Total Variation and Cheeger Cuts

Arthur Szlam ASZLAM@COURANT.NYU.EDU
The Courant Institute, NYU, 715 Broadway, New York, NY 10003

Xavier Bresson XBRESSON@MATH.UCLA.EDU
Department of Mathematics, UCLA, 520 Portola Plaza, Los Angeles, CA 90095

Abstract

In this work, inspired by (Bühler & Hein, 2009), (Strang, 1983), and (Zhang et al., 2009), 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. We then describe an algorithm for finding good cuts suggested by the similarities of the energy of the relaxed problem and various well studied energies in image processing. Finally we provide experimental validation of the proposed algorithm, demonstrating its efficiency in finding high quality cuts.

1. Introduction

Many clustering methods start with a (nonnegative, symmetric) matrix W which collects the relative similarities between a set of points V to be clustered, and make 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 size. This motivates the definition of the Cheeger cut value of a partition $P = \{S, S^c\}$ of V:

$$\mathcal{C}(S) = \frac{\text{Cut}(S, S^c)}{\min(|S|, |S^c|)}, \tag{1}$$

where

$$\text{Cut}(A, B) = \sum_{i \in A, j \in B} W_{ij},$$

and where $S \subset V$, S^c is the complement of S in V, and $|S|$ is the cardinality of S.

Appearing in *Proceedings of the 27th International Conference on Machine Learning*, Haifa, Israel, 2010.

Computing the optimal partition $\mathcal{C}_* = \min_{S \subset V} \mathcal{C}(S)$ is unfortunately NP-hard. However it turns out that $\mathcal{C}_*$ is approximated by the second eigenvalue of the combinatorial Laplacian $D - W$, where $D_{ii} = \sum_j W_{ij}$:

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

See (Cheeger) for the continuous version, and for the discrete version, see (Chung, 1997). This motivates "spectral clustering", which in its simplest from, thresholds the second eigenvector ϕ_2 of the Laplacian to get an approximation to the clustering with the smallest cut (see (von Luxburg, 2006) for an excellent introduction). In symbols, we find the threshold $\gamma_* = arg \min_\gamma \mathcal{C}(S_\gamma)$, where $S_\gamma = \{v \in V, \phi_2(v) > \gamma\}$. Unfortunately, the bounds for the quality of $\mathcal{C}(S_{\gamma_*})$ have the quadratic gap present in (2).

Using the Rayleigh quotient formula for the eigenvalue gives

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

where

$$\mathcal{H}_p(f) = \frac{\sum ||\nabla f||^p}{\min_{c \in \mathbb{R}} ||f - c||_p^p},$$

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

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

Recently, in (Bühler & Hein, 2009) (also see (Amghibech, 2003)), it was shown that

$$\lim_{p \mapsto 1^+} \min_{f \in \mathcal{L}^2(V)} \mathcal{H}_p(f) = \mathcal{C}_*. \tag{3}$$

In fact they show something stronger: for any $p > 1$, denote by S_p the set obtained from the optimal thresholding of $arg \min_{f \in \mathcal{L}^2(V)} \mathcal{H}_p(f)$. They prove

$$\mathcal{C}_* \leq \mathcal{C}(S_p) \leq p \left(\max_i D_{ii} \right)^{\frac{p-1}{p}} (\mathcal{C}_*)^{\frac{1}{p}}. \tag{4}$$

Thus, by finding the minimizer of $\mathcal{H}_p$ for small p and thresholding it, we get a cut closely approximating $\mathcal{C}_*$.

In this work we will consider what happens when $p = 1$. We start by noting that minimizing $\mathcal{H}_1$ over $\mathcal{L}^2(V)$ is a relaxation of minimizing (1) over partitions of V. On the other hand, we will show that as one might suspect from (3) and (4), the two problems are actually equivalent, and there is a minimizer of the relaxed problem which is the indicator of a set. While this might seem to scuttle hopes of using $\mathcal{H}_1$ for algorithmic advantage, it turns out that the extra space afforded by the continuous domain can be put to practical use. We will describe an iterative algorithm for minimizing $\mathcal{H}_1$ based on ideas in (Goldstein & Osher, 2009) which finds high quality cuts rapidly in practice; for cuts of comparable quality, it often runs ten to a hundred times faster than the algorithm used in (Bühler & Hein, 2009). 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 of size n and a similarity matrix W on V. We can relax the problem

$$\min_{S \subset V} \mathcal{C}(S)$$

as follows: for any binary valued function $f = \chi_S$, $S \subsetneq V$,

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

where $\mathrm{m}(f)$ is the median of f. Then

$$\frac{\sum_i ||\nabla f||(i)}{||f - \mathrm{m}(f)||_1} = 2\frac{\sum_{v_i \in S^c} \sum_{v_j \in S} W_{ij}}{\min(|S|, |S^c|)}$$
$$= 2\mathcal{C}(S).$$

Thus

$$\min_f \frac{\sum_i ||\nabla f||(i)}{||f - \mathrm{m}(f)||_1} \tag{5}$$

is a relaxation of the Cheeger cut problem, and

$$\min_f \frac{\sum_i ||\nabla f||(i)}{||f - \mathrm{m}(f)||_1} \leq \min_S \mathcal{C}(S). \tag{6}$$

In this work we will show that the inequality (6) is actually an equality, and for any solution f of the relaxed minimization, 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. While the equivalence of the relaxation and the original problem is not new, it seems that the explicit form of the minimizer has not been described before. Our approach is in part inspired by the analysis of similar problems in the continuous setting by Strang in (Strang, 1983).

For a function $f : V \mapsto \mathbb{R}$, denote

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

Note that this is a norm on the set of function modulo constants. TV stands for "total variation"; this norm has been well studied in many contexts.

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

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

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 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 point 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. □

Remark 1. *If instead of using $f - m(f)$, we use $f - m_j(f)$ with $j < n/2$, where $m_j(f)$ is the jth largest value of f, we can encode an expectation of clusters of a specific size. That is, the energy of a partition becomes $\mathcal{C}_j(S) = Cut(S, S^c)/\alpha_j(S)$, where $\alpha_j(S) = |S|$ if $|S| < |V| - j$, and $|S^c|$ otherwise.*

3. A Split-Bregman algorithm for ratio minimization

In this section, we give a heuristic algorithm for minimizing the ratio cut energy 5. The algorithm will be based on the split-Bregman method introduced in (Goldstein & Osher, 2009) to solve the TV/L^2 problem for image denoising.

3.1. The alternating direction method, a.k.a Bregman iteration

Here we give a brief review of the Bregman iterative method for solving $l1$ regularized linear constrained problems. Suppose we want to solve

$$arg\min_u ||u||_1,$$

$$Au = g,$$

where A is fat $n \times m$ matrix A (i.e. $m > n$) and g is a fixed m vector. We can solve the sequence of unconstrained problems

1. $g^{k+1} = g - Au^k + g^k$
2. $u^{k+1} = arg\min_u ||u||_1 + \eta||Au - g^{k+1}||^2,$

where η is fixed penalty parameter. In this case (Yin et al.), the u^k can be proved to converge to the solution of the constrained problem, and the unconstrained problem in the second step is easier to solve than the original problem. This method can be interpreted as alternating updates in the dual variables g^k and primal variables u^k. In more pedestrian language, the update of g^k is designed to increase the penalty

on the coordinates where the previous unconstrained solution was away from the constraints.

In (Goldstein & Osher, 2009) (and later (Zhang et al., 2009) in the non-local/graph setting), this technique was used for the solution of various total variation regularized problems in image processing, for example

$$arg\min_f ||f||_{\mathrm{TV}} + \beta||f - f_0||^2, \tag{7}$$

where f_0 is a fixed function and β is a fixed parameter. To put this kind of problem in the form above, they introduced the dummy variable

$$d = \nabla f,$$

and solved

$$arg\min_{d,f} ||d||_1 + \beta||f - f_0||^2,$$

$$\nabla f = d.$$

using the Bregman method:

