Globally Optimizing Graph Partitioning Problems Using Message Passing

Elad Mezuman
Interdisciplinary Center for Neural Computation
Edmond & Lily Safra Center for Brain Sciences
Hebrew University of Jerusalem
http://www.cs.huji.ac.il/~mezuman

Yair Weiss
School of Computer Science and Engineering
Edmond & Lily Safra Center for Brain Sciences
Hebrew University of Jerusalem
http://www.cs.huji.ac.il/~yweiss

Abstract

Graph partitioning algorithms play a central role in data analysis and machine learning. Most useful graph partitioning criteria correspond to optimizing a ratio between the cut and the size of the partitions, this ratio leads to an NP-hard problem that is only solved approximately. This makes it difficult to know whether failures of the algorithm are due to failures of the optimization or to the criterion being optimized.

In this paper we present a framework that seeks and finds the optimal solution of several NP-hard graph partitioning problems. We use a classical approach to ratio problems where we repeatedly ask whether the optimal solution is greater than or less than some constant - λ. Our main insight is the equivalence between this "λ question" and performing inference in a graphical model with many local potentials and one high-order potential. We show that this specific form of the high-order potential is amenable to message-passing algorithms and how to obtain a bound on the optimal solution from the messages. Our experiments show that in many cases our approach yields the global optimum and improves the popular spectral solution.

1 Introduction

Graph partitioning is the problem of dividing the vertices of a graph into sets, minimizing the number (or weight) of edges between sets (the cut), while penalizing for "too small" sets. Graph partitioning has many applications starting from clustering genes, through optimizing financial problems and parallel scientific computing to image segmentation.

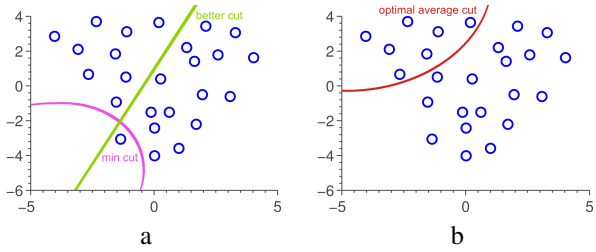


Figure 1: Graph partitioning. (a) Minimal cut yields a bad partition of the graph. (b) The global optimum of average cut, found using our method.

Although there are an exponential number of graph partitions, finding the minimum cut of a graph, without size considerations, is a well-studied problem and efficient algorithms exist for solving it. However, the minimum cut criteria favors cutting small sets of isolated vertices in the graph (Wu and Leahy, 1993, Shi and Malik, 2000), as can be seen in Figure 1. To avoid this unnatural bias for partitioning out small sets of points, other measures that involve optimization on some attribute (e.g. the size) of the sets, in addition to the size of the cut, are usually used. Probably the most well known out of them is normalized cut (Shi and Malik, 2000). Normalized-cut seeks to maximize the similarity within the sets while minimizing the dissimilarity between the sets. Other measures include size-normalized cut (or average cut) that maximizes the sizes of the sets, and the Cheeger cut. The optimization function of these 3 mentioned measures, and of many other graph partitioning measures, is NP-hard to solve. This NP-hardness of the problem yielded approaches that try to find other similar criteria that are easy to optimize. Hochbaum (2010) showed that a variant of normalized-cut can be solved in polynomial time and achieves good image segmentation. However, most approaches try to tackle this hardness by solving a relaxed version of the problem, for example using spectral methods. Solving the relaxed problem usually leaves us with a big interval in which the optimal solution

Appearing in Proceedings of the 15^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2012, La Palma, Canary Islands. Volume XX of JMLR: W&CP XX.

can be found. This makes it difficult to know whether failures of the algorithm are due to failures of the optimization or to the criterion being optimized.

In this paper we develop a framework to find the optimal solution for graph partitioning problems that are ratios of the form $f(x)/g(x)$ where $f(x)$ is the cut size and $g(x)$ is a function of the size of the partitions. We use a classical approach where we repeatedly solve the λ question: $min_x f(x) - \lambda g(x)$. The answer to the λ question tells us whether the optimal solution is better than λ or not. Our main insight is that we can solve the λ question efficiently using recently developed techniques for Markov Random fields (MRFs) with high order potentials (HOPs) (Tarlow et al., 2010, Rother et al., 2007, Weiss et al., 2007). We show that the specific form of the HOP is amenable to message passing and show how to derive a bound on the optimal solution from the messages. Our experiments show that message passing often succeeds in solving the λ question in short time. Using a bisection algorithm over λ we succeed in improving the bounds on the optimal solution and in some examples to find it.

2 Notations and Preliminaries

The set of points in an arbitrary feature space are represented as a weighted undirected graph $G = (V, E)$, $|V| = n$, where the vertices (or nodes) of the graph are the points in the feature space, and edges are formed between pairs of nodes. The weight on each edge, $w(i, j) = w_{i,j}$, is a function of the similarity (or affinity) between nodes i and j. We use the notations $d(i) = \sum_j w_{i,j}$ for the sum of edges of node i and $deg(i)$ as the degree of vertex i. We define as the neighbors of i, $Nei(i)$, the set of nodes that are connected to i by an edge. The Laplacian of the graph is defined as $L = (D - W)$, where D is a $n \times n$ diagonal matrix with d on its diagonal, and W is a $n \times n$ symmetrical matrix with $W(i, j) = w_{i,j}$.

The graph can be partitioned into two disjoint sets, A, B $s.t$ $A \cup B = V$, $A \cap B = \emptyset$ by simply removing edges connecting the two parts. The degree of similarity between these two parts can be computed as the total weight of the edges that have been removed. In graph theoretic language, it is called the cut: $cut(A, B) = \sum_{i \in A, j \in B} w_{i,j}$. We will use the indicator vector $x \in \{0, 1\}^n$ to indicate to which group each node belong and get $cut(x) = \sum_{i,j} x_i(1 - x_j)w_{i,j}$.

2.1 Ratio Optimization Problems for Graph Partitioning

We can find in several graph partitioning measures 2 opposite goals: The first is minimizing the cut and the second is maximizing some property of the sets (e.g. their size). Usually these two goals are combined to one ratio optimization problem. We mention here graph partitioning measures which are ratio problems:

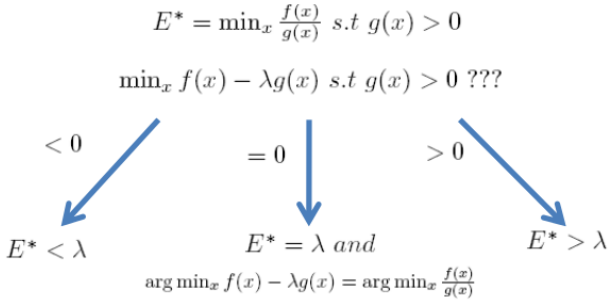


Figure 2: The λ Question.

1. Average cut $\frac{cut(x)}{\|x\|} + \frac{cut(x)}{(n-\|x\|)} = \frac{n*cut(x)}{\|x\|(n-\|x\|)}$
2. Normalized cut $\frac{cut(x)}{\sum_{i|x_i=0} d_i} + \frac{cut(x)}{\sum_{i|x_i=1} d_i} = \frac{\sum_i d_i * cut(x)}{\sum_{i|x_i=0} d_i \sum_{j|x_j=1} d_j}$
3. Cheeger cut $\frac{cut(x)}{min(\|x\|, n-\|x\|)}$.

2.2 The λ Question

We use a classical approach (Hochbaum, 2010) for maximizing a fractional objective function with positive denominator (see figure 2). Given a problem of the form: $\min_x \frac{f(x)}{g(x)}$, we reduce it to a sequence of calls to an oracle that provides the answer to the λ-question: Is $\min_x f(x) - \lambda g(x)$ less than, greater than or equal to 0? If the answer is equal to 0, the optimal solution to the original fractional problem is λ and the same x^* that minimizes $f(x) - \lambda g(x)$ minimizes also the fractional objective function. If the answer is less than zero, then the optimal solution has a value smaller than λ and otherwise, the optimal value is greater than λ (that is because $f(x) - \lambda g(x) < 0 \Leftrightarrow \frac{f(x)}{g(x)} < \lambda$ given that $g(x) > 0$). Assuming we have an initial upper bound U, and lower bound L on the optimal solution, we can use a bisection method to find the optimal solution. Using the bisection method we can get as close as ϵ to the optimal solution solving $O(\log(\frac{U-L}{\epsilon}))$ times the λ-question. Therefore, if the linearized version of the problem, i.e. the λ-question, is solved in polynomial time, then so is the ratio problem.

3 The λ Question as a MRF

Although the λ question gets rid of the ratio $min_x f(x)/g(x)$ and replaces it with the simpler form $min_x f(x) - \lambda g(x)$ we are still faced with minimizing over x and the number of possible values of x is still exponential in the graph size. The fundamental insight

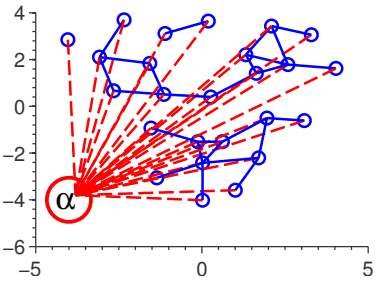


Figure 3: The Graphical Model. We have pairwise potentials for pairs of nodes with edge between them and a global node, the α node. The α node is connected to all the other nodes and is a cardinality potential.

behind our algorithm is that we can efficiently solve the λ question by using message passing algorithms with tractable high order potentials.

In order to solve the λ question for the three cut problems mentioned above, we need to find $\arg\min_{x\in\{0,1\}^n,\|x\|>0} cut(x) - \lambda g(x)$. Notice that we have turned to finding the argument that minimize the objective function since we will need it to get the partitioning. Defining the potentials: $\phi_\alpha(x) = -\lambda g(x)$ and $\psi_{i,j}(x_i, x_j) = \mathbf{1}[\mathbf{x_i} \neq \mathbf{x_j}]w_{i,j}$ we can rewrite it as:

$$x^* = \arg\min_{x\in\{0,1\}^n} \sum_{<i,j>} \psi_{i,j}(x_i, x_j) + \phi_\alpha(x) \quad (1)$$

We can formulate this optimization problem using a graphical model over binary random variables $\{x_i\}_{i=0}^n$. We define the MRF with the above potentials (see Figure 3 for graphical view):

$$P(x) \propto \prod_{<i,j>} \exp(-\psi_{i,j}(x_i, x_j)) \exp(-\phi_\alpha(x)) \quad (2)$$

We wish to find $x^* = \arg\max_x P(x)$. In principle we can use any inference algorithm for MRFs, but note that for the bisection algorithm to work, it is not enough to solve the λ question approximately. We need an algorithm that can give a rigorous bound on the optimal solution.

A classical approach for obtaining bounds on the optimal solution in such problems is linear programming relaxations (e.g. Wainwright and Jordan, 2008) but it is easy to show that due to the high order potential $\phi_\alpha(x)$, even a first-order LP relaxation will have an exponentially large state space. Instead we follow a number of recent works: Tarlow et al. (2010), Weiss et al. (2007), Werner (2007), Globerson and Jaakkola (2007) in which message passing is used to solve the dual of the LP relaxation.

3.1 Convex Belief-Propagation Message Passing

We use the "default" convex belief propagation (BP) messages from Weiss et al. (2007, 2011). These are based on approximating the joint entropy of all variables with a combination of entropies over single variables and pairs of variables: $H \approx \sum_i c_i H_i + c_\alpha H_\alpha + \sum_{<ij>} H_{ij} + \sum_i H_{i\alpha}$. We choose $c_i = \frac{-deg(i)}{2}$, $c_\alpha = -n + 1$, where $deg(i)$ is the degree of node i. It can be shown that this combination yields a convex entropy approximation. Substituting these constants into equations 6.18-6.20 from Weiss et al. (2011), using $\rho_i = \frac{2}{deg(i)+2}$, $F_{i,j} = \exp(-\psi_{i,j})$ and $F_\alpha = \exp(-\phi_\alpha)$ we get the following message passing and beliefs equations:

$$m_{i\to j}(x_j) = \max_{x_i} F_{i,j}(x_i, x_j) F_i(x_i)^{\rho_i} \; m_{\alpha\to i}(x_i) \prod_{k\in Nei(i)\setminus j} m_{k\to i}^{\rho_i}(x_i) m_{j\to i}^{\rho_i - 1}(x_i) \quad (3)$$

$$m_{i\to\alpha}(x_i) = F_i^{\rho_i}(x_i) \prod_{k\in Nei(i)} m_{k\to i}^{\rho_i}(x_i) m_{\alpha\to i}^{\rho_i - 1}(x_i) \quad (4)$$

$$m_{\alpha\to i}(x_i) = \max_{x\setminus x_i} F_\alpha(x) \prod_{k\neq i} m_{k\to\alpha}(x_k) \quad (5)$$

$$b_i(x_i) = F_i^{\rho_i}(x_i) m_{\alpha\to i}(x_i) \prod_{k\in Nei(i)} m_{k\to i}^{\rho_i}(x_i) \quad (6)$$

$$b_\alpha(x) = F_\alpha(x) \prod_j m_{j\to\alpha}(x_j) \quad (7)$$

$$b_{ij}(x_i, x_j) = F_{ij}(x_i, x_j) \frac{b_i(x_i) b_j(x_j)}{m_{j\to i}(x_i) m_{i\to j}(x_j)} \quad (8)$$

$$b_{i\alpha}(x) = \frac{b_i(x_i) b_\alpha(x)}{m_{\alpha\to i}(x_i) m_{i\to\alpha}(x_i)} \quad (9)$$

Given the messages and beliefs after each iteration we can compute the labeling of x_i as described in Kolmogorov (2006): We order the nodes by the value of their maximal belief in descending order, let $S(i)$ be this order. We then go by this order over the nodes choosing label $x_i^{*(t)}$ that maximizes:

$$x_i^{*(t)} = \arg\max_{x_i} F_i(x_i) m_{\alpha\to i}^{(t)}(x_i) \prod_{S(j)<S(i)} F_{ij}(x_i, x_j) \prod_{S(j)>S(i)} m_{j\to i}^{(t)}(x_i) \quad (10)$$

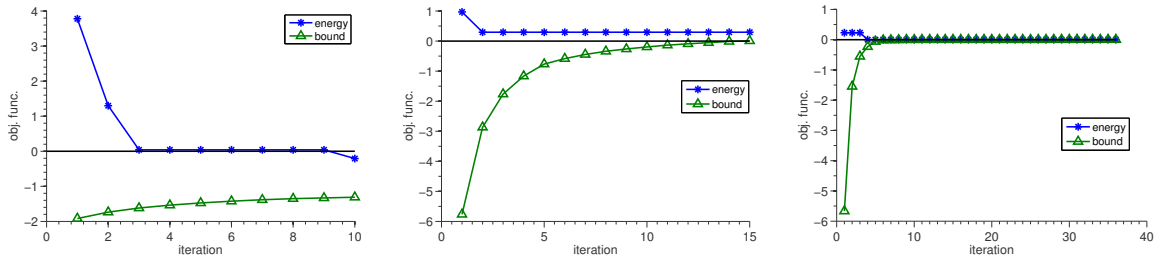


Figure 4: Energy and lower bound as a function of the iteration of the BP. (a) energy reached a value below zero → the λ chosen is greater than the optimal solution (b) bound reached a value above zero → the λ chosen is smaller than the optimal solution (c) the bound coincides with the energy→ we have found the global optimum.

