposed

*Department of Computer Science, City University of Hong Kong, Hong Kong (xbresson@cityu.edu.hk).
†Department of Mathematics, University of California Riverside, Riverside CA 92521 (laurent@math.ucr.edu)

to use ℓ^1 optimization techniques from image processing to directly work with the ℓ^1 problem

$$\mathcal{E}(f) = \frac{\frac{1}{2}\sum_{i,j} w_{i,j}|f_i - f_j|}{\sum_i |f_i - m_1(f)|} \tag{4}$$

where $m_1(f)$ is the median of f. The minimum of this energy is achieved by the indicator function of the Cheeger cut, therefore minimizing (4) provides a tight relaxation of the Cheeger cut problem (1). Unfortunately, there is no algorithm that guarantees to find solutions to this ℓ^1 relaxation problem. Nevertheless, experiments in [16, 10] report high quality cuts for data clustering, outperforming spectral clustering methods. Solving problem (4) is not as fast as solving problem (2), but recent advances in ℓ^1 optimization offer powerful tools to design fast and accurate algorithms to solve the minimization problem (4). Other ℓ^1 related approach based on phase field modeling has been introduced in [1]. We also notice that total variation-based algorithms on graph have been used in image processing in the context of image denoising with intensity patches [8, 12].

Minimizing (1) favors a 50\50 partition of the graph. In this work we introduce an asymmetric variant of the Cheeger cut which favors a $\theta\backslash(1-\theta)$ partition. This asymmetric Cheeger cut provides a powerful tool for multi-class data partitioning. It allows one to detach each class one after the other as opposed to recursively dividing the data set into two equal groups of classes as it is done with symmetric cut. Whereas recursive bipartitioning with symmetric cut is naturally better suited for 2^m classes, our asymmetric partitioning algorithm handles equally well any number of classes. In a recent work [11], other interesting variants of the Cheeger cut have been investigated.

In section 2 we present the asymmetric Cheeger cut as well as its tight relaxation. In section 3 we present our asymmetric recursive bipartitioning algorithm and we illustrate on a simple example how it outperforms symmetric recursive bipartitioning algorithm when the number of class is not dyadic (i.e. a power of two). In section 4 we introduce the optimization scheme and in section 5 we provide experimental results on the MNIST and USPS datasets, demonstrating drastic improvements compared to previous Cheeger cut bipartitioning algorithms [16, 10, 11, 2].

2 Asymmetric Cheeger cut

Fix a set of points $V = \{x_1, \ldots, x_n\}$ and a nonnegative symmetric matrix $\{w_{i,j}\}_{1\le i,j\le n}$ which collects the relative similarities between the points of V. Let $\lambda > 0$ and $\theta = (1+\lambda)^{-1}$. The λ-asymmetric Cheeger cut problem is:

$$\text{minimize } \mathcal{C}_\lambda(S) = \frac{\sum_{x_i\in S}\sum_{x_j\in S^c} w_{i,j}}{\min\{\lambda|S|, |S^c|\}} \tag{5}$$

over all the subset $S \subset V$. Note that in order to maximize the denominator, one need to choose a subset S which satisfies

$$\lambda|S| = |S^c|, \text{ or equivalently } |S| = \theta n$$

from which we see that the λ-asymmetric Cheeger cut favors a $\theta\backslash(1-\theta)$ partition of the graph. Note that the energy $\mathcal{C}_\lambda$ is asymmetric in the sense that $\mathcal{C}_\lambda(S) \neq \mathcal{C}_\lambda(S^c)$.

Problem (5) has the following tight continuous relaxation:

$$\text{minimize } \mathcal{E}_\lambda(f) = \frac{\frac{1}{2}\sum_{i,j} w_{i,j}|f_i - f_j|}{\inf_c \sum_i |f_i - c|_\lambda} = \frac{\frac{1}{2}\sum_{i,j} w_{i,j}|f_i - f_j|}{\sum_i |f_i - m_\lambda(f)|_\lambda} \tag{6}$$

over the nonconstant functions $f : V \to \mathbb{R}$. Here the asymmetric absolute value $|\cdot|_\lambda$ is defined by

$$|x|_\lambda = \begin{cases} x & \text{if } x \geq 0 \\ -\lambda x & \text{if } x < 0 \end{cases}$$

and the λ-median $\mu_\lambda(f)$ is defined by

$$\mu_\lambda(f) = \min\{c \in \text{range}(f) \text{ satisfying } |\{f \le c\}| \ge \theta n\}. \tag{7}$$

In (6) the notation f_i stands for $f(x_i)$ and in (7) the notation $\{f \le c\}$ stands for $\{x_i \in V : f(x_i) \le c\}$. The following theorem precises in which sense the relaxation (6) is tight.

Theorem 1. *Let S^* be a global minimizer of $\mathcal{C}_\lambda$ over the subset $S \subset V$. Then for any $a < b$, the binary function*

$$f^*(x_i) = \begin{cases} a & \text{if } x_i \in S^* \\ b & \text{if } x_i \in (S^*)^c \end{cases} \tag{8}$$

is a global minimizer of $\mathcal{E}_\lambda$ over the non-constant functions $f : V \to \mathbb{R}$.

The proof of this result, which can be found in the appendix, is standard and follows the same steps than in [6]. See also [16] for an alternative proof based on [14].

3 Asymmetric recursive bipartitioning

Suppose that we want to divide a data set $V = \{x_1, \ldots, x_n\}$ into k groups of comparable size – that is the size of every group is somewhere around n/k. By using the asymmetric Cheeger cut with $\theta = 1/k$, we first try to detach from the data set a group of approximatively n/k points. We then repeat the process with the remaining points (and with suitable θ in order to keep aiming for groups of approximatively n/k points).

In practice the above algorithm may detach two or three groups at once. In order to remedy this problem, each extracted group V_i is revisited and tested for possible cutting. This leads to the recursive bipartitioning algorithm described in 1.

Algorithm 1 Asymmetric Cheeger cut recursive bipartitioning algorithm

---

while number of clusters $l < k$ **do**

Let $V_1, \ldots, V_l$ be the clusters and let $n_1, \ldots, n_l$ be their size.

Tentatively divide each cluster V_i with a λ-asymmetric Cheeger cut where λ satisfies:

$$\frac{n_i}{\lambda + 1} = \frac{n}{k}, \quad \text{that is} \quad \lambda = k(n_i/n) - 1.$$

Among all these l divisions, keep the one with smallest λ-asymmetric Cheeger cut value $\mathcal{C}_\lambda$.

Now we have $l + 1$ clusters.

end while

Compared to previous symmetric Cheeger–based bipartioning algorithms which are naturally better suited for a dyadic number classes [16, 10, 11, 2], our asymmetric bipartitioning algorithm allows us to handle arbitrary number of classes and greatly outperforms symmetric bipartitioning algorithms when the number of classes is not 2^m. This is illustrated on a five-moon example on Figure 1. Each moon has 1000 data points in $\mathbb{R}^{100}$.

4 Algorithm

This section presents an algorithm for

$$\min_{f:V\to\mathbb{R}} \frac{\sum_{i,j} w_{i,j}|f_i - f_j|}{\sum_i |f_i - m_\lambda(f)|_\lambda}. \tag{9}$$

No algorithm can guarantee to compute global minimizers of (9) as the problem is non-convex. However, recent advances in ℓ^1 optimization offer powerful tools to design fast and accurate algorithms to solve problems of the form (9). We develop here an algorithm based on [7, 17, 15]. Let $T(f) := \sum_{i,j} w_{i,j}|f_i - f_j|$ and $L(f) := \sum_i |f_i|_\lambda$. Observe now that minimizing (9) is equivalent to:

$$\min_{f:V\to\mathbb{R}} \frac{T(f)}{L(f)} \quad \text{s.t.} \quad m_\lambda(f) = 0, \tag{10}$$

as the ratio energy is unchanged by adding a constant. So the minimization problem can be restricted to functions with zero λ-median. The next step applies the method of Dinkelbach [7] to replace the ratio

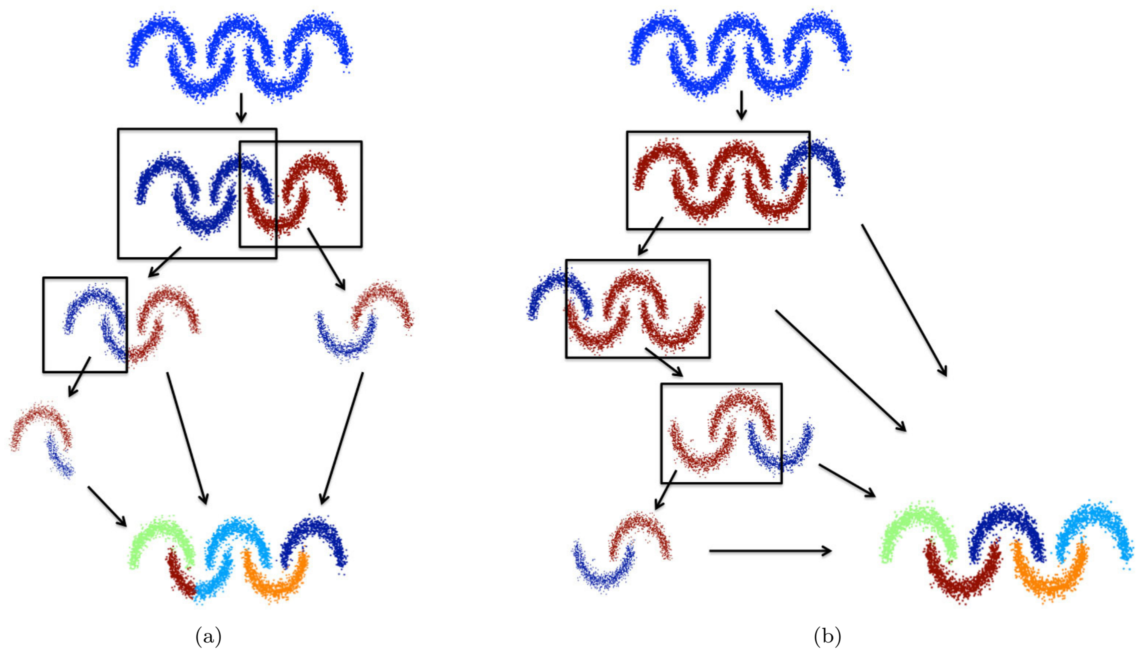

Figure 1: Comparison between symmetric and asymmetric Cheeger cut recursive bipartioning algorithms. The asymmetric algorithm aims at extracting one class at a time as opposed to the symmetric algorithm which divides at each stage every group into two sub-groups of comparable size. In the above five-moon example, the symmetric algorithm fails because five is not a dyadic integer.

minimization problem into a sequence of parametric problems:

$$\left\{ \begin{array}{lcl} f^{k+1} & = & \arg\min_f \; T(f) - \eta^k L(f) \text{ s.t. } m_\lambda(f) = 0 \\ \eta^{k+1} & = & \frac{T(f^{k+1})}{L(f^{k+1})} \end{array} \right. \tag{11}$$

The previous iterative scheme is not guaranteed to converge to a solution of (9) because of the nonlinear λ-median constraint. Without considering the median constraint, the ℓ^1 minimization problem in (11) is a minimization problem of a difference of convex functions, which can be solved accurately using a proximal method as proposed e.g. in [17, 15]. We propose the following approximate algorithm to minimize $T(f) - \eta L(f)$ s.t. $m_\lambda(f) = 0$. The implicit explicit gradient flow is:

$$\frac{f^{n+1} - f^n}{\tau^n} = -\big(\partial T(f^{n+1}) - \eta \partial L(f^n)\big) \quad \text{s.t.} \quad m_\lambda(f) = 0, \tag{12}$$

or equivalently $\frac{f^{n+1} - (f^n + \tau^n \eta \partial L(f^n))}{\tau^n} = -\partial T(f^{n+1})$ s.t. $m_\lambda(f) = 0$, which leads to the iterative scheme:

$$\left\{ \begin{array}{lcl} e^{n+1} & = & f^n + \tau^n \eta \partial L(f^n) \\ f^{n+1} & = & \arg\min_f \; \{ \; T(f) + \frac{1}{2\tau^n} ||f - e^{n+1}||_2^2 \;\; \text{s.t.} \;\; m_\lambda(f) = 0 \; \}. \end{array} \right. \tag{13}$$