1. $d^{k+1}, f^{k+1} =$

   $$arg\min_{d,f} ||d||_1 + \beta||f - f_0||^2 + \eta||d - \nabla f + g^k||^2$$

2. $g^{k+1} = g^k + d^{k+1} - \nabla f^{k+1},$

which is further divided into

1. $d^{k+1} = arg\min_d ||d||_1 + ||d - \nabla f + g^k||^2$
2. $f^{k+1} = arg\min_f \beta||f - f_0||^2 + ||d - \nabla f + g^k||^2$
3. $g^{k+1} = g^k + d^{k+1} - \nabla f^{k+1}.$

This "splitting" trick has a long history (see (Esser, 2009) and the references therein for a nice account in this context). The point is that the d update is now a simple shrinkage:

$$d^{k+1} = S(-\nabla f + g^k, \eta),$$

where the shrinkage operator S is defined by

$$S(x, \eta) = \begin{cases} x - \mathrm{sign}(x)\eta & |x| > \eta \\ 0 & |x| \leq \eta \end{cases} \tag{8}$$

and the f update is the solution of a linear system. Thus the $l1$ part of the energy has been decoupled from the ∇f part.

3.2. Ratio minimization

In this section we will describe a heuristic based on split Bregman for minimizing energy (5). While we will not be able to prove the convergence of the method, experimentally it is very fast gives high quality cuts. We use the method of (Dinkelbach, 1967) to convert the ratio problem into a sequence of problems resembling (7).

As before, note that the minimization problem (5) is invariant to shifts of f by constant functions, and so it is equivalent to the constrained minimization problem

$$\min_{f \in \mathbb{R}^n} \frac{|f|_{\mathrm{TV}}}{||f||_1} \text{ s.t. } \mathrm{m}(f) = 0. \tag{9}$$

We can almost use Dinkelbach's method to reformulate this as

$$\begin{aligned} \max_\lambda \min_f |f|_{\mathrm{TV}} - \lambda||f||_1 \\ \mathrm{m}(f) = 0, \end{aligned} \tag{10}$$

and alternate between the steps

$$\begin{aligned} f_{n+1} = \min_f |f|_{\mathrm{TV}} - \lambda_n||f||_1 \\ \text{s.t. } \mathrm{m}(f) = 0, \\ \lambda_{n+1} = |f_{n+1}|_{\mathrm{TV}}/||f_{n+1}||_1. \end{aligned} \tag{11}$$

However, if the domain of f is unbounded, the objective in (10) may also be unbounded; and for 9 to make sense the domain should not contain the origin. Since energy (5) is homogenous of degree 0, we can add the constraint $||f||_2 = 1$ without changing the solution. This fixes the subproblem (10), and the results in (Dinkelbach, 1967) would guarantee convergence to the global minimum if we could solve this subproblem. Unfortunately, the median zero and l^2 constraints make it difficult to prove that the split Bregman methods converge. In practice, we will find that instead of attempting to solve problem (11) to completion with the constraints, it will be more effective to take a few steps towards the unconstrained (and therefore unbounded) objective, update λ, take a few more steps, etc.

We now describe the algorithm we will use in more detail. Recall we are working on a graph with vertices V and weights W. Suppose V has m points, and let E be the nonzero entries in the upper-triangular part of W. Let D be the $|E| \times m$ matrix given by $D_{rs} = W_{ij}$ if r corresponds to ij and $s = j$, $D_{rs} = -W_{ij}$ if r corresponds to ij and $s = i$, and zero otherwise. Then $|f|_{TV} = |Df|_1$. We introduce the dummy variables $d = Df$, and $e = f$, and rewrite the norm constrained

version of problem (11) as

$$\min_{d,e,f} ||d||_1 - \lambda_n ||e||_1$$

$$d = Df,\ e = f,$$

$$\mathrm{m}(e) = 0,\ ||e||_2 = 1.$$

This leads to the introduction of the dual variables b_n^k and c_n^k and the following subproblems, as in the previous section:

1. $d_n^{k+1}, f_n^{k+1}, e_n^{k+1} = \min_{d,e,f} ||d||_1 - \lambda^n ||e||_1$
   $+\eta_1 ||Df - d + b_n^k||^2 + \eta_2 ||f - e + c_n^k||^2$
   s.t. $m(e) = 0$, and $||e||_2 = 1$,
2. $b_n^{k+1} = b_n^k + Df_n^{k+1} - d_n^{k+1}$,
3. $c_n^{k+1} = c_n^k + f_n^{k+1} - e_n^{k+1}$.

Here n is as in (11), and k indexes the steps towards a solution of (11) for a fixed n. The reader may wonder at the use of two dummy variables instead of one. The reason for this becomes apparent when we continue as in section 3.1, and further split step 1 into three updates, leading to three easy to solve problems:

1. $d_n^{k+1} = \min_d ||d||_1 + \eta_1 ||Df_n^k - d + b_n^k||^2$,
2. $e_n^{k+1} = \min_e -\lambda^n ||e||_1 + \eta_2 ||f_n^k - e + c_n^k||^2$,
   such that $m(e) = 0$, $||e||_2^2 = 1$,
3. $f_n^{k+1} = \min_f \eta_1 ||Df_n^k - d_n^{k+1} + b_n^k||^2 + \eta_2 ||f_n^k - e_n^{k+1} + c_n^k||^2$.

Each of these subproblems has an explicit solution. The solution to 1 is given by $d_n^{k+1} = S(Df_n^k + b_n^k, 1/\eta_1)$, where S is the shrinkage operator defined in 8. Subproblem 3 is a linear system, and has explicit solution

$$(\eta_1 D^T D + \eta_2 I)^{-1}(\eta_1 D^T (d_n^{k+1} - b_n^{k+1}) + \eta_2 (e_n^{k+1} - c_n^{k+1}).$$

Subproblem 2 has a similar solution to subproblem 1, but its description is more involved. Suppose for a moment that the median zero condition is removed, and set $r = f_n^k + c_n^k$. Then the solution is simply an "unshrinkage"

$$E(r, \lambda_n/\eta_2) = r + \frac{\lambda_n}{\eta_2} \mathrm{sign}(r), \tag{12}$$

projected onto the l^2 unit sphere. To deal with the median zero constraint, note that the sign of the solution will still be the sign of r. Thus enforcing median zero simply specifies that some coordinates of the solution corresponding to the overrepresented sign of r will be zero. Once these coordinates are fixed, the solution to the problem is exactly as above; note that independent of the choice of coordinates, the quantity $||e_s - r||^2$ will be the same. Therefore, we simply choose the coordinates so that after unshrinkage and projection, the resulting vector has largest $l1$ norm. This can be done as follows: sort r, and without loss, assume there are m more positive entries than negative entries. Now we mask out the contiguous block of m sorted entries so that the remaining positive values have the smallest $l1$ norm after projection onto the l^2 ball. We can find this block quickly by keeping a running count of the absolute and square sums. The solution $E_c(r, \lambda, \eta)$ to subproblem 2 is then given by

$$E_c(r, \lambda, \eta) = \begin{cases} \frac{E(r, \lambda_n/\eta_2)(j)}{||E(r, \lambda_n/\eta_2)||_2} & j \notin I \\ 0 & j \in I \end{cases},$$

where I is the set of masked indices.

In practice, we have found it is better to solve subproblem 3 iteratively, rather than using the explicit solution. Computing $(\eta_1 D^T D + \eta_2 I)^{-1}$ is expensive and unecessary, as D is extremely sparse. Instead, we run a few steps of conjugate gradient descent. As mentioned above, we have also found that enforcing the constraints in subproblem 2 is not usually as efficient as just subtracting the median from f after its update, and either projecting back to the sphere after every few iterations, or ignoring the artificial l^2 constraint altogether.

The algorithm is summarized in 3.1. Although we cannot prove that it converges, we will see in the next section that experimentally, the algorithm performs extremely well.

Algorithm 3.1: RATIO MIN$(f_0, \eta_1, \eta_2, N_i, N_{cg}, N, D)$

$f = f_0 - \mathrm{m}(f_0),\ d = 0,\ e = 0,$
$b = 0,\ c = 0,\ \lambda = ||Df||_1 / ||f||_1$
for $i = 1$ **to** N
  for $j = 1$ **to** N_i
    $d \leftarrow S(f + b, 1/\eta 1)$
    $e \leftarrow E(f + c, \lambda/\eta_2)$
    $f \leftarrow \overline{f} - m(\overline{f})$, where $\overline{f}$ is the output of N_{cg}
      conjugate gradient steps for solving
      $(\eta_1 D^T D + \eta_2 I) f = \eta_1 D^T (d - b) + \eta_2 (e - c)$
    $b \leftarrow b + Df - d$
    $c \leftarrow c + f - e$
  end(j)
  $\lambda \leftarrow ||Df||_1 / ||f||_1$
  $(optional)\ u \leftarrow ||f||_2,\ f \leftarrow f/u,$
  $d \leftarrow d/u,\ e \leftarrow e/u,\ b \leftarrow b/u,\ c \leftarrow c/u$
end(i)

4. Experiments

In all experiments we use a 10-NN graph with the self-tuning weights as in (Zelnik-Manor & Perona, 2004), and the neighbor parameter set to 10. The optimization parameters for algorithm 3.1 for all experiments are fixed as follows: the total number of iterations $N = 120$, the number of inner iterations before a λ update $N_i = 1$, the number of conjugate gradient steps $N_{cg} = 5$, and the penalty parameters $\eta_1 = 1$ and $\eta_2 = 1$. We do not do the optional normalization step. The method is always initialized using the second eigenvector of the normalized Laplacian.

We compare our method against the one in (Bühler & Hein, 2009), using code downloaded from `http://www.ml.uni-saarland.de/code/pSpectralClustering/pSpectralClustering.html`, and against thresholding the second eigenvector of the normalized Laplacian. For each method, we use the threshold with the lowest Cheeger cut value. The timings do not include the cost of finding the threshold or of building the graph.

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.

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 using algorithm 3.1 and the second eigenvector method are presented in Figure 1, where each row is a cluster, and the number in the leftmost column of each row is the dominant label of that cluster.

We see that the method does well. The 4's and 9's are merged, but otherwise the clustering is very accurate. The total computation time (not including constructing the weights) is 214 seconds. This experiment is interesting because it shows that MNIST has quite good clusters, suggesting that the cluster structure, and not the manifold structure, is behind the success of many of the SSL techniques that have used this data set as a benchmark. The second eigenvector method does not do as well, merging more classes; but still its performance is good. The computation time of this method is 54 seconds. The results using the method in (Bühler & Hein, 2009) are roughly the same quality as our method, merging the 9's and 4's, but otherwise separating all of the classes; however, the run time is 7303 seconds.

Table 1. Top: the confusion matrix for the clustering of MNIST using Algorithm 3.1 iterated as described in 4.1. Each row is a cluster; the number in the leftmost column of each row is the dominant label of that cluster. The 4's and 9's are merged, but otherwise the clustering is very accurate. The total computation time (not including constructing the weights) was 214 seconds. Middle: the confusion matrix using (Bühler & Hein, 2009); the total computation time is 7303 seconds. Bottom: the confusion matrix using the iterated second eigenvector cut. More classes have been merged, and the ones broken into 4 clusters. The computation time was 54 seconds

mode/true	0	1	2	3	4	5	6	7	8	9
0	6857	1	22	1	4	10	11	7	3	13
1	0	4011	2	1	4	0	2	9	18	5
1	2	3618	4	2	1	0	1	8	12	0
2	5	125	6860	50	3	1	6	31	55	3
3	2	1	14	6919	1	42	0	2	60	128
4	3	62	11	32	6785	36	6	86	62	6735
5	7	3	0	53	1	6162	43	0	44	18
6	22	4	6	1	18	43	6781	0	5	5
7	1	49	56	36	5	3	0	7148	12	34
8	4	3	15	46	2	16	26	2	6554	17

mode/true	0	1	2	3	4	5	6	7	8	9
0	6867	2	28	3	4	22	18	6	6	13
1	2	3602	4	2	1	0	2	9	12	0
1	0	4027	2	1	4	0	2	8	17	5
2	7	155	6842	42	3	1	1	19	12	2
3	1	2	11	6922	1	58	0	2	53	119
4	3	62	10	33	6784	40	6	78	59	6723
5	3	3	0	38	1	6109	11	0	33	33
6	15	8	8	3	17	62	6834	0	19	5
7	1	12	66	42	8	4	0	7168	11	38
8	4	4	19	55	1	17	2	3	6603	20

mode/true	0	1	2	3	4	5	6	7	8	9
0	6871	1	66	7	14	35	15	22	17	98
1	0	1830	0	1	0	0	2	8	4	0
1	0	2034	0	0	2	0	0	4	4	0
1	0	2044	0	0	1	0	1	4	6	4
1	1	1738	2	0	1	0	3	4	2	0
2	1	82	6809	31	8	1	6	38	6	5
3	6	83	54	7042	6	6221	90	10	6713	175
4	2	47	7	26	6726	17	2	144	49	6500
6	22	4	6	5	60	37	6757	2	11	26
7	0	14	46	29	6	2	0	7057	13	150

4.2. Two moons

We construct the two moons data set as in (Bühler & Hein, 2009). 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 using algorithm 3.1, the second eigenvector method, and using the algorithm from (Bühler & Hein, 2009). The results displayed in Figures 1 are averaged over 100 instantiations of the dataset.

Our algorithm gives slightly better cuts on average than (Bühler & Hein, 2009), but runs more than 50 times as fast; on the other hand, it runs about 20 times slower than the second eigenvector method, which gives poor results here, both in terms of cut value and classification error.

In Figure 3 we show the results of modifying algorithm 3.1 as in Remark 1; that is, instead of subtracting off the median in the f update, we specify a position m, and subtract off the mth sorted value of f. If $m = n/2$, where n is the number of data points, this would be the normal version of the algorithm. This modification results in favoring a partition with an element roughly the size of m, instead of favoring a partition of size $n/2$. Our experience has been that picking very small (or very large) values for m can destablize the algorithm, but it is tolerant of moderate values of m.

4.3. CIFAR-10

The CIFAR-10 dataset, available at `http://www.cs.utoronto.ca/~kriz/cifar.html` and described in (Krizhevsky, 2009), is a labeled subset of the 80 million tiny images dataset, described in (A. Torralba, 2008). There are 5000 32×32 color images of various objects, grouped into 10 classes. We preprocess the data by projecting each image (considered as a 3072 vector) onto the unit sphere in rgb space, and then project onto 500 principal components. The results, displayed in Figure 1, are averaged over the 45 binary problems.

Our algorithm again gives slightly better cuts on average than (Bühler & Hein, 2009), but on average, runs more than 100 times as fast. On the other hand, it runs about 10 times slower than the second eigenvector method, which on this dataset gives poorer cut values, but not significantly poorer classification errors.

5. Conclusions

In this work we have shown an equivalence between the Cheeger ratio cut clustering objective and the minimization of the energy (5). We develop a algorithm using the split Bregman technique for minimizing this energy which experimentally gives higer quality cuts than the second eigenvector method, while

Figure 1. Results on two moons and CIFAR-10. Two moons is averaged over 100 instances, and CIFAR-10 is averaged over the 45 pairs of two class problems.

	Two moons	CIFAR-10
Algorithm 3.1	.046	.382
p-Laplacian, $p = 1.1$	.052	.381
Eigenvector method	.171	.390

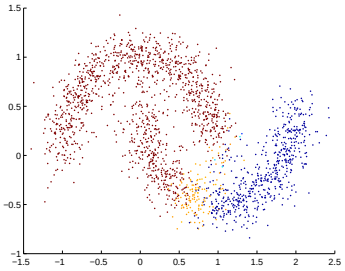
(a) Average error in percent on the two moons and CIFAR-10 data sets.

	Two moons	CIFAR-10
Algorithm 3.1	.81	5.2
p-Laplacian, $p = 1.1$	48	580
Eigenvector method	.07	.450

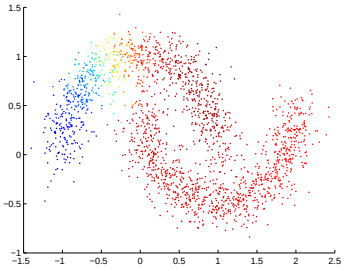
(b) Average run time on the two moons and CIFAR-10 data sets in seconds.

	Two moons	CIFAR-10
Algorithm 3.1	.341	.308
p-Laplacian, $p = 1.1$	.342	.309
Eigenvector method	.448	.338

(c) Average Cheeger cut value on the two moons and CIFAR-10 data sets.

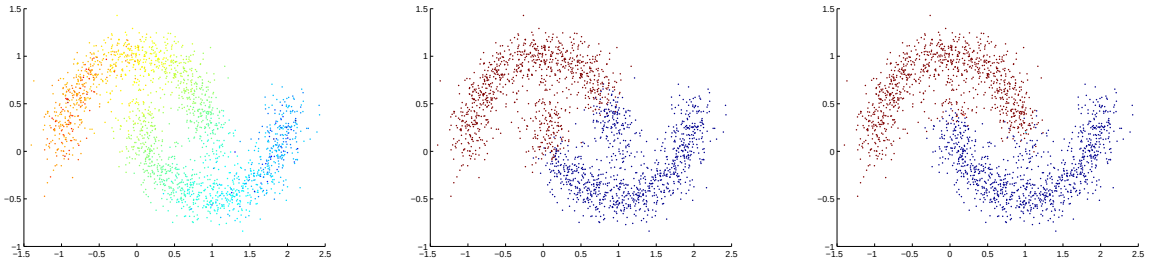


(a) Output of algorithm 3.1 with the "median" set at element 667 as in remark 1. The thresholded partition has 695 elements in one cluster and 1305 in the other.



(b) Output of algorithm 3.1 with the "median" set at element 333 as in remark 1. The thresholded partition has 340 elements in one cluster and 1660 in the other.

Figure 3. Results for an instantiation of the two moons dataset, 2000 points in 100 dimensions, with the modified algorithm 3.1 as in Remark 1 .



(d) Second eigenvector of the Laplacian.

(e) Optimal Cheeger cut obtained by thresholding the second eigenvector of the Laplacian from a.

(f) Output of proposed algorithm 3.1.

Figure 2. Results for an instantiation of the two moons dataset, 2000 points in 100 dimensions.

running significantly faster than the method presented in (Bühler & Hein, 2009)

There is of course much left to be done here. We suspect that some mild modifications of the algorithm would lead to something which provably converges. Furthermore, we suspect quite strong convergence results are possible for data that admit a good cut. We also think that there is plenty of room to speed up the algorithm. While it is seems unlikely that there is a way to get faster than finding the second eigenvector, we also suspect finding a good cut should not take much longer than this.

6. Acknowledgments

ADS and XB thank the NSF for generous support under grants DMS-0811203 and DMS-0610079. We also thank the reviewers for their useful suggestions.

References

A. Torralba, R. Fergus, W. T. Freeman. 80 million tiny images: a large dataset for non-parametric object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11): 1958–1970, 2008.

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

Bühler, Thomas and Hein, Matthias. Spectral clustering based on the graph p-laplacian. In Bottou, Léon and Littman, Michael (eds.), *Proceedings of the 26th International Conference on Machine Learning*, pp. 81–88, Montreal, June 2009. Omnipress.

Cheeger, J. A lower bound for the smallest eigenvalue of the laplacian. In Gunning, RC (ed.), *Problems in Analysis*, pp. 195–199. Princeton Univ. Press.

Chung, F. R. K. *Spectral graph theory*, volume 92 of *CBMS Regional Conference Series in Mathematics*. Published for the Conference Board of the Mathematical Sciences, Washington, DC, 1997. ISBN 0-8218-0315-8.

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

Esser, E. Applications of lagrangian-based alternating direction methods and connections to split bregman. Technical Report TR09-31, UCLA CAM, 2009.

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

Krizhevsky, Alex. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

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

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

Yin, Wotao, Osher, Stanley, Goldfarb, Donald, and Darbon, Jerome. Bregman iterative algorithms for 1minimization with applications to compressed sensing. *SIAM J. Imaging Sci*, 1:143–168.

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

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