Using this computed labeling, $x^{*(t)}$, we can infer the probability (and the energy) after each iteration. It is shown in Weiss et al. (2007) that at each iteration we can compute the bound on the optimal solution:

$$\Pr(x) \leq \frac{1}{z}\prod_i \prod_{j\in Nei(i)} max_{x_i,x_j}\left(\frac{b_{ij}^{(t)}(x_i,x_j)}{b_i^{(t)}(x)}\right)^{\frac{1}{2}}$$

$$max_x F_\alpha(x)\prod_i \frac{b_i^{(t)}(x_i)}{m_{\alpha\to i}^{(t)}(x_i)} = B(t) \quad (11)$$

If this bound coincides with the energy of our current $x^{*(t)}$ then we know we have found the global optimum.

Figure 4 shows the usual behaviors of the energy and bound as a function of the iteration of the BP.

3.2 Tractable High Order Potentials

Our MRF includes pairwise potentials - $\exp(-\psi_{i,j}(x_i,x_j))$ and a n order potential - $\exp(-\phi_\alpha(x))$. Naively computing messages using max product message passing algorithm for n-order potential takes $O(2^n)$. But examining the global factor $\phi_\alpha(x)$ for the three cases discussed above shows that both the computation of the messages and the computation of the bound can be performed in polynomial time:

1. Average cut $\frac{cut(x)}{\|x\|} + \frac{cut(x)}{(n-\|x\|)} = \frac{n*cut(x)}{\|x\|(n-\|x\|)}$. Here the global factor α is a cardinality based potential, i.e. all partitions that have the same size have exactly the same global factor $\phi(x_1,\ldots,x_n) = f(\sum_i x_i)$. For cardinality-based potentials, it can be shown that all the n messages outgoing from this potential and the bound can be computed in $O(n\log n)$ (Tarlow et al., 2010).

   Let us look more carefully on how we compute the second part of the bound from equation 11 (the first part, i.e. the one that involves the pairwise beliefs, can be computed in $O(|E|)$ that in a sparse graph turns to $O(n)$). We want to find $max_x F_\alpha(x)\prod_i \frac{b_i^{(t)}(x_i)}{m_{\alpha\to i}^{(t)}(x_i)} = max_x F_\alpha(x)\prod_i v^{(t)}(x_i)$. In the general case this computation requires $O(2^n)$, the number of different possibilities for x, but in the case where $F_\alpha(x)$ is a cardinality potential we will now show how to compute it in $O(n\log n)$. Moving to log space and plugging in the global factor of average cut we get:

$$\max_x \lambda\|x\|(n-\|x\|) + \sum_i v^{(t)}(x_i) =$$

$$\max_k \lambda k(n-k) + \max_{x\ s.t\ \|x\|=k}\sum_i v^{(t)}(x_i) \quad (12)$$

   Notice that the last maximization in equation 12 resembles the famous knapsack problem (e.g. see Cormen (2001)): Given k we wish to pick a set of nodes such that their total weight equals k and will maximize the total value. This is a special case of the knapsack problem since all the weights are equal, using sorting we can solve it in $O(nlogn)$.

2. Normalized cut $\frac{cut(x)}{\sum_{i|x_i=0} d_i} + \frac{cut(x)}{\sum_{i|x_i=1} d_i} = \frac{\sum_i d_i * cut(x)}{\sum_{i|x_i=0} d_i \sum_{i|x_i=1} d_i}$. Here the global factor α is a weighted-cardinality based potential, where the weights are the degrees of the nodes - $\phi(x_1,\ldots,x_n) = f(\sum_i d_i x_i)$. In a similar way to the way we developed equation 12 it can be shown that for computing the bound we need to solve

$$\max_d \lambda d(\sum_i d_i - d) + \max_{x\ s.t\ \sum_i x_i d_i = d}\sum_i v^{(t)}(x_i) \quad (13)$$

   This is again a knapsack problem: we wish to find a set of nodes whose total weight is d and maximize the overall value. Without any constraints on the weights, this problem is intractable, but if we assume the affinities (and hence the weights) are integers, then we can use dynamic programming as in Cormen (2001), to solve the bound (and the messages in a similar way)

in $O(n^2D)$ where D is the maximal degree of a node in the graph.

3. Cheeger cut $\frac{cut(x)}{min(\|x\|,min(n-\|x\|)}$. Here again this is a cardinality based potential so messages and bound can be computed in $O(n \log n)$ (Tarlow et al., 2010).

3.3 Tightening by Conditioning

As mentioned above, the message-passing algorithm we use is equivalent to solving the dual of the linear programming relaxation. When the LP relaxation is not tight, the message-passing may converge to "tied beliefs", when the maximization for x_i^* does not have a unique maximum and the bound (Eq. 11) will not agree with $\Pr(x^*)$ (Eq. 2). Several approaches to tightening the LP relaxation have been proposed: Komodakis et al. (2007), Sontag et al. (2008), Rother et al. (2007), Boros et al. (2006). We use a simple algorithm adapted from the probing algorithm of Rother et al. (2007), Boros et al. (2006). The basic idea is that, for any function over a set of variables x, $\min_x E(x) = \min(\min_x E(x|x(i) = 0), \min_x E(x|x(i) = 1))$, so if we cannot find $\min E(x)$ directly we can condition over one of its nodes, fix it once to 1 and once to 0 and then solve the new optimization problems. If we still did not solve the original optimization problem (i.e. solved all the new optimization problems) we can add more and more variables. In general if we condition over k nodes we have 2^{k-1}(minus 1 because of the symmetry for binary variables) optimization problems to solve. Partial solutions (i.e. for some of the conditioned problems) enable us to prune some branches from this tree of conditioned problems. For example, if we found $\min_x E(x|x(1) = 0)$ but did not find $\min_x E(x|x(1) = 1)$ when we add another variable to the conditioning we only need to solve $\min_x E(x|x(1) = 1, x(2) = 0)$, $\min_x E(x|x(1) = 1, x(2) = 1)$ since $\min_x E(x|x(1) = 0) = \min(\min_x E(x|x(1) = 1, x(2) = 0), \min_x E(x|x(1) = 1, x(2) = 1))$.

In order to choose which node to fix (after the first random choice) we used the following heuristic: we compute the entropy of the beliefs of each node after each full BP run and choose the node that its total entropy is the largest. That is, we try to fix the nodes most uncertain about their labeling.

3.4 Algorithm Summary

We summarize our algorithm for finding the optimal solution for graph partitioning with fractional objective function in Algorithm 1. Please notice that in order to solve the λ question we usually do not need to find $\tilde{E}^*_{current}$ exactly. In order to know $\tilde{E}^*_{current}$ is below zero all we need is to find a specific $x^{(t)}$ for which $\tilde{E}(x^{(t)}) < 0$ (see Figure 4a). The bound gives us this service from the other end (Figure 4b). We compute the current energy and bound efficiently every few iterations, usually, this allows us to terminate our BP before its convergence. We will wait until convergence when $\tilde{E}^*_{current} = 0$, in this case we will know we have found the optimal solution (Figure 4c). We emphasize that without the bound we could answer the λ question only in cases where we found an example for which $\tilde{E}(x^{(t)}) < 0$.

Algorithm 1

$\tilde{E}(x;\lambda) = f(x) - \lambda g(x)$
$\lambda_{lower} \leftarrow$ initial lower bound (default: 0)
$\lambda_{upper} \leftarrow$ initial upper bound (default: guess x, $\frac{f(x)}{g(x)}$)
while $\lambda_{lower} + \epsilon < \lambda_{upper}$ **do**
  $\lambda_{current} = \frac{\lambda_{lower}+\lambda_{upper}}{2}$
  Using convex BP with HOP and conditioning find: $\tilde{E}^*_{current}$=$\min_x \tilde{E}(x;\lambda_{current})$
  if $\tilde{E}^*_{current}$==0 **then**
    $\lambda_{lower} = \lambda_{upper} = \lambda_{current}$
  else
    if $\tilde{E}^*_{current} < 0$ **then**
      $\lambda_{upper} = \lambda_{current}$
    else
      $\lambda_{lower} = \lambda_{current}$
    end if
  end if
end while

4 Experiments

We used our method to find the optimum of average cut problem on several benchmark problems: from clustering two dimensional points, through image segmentation to financial optimization. In all the experiments the input was the symmetric affinity matrix containing the affinities between each pair of data points. Our initial upper bound was the spectral solution (using zero as a threshold on the second smallest eigenvector of the Laplacian to partition the points) and the lower bound was the second smallest eigenvalue (the Fiedler value). We also provided to the method the required interval between the upper bound to the lower bound, if we achieved it we announced we got to the optimal solution[1].

Notice that for a fixed λ our method has a random component - the first conditioned variable. It might be that in two runs using the same λ one run of the algorithm will answer the λ question and the other will not. Because of that, if our algorithm did not succeed to answer the λ question we do 3 more trials before terminating the entire run announcing we failed to find the optimal solution.

[1]Using the bisection algorithm we cut by half the interval between the lower and upper bounds on each successful iteration. Since the computer has limited accuracy we can announce that we have found the optimal solution when the interval is small enough.

Though we did not put an effort in optimizing our code, we mention here that running the experiments took from several seconds (when the number of nodes, n was 25) to several tens of minutes ($n = 37,376$).

We have made the code to reproduce these results available online, it can be downloaded at `http://www.cs.huji.ac.il/~mezuman/code/avgcut.tar`.

Clustering Two-Dimensional Data Points

The similarity between every pair of 25 two dimensional data points was set to the exponent of the negative squared distance between the points divided by $\sigma^2 = 6.25$, that is, $w_{i,j} = \exp(\frac{-\|x_i - x_j\|^2}{\sigma^2})$. We used the same two dimensional data points as in Frey and Dueck (2007). As can be seen in Figure 5 our method got the the optimal average cut.

Clustering Images Derived from Olivetti Face Database

We took the data for the faces images from Frey and Dueck (2007): "Each 64×64 face image from the first 100 images in the Olivetti database was smoothed using a Gaussian kernel with σ=0.5 and then rotated by -10°, 0° and 10° and scaled by a factor of 0.9, 1.0 and 1.1 (using nearest-neighbor interpolation), to produce a total of 900 images. To avoid including the background behind each face, a central window of size 50×50 pixels was extracted. Finally, the pixels in each 50×50 image were normalized to have mean 0 and variance 0.1. The similarity between two images was set to the negative sum of squared pixel differences". The input affinities to our method were the exponent of the similarities between images, which was just described, divided by $\sigma^2 = 1.69$. In this experiment our method proved that the solution received from the spectral method is the optimal average cut by improving the lower bound. The best average cut divided the 900 images to 810 images of different people and 90 images of the same person. The results can be seen in Figure 6.

Image Segmentation

Our next experiment was on images. Each pixel in the image was a node in the graph and the weights of the edges were computed using intervening contours (Leung and Malik, 1998) using the implementation provided by Cour et al. (2010).

The first image we examined was a baby image (Figure 7d), which was taken from Cour et al. (2010). When we used the image in its original size (132x130) the optimal average cut that was found using our method (Fig. 7a) was not good for image segmentation - the cut separated 2 pixels from all the other pixels. We must emphasize that these 2 pixels are connected to the rest of the image. When we resized the image to half of its original size, our method found the global optimum (Fig. 7b) that separated the baby from the background (Fig. 7e). The spectral method also finds this partition, but cannot prove that it is optimal as there is still a gap between the spectral lower and upper bounds.

The second image we experimented on, 'a man with a hat' (Fig. 7f), was taken from the Berkeley segmentation dataset (Martin et al., 2001). The image's size was first reduced to 160x107 pixels. Looking on the image segmentation that we got (Fig. 7h) we can see that although the optimal solution is 10% better in value than the spectral solution, the differences are in only few pixels. This difference can merely be noticed (Fig. 7g).

To obtain a better measure of the success of our method, we ran our method over the 200 training images of Berkeley segmentation dataset, after resizing them to 100x67 pixels (having total of 6,700 nodes). We compared the interval between the upper and lower bounds on the optimal solution before using our method (i.e. using the spectral method) and after using our method. For 59% of the images we improved this interval finding for 88% of the images the global optimum up to accuracy of 0.1, see Figure 8.

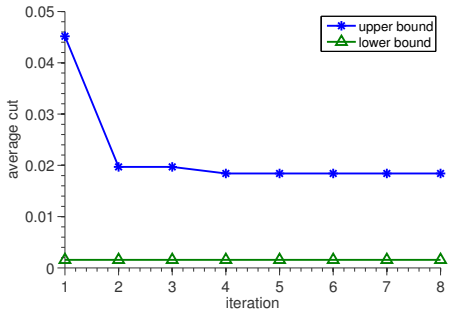


Figure 9: Finance256. The lower and upper bounds on the optimal solution for average cut after each iteration of the bisection algorithm. As can be seen our method succeeded to improve the upper bound (found a better average cut than the spectral one) but failed to improve the lower bound.

Financial Optimization

We downloaded the affinity matrix known as finance256 (its graph partition has application for financial optimization), from The University of Florida Sparse Matrix Collection (Davis et al., 1997). It is one of the graph partitioning benchmark problems experimented on by Dhillon et al. 2007. This problem contains $37,376$ nodes and $130,560$ edges. In this experiment we failed to find the optimal average cut, but did succeed to find a better cut than the spectral solution and improve the upper bound, see Figure 9.

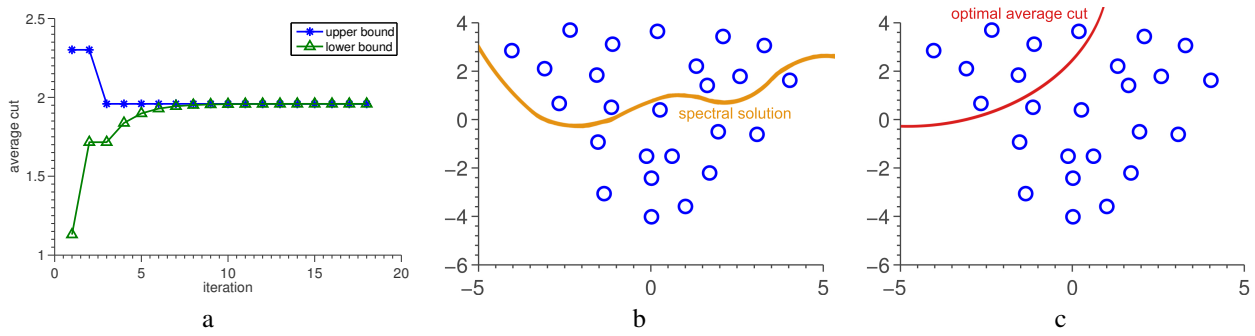


Figure 5: Clustering two dimensional data points. (a) The lower and upper bounds on the optimal solution for average cut after each iteration of the bisection algorithm. (b) The spectral solution for average cut for this problem. (c) The optimal average cut found using our method