The first step of the above scheme is simply given by $e^{n+1} = f^n + \tau^n \eta \, \mathrm{sign}_\lambda(f^n)$ where

$$\mathrm{sign}_\lambda(x) = \begin{cases} -\lambda & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}.$$

Without the λ-median constraint the second step would be a standard ROF problem [13] that can be solved efficiently using approaches such as augmented Lagrangian method [9] or primal-dual method [4]. We artificially enforce the λ-median constrain by substracting at each iteration of the ROF algorithm the λ-median from the current function. To summarize, the proposed algorithm to solve the asymmetric Cheeger minimization problem (9) is given by Algorithm 2.

Algorithm 2 Asymmetric Cheeger cut (9)

$f^{k=0}$ indicator function of a random data point
while outer loop not converged **do**
  f^{k+1} given by inner loop
  while inner loop not converged **do**
    $e^{n+1} = f^n + \tau^n \eta^k \mathrm{sign}_\lambda(f^n)$
    $f^{n+1} = \arg\min_f \{ T(f) + \frac{1}{2\tau^n} ||f - e^{n+1}||_2^2 \;\; \text{s.t.} \;\; m_\lambda(f) = 0 \}$
  end while
  $\eta^{k+1} = \frac{T(f^{k+1})}{L(f^{k+1})}$
end while

	Asymmetric algorithm	Symmetric algorithm
MNIST	2.35%	11.72%
USPS	4.07%	28.49%

Table 1: Comparison between the asymmetric recursive bipartioning algorithm and the symmetric recursive bipartioning algorithm with all 10 classes.

	Asymmetric algorithm	Symmetric algorithm
8-class MNIST	2.31%	2.28%
8-class USPS	4.43%	4.32%

Table 2: Comparison between the asymmetric recursive bipartioning algorithm and the symmetric recursive bipartioning algorithm with 2^3 classes (the 0s and 1s were taken out).

5 Experiments

In all experiments we use a 10 nearest neighbors graph with the self-tuning weights as in [18] (the neighbor parameter in the self-tuning is set to 7 and the universal scaling to 1).

We test our asymmetric Cheeger cut recursive bipartioning algorithm on the MNIST and USPS datasets. The MNIST dataset is available at http://yann.lecun.com/exdb/mnist/. This dataset consists of 70,000 28×28 images of handwritten digits, 0 through 9. Each digit is approximatively equally represented. Note that we combine the training and test samples. The data was preprocessed by projecting onto 50 principal components. The USPS dataset is available at http://www-stat-class.stanford.edu/~tibs/ElemStatLearn/. This dataset consists of 9,298 16×16 images of handwritten digits, 0 through 9. The 0s and the 9s are twice more represented than other digits. We also combine the training and test samples. The data was not preprocessed.

Tables 1 and 2 compare the asymmetric Cheeger cut recursive bipartioning algorithms, Algorithm 1, and the symmetric algorithm (that is Algorithm 1 where λ is always equal to 1). Table 1 consider the original MNIST and USPS datasets with 10 classes and Table 2 consider modified MNIST and USPS datasets with only 2^3 classes (the 0s and 1s were taken out). Note that the asymmetric and symmetric algorithms perform comparably for 2^3 classes, but the asymmetric algorithm greatly outperforms the symmetric one when the original datasets are considered.

Tables 3 and 4 show the confusion matrix for the MNIST and USPS using the asymmetric recursive bipartioning algorithm.

Recently reported results in multi-class unsupervised Cheeger-based classification [16, 10, 11, 2] obtain error rates above 11.7% for MNIST and 13% for USPS. Therefore, the 2.35% and 4.07% error rates reported in this work represent a significant improvement.

mode/true	0	1	2	3	4	5	6	7	8	9
0	6853	1	19	1	3	11	12	6	3	13
1	3	7631	6	3	5	0	4	17	30	5
2	8	160	6860	37	3	3	3	27	12	3
3	0	2	11	6901	1	59	0	1	45	103
4	0	19	4	1	6650	7	6	11	10	50
5	4	1	0	45	0	6130	31	0	19	14
6	20	4	6	1	20	42	6806	0	10	5
7	1	12	58	37	11	3	0	7153	8	29
8	10	5	21	73	5	27	14	6	6664	29
9	4	42	5	42	126	31	0	72	24	6707

Table 3: The confusion matrix for the clustering of MNIST using the asymmetric recursive bipartioning algorithm. Compared to the confusion matrix of the symmetric recursive bipartioning algorithm available in [16], the asymmetric algorithm does not merge the 4s and 9s, producing an accurate classification result.

mode/true	0	1	2	3	4	5	6	7	8	9
0	1250	0	0	3	0	1	0	2	0	0
1	3	889	7	2	3	3	1	2	0	5
2	0	5	775	0	5	0	1	30	2	1
3	3	6	0	789	2	4	13	4	28	1
4	0	1	23	1	687	3	0	8	1	3
5	3	1	0	6	3	810	0	0	0	4
6	1	9	3	2	0	0	758	5	14	0
7	0	3	9	0	2	5	2	649	2	0
8	9	1	2	47	4	0	17	2	774	1
9	0	14	5	2	10	8	0	6	0	1538

Table 4: The confusion matrix for the clustering of USPS using the asymmetric recursive bipartioning algorithm. The asymmetric algorithm produces an accurate classification result.

6 Appendix

We first enumerate some elementary properties of the λ-median $m_\lambda = m_\lambda(f)$:

Lemma 1. *(i)* $m_\lambda \in \mathit{argmin}_c \sum_i |f_i - c|_\lambda$.

(ii) $\lambda|\{f < m_\lambda\}| < |\{f \geq m_\lambda\}|$ *and* $\lambda|\{f \leq m_\lambda\}| \geq |\{f > m_\lambda\}|$.

(iii) $\lambda|\{f < c\}| < |\{f \geq c\}|$ *for all* $c < m_\lambda$ *and* $\lambda|\{f < c\}| \geq |\{f \geq c\}|$ *for all* $c > m_\lambda$.

Proof. Let $\mathrm{range}(f) = \{c_1, \ldots, c_l\}$ with $c_1 < c_2 < \ldots < c_l$. Also let $n_k = |\{f \leq c_k\}| = |\{f < c_{k+1}\}|$. Clearly $0 < n_1 < n_2 < \ldots < n_l = n$. From the definition of the λ-median (7), we see that $m_\lambda = c_{k_0}$ where $\frac{n}{\lambda+1} \in (n_{k_0-1}, n_{k_0}]$. So we have

$$\begin{aligned} \lambda n_{k_0-1} < n - n_{k_0-1} \quad &\text{and} \quad \lambda n_{k_0} \geq n - n_{k_0} \\ \lambda|\{f < c_{k_0}\}| < |\{f \geq c_{k_0}\}| \quad &\text{and} \quad \lambda|\{f \leq c_{k_0}\}| \geq |\{f > c_{k_0}\}|. \end{aligned}$$

This proof (ii). Statement (iii) is direct consequence of (ii). We now turn to the proof of (i). Define the convex function $\Phi(c) = \sum_i |f_i - c|_\lambda$. For $\epsilon > 0$ small enough we have

$$\begin{aligned} \Phi(c_{k_0} + \epsilon) &= \epsilon\left(\lambda|\{f \leq c_{k_0}\}| - |\{f > c_{k_0}\}|\right) \geq 0 \\ \Phi(c_{k_0} - \epsilon) &= \epsilon\left(|\{f \geq c_{k_0}\}| - \lambda|\{f < c_{k_0}\}|\right) > 0 \end{aligned}$$

and therefore c_{k_0} is a global minimizer. □

We now turn to the proof of Theorem 1.

Proof of Theorem 1. Let $h_\lambda = \min_{S\subset V} \mathcal{C}_\lambda(S)$ and let $f : V \to \mathbb{R}$ be a nonconstant function with λ-median m_λ. Then, following [6, Theorem 2.9],

$$\frac{1}{2}\sum_{i,j} w_{i,j}|f_i - f_j| = \int_{-\infty}^{+\infty} \mathrm{Cut}(\{f < r\}, \{f \geq r\})\, dr \tag{14}$$

$$= \int_{-\infty}^{m_\lambda} \frac{\mathrm{Cut}(\{f < r\}, \{f \geq r\})}{\lambda|\{f < r\}|} \lambda|\{f < r\}|dr + \int_{m_\lambda}^{+\infty} \frac{\mathrm{Cut}(\{f < r\}, \{f \geq r\})}{|\{f \geq r\}|} |\{f \geq r\}|dr \tag{15}$$

$$\geq h_\lambda \left(\int_{-\infty}^{m_\lambda} \lambda|\{f < r\}|dr + \int_{m_\lambda}^{+\infty} |\{f \geq r\}|dr \right) \tag{16}$$

$$= h_\lambda \sum_i |f_i - m_\lambda|_\lambda \tag{17}$$

where we have used the notation $\mathrm{Cut}(S, S^c) = \sum_{x_i \in S} \sum_{x_j \in S^c} w_{i,j}$. Equality (14) is simply the discrete coarea formula. To go from (15) to (16) we have used statement (iii) of Lemma 1. To go from (16) to (17) we have used the discrete layer cake formula. So $\mathcal{E}_\lambda(f) \geq h_\lambda$ for all nonconstant $f : V \to \mathbb{R}$. To conclude the proof, one need to observe that if f^* is the binary function defined by (8), then $\mathcal{E}_\lambda(f^*) = \mathcal{C}_\lambda(S^*) = h_\lambda$. So f^* is a global minimizer of $\mathcal{E}_\lambda$. $\square$

References

[1] A. Bertozzi and A. Flenner. Diffuse Interface Models on Graphs for Classification of High Dimensional Data. *Accepted in Multiscale Modeling and Simulation*, 2012.

[2] X. Bresson, X.-C. Tai, T.F. Chan, and A. Szlam. Multi-Class Transductive Learning based on ℓ^1 Relaxations of Cheeger Cut and Mumford-Shah-Potts Model. *UCLA CAM Report*, 2012.

[3] T. Bühler and M. Hein. Spectral Clustering Based on the Graph p-Laplacian. In *International Conference on Machine Learning*, pages 81–88, 2009.

[4] A. Chambolle and T. Pock. A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011.

[5] J. Cheeger. A Lower Bound for the Smallest Eigenvalue of the Laplacian. *Problems in Analysis*, pages 195–199, 1970.

[6] 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.

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

[8] G. Gilboa and S. Osher. Nonlocal Operators with Applications to Image Processing. *SIAM Multiscale Modeling and Simulation (MMS)*, 7(3):1005–1028, 2007.

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

[10] M. Hein and T. Bühler. An Inverse Power Method for Nonlinear Eigenproblems with Applications in 1-Spectral Clustering and Sparse PCA. In *In Advances in Neural Information Processing Systems (NIPS)*, pages 847–855, 2010.

[11] M. Hein and S. Setzer. Beyond Spectral Clustering - Tight Relaxations of Balanced Graph Cuts. In *In Advances in Neural Information Processing Systems (NIPS)*, 2011.

[12] Y. Lou, X. Zhang, S. Osher, and A. Bertozzi. Image Recovery via Nonlocal Operators. *Journal of Scientific Computing*, 42(2):185–197, 2010.

[13] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear Total Variation Based Noise Removal Algorithms. *Physica D*, 60(1-4):259 – 268, 1992.

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

[15] W. Sun, R.J.B. Sampaio, and M.A.B. Candido. Proximal Point Algorithm for Minimization of DC Function. *Journal of Computational Mathematics, 21 (2003), pp.* ., 21(4):451–462, 2003.

[16] A. Szlam and X. Bresson. Total variation and cheeger cuts. In *Proceedings of the 27th International Conference on Machine Learning*, pages 1039–1046, 2010.

[17] T. Teuber, G. Steidl, P. Gwosdek, C. Schmaltz, and J. Weickert. Dithering by Differences of Convex Functions. *SIAM Journal on Imaging Sciences*, 4:79–108, 2011.

[18] L. Zelnik-Manor and P. Perona. Self-tuning Spectral Clustering. In *In Advances in Neural Information Processing Systems (NIPS)*, 2004.