Biased Normalized Cuts

Subhransu Maji[1], Nisheeth K. Vishnoi[2], and Jitendra Malik[1]
University of California, Berkeley[1]
Microsoft Research India, Bangalore[2]
smaji@eecs.berkeley.edu, nisheeth.vishnoi@gmail.com, malik@eecs.berkeley.edu

Abstract

We present a modification of "Normalized Cuts" to incorporate priors which can be used for constrained image segmentation. Compared to previous generalizations of "Normalized Cuts" which incorporate constraints, our technique has two advantages. First, we seek solutions which are sufficiently "correlated" with priors which allows us to use noisy top-down information, for example from an object detector. Second, given the spectral solution of the unconstrained problem, the solution of the constrained one can be computed in small additional time, which allows us to run the algorithm in an interactive mode. We compare our algorithm to other graph cut based algorithms and highlight the advantages.

1. Introduction

Consider Figure 1. Suppose that we want to segment the cat from its background. We can approach this

1. Bottom-up, we might detect contours corresponding to significant changes in brightness, color or texture. The output of one such detector is shown in Figure 1(b). Note that there will be internal contours, e.g. corresponding to the colored spots, that are of higher contrast than the external contour separating the cat from its background. This makes it difficult for any bottom-up segmentation process to separate the cat from its background in the absence of high-level knowledge of what cats look like.

2. Top-down, we might look for strong activation of, say, a sliding window cat detector. The output of one such part-based detector is shown in Figure 1(c). But note that this too is inadequate for our purpose. The outputs of these detectors are typically not sharply localized; indeed that is almost a consequence of the desire to make them invariant to small deformations.

In this paper we present an approach, "Biased Normalized Cuts", where we try to get the best of both worlds,

(a) Image (b) Bottom-up (c) Top-down (d) Biased Ncut

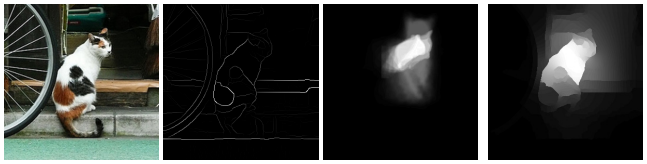


Figure 1. Segmentation using bottom-up & top-down information.

with results as shown in Figure 1(d). The formalism is that of graph partitioning approached using tools and techniques from spectral graph theory. As in Normalized Cuts [18], we begin by computing eigenvectors of the Normalized Laplacian of a graph where the edge weights w_{ij} correspond to the "affinity" between two pixels based on low-level similarity cues. This is illustrated in Figure 2 for the cat image, where the top row shows the second to fifth eigenvectors, u_i. Note that none of them is a particularly good indicator vector from which the cat can be extracted by thresholding.

We are interested in cuts which not only minimize the normalized cut value but, at the same time, one of the sets in the partition $(S, \bar{S})$ is required to have a sufficient overlap with a "bias" or a "prior guess". In Figure 1, this was the output of a top-down detector. In Figure 2, bottom row, the user has specified the bias by indicating various point sets T in the different panels. We will define a seed-vector s_T associated to T. Let D_G be the diagonal matrix of the graph, and u_i, λ_i be the eigenvectors and eigenvalues of the Normalized Laplacian. Theorem 3.1 (originally due to [15]) shows that we can get cuts that meet the twin goals of having a small normalized cut, and being sufficiently correlated with s_T by a remarkably simple procedure. Construct the biased normalized cut vector, $x^\star$, where $x^\star = c\sum_{i=2}^{n} \frac{1}{\lambda_i - \gamma} u_i \left(u_i^T D_G s_T\right)$. Intuitively the eigenvectors are linearly combined such that the ones that are well correlated with the seed vector are up weighted, while those that are inversely correlated have their sign flipped. In Figure 2 bottom row, these vectors are shown as images, and the bias point sets are overlaid. This vector $x^\star$ can be thresholded to find the desired "biased normalized cut". One can

run the algorithm in an interactive mode by computing the eigenvectors once and generating the biased normalized cut for any bias.

2. Previous Work

Spectral graph theory originated in the 1970s with work such as Donath and Hoffman [10] and Fiedler [13]. By the 1990s, there was a well developed literature in mathematics and theoretical computer science, and Chung [9] provides a monograph level treatment. In computer vision and machine learning, the work of Shi and Malik on normalized cuts [18, 19] was very influential, and since then a significant body of work has emerged in these fields that we cannot possibly do justice here. The best current approaches to finding contours and regions bottom up are still based on variations of this theme [16].

Given our interest in this work of biasing the solution to satisfy additional constraints, two papers need particular mention. The problem in the constrained setting for image segmentation has been studied by Yu and Shi [21] where they find the solution to the normalized cuts problem subject to a set of linear constraints of the form $U^T x = 0$. This problem can be reduced to an eigenvalue problem which can be solved using spectral techniques as well. Ericksson *et al*. [11] generalize the set of constraints to $U^T x = b$. We will see in Section 5 that enforcing constraints in this manner is not robust when the constraints are noisy. The computational complexity of these approaches is also significantly higher than that of solving the basic normalized cuts problem.

In the last decade the most popular approach to interactive image segmentation in computer vision and graphics has followed in the steps of the work of Boykov and Jolly [7] who find an image segmentation by computing a min-cut/max-flow on a graph which encodes both the user constraints and pairwise pixel similarity. This line of work has been further investigated by Blake and Rother [3] where they experiment with ways to model the foreground and background regions. In the GrabCut framework [17], the process of segmentation and foreground/background modeling is repeated till convergence. With the advances in the min-cut/max-flow algorithms like [6], these methods have become computationally attractive and can often be used in an interactive mode. However, these methods fail when the constraints are sparse making it difficult to construct good foreground/background models and these methods tend to produce isolated cuts.

From a theoretical perspective, there has been a significant interest in the cut improvement problem: given as input a graph and a cut, find a subset that is a better cut. This started with Gallo, Grigoriadis and Tarjan [14] and has attained more recent attention in the work of Andersen and Lang [2], who gave a general algorithm that uses a small number of single-commodity maximum-flows to find low-conductance cuts not only inside the input subset T, but among all cuts which are well-correlated with $(T, \bar{T})$. Among spectral methods, local graph partitioning was introduced by Spielman and Teng [20], who were interested in finding a low-conductance cut in a graph in time nearly-linear in the volume of the output cut. They used random walk based methods to do this; and subsequently this result was improved by Andersen, Chung and Lang [1] by doing certain Personalized PageRank based random walks.

3. Biased Graph Partitioning

Image as a Weighted Graph and Normalized Cuts. The image is represented as a weighted undirected graph $G = (V, E)$ where the nodes of the graph are the points in the feature space, and an edge is formed between every pair of nodes. The weight function on the edges $w : E \mapsto \mathbb{R}_{\geq 0}$ is a function of the similarity between the end points of the edge. The volume of a set of vertices S is the total weight of the edges incident to it: $\mathrm{vol}(S) \stackrel{\text{def}}{=} \sum_{i \in S, j \in V} w(i,j)$. $\mathrm{vol}(G) \stackrel{\text{def}}{=} \sum_{i,j \in V} w(i,j)$ be the total weight of the edges in the graph. Normalized cut measure, defined next, is a standard way to measure the degree of dissimilarity between two pieces of an image. For $S \subseteq V$, let $\bar{S}$ denote $V \backslash S$. Let $\mathrm{cut}(S, \bar{S}) \stackrel{\text{def}}{=} \sum_{i \in S, j \in \bar{S}} w(i,j)$ be the weight of edges crossing the cut $(S, \bar{S})$. Then, the normalized cut value corresponding to $(S, \bar{S})$ is defined as

$$\mathrm{Ncut}(S) \stackrel{\text{def}}{=} \frac{\mathrm{cut}(S, \bar{S})}{\mathrm{vol}(S)} + \frac{\mathrm{cut}(S, \bar{S})}{\mathrm{vol}(\bar{S})}.$$

Noting that $\mathrm{vol}(S) + \mathrm{vol}(\bar{S}) = \mathrm{vol}(G)$, it can be seen that this notion of a normalized cut is the same as that of the traditional graph conductance defined for a set S as

$$\phi(S) \stackrel{\text{def}}{=} \mathrm{vol}(G) \cdot \frac{\mathrm{cut}(S, \bar{S})}{\mathrm{vol}(S) \cdot \mathrm{vol}(\bar{S})}.$$

The *conductance of the graph* G is $\phi(G) \stackrel{\text{def}}{=} \min_{S \subseteq V} \phi(S)$ and is an extremely well studied quantity in graph theory, computer science and machine learning. It is NP-hard to compute exactly and one of the earliest and most popular approach to compute an approximation to it is to write down a relaxation which boils down to computing the second eigenvalue of the Laplacian matrix associated to a graph. We discuss this next.

Graphs and Laplacians. For a graph $G = (V, E)$ with edge weights function w, let $A_G \in \mathbb{R}^{V \times V}$ denote its adjacency matrix with $A_G(i,j) = w(i,j)$; D_G denotes the diagonal degree matrix of G, *i.e.*, $D_G(i,i) = \sum_{j \in V} w(i,j)$ $D_G(i,j) = 0$, for all $i \neq j$; $L_G \stackrel{\text{def}}{=} D_G - A_G$ will

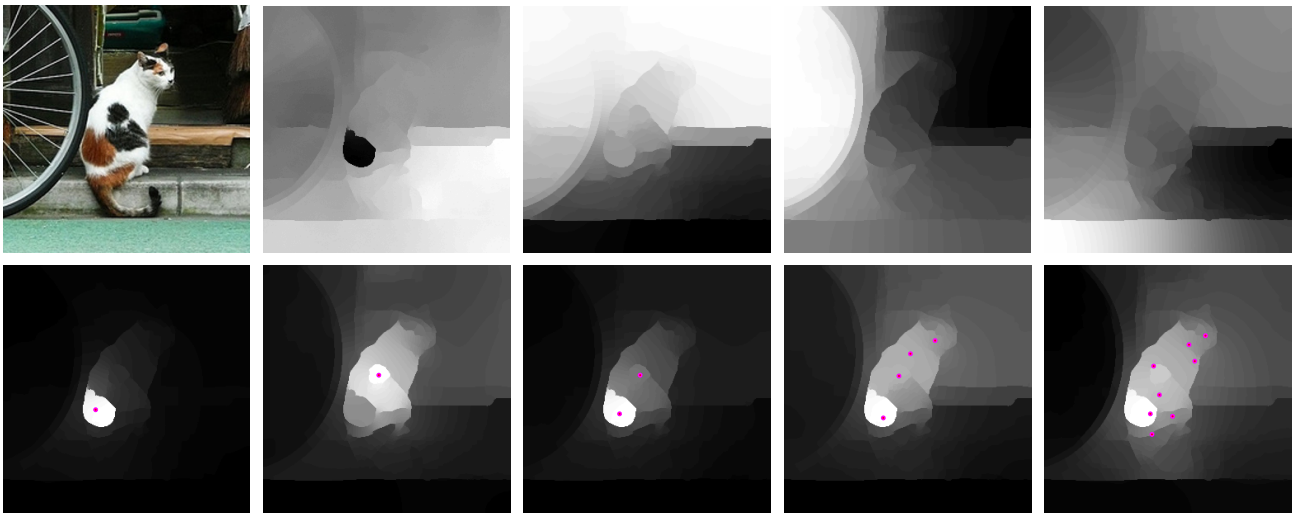


Figure 2. **Top Row:** Input Image and the top 4 eigenvectors computed using the *intervening contour cue* [16]. **Bottom Row:** Biased normalized cuts for various seed sets T. The marked points in each image are the set of points in T.

denote the (combinatorial) Laplacian of G; and $\mathcal{L}_G \stackrel{\text{def}}{=} D_G^{-1/2} L_G D_G^{-1/2}$ will denote the normalized Laplacian of G. We will assume G is connected, in which case the eigenvalues of $\mathcal{L}_G$ are $0 = \lambda_1 < \lambda_2 \leq \cdots \leq \lambda_n$. We will denote by $\lambda_2(G)$ as this second eigenvalue of the normalized Laplacian of G. If $u_1, \ldots, u_n$ are the corresponding eigenvectors of $\mathcal{L}_G$, then we define $v_i \stackrel{\text{def}}{=} D_G^{-\frac{1}{2}} u_i$ and think of them as the associated eigenvectors of L_G; $L_G v_i = \lambda_i D_G v_i$. The spectral decomposition of $\mathcal{L}_G$ can be written as $\sum_{i=2}^{n} \lambda_i u_i u_i^T$. We can also define the Moore-Penrose pseudo-inverse of $\mathcal{L}_G$ as $\mathcal{L}_G^+ \stackrel{\text{def}}{=} \sum_{i=2}^{n} \frac{1}{\lambda_i} u_i u_i^T$.

The Spectral Relaxation to Computing Normalized Cuts. Spectral methods approximate the solution to normalized cuts by trying to find a $x \in \mathbb{R}^V$ which minimizes $x^T L_G x$ subject to $\sum_{i,j \in V} d_i d_j (x_i - x_j)^2 = \text{vol}(G)$. To see why this is a relaxation first, for a subset S of vertices, let 1_S be the indicator vector of S in $\mathbb{R}^V$. Then $1_S^T L_G 1_S = \sum_{ij \in E} w_{i,j} (1_S(i) - 1_S(j))^2 = \text{cut}(S, \bar{S})$ and $\sum_{i,j \in V} d_i d_j (1_S(i) - 1_S(j))^2 = \text{vol}(S)\text{vol}(\bar{S})$. Hence, if we let $x = \sqrt{\frac{\text{vol}(G)}{\text{vol}(S) \cdot \text{vol}(\bar{S})}} \cdot 1_S$, then x satisfies the above constraints and has objective value exactly $\phi(S)$. Hence, the minimum of the quadratic program above can be at-most the normalized cut value of G.

In fact, it is easy to see that the optimal value of this optimization problem, if G is connected, is $\lambda_2(G)$ and the optimal vector to this program is $D_G^{-\frac{1}{2}} u_2$, where u_2 is the eigenvector corresponding to the smallest non-zero eigenvalue ($\lambda_2(G)$) of $\mathcal{L}_G$. In particular, the optimal vector $x^\star$ has the property that $\sum_{i \in V} x_i^\star d_i = 0$.

Biased Normalized Cuts. Now we move on to incorporating the prior information given to us about the image to define the notion of biased normalized cuts. Recall that our problem is: we are given a region of interest in the image and we would like to segment the image so that the segment is biased towards the specified region. A region is modeled as a subset $T \subseteq V$, of the vertices of the image. We would be interested in cuts $(S, \bar{S})$ which not only minimize the normalized cut value but, at the same time, have sufficient correlation with the region specified by T. To model this, we will first associate a vector s_T to the set T as follows: $s_T(i) = \sqrt{\frac{\text{vol}(T)\text{vol}(\bar{T})}{\text{vol}(G)}} \cdot \frac{1}{\text{vol}(T)}$, if $i \in T$, and $s_T(i) = -\sqrt{\frac{\text{vol}(T)\text{vol}(\bar{T})}{\text{vol}(G)}} \cdot \frac{1}{\text{vol}(\bar{T})}$ if $i \in \bar{T}$; or equivalently, $s_T \stackrel{\text{def}}{=} \sqrt{\frac{\text{vol}(T)\text{vol}(\bar{T})}{\text{vol}(G)}} \left(\frac{1_T}{\text{vol}(T)} - \frac{1_{\bar{T}}}{\text{vol}(\bar{T})} \right)$. We have defined it in a way such that $\sum_{i \in V} s_T(i) d_i = 0$ and $\sum_{i \in V} s_T(i)^2 d_i = 1$.

This notion of biased normalized cuts is quite natural and motivated from the theory of local graph partitioning where the goal is to find a low-conductance cut well correlated with a specified input set. The correlation is specified by a parameter $\kappa \in (0, 1)$. This allows us to explore the possibility of image segments which are well-correlated with the prior information which may have much less normalized cut value than T itself and, hence, refine the initial guess. In particular, we consider the spectral relaxation in Figure 3 to a κ-biased normalized cut around T.

Note that $x = s_T$, is a feasible solution to this spectral relaxation. Also note that if v_2 satisfies the correlation constraint with s_T, then it will be the optimal to this program. What is quite interesting is that one can characterize the optimal solution to this spectral program under mild condi-

$$\begin{aligned} \text{minimize} \quad & x^T L_G x \\ \text{s.t.} \quad & \sum_{i,j \in V} d_i d_j (x_i - x_j)^2 = \text{vol}(G) \\ & (\sum_{i \in V} x_i s_T(i) d_i)^2 \geq \kappa \end{aligned}$$

Figure 3. BiasedNcut(G, T, κ)- Spectral relaxation to compute κ-biased normalized cuts around T

tions on the graph G and the set T and, as it turns out, one has to do very little effort to compute the optimal solution if one has already computed the spectrum of $\mathcal{L}_G$. This is captured by the following theorem which is due to [15]. We include a proof of this in the appendix for completeness.

Theorem 3.1. *Let G be a connected graph and T be such that $\sum_{i \in V} s_T(i) v_2(i) d_i \neq 0$. Further, let $1 \geq \kappa \geq 0$ be a correlation parameter. Then, there is an optimal solution, $x^\star$, to the spectral relaxation to the κ-biased normalized cuts around T such that $x^\star = c \sum_{i=2}^{n} \frac{1}{\lambda_i - \gamma} u_i u_i^T D_G s_T$ for some $\gamma \in (-\infty, \lambda_2(G))$ and a constant c.*

4. Algorithm

Theorem 3.1 shows that the final solution is a weighted combination of the eigenvectors and the weight of each eigenvector is proportional to the "correlation" with the seed, $u_i^T D_G s_T$ and inversely proportional to $\lambda_i - \gamma$. Intuitively the eigenvectors that are well correlated with the seed vector are up weighted, while those that are inversely correlated have their sign flipped.

Often for images the $\lambda_i - \gamma$ grows quickly and one can obtain a good approximation by considering eigenvectors for the K smallest eigenvalues. In our experiments with natural images we set $K = 26$, i.e. use the top 25 eigenvectors ignoring the all ones vector. We also set γ parameter which controls the amount of correlation implicitly to $\gamma = -\tau \times \lambda_{avg}$, where $\tau = 1$ and λ_{avg} is the average of the top K eigenvalues. This could also be user defined parameter in an interactive setting. Algorithm 1 describes our method for computing the biased normalized cuts. Steps $1, 2, 3$ are also the steps for computing the segmentations using normalized cuts which involve computing the K smallest eigenvectors of the normalized graph laplacian $\mathcal{L}_G$. The biased normalized cut for any seed vector s_T is the weighted combination of eigenvectors where the weights are computed in step 4. In the interactive setting only steps 4 and 5 need to be done when the seed vector changes which is very quick.

The time taken by the algorithm is dominated by the time taken to compute the eigenvectors. In an interactive setting one can use special purpose hardware accelerated methods to compute the eigenvectors of typical images in fraction of a second [8]. Our method can be faster than the min-cut/max-flow cut based approaches in an interactive setting as these eigenvectors need to be computed just once. In addition the real valued solution like the one shown in Figure 2 might provide the user better guidance than a hard segmentation produced by a min-cut algorithm.

Algorithm 1 Biased Normalized Cuts (G, w, s_T, γ)

Require: Graph $G = (V, E)$, edge weight function w, seed s_T and a correlation parameter $\gamma \in (-\infty, \lambda_2(G))$

1: $A_G(i,j) \leftarrow w(i,j)$, $D_G(i,i) \leftarrow \sum_j w(i,j)$

2: $L_G \leftarrow D_G - A_G$, $\mathcal{L}_G \leftarrow D_G^{-1/2} L_G D_G^{-1/2}$

3: Compute $u_1, u_2, \ldots, u_K$ the eigenvectors of $\mathcal{L}_G$ corresponding to the K smallest eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_K$.

4: $w_i \leftarrow \frac{u_i^T D_G s_T}{\lambda_i - \gamma}$, for $i = 2, \ldots, K$

5: Obtain the biased normalized cut, $x* \propto \sum_{i=2}^{K} w_i u_i$

5. Constrained Normalized Cuts

We compare our approach to the constrained normalized cut formulation of Yu and Shi [21] and Ericksson *et al.* [11]. The later generalized the linear constraints to $U^T P x = b$, where b could be an arbitrary non-zero vector. We consider a toy example similar to the one used by Yu and Shi. The authors observe that when the set of constraints are small it is better to use "conditioned constraints", $U^T P x = 0$, where $P = D^{-1} W$, instead of the original constraints, $U^T x = 0$. The "conditioned constraints" propagate the constraints to the neighbors of the constrained points avoiding solutions that are too close to the unconstrained one.

To illustrate our approach, let us consider the points $p_1, p_2, \ldots p_n$ grouped into three sets S_1, S_2 and S_3 from left to right, as shown in Figure 4. We construct a graph with edge weights between points p_i and p_j

$$w_{ij} = \exp\left(-d(p_i, p_j)^2 / 2\sigma^2\right) \tag{1}$$

with $\sigma = 3$, where $d(x, y)$ is the Euclidean distance between the points x and y. The unconstrained solution to the normalized cuts correctly groups the points as shown in Figure 4.

Now suppose we want to group S_1 and S_3 together. This can be done by adding a constraint that the circled points belong to the same group and can encoded as a linear constraint of the form, $x_i - x_j = 0$, where i and j are the indices of the constrained points. The constrained cut formulation is able to correctly separate S_1 and S_3 from S_2 as shown in Figure 5(a). For our approach we construct the vector s_T with $i, j \in T$ defined earlier and use the top 16 generalized eigenvectors. The biased cut solution separates S_1 and S_3 from S_2 as well.

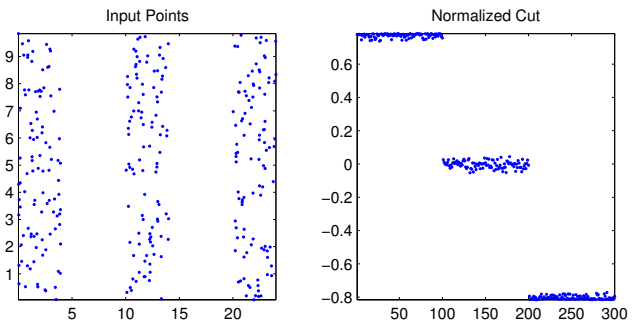


Figure 4. Input points (left) clustered using normalized cuts (right).

Next we increase the number of such constraints by randomly sampling points from $S_1 \cup S_3$ and adding constraints that they belong together. Given a set of n points we generate $n-1$ constraints by ordering the points and adding a constraint for each consecutive pair similar to the approach of [21]. Instead of improving the solution, the solution to the constrained cuts deteriorates when the number of constraints are large as the solution that separates S_1 and S_3 from S_2 is no longer feasible. Our method on the other hand gets better with more constraints as seen in Figure 5(b).

We also test the robustness of the algorithm to outliers by adding some points from S_2 into the constraints. The constrained cut solution deteriorates even when there is one outlier as seen in Figure 5(c). Our method on the other hand remains fairly robust even when there are two outliers Figure 5(d).

6. Qualitative Evaluation

We present qualitative results on images taken from PASCAL VOC 2010 dataset [12]. For all our experiments we use the intervening contour cue [16] for computing the weight matrix and 25 eigenvectors to compute the biased normalized cut. We refrain from doing quantitive comparison on segmentation benchmarks as the goal of this paper is to introduce a new algorithm for computing biased normalized cuts, which is only one of the components of a good segmentation engine.

Effect of γ. The correlation parameter γ when set smaller values increases the correlation κ with the seed vector. Figure 6 shows the effect of γ on the result of the biased normalized cut. One can obtain tighter cuts around the seed by setting γ to smaller values. In an interactive setup this could be adjusted by the user.

Bias from user interaction. We first test the algorithm in the interactive setting by automatically sampling a set of points inside the figure mask. Figure 7 shows the several example outputs. All these images are taken from the animal categories of the Pascal VOC 2010 detection dataset.

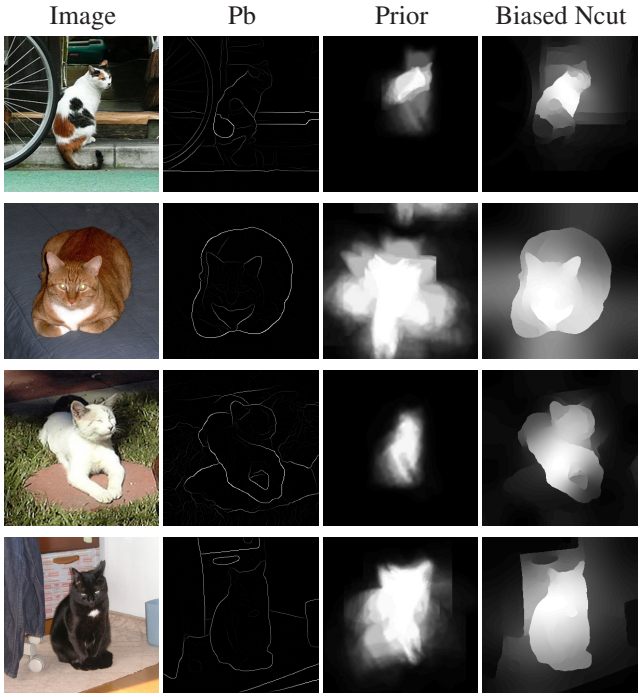


Figure 8. Biased normalized cuts using top-down priors from an object detector.

Bias from object detectors. Although in our formulation the seed vector s_T was discrete, it is not necessary. In particular we can use the probability estimates produced by an object detector as a bias. Figure 8 shows some examples of a top down segmentation mask generated by the detector of [4] used as a seed vector directly after normalization.

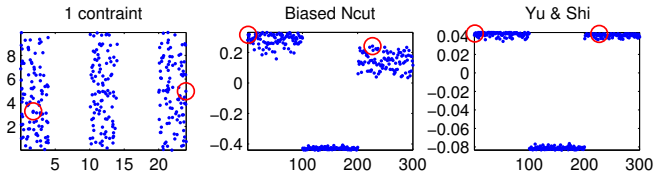
7. Conclusion

We present a simple and effective method for computing the biased normalized cuts of the images to incorporate top-down priors or user input. The formulation is attractive as it allows one to incorporate these priors without additional overhead. Linear combinations of eigenvectors naturally "regularizes" the combinatorial space of segmentations. Code for computing biased normalized cuts interactively on images can be downloaded at the author's website.

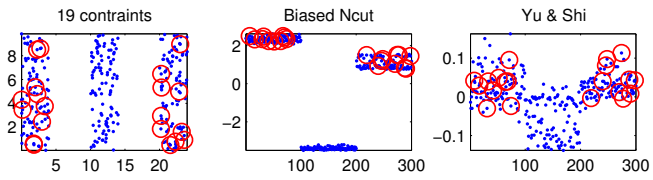
Acknowledgements: Thanks to Pablo Arbelaez. Subhransu Maji is supported by a fellowship from Google Inc. and ONR MURI N00014-06-1-0734. The work was done when the authors were at Microsoft Research India.

A. Appendix

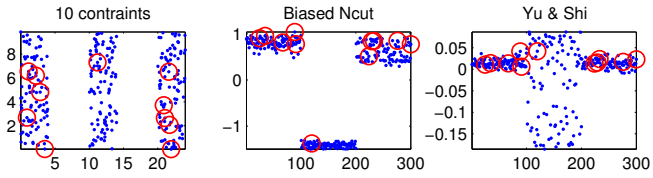
Proof. [of Theorem 3.1] To start off, note that $\mathsf{BiasedNcut}(G, T, \kappa)$ is a non-convex program. One can relax it to $\mathsf{SDP}_p(G, T, \kappa)$ of Figure 9. In the same figure the dual of this SDP appears: $\mathsf{SDP}_d(G, T, \kappa)$. Here, for matrices $A, B \in \mathbb{R}^{n \times n}$, $A \circ B \stackrel{\text{def}}{=} \sum_{i,j} A_{i,j} B_{i,j}$. Also $L_n \stackrel{\text{def}}{=} D_G - \frac{1}{\text{vol}(G)} D_G J D_G$, where J is the matrix



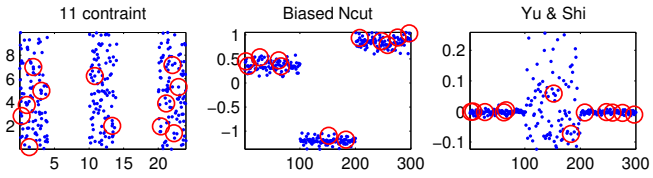
(a) One point each sampled from S_2 and S_3



(b) 10 points each sampled from S_2 and S_3



(c) 5 points each from sets S_1 and S_3 with 1 outlier from S_2



(d) 5 points each from sets S_1 and S_3 with 2 outliers from S_2

Figure 5. Comparison of our biased normalized cut approach to constrained cut approach of [21, 11] for various constraints shown by red circled points. When the number of constraints is large **(b)** or there are outliers **(c, d)**, our method produces better solutions than the solution of constrained cuts.

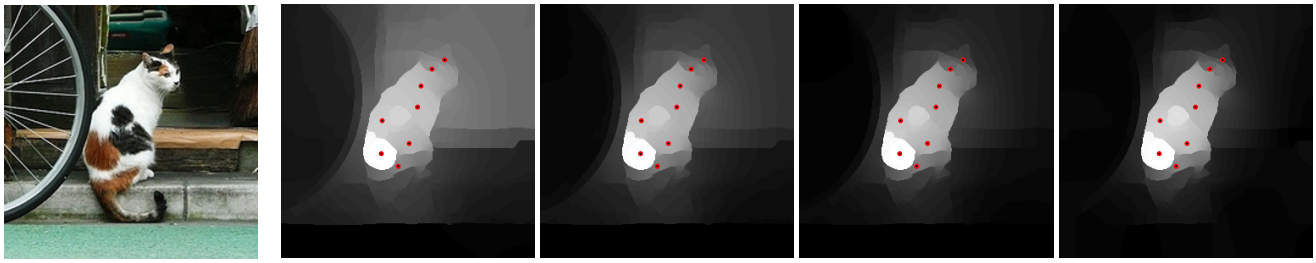


Figure 6. **Effect of γ.** Input image and biased cuts for decreasing values of γ from left to right. The local cuts are more and more correlated with the seed set (shown as dots) as γ decreases.

with $J_{i,j} = 1$ for all i, j. Verifying Slater's condition, one can observe that strong duality holds for this SDP relaxation. Then, using strong duality and the complementary slackness conditions implied by it, we will argue that the $\mathsf{SDP}_p(G, T, \kappa)$ has a rank one optimal solution under the conditions of the theorem. This implies that the optimal solution of $\mathsf{SDP}_p(G, T, \kappa)$ is the same as the optimal solution of BiasedNcut. Combining this with the complementary slackness condition obtained from the dual $\mathsf{SDP}_d(G, T, \kappa)$, one can derive that the optimal rank one solution has, up to a constant, the desired form promised by the theorem. Now we expand the above steps in claims.

Claim A.1. *The primal* $\mathsf{SDP}_p(G, T, \kappa)$ *is a relaxation of the vector program* $\mathsf{BiasedNcut}(G, T, \kappa)$.

Proof. Consider a vector x that is a feasible solution to $\mathsf{BiasedNcut}(G, T, \kappa)$, and note that $X = xx^T$ is a feasible solution to $\mathsf{SDP}_p(G, T, \kappa)$. □

Claim A.2. *Strong duality holds between* $\mathsf{SDP}_p(G, T, \kappa)$ *and* $\mathsf{SDP}_d(G, T, \kappa)$.

Proof. Since $\mathsf{SDP}_p(G, T, \kappa)$ is convex, if suffices to verify that Slater's constraint qualification condition is true for this primal SDP. Consider $X = s_T s_T{}^T$. Then, $(D_G s_T)(D_G s_T)^T \circ s_T s_T{}^T = (s_T{}^T D_G s_T)^2 = 1 > \kappa$. □

$$
\begin{aligned}
\text{minimize} \quad & L_G \circ X \\
\text{subject to} \quad & L_n \circ X = 1 \\
& (D_G s_T)(D_G s_T)^T \circ X \geq \kappa \\
& X \succeq 0
\end{aligned}
$$

$$
\begin{aligned}
\text{maximize} \quad & \alpha + \kappa\beta \\
\text{subject to} \quad & L_G \succeq \alpha L_n + \beta (D_G s_T)(D_G s_T)^T \\
& \alpha \in \mathbb{R},\ \beta \geq 0
\end{aligned}
$$

Figure 9. Top: $\mathsf{SDP}_p(G, T, \kappa)$ Primal SDP relaxation. Bottom: $\mathsf{SDP}_d(G, T, \kappa)$ Dual SDP

Claim A.3. *The feasibility and complementary slackness conditions for a primal-dual pair* $X^\star, \alpha^\star, \beta^\star$ *listed in Figure 10 are sufficient for them to be an optimal solution.*

Proof. This follows from the convexity of $\mathsf{SDP}_p(G, T, \kappa)$ and Slater's condition [5]. □

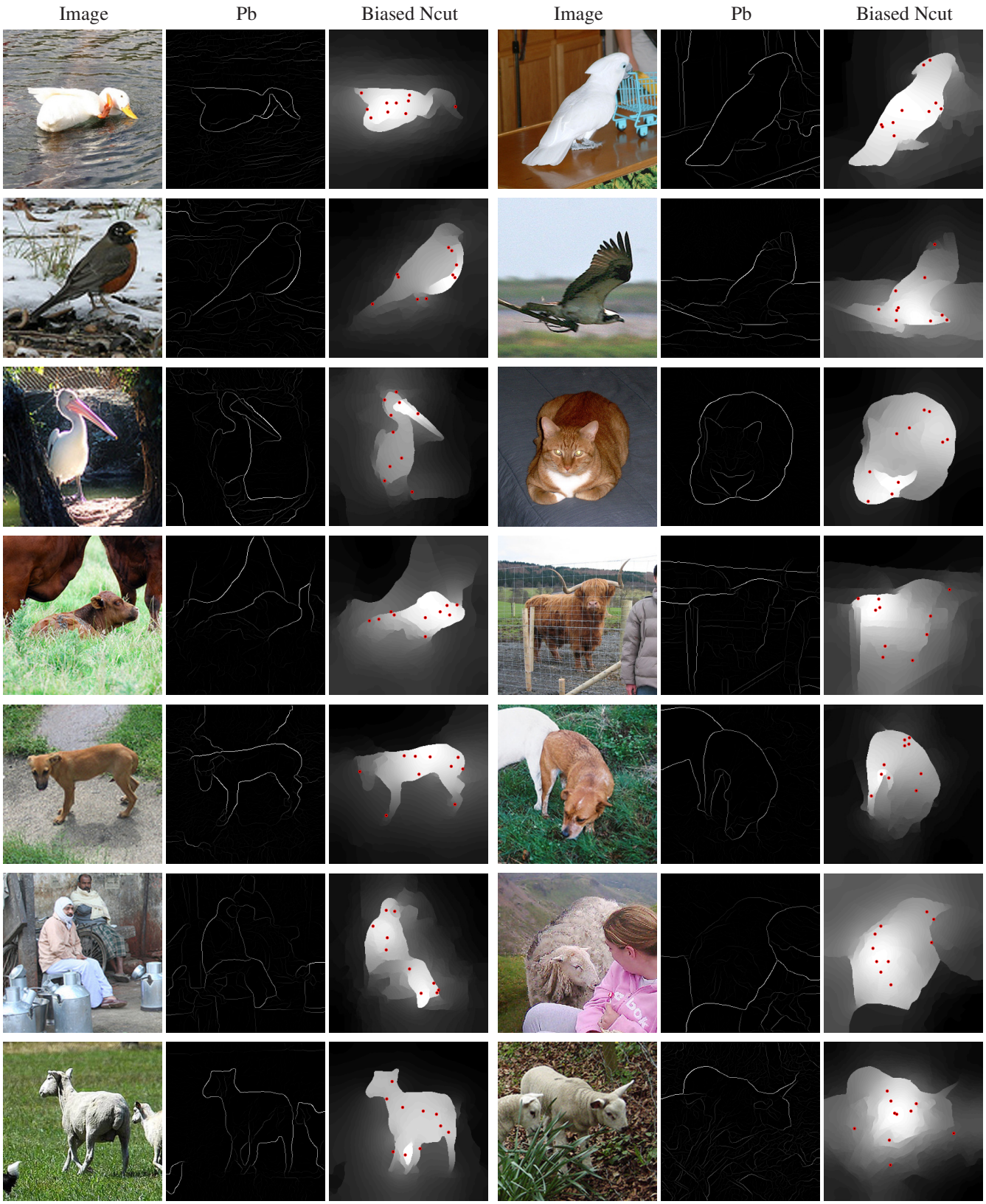


Figure 7. **Example biased cuts.** Images shown with the "probability of boundary" map (Pb) computed using [16] and the biased normalized cut for various seed sets (marked as red dots).

$$
\begin{aligned}
L_n \circ X^\star &= 1 \\
(D_G s_T)(D_G s_T)^T \circ X^\star &\geq \kappa \\
L_G - \alpha^\star L_n - \beta^\star (D_G s_T)(D_G s_T)^T &\succeq 0 \\
\beta^\star &\geq 0
\end{aligned}
$$

$$
\begin{aligned}
\alpha^\star (L_n \circ X^\star - 1) &= 0 \\
\beta^\star ((D_G s_T)(D_G s_T)^T \circ X^\star - \kappa) &= 0 \\
X^\star \circ (L_G - \alpha^\star L_n - \beta^\star (D_G s_T)(D_G s_T)^T) &= 0
\end{aligned}
$$

Figure 10. Top: Feasibility conditions. Bottom: Complementary slackness conditions.

Claim A.4. *These feasibility and complementary slackness conditions, coupled with the assumptions of the theorem, imply that $X^\star$ must be rank 1 and $\beta^\star > 0$.*

Now we complete the proof of the theorem. From Claim A.4 it follows that, $X^\star = x^\star x^{\star T}$ where $x^\star$ satisfies the equation $(L_G - \alpha^\star L_n - \beta^\star (D_G s_T)(D_G s_T)^T) x^\star = 0$. From the second complementary slackness condition in Figure 10, and the fact that $\beta^\star > 0$, we obtain that $\sum_i x_i^\star s_T(i) d_i = \pm\sqrt{\kappa}$. Thus, $x^\star = \pm\beta^\star \sqrt{\kappa} (L_G - \alpha^\star L_n)^+ D_G s_T$. This proves the theorem. □

A.1. Proof of Claim A.4

Proof. We start by stating two facts. The second is trivial.

Fact A.5. *$\alpha^\star \leq \lambda_2(G)$. Moreover if $\lambda_2 = \alpha^\star$ then $\sum_i v_2(i) s_T(i) d_i = 0$.*

Proof. Recall that $v_2 = D_G^{-\frac{1}{2}} u_2$ where u_2 is the unit length eigenvector corresponding to $\lambda_2(G)$ of $\mathcal{L}_G$. Plugging in v_2 in Equation 3 from the feasibility conditions, we obtain that $v_2^T L_G v_2 - \alpha^\star - \beta^\star (\sum_i v_2(i) s_T(i) d_i)^2 \geq 0$. But $v_2^T L_G v_2 = \lambda_2(G)$ and $\beta^\star \geq 0$. Hence, $\lambda_2(G) \geq \alpha^\star$. It follows that if $\lambda_2 = \alpha^\star$ then $\sum_i v_2(i) s_T(i) d_i = 0$. □

Fact A.6. *We may assume that the optimal $X^\star$ satisfies $1^T D_G^{\frac{1}{2}} X^\star D_G^{\frac{1}{2}} 1 = 0$, where 1 is the all ones vector.*

Now we return to the proof of Claim A.4. If we assume $\sum_i v_2(i) s_T(i) d_i \neq 0$, then we know that $\alpha^\star < \lambda_2(G)$ from Fact A.5. Note that since G is connected and $\alpha^\star < \lambda_2(G)$, $L_G - \alpha^\star L_n$ has rank exactly $n-1$. From the complementary slackness condition 2 we can deduce that the image of $X^\star$ is in the kernel of $L_G - \alpha^\star L_n - \beta^\star (D_G s_T)(D_G s_T)^T$. But $\beta^\star (D_G s_T)(D_G s_T)^T$ is a rank one matrix and since $\sum_i s_T(i) d_i = 0$, it reduces the rank of $L_G - \alpha^\star L_n$ by one precisely when $\beta^\star > 0$. If $\beta^\star = 0$ then $X^\star$ must be 0 which is not possible if $\mathsf{SDP}_p(G, T, \kappa)$ is feasible. Hence, the rank of $L_G - \alpha^\star L_n - \beta^\star (D_G s_T)(D_G s_T)^T$ must be exactly $n-2$ and since $X^\star$ cannot have any component along the all ones vector, $X^\star$ must be rank one. This proves the claim. □

References

[1] R. Andersen, F. Chung, and K. Lang. Local graph partitioning using PageRank vectors. In *FOCS*, pages 475–486, 2006. 2058

[2] R. Andersen and K. Lang. An algorithm for improving graph partitions. In *SODA*, pages 651–660, 2008. 2058

[3] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr. Interactive image segmentation using an adaptive gmmrf model. In *ECCV*, pages 428–441, 2004. 2058

[4] L. Bourdev, S. Maji, T. Brox, and J. Malik. Detecting people using mutually consistent poselet activations. In *ECCV*, sep 2010. 2061

[5] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK, 2004. 2062

[6] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *TPAMI*, 26:1124–1137, September 2004. 2058

[7] Y. Y. Boykov and M. P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In *ICCV*, 2001. 2058

[8] B. Catanzaro, B.-Y. Su, N. Sundaram, Y. Lee, M. Murphy, and K. Keutzer. Efficient, high-quality image contour detection. In *ICCV*, pages 2381 –2388, 2009. 2060

[9] F. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997. 2058

[10] W. E. Donath and A. J. Hoffman. Lower bounds for the partitioning of graphs. *IBM J. Res. Dev.*, 17:420–425, September 1973. 2058

[11] A. Eriksson, C. Olsson, and F. Kahl. Normalized cuts revisited: A reformulation for segmentation with linear grouping constraints. In *ICCV*, oct. 2007. 2058, 2060, 2062

[12] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, June 2010. 2061

[13] M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(98):298–305, 1973. 2058

[14] G. Gallo, M. D. Grigoriadis, and R. E. Tarjan. A fast parametric maximum flow algorithm and applications. *SIAM J. Comput.*, 18(1):30–55, 1989. 2058

[15] M. W. Mahoney, L. Orecchia, and N. K. Vishnoi. A spectral algorithm for improving graph partitions. *CoRR*, abs/0912.0681, 2009. 2057, 2060

[16] M. Maire, P. Arbelaez, C. Fowlkes, and J. Malik. Using contours to detect and localize junctions in natural images. In *CVPR*, pages 1 –8, 2008. 2058, 2059, 2061, 2063

[17] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23:309–314, 2004. 2058

[18] J. Shi and J. Malik. Normalized cuts and image segmentation. In *CVPR*, jun. 1997. 2057, 2058

[19] J. Shi and J. Malik. Normalized cuts and image segmentation. *TPAMI*, 22(8):888 –905, Aug. 2000. 2058

[20] D. Spielman and S.-H. Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *STOC*, pages 81–90, 2004. 2058

[21] S. X. Yu and J. Shi. Grouping with bias. In *NIPS*. MIT Press, 2001. 2058, 2060, 2061, 2062