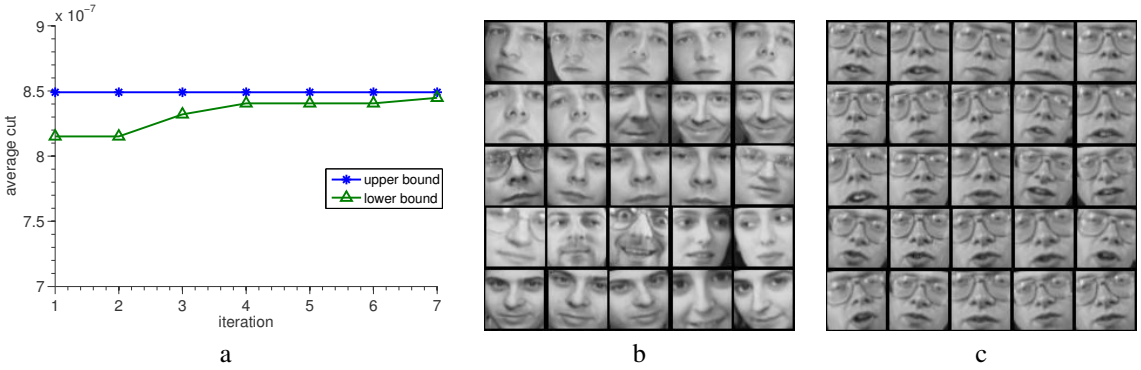


Figure 6: Clustering Images Derived from Olivetti Face Database. (a) The lower and upper bounds on the optimal solution for average cut after each iteration of the bisection algorithm. (b) Random sample of 25 faces from the larger part of the cut (810 images). (c) Random sample of 25 faces from the smaller part of the cut (90 images).

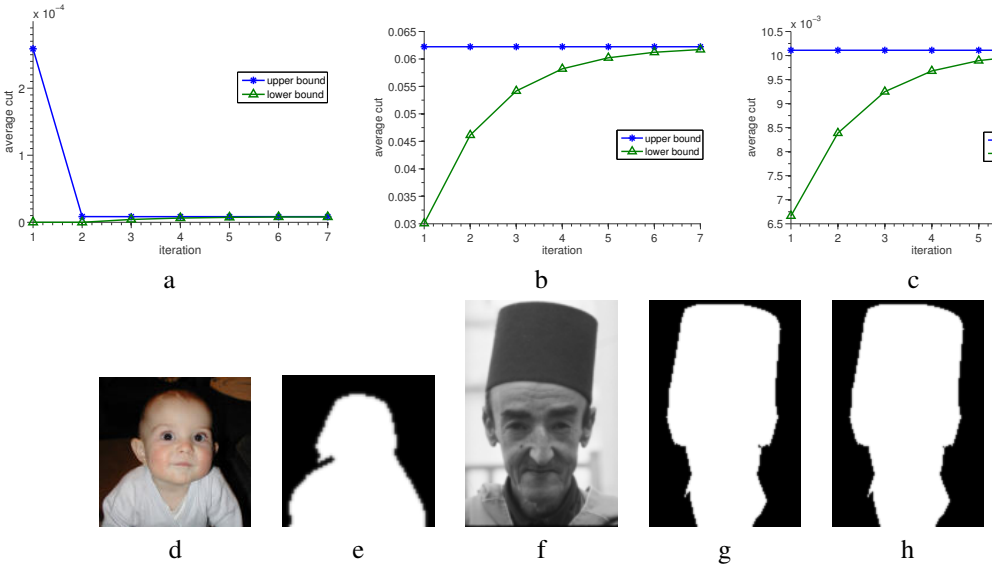
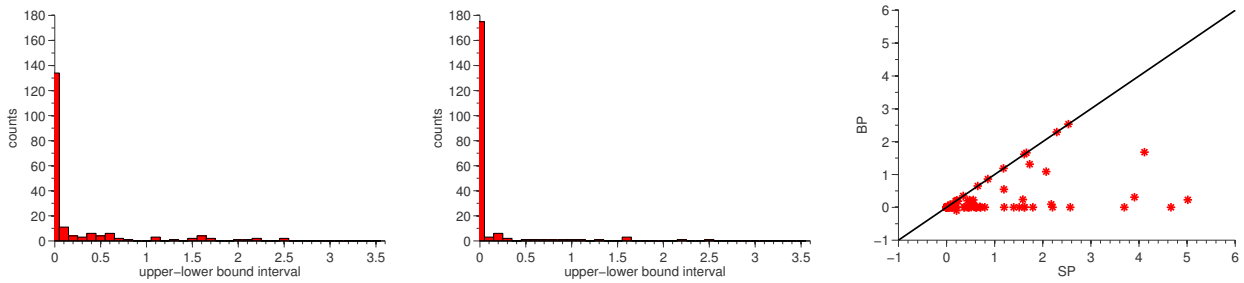


Figure 7: Image Segmentation. The lower and upper bounds on the optimal solution for average cut at each iteration of the bisection algorithm, for the (a) original and (b) resized baby images and (c) 'a man with a hat' image. (d) Baby input image (e) Segmentation result on the resized image (f) 'A man with a hat' input image. Segmentation results using: (g) the spectral method (h) our method .



a b c

Figure 8: Statistics on the success of our method to find the global optimum of average cut. The statistics were done on 180 resized images from Berkeley segmentation dataset. Histograms showing the differences between the upper and lower bounds for average cut when using (a) spectral method - SP (b) our method - BP. (c) BP interval vs. SP interval, points below the diagonal are images for which our method decreased the upper-lower bounds interval.

	2d Points	Faces	Baby	Baby (small)	Man	Finance
Cour et al. (2010) code for normalized-cut (ncut)	3.5e-01	4.3e-03	1.0e-03	9.8e-04	1.9e-04	9.9e-02
ncut using the partitioning from average cut	3.6e-01	7.8e-07	4.2e-06	9.8e-04	1.4e-04	2.3e-03
ratio	104.40%	0.02%	0.42%	100.00%	72.63%	2.33%

Table 1: Comparison of the normalized cut value achieved using the spectral method (using the code from Cour et al.) and using the partition found for average cut using our method.

Normalized cut

As mentioned above, when the affinities are integers, our framework allows solving the normalized cut. For the case when the affinities are not integers, we experimented with simply using the average cut algorithm since it has been shown, e.g by Soundararajan and Sarkar (2001), that average cut and normalized cut often have very similar results. We compared our results to the spectral method that directly attempts to approximately optimize the normalized cut (Shi and Malik, 2000). As can be seen in Table 1, usually, when the spectral average cut was improved using our method the same partitioning also improved the spectral normalized cut.

5 Discussion

In this paper we presented a new framework to find the global optimum of graph partitioning problems. The basic building blocks of our method are: (1) Linearizing a ratio problem to get the λ question (Hochbaum, 2010). (2) Convex message passing algorithm with a bound on the objective function (Weiss et al., 2007).(3) Efficient MAP inference with high order potentials (Tarlow et al., 2010).(4) Tightening linear programming relaxation using conditioning. (Rother et al., 2007, Boros et al., 2006). By using tools from graphical models we were able to efficiently answer the λ question and provably find the global optimum for fractional graph partitioning problems.

For some of the experiments we have conducted, the simple conditioning algorithm we have used did not tighten the LP relaxation enough: our convex BP method converged to beliefs with "ties". Specifically, when we conducted experiments on larger images (e.g. 240x160 pixels, 38k nodes) conditioning on only few (up to 10) variables was not enough and we got beliefs with ties. We plan to deal with these cases by adapting ideas from Sontag et al. (2008) who tighten the relaxation by adding an explicit treatment for "frustrated cycles" (i.e. 3). In our problems, we have found that no frustrated cycles exist between nodes that correspond to datapoints, so we need to modify the algorithm in Sontag et al. (2008) to deal with the high order potential. We also plan to improve our conditioning and probing techniques (Rother et al., 2007).

We established a framework to find the global optimum of graph partitioning problems. This framework should now be used to examine the true compatibility of graph partitioning algorithms to the application they were used for. We hope that questions like: "Is the fault in the graph-partitioning criteria (e.g. normalized cut) or is the fault in the relaxation (e.g. the spectral solution)?", will now be answered.

Acknowledgments

The authors wish to thank David Sontag for helpful discussions and Danny Rosenberg for his help in writing the code. This research was funded by the Israel Science Foundation.

References

E. Boros, P.L. Hammer, and G. Tavares. Preprocessing of unconstrained quadratic binary optimization. *RUTCOR Research Report, RRR*, pages 10–2006, 2006.

T.H. Cormen. *Introduction to algorithms*. The MIT press, 2001.

T. Cour, S. Yu, and J. Shi. Matlab normalized cuts segmentation code. `http://www.seas.upenn.edu/~timothee/software/ncut/ncut.html`, 2010.

T. Davis et al. University of florida sparse matrix collection. *NA digest*, 97(23):7, 1997.

I.S. Dhillon, Y. Guan, and B. Kulis. Weighted graph cuts without eigenvectors a multilevel approach. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(11):1944–1957, 2007.

Brendan J. Frey and Delbert Dueck. Clustering by passing messages between data points. *Science*, 315:972–976, 2007.

A. Globerson and T. Jaakkola. Fixing max-product: Convergent message passing algorithms for map lp-relaxations. *Advances in Neural Information Processing Systems*, 21(1.6), 2007.

D.S. Hochbaum. Polynomial time algorithms for ratio regions and a variant of normalized cut. *IEEE transactions on pattern analysis and machine intelligence*, pages 889–898, 2010.

V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1568–1583, 2006.

N. Komodakis, N. Paragios, and G. Tziritas. Mrf optimization via dual decomposition: Message-passing revisited. 2007.

T. Leung and J. Malik. Contour continuity in region based image segmentation. *Computer Vision ECCV98*, pages 544–559, 1998.

D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.

C. Rother, V. Kolmogorov, V. Lempitsky, and M. Szummer. Optimizing binary mrfs via extended roof duality. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. Ieee, 2007.

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

D. Sontag, T. Meltzer, A. Globerson, T. Jaakkola, and Y. Weiss. Tightening lp relaxations for map using message passing. 2008.

P. Soundararajan and S. Sarkar. Analysis of mincut, average cut, and normalized cut measures. In *Proc. Third Workshop Perceptual Organization in Computer Vision*. Citeseer, 2001.

D. Tarlow, I.E. Givoni, and R.S. Zemel. Hopmap: Efficient message passing with high order potentials. In *Proc. of AISTATS*, 2010.

M.J. Wainwright and M.I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1-2):1–305, 2008.

Y. Weiss, C. Yanover, and T. Meltzer. Map estimation, linear programming and belief propagation with convex free energies. In *Uncertainty in Artificial Intelligence*. Citeseer, 2007.

Y. Weiss, C. Yanover, and T. Meltzer. *Linear Programming and variants of Belief Propagation in (Blake and Rother, ed.) Markov Random Fields for Vision and Image Processing*, chapter 6. Mit Pr, 2011.

T. Werner. A linear programming approach to max-sum problem: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1165–1179, 2007.

Z. Wu and R. Leahy. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, pages 1101–1113, 1993.