Semi-Supervised Normalized Cuts for Image Segmentation

Selene E. Chew and Nathan D. Cahill
School of Mathematical Sciences, Rochester Institute of Technology
sec6971@rit.edu, nathan.cahill@rit.edu

Abstract

Since its introduction as a powerful graph-based method for image segmentation, the Normalized Cuts (NCuts) algorithm has been generalized to incorporate expert knowledge about how certain pixels or regions should be grouped, or how the resulting segmentation should be biased to be correlated with priors. Previous approaches incorporate hard must-link *constraints on how certain pixels should be grouped as well as hard* cannot-link *constraints on how other pixels should be separated into different groups. In this paper, we reformulate NCuts to allow both sets of constraints to be handled in a soft manner, enabling the user to tune the degree to which the constraints are satisfied. An approximate spectral solution to the reformulated problem exists without requiring explicit construction of a large, dense matrix; hence, computation time is comparable to that of unconstrained NCuts. Using synthetic data and real imagery, we show that soft handling of constraints yields better results than unconstrained NCuts and enables more robust clustering and segmentation than is possible when the constraints are strictly enforced.*

1. Introduction

One of the most popular algorithms for image segmentation and clustering is *Normalized Cuts* [13, 14] (NCuts), which generates segments by partitioning a graph that models the image. Although NCuts has been successful in many settings, it is a fully automatic algorithm, and no fully automatic segmentation algorithm currently exists that is as good as the human brain at handling the complexity and variety inherent in real-world images. Because of this, one vibrant area of research over the last decade has been into the development of *interactive* image segmentation algorithms that rely on user input to help guide the segmentation process.

Like NCuts, many of the modern interactive image segmentation techniques are based on graph partitioning and labeling algorithms [4, 3, 6, 8, 12, 18]. User input about pixels or regions of the image can be provided as *must-link* or *cannot-link* constraints [16], which specify that two or more vertices of the graph should be grouped in the same partition or separated into different partitions.

Two extensions of NCuts have been developed that handle *hard* constraints: Yu and Shi [18] enable hard must-link constraints, and Eriksson et al. [6] enable both hard must-link and cannot-link constraints. A different generalization is the *Biased* Normalized Cut, by Maji et al. [10], in which the resulting clustering or segmentation can be biased towards being correlated with some predefined template or function. If this function is defined by the user to indicate pixels/regions that should be linked, Biased Normalized Cuts can be thought of as a way of modifying the NCuts solution towards satisfying must-link constraints in a *soft* manner.

In this paper, we present a different generalization of NCuts that allows *soft* versions of both must-link and cannot-link constraints to be satisfied, enabling an expert user to vary the desired influence of specific constraints on the partitioning process. Soft versions of the constraints allow for flexibility to be robust to situations when user input is not guaranteed to be completely accurate. Our generalization, which we refer to as *Semi-Supervised Normalized Cuts* (SSNCuts), has an approximate spectral solution that does not require explicit construction of a large, dense matrix; hence, computation time is comparable to that of the original NCuts algorithm. Furthermore, it directly computes the constrained solution without relying on an initial unconstrained solution (as opposed to the approach of Biased NCuts). If desired, SSNCuts can form the basis of a foreground extraction system like GrabCuts [12], by iteratively updating the manual labels and then estimating the alpha-matte around the region border.

The remainder of this paper is organized as follows. Section 2 reviews the original NCuts algorithm, and Section 3 shows how to generalize it to incorporate soft must-link and cannot-link constraints. Section 4 uses the toy problem presented in [10] to explore the behavior of our generalization compared to methods requiring strict constraint satisfaction [6, 18]. Section 5 illustrates image segmentation results on a variety of images from the PASCAL VOC dataset [7].

A preliminary version of this paper appeared in [5], in which we proposed a generalization of NCuts to handle only soft must-link constraints.

2. Normalized Cuts

Consider an undirected weighted graph $\mathcal{G} = (V, \mathcal{E})$ that we wish to partition into two disjoint subgraphs $\mathcal{G}_A = (A, \mathcal{E}_A)$, $\mathcal{G}_B = (B, \mathcal{E}_B)$, where $A \bigcup B = V$. Partitioning can be achieved by removing the edges connecting A to B; the cost of partitioning $\mathcal{G}$ is called the *cut* cost and is defined by the total weight of the edges that have been removed:

$$\text{cut}(A, B) = \sum_{v_i \in A, v_j \in B} W_{i,j} \quad , \tag{1}$$

where the vertex set $V = \{v_1, v_2, \cdots, v_n\}$, and where $\mathbf{W}$ is the weighted adjacency matrix of $\mathcal{G}$.

To find an optimal partitioning of $\mathcal{G}$, one strategy is to minimize (1) to find the *minimum cut*. As described in [17], however, such a minimum cut can be unnaturally biased towards partitionings in which one of the subgraphs has a single vertex. To yield more balanced partitionings, other related cut costs have been proposed, one of the most popular of which is the *Normalized Cut* [14]:

$$\text{NCut}(A, B) = \frac{\text{cut}(A, B)}{\text{assoc}(A, X)} + \frac{\text{cut}(A, B)}{\text{assoc}(B, X)} \quad , \tag{2}$$

where

$$\text{assoc}(S, X) = \sum_{v_i \in S, v_j \in X} W_{i,j} \tag{3}$$

is the total connection from all vertices in S to all vertices in the graph $\mathcal{G}$.

Shi and Malik [14] show that minimizing (2) is equivalent to solving the following discrete minimization problem:

$$\begin{aligned} \min_{\mathbf{y}} \quad & \frac{\mathbf{y}^\mathbf{T}(\mathbf{D} - \mathbf{W})\,\mathbf{y}}{\mathbf{y}^\mathbf{T}\mathbf{D}\mathbf{y}} \\ \text{subject to} \quad & y_i \in \{1, -b\} \quad , \quad i = 1, 2, \ldots, n \quad , \\ & \mathbf{y}^\mathbf{T}\mathbf{D}\mathbf{1} = 0 \quad , \end{aligned} \tag{4}$$

where $\mathbf{D}$ is the diagonal weighted degree matrix defined componentwise by $d_i = D_{i,i} = \sum_j W_{i,j}$, $b = \left(\sum_{x_i > 0} d_i\right) / \left(\sum_{x_i < 0} d_i\right)$, and $\mathbf{y} = (\mathbf{1} + \mathbf{x})/2 - b(\mathbf{1} - \mathbf{x})/2$, where $\mathbf{x}$ is an n-dimensional indicator vector such that $x_i = 1$ if vertex v_i is in A and $x_i = -1$ otherwise. They further show that if (4) is relaxed so that the components of $\mathbf{y}$ are real-valued, its solution is the generalized eigenvector corresponding to the smallest nontrivial eigenvalue of:

$$(\mathbf{D} - \mathbf{W})\,\mathbf{y} = \lambda \mathbf{D}\mathbf{y} \quad . \tag{5}$$

The components of the resulting generalized eigenvector can then be thresholded to assign them one of the discrete labels.

An alternate way to formulate the discrete minimization problem (4) is presented in Eriksson et al. [6] and involves parameterizing directly with $\mathbf{x}$:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \frac{\left(\mathbf{x}^\mathbf{T}(\mathbf{D} - \mathbf{W})\,\mathbf{x}\right)\mathbf{d}^\mathbf{T}\mathbf{1}}{\mathbf{x}^\mathbf{T}\left((\mathbf{d}^\mathbf{T}\mathbf{1})\,\mathbf{D} - \mathbf{d}\mathbf{d}^\mathbf{T}\right)\mathbf{x}} \\ \text{subject to} \quad & x_i \in \{1, -1\} \quad , \quad i = 1, 2, \ldots, n \quad , \end{aligned} \tag{6}$$

where $\mathbf{d} = \mathbf{D}^\mathbf{T}\mathbf{1}$ is the vector containing the diagonal elements of $\mathbf{D}$. We go one step further; noting that:

$$\begin{aligned} \frac{(\mathbf{d}^\mathbf{T}\mathbf{1})\,\mathbf{D} - \mathbf{d}\mathbf{d}^\mathbf{T}}{\mathbf{d}^\mathbf{T}\mathbf{1}} &= \mathbf{D} - \frac{\mathbf{d}\mathbf{d}^\mathbf{T}}{\mathbf{d}^\mathbf{T}\mathbf{1}} = \mathbf{D} - \frac{\mathbf{D}\mathbf{1}\mathbf{1}^\mathbf{T}\mathbf{D}}{\mathbf{1}^\mathbf{T}\mathbf{D}\mathbf{1}} \\ &= \mathbf{D}^{1/2}\left(\mathbf{I} - \frac{\mathbf{D}^{1/2}\mathbf{1}}{\|\mathbf{D}^{1/2}\mathbf{1}\|}\left(\frac{\mathbf{D}^{1/2}\mathbf{1}}{\|\mathbf{D}^{1/2}\mathbf{1}\|}\right)^\mathbf{T}\right)\mathbf{D}^{1/2} \quad , \end{aligned} \tag{7}$$

which allows us to state (6) as:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \frac{\mathbf{x}^\mathbf{T}(\mathbf{D} - \mathbf{W})\,\mathbf{x}}{\mathbf{x}^\mathbf{T}\mathbf{D}^{1/2}(\mathbf{I} - \mathbf{q}\mathbf{q}^\mathbf{T})\,\mathbf{D}^{1/2}\mathbf{x}} \\ \text{subject to} \quad & x_i \in \{1, -1\} \quad , \quad i = 1, 2, \ldots, n \quad , \end{aligned} \tag{8}$$

where $\mathbf{q}$ is the unit vector in the direction of $\mathbf{D}^{1/2}\mathbf{1}$.

There are two reasons we prefer to represent NCut minimization by (8) as opposed to (4). First, b is not actually constant with respect to $\mathbf{y}$; hence, the class labels for $\mathbf{y}$ are actually a function of $\mathbf{y}$ itself, whereas the class labels for $\mathbf{x}$ are fixed. Second, the balance constraint $\mathbf{y}^\mathbf{T}\mathbf{D}\mathbf{1} = 0$, while required by (4), is not actually required by (8) because it is implicitly satisfied by the solution to the relaxed version of (8).

To see why the latter point is true, we first note that taking the gradient of the objective function in (8) and setting it equal to zero shows us that any critical point $\mathbf{x}^*$ of the relaxed version of (8) must satisfy the generalized eigenvector problem:

$$(\mathbf{D} - \mathbf{W})\,\mathbf{x}^* = \lambda \mathbf{D}^{1/2}\left(\mathbf{I} - \mathbf{q}\mathbf{q}^\mathbf{T}\right)\mathbf{D}^{1/2}\mathbf{x}^* \quad . \tag{9}$$

This is equivalent to the generalized eigenvector problem:

$$\mathbf{D}^{-1/2}\,(\mathbf{D} - \mathbf{W})\,\mathbf{D}^{-1/2}\mathbf{z}^* = \left(\mathbf{I} - \mathbf{q}\mathbf{q}^\mathbf{T}\right)\mathbf{z}^* \quad , \tag{10}$$

where $\mathbf{z}^* = \mathbf{D}^{1/2}\mathbf{x}^*$.

Using the orthogonal decomposition theorem, we write $\mathbf{z}^* = \alpha_1 \mathbf{q} + \alpha_2 \mathbf{z}^*_{\mathbf{q}^\perp}$, where $\mathbf{q}^\mathbf{T}\mathbf{z}^*_{\mathbf{q}^\perp} = 0$. First, note that if $\alpha_2 = 0$, $\mathbf{x}^* = \mathbf{D}^{-1/2}\mathbf{z}^*$ represents a trivial cut of $\mathcal{G}$, so to avoid this case we assume $\alpha_2 \neq 0$. Now, noting that $\mathbf{q}$ is an eigenvector of $\mathbf{D}^{-1/2}\,(\mathbf{D} - \mathbf{W})\,\mathbf{D}^{-1/2}$ with corresponding eigenvalue 0 and that $\mathbf{I} - \mathbf{q}\mathbf{q}^\mathbf{T}$ projects $\mathbf{q}$ to zero and preserves any vector orthogonal to $\mathbf{q}$, we have:

$$\mathbf{D}^{-1/2}\,(\mathbf{D} - \mathbf{W})\,\mathbf{D}^{-1/2}\mathbf{z}^*_{\mathbf{q}^\perp} = \lambda \mathbf{z}^*_{\mathbf{q}^\perp} \quad . \tag{11}$$

Furthermore, the objective function evaluated at $\mathbf{z}^*$ simplifies to:

$$\frac{\mathbf{z}^{*\mathbf{T}}\mathbf{D}^{-1/2}\left(\mathbf{D}-\mathbf{W}\right)\mathbf{D}^{-1/2}\mathbf{z}^*}{\mathbf{z}^{*\mathbf{T}}\left(\mathbf{I}-\mathbf{q}\mathbf{q}^{\mathbf{T}}\right)\mathbf{z}^*} \tag{12}$$
$$= \frac{\mathbf{z}^{*\mathbf{T}}\mathbf{D}^{-1/2}\left(\mathbf{D}-\mathbf{W}\right)\mathbf{D}^{-1/2}\mathbf{z}^*_{\mathbf{q}^\perp}}{\mathbf{z}^{*\mathbf{T}}\mathbf{z}^*_{\mathbf{q}^\perp}} = \frac{\lambda\mathbf{z}^{*\mathbf{T}}\mathbf{z}^*_{\mathbf{q}^\perp}}{\mathbf{z}^{*\mathbf{T}}\mathbf{z}^*_{\mathbf{q}^\perp}} = \lambda \ .$$

Since $\lambda = 0$ would correspond to $\mathbf{z}^*_{\mathbf{q}^\perp} = \mathbf{q}$ (which is impossible because they are orthogonal), $\mathbf{z}^*_{\mathbf{q}^\perp}$ must be the generalized eigenvector corresponding to the second smallest eigenvalue of (11). Once $\mathbf{z}^*_{\mathbf{q}^\perp}$ is determined, α_1 can be free to be chosen to be any value without impacting the objective function nor changing the fact that $\mathbf{z}^*$ is a critical point. Thus, we can choose $\alpha_1 = 0$; this ensures that $\mathbf{z}^*$ itself is the eigenvector corresponding to the second smallest eigenvalue of:

$$\mathbf{D}^{-1/2}\left(\mathbf{D}-\mathbf{W}\right)\mathbf{D}^{-1/2}\mathbf{z} = \lambda\mathbf{z} \tag{13}$$

and consequently, that $\mathbf{x}^*$ is the generalized eigenvector corresponding to the second smallest eigenvalue of:

$$\left(\mathbf{D}-\mathbf{W}\right)\mathbf{x} = \lambda\mathbf{D}\mathbf{x} \ . \tag{14}$$

This also ensures that the balance constraint is implicitly enforced (because generalized eigenvectors of $\mathbf{A}\mathbf{x} = \lambda\mathbf{B}\mathbf{x}$ corresponding to distinct eigenvalues are $\mathbf{B}$-orthogonal if $\mathbf{B}$ is symmetric positive definite [11]).

3. Incorporating Soft Constraints

In many situations, prior knowledge exists about how some pairs of vertices should or should not be grouped in the resulting graph partitioning procedure. Wagstaff et al. [16] describes this type of knowledge in terms of *must-link* and *cannot-link* constraints; must-link constraints specify that two vertices are grouped in the same partition, and cannot-link constraints specify that two vertices are separated into different partitions. Eriksson et al. [6] show how NCuts can be generalized to handle *hard* versions of must-link and cannot-link constraints. In this article, we present a generalization that allows *soft* versions of both types of constraints, enabling an expert user to vary the desired influence of specific constraints on the partitioning process.

3.1. Must-Link Constraints

Prior knowledge in the form of soft must-link constraints is straightforward to incorporate in a semi-supervised version of NCuts. Suppose we know that each of the ordered pairs of vertices in the set $\mathcal{C} = \{(v_{i_\ell}, v_{j_\ell})\,|\,\ell = 1,\ldots,m\}$ represents two vertices that should be grouped in the same partition. A modified cut cost can be formulated that penalizes violations of the must-link constraints:

$$\text{cut}_{\text{ML}}(A,B) = \text{cut}(A,B) + \tfrac{1}{2}\sum_{\ell=1}^{m}\gamma_\ell\cdot\theta(v_{i_\ell},v_{j_\ell}) \ , \tag{15}$$

where $\theta(v_i, v_j)$ is the *must-link penalty* defined to be 1 if v_i and v_j are in different subgraphs (i.e., if $\{v_i \in A \text{ and } v_j \in B\}$ or $\{v_i \in B \text{ and } v_j \in A\}$) and 0 if they are in the same subgraph. The γ_ℓ's specify the strength of each must-link constraint.

Using (15) in place of (1) in (2), we can further modify the NCut cost by defining:

$$\text{NCut}_{\text{ML}}(A,B) = \frac{\text{cut}_{\text{ML}}(A,B)}{\text{assoc}(A,X)} + \frac{\text{cut}_{\text{ML}}(A,B)}{\text{assoc}(B,X)} = \tag{16}$$
$$\text{NCut}(A,B) + \frac{\sum_{\ell=1}^{m}\gamma_\ell\cdot\theta(v_{i_\ell},v_{j_\ell})}{2\cdot\text{assoc}(A,X)} + \frac{\sum_{\ell=1}^{m}\gamma_\ell\cdot\theta(v_{i_\ell},v_{j_\ell})}{2\cdot\text{assoc}(B,X)} \ .$$

The must-link penalty function can also be written in terms of the corresponding components of the indicator vector $\mathbf{x}$; i.e., $\theta(v_i, v_j) = (x_i - x_j)^2/4$. This allows us to express $\sum_{\ell=1}^{m}\gamma_\ell\cdot\theta(v_{i_\ell},v_{j_\ell})$ as $\mathbf{x}^{\mathbf{T}}\mathbf{U}^{\mathbf{T}}\mathbf{\Gamma}\mathbf{U}\mathbf{x}$, where $\mathbf{U}$ is the $m\times n$ matrix defined row-wise so that row ℓ contains a $1/2$ in column i_ℓ, a $-1/2$ in column j_ℓ, and 0's in every other column, and where $\mathbf{\Gamma}$ is the diagonal matrix containing the weights on its main diagonal. (Note that in the special case where all weights are equal to γ, $\mathbf{x}^{\mathbf{T}}\mathbf{U}^{\mathbf{T}}\mathbf{\Gamma}\mathbf{U}\mathbf{x}$ reduces to $\gamma\mathbf{x}^{\mathbf{T}}\mathbf{U}^{\mathbf{T}}\mathbf{U}\mathbf{x}$.) Hence, minimizing (16) is equivalent to solving the discrete minimization problem:

$$\min_{\mathbf{x}} \quad \frac{\mathbf{x}^{\mathbf{T}}\left(\mathbf{D}-\mathbf{W}+\mathbf{U}^{\mathbf{T}}\mathbf{\Gamma}\mathbf{U}\right)\mathbf{x}}{\mathbf{x}^{\mathbf{T}}\mathbf{D}^{1/2}\left(\mathbf{I}-\mathbf{q}\mathbf{q}^{\mathbf{T}}\right)\mathbf{D}^{1/2}\mathbf{x}} \tag{17}$$
$$\text{subject to} \quad x_i \in \{1,-1\} \ , \ i = 1,2,\ldots,n \ .$$

Since $\mathbf{U}\mathbf{1} = \mathbf{0}$, $\mathbf{1}$ is an eigenvector of $\mathbf{D}-\mathbf{W}+\mathbf{U}^{\mathbf{T}}\mathbf{\Gamma}\mathbf{U}$ corresponding to eigenvalue 0, and therefore we can use an argument similar to that of Section 2 to show that the solution to the relaxed (unconstrained) version of (17) is the generalized eigenvector corresponding to the smallest nontrivial eigenvalue of:

$$\left(\mathbf{D}-\mathbf{W}+\mathbf{U}^{\mathbf{T}}\mathbf{\Gamma}\mathbf{U}\right)\mathbf{x} = \lambda\mathbf{D}\mathbf{x} \ . \tag{18}$$

As with unconstrained NCuts, the components of the resulting generalized eigenvector can be thresholded in order to assign classification labels.

3.2. Cannot-Link Constraints

Incorporating soft cannot-link constraints into the NCuts framework is not quite as straightforward as incorporating soft must-link constraints. Suppose now that each of the ordered pairs of vertices in the set $\tilde{\mathcal{C}} = \left\{\left(v_{\tilde{i}_\ell}, v_{\tilde{j}_\ell}\right)|\,\ell = 1,\ldots,\tilde{m}\right\}$ represents two vertices that should be grouped in *different* partitions. Another modified cut cost can be formulated, this one penalizing violations of the cannot-link constraints:

$$\text{cut}_{\text{CL}}(A,B) = \text{cut}(A,B) + \tfrac{1}{2}\sum_{\ell=1}^{\tilde{m}} \tilde{\gamma}_\ell \cdot \tilde{\theta}\left(v_{\tilde{i}_\ell}, v_{\tilde{j}_\ell}\right) \quad , \quad (19)$$

where $\tilde{\theta}(v_i, v_j) = 1 - \theta(v_i, v_j) = (x_i + x_j)^2/4$ is the *cannot-link penalty* that equals 1 if v_i and v_j are in the same subgraph and 0 if they are in different subgraphs. The $\tilde{\gamma}_\ell$'s specify the strength of each cannot-link constraint. The modified cut cost can be used to create a modified NCut:

$$\text{NCut}_{\text{CL}}(A,B) = \frac{\text{cut}_{\text{CL}}(A,B)}{\text{assoc}(A,X)} + \frac{\text{cut}_{\text{CL}}(A,B)}{\text{assoc}(B,X)} = \quad (20)$$

$$\text{NCut}(A,B) + \frac{\sum_{\ell=1}^{\tilde{m}} \tilde{\gamma}_\ell \cdot \tilde{\theta}\left(v_{\tilde{i}_\ell}, v_{\tilde{j}_\ell}\right)}{2 \cdot \text{assoc}(A,X)} + \frac{\sum_{\ell=1}^{\tilde{m}} \tilde{\gamma}_\ell \cdot \tilde{\theta}\left(v_{\tilde{i}_\ell}, v_{\tilde{j}_\ell}\right)}{2 \cdot \text{assoc}(B,X)}$$

$$= \frac{\mathbf{x}^{\mathbf{T}}\left(\mathbf{D} - \mathbf{W} + \tilde{\mathbf{U}}^{\mathbf{T}}\tilde{\mathbf{\Gamma}}\tilde{\mathbf{U}}\right)\mathbf{x}}{\mathbf{x}^{\mathbf{T}}\mathbf{D}^{1/2}(\mathbf{I} - \mathbf{q}\mathbf{q}^{\mathbf{T}})\,\mathbf{D}^{1/2}\mathbf{x}} \quad , \quad (21)$$

where $\tilde{\mathbf{U}}$ is the $\tilde{m} \times n$ matrix defined row-wise so that row ℓ contains $1/2$'s in columns $\tilde{i}_\ell$ and $\tilde{j}_\ell$, and 0's in every other column, and where $\tilde{\mathbf{\Gamma}}$ is the diagonal matrix containing the weights on its main diagonal.

Critical points of (21) (assuming the constraints $x_i \in \{1, -1\}, i = 1, 2, \ldots, n$, are relaxed), satisfy the generalized eigenvector problem:

$$\left(\mathbf{D} - \mathbf{W} + \tilde{\mathbf{U}}^{\mathbf{T}}\tilde{\mathbf{\Gamma}}\tilde{\mathbf{U}}\right)\mathbf{x} = \lambda \mathbf{D}^{1/2}(\mathbf{I} - \mathbf{q}\mathbf{q}^{\mathbf{T}})\mathbf{D}^{1/2}\mathbf{x} \quad . \quad (22)$$

However, solving (22) directly is problematic in a way that solving (14) or (18) is not. Consider that the matrix $\mathbf{I} - \mathbf{q}\mathbf{q}^{\mathbf{T}}$ projects vectors onto the subspace orthogonal to $\mathbf{q}$, so the inner product of any scalar multiple of $\mathbf{q}$ with the vector $(\mathbf{I} - \mathbf{q}\mathbf{q}^{\mathbf{T}})\mathbf{D}^{1/2}\mathbf{x}$ must be zero. $\mathbf{D}^{1/2}\mathbf{1}$ is a scalar multiple of $\mathbf{q}$, so $\mathbf{1}^{\mathbf{T}}\mathbf{D}^{1/2}(\mathbf{I} - \mathbf{q}\mathbf{q}^{\mathbf{T}})\mathbf{D}^{1/2}\mathbf{x}$ vanishes for any $\mathbf{x}$. Hence, the right-hand side of (22) is orthogonal to $\mathbf{1}$.

This means that the left-hand side of (22) must also be orthogonal to $\mathbf{1}$ whenever $\lambda \neq 0$. Even though this same analysis is true for (14) or (18), the row sums of $\mathbf{D} - \mathbf{W}$ and of $\mathbf{U}^{\mathbf{T}}\mathbf{\Gamma}\mathbf{U}$ vanish, guaranteeing that the left-hand sides of (14) or (18) are orthogonal to $\mathbf{1}$. However, the row sums of $\tilde{\mathbf{U}}^{\mathbf{T}}\tilde{\mathbf{\Gamma}}\tilde{\mathbf{U}}$ *do not vanish*, and so we must explicitly enforce the constraint $\mathbf{x}^{\mathbf{T}}\tilde{\mathbf{U}}^{\mathbf{T}}\tilde{\mathbf{\Gamma}}\tilde{\mathbf{U}}\mathbf{1} = 0$. The relaxed version of the minimization of (21) is therefore:

$$\min_{\mathbf{x}} \frac{\mathbf{x}^{\mathbf{T}}\left(\mathbf{D} - \mathbf{W} + \tilde{\mathbf{U}}^{\mathbf{T}}\tilde{\mathbf{\Gamma}}\tilde{\mathbf{U}}\right)\mathbf{x}}{\mathbf{x}^{\mathbf{T}}\mathbf{D}^{1/2}(\mathbf{I} - \mathbf{q}\mathbf{q}^{\mathbf{T}})\,\mathbf{D}^{1/2}\mathbf{x}} \quad (23)$$
$$\text{subject to} \quad \mathbf{x}^{\mathbf{T}}\tilde{\mathbf{U}}^{\mathbf{T}}\tilde{\mathbf{\Gamma}}\tilde{\mathbf{U}}\mathbf{1} = 0 \quad .$$

To make (23) easier to solve, we follow the strategy of [6] and judiciously choose a $n \times (n-1)$ matrix $\mathbf{B}$ whose columns form a basis for the subspace orthogonal to the vector $\mathbf{p} = \tilde{\mathbf{U}}^{\mathbf{T}}\tilde{\mathbf{\Gamma}}\tilde{\mathbf{U}}\mathbf{1}$, and we make the substitution $\mathbf{x} = \mathbf{B}\mathbf{h}$, where $\mathbf{h}$ is $(n-1) \times 1$. Minimizing the relaxed version of (23) is then equivalent to solving the unconstrained minimization problem:

$$\min_{\mathbf{h}} \frac{\mathbf{h}^{\mathbf{T}}\mathbf{B}^{\mathbf{T}}\left(\mathbf{D} - \mathbf{W} + \tilde{\mathbf{U}}^{\mathbf{T}}\tilde{\mathbf{\Gamma}}\tilde{\mathbf{U}}\right)\mathbf{B}\mathbf{h}}{\mathbf{h}^{\mathbf{T}}\mathbf{B}^{\mathbf{T}}\mathbf{D}^{1/2}(\mathbf{I} - \mathbf{q}\mathbf{q}^{\mathbf{T}})\,\mathbf{D}^{1/2}\mathbf{B}\mathbf{h}} \quad . \quad (24)$$

A variety of choices can be made for $\mathbf{B}$; for computational purposes, we make the sparse choice $\mathbf{B} = \mathbf{P}_{\hat{i}}\mathbf{M}$, where $\hat{i} = \arg\max \mathbf{p}$, $\mathbf{P}_{\hat{i}}$ is the $n \times n$ permutation matrix that swaps rows 1 and $\hat{i}$, and $\mathbf{M}$ is the $n \times (n-1)$ matrix given by:

$$\mathbf{M} = \begin{bmatrix} \hat{p}_2 & \hat{p}_3 & \cdots & \hat{p}_n \\ & -\hat{p}_1\mathbf{I} & & \end{bmatrix} \quad , \quad (25)$$

with $\hat{\mathbf{p}} = \mathbf{P}_{\hat{i}}\mathbf{p}$.

With this choice of $\mathbf{B}$, we find that critical points of (24) satisfy the generalized eigenvector problem:

$$\mathbf{M}^{\mathbf{T}}\mathbf{P}_{\hat{i}}\left(\mathbf{D} - \mathbf{W} + \tilde{\mathbf{U}}^{\mathbf{T}}\tilde{\mathbf{\Gamma}}\tilde{\mathbf{U}}\right)\mathbf{P}_{\hat{i}}\mathbf{M}\mathbf{h} \quad (26)$$
$$= \lambda \mathbf{M}^{\mathbf{T}}\mathbf{P}_{\hat{i}}\mathbf{D}^{1/2}(\mathbf{I} - \mathbf{q}\mathbf{q}^{\mathbf{T}})\,\mathbf{D}^{1/2}\mathbf{P}_{\hat{i}}\mathbf{M}\mathbf{h} \quad .$$

The solution to (24) is therefore the generalized eigenvector of (26) corresponding to the smallest eigenvalue; this eigenvector can then be premultiplied by $\mathbf{P}_{\hat{i}}\mathbf{M}$ and thresholded to approximately minimize $\text{NCut}_{\text{CL}}(A,B)$.

Note that all matrices in (26) are sparse except for $\mathbf{I} - \mathbf{q}\mathbf{q}^{\mathbf{T}}$; however, if the Lanczos algorithm [9] is used to solve (26), $\mathbf{I} - \mathbf{q}\mathbf{q}^{\mathbf{T}}$ never needs to be formed explicitly. Only the matrix-vector product $(\mathbf{I} - \mathbf{q}\mathbf{q}^{\mathbf{T}})\,\mathbf{f}$ must be computed repeatedly for various vectors $\mathbf{f}$, and this product is equivalent to $\mathbf{f} - (\mathbf{q}^{\mathbf{T}}\mathbf{f})\,\mathbf{q}$.

3.3. Both Types of Constraints in the Same Problem

Soft versions of must-link and cannot-link constraints can easily be incorporated into the same problem. If we first define the semi-supervised cut cost $\text{cut}_{\text{SS}}(A,B)$ to enable both types of constraints; i.e.,

$$\text{cut}_{\text{SS}}(A,B) = \quad (27)$$
$$\text{cut}(A,B) + \tfrac{1}{2}\sum_{\ell=1}^{m} \gamma_\ell \cdot \theta(v_{i_\ell}, v_{j_\ell}) + \tfrac{1}{2}\sum_{\ell=1}^{\tilde{m}} \tilde{\gamma}_\ell \cdot \tilde{\theta}\left(v_{\tilde{i}_\ell}, v_{\tilde{j}_\ell}\right)$$
$$= \text{cut}_{\text{ML}}(A,B) + \text{cut}_{\text{CL}}(A,B) - \text{cut}(A,B) \quad ,$$

then we can define the *Semi-Supervised Normalized Cut* cost by:

$$\text{NCut}_{\text{SS}}(A,B) = \frac{\text{cut}_{\text{SS}}(A,B)}{\text{assoc}(A,X)} + \frac{\text{cut}_{\text{SS}}(A,B)}{\text{assoc}(B,X)} \quad . \quad (28)$$

Combining the analyses of Sections 3.1 and 3.2, we find that minimizing (28) can be relaxed into the equivalent minimization problem:

$$\begin{aligned} &\min_{\mathbf{x}} \quad \frac{\mathbf{x}^{\mathbf{T}}\left(\mathbf{D}-\mathbf{W}+\mathbf{U}^{\mathbf{T}}\boldsymbol{\Gamma}\mathbf{U}+\tilde{\mathbf{U}}^{\mathbf{T}}\tilde{\boldsymbol{\Gamma}}\tilde{\mathbf{U}}\right)\mathbf{x}}{\mathbf{x}^{\mathbf{T}}\mathbf{D}^{1/2}\left(\mathbf{I}-\mathbf{q}\mathbf{q}^{\mathbf{T}}\right)\mathbf{D}^{1/2}\mathbf{x}} \\ &\text{subject to} \quad \mathbf{x}^{\mathbf{T}}\tilde{\mathbf{U}}^{\mathbf{T}}\tilde{\boldsymbol{\Gamma}}\tilde{\mathbf{U}}\mathbf{1}=0\ . \end{aligned} \tag{29}$$

This can be solved by finding the generalized eigenvector corresponding to the smallest eigenvalue of:

$$\begin{aligned} &\mathbf{M}^{\mathbf{T}}\mathbf{P}_{\hat{i}}\left(\mathbf{D}-\mathbf{W}+\mathbf{U}^{\mathbf{T}}\boldsymbol{\Gamma}\mathbf{U}+\tilde{\mathbf{U}}^{\mathbf{T}}\tilde{\boldsymbol{\Gamma}}\tilde{\mathbf{U}}\right)\mathbf{P}_{\hat{i}}\mathbf{M}\mathbf{h} \\ &\quad =\lambda\mathbf{M}^{\mathbf{T}}\mathbf{P}_{\hat{i}}\mathbf{D}^{1/2}\left(\mathbf{I}-\mathbf{q}\mathbf{q}^{\mathbf{T}}\right)\mathbf{D}^{1/2}\mathbf{P}_{\hat{i}}\mathbf{M}\mathbf{h}\ , \end{aligned} \tag{30}$$

and then premultiplying the result by $\mathbf{P}_{\hat{i}}\mathbf{M}$.

As in Section 3.2, all matrices in (30) are sparse except for $\mathbf{I}-\mathbf{q}\mathbf{q}^{\mathbf{T}}$, which never needs to be formed explicitly.

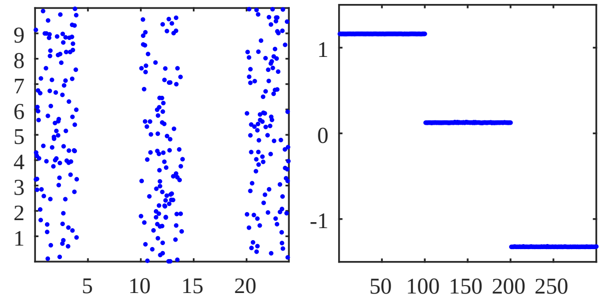
3.4. Constraint Conditioning

Yu and Shi [18] show that when the set of constraints is sparse, better solutions to the hard must-link constrained Normalized Cuts problem are obtained by propagating (or *conditioning*) the constraints to neighboring vertices of the graph by replacing the constraint matrix $\mathbf{U}$ with $\mathbf{U}\mathbf{D}^{-1}\mathbf{W}$. In the context of this paper, sparse constraints can be conditioned in the same way, by replacing $\mathbf{U}$ with $\mathbf{U}\mathbf{D}^{-1}\mathbf{W}$ and $\tilde{\mathbf{U}}$ with $\tilde{\mathbf{U}}\mathbf{D}^{-1}\mathbf{W}$ in the generalized eigenvector problems (18), (26), and (30).

3.5. Choosing Constraint Weights

The values assigned to the constraint weight matrices $\boldsymbol{\Gamma}$ and $\tilde{\boldsymbol{\Gamma}}$ can have unpredictable effects if the number of constraints is changed. A recommended strategy is to define equal weights for each constraint (i.e., set $\boldsymbol{\Gamma}=\gamma\mathbf{I}$ and $\tilde{\boldsymbol{\Gamma}}=\tilde{\gamma}\mathbf{I}$), and then rescale $\mathbf{U}$ and $\tilde{\mathbf{U}}$ so that $\operatorname{tr}\left(\mathbf{U}^{\mathbf{T}}\mathbf{U}\right)=\operatorname{tr}\left(\tilde{\mathbf{U}}^{\mathbf{T}}\tilde{\mathbf{U}}\right)=\operatorname{tr}(\mathbf{D})$. This rescaling normalizes the problem so that the choice $\gamma=\tilde{\gamma}=1$ always indicates equal influence across all terms in the cost function. For all of the images in section 5, after rescaling $\mathbf{U}$ and $\tilde{\mathbf{U}}$, SSNCuts is carried out with $\gamma=\tilde{\gamma}=100$.

For greater flexibility in interactive segmentation systems, an alternative is to choose constraint weights to equally influence must-link constraints in the foreground with those in the background. By reordering and partitioning $\mathbf{U}$ so that the top rows ($\mathbf{U}_f$) correspond to foreground ML constraints and the bottom rows ($\mathbf{U}_b$) to background ML constraints, we can separately rescale $\mathbf{U}_f$ and $\mathbf{U}_b$ so that $\operatorname{tr}\left(\mathbf{U}_f^{\mathbf{T}}\mathbf{U}_f\right)=\operatorname{tr}\left(\mathbf{U}_b^{\mathbf{T}}\mathbf{U}_b\right)=\operatorname{tr}\left(\tilde{\mathbf{U}}^{\mathbf{T}}\tilde{\mathbf{U}}\right)=\operatorname{tr}(\mathbf{D})$. This allows for the choice of separate weights γ_f and γ_b for the two groups of must-link constraints.



(a) Input Points

(b) Normalized Cut

Figure 1: Synthetic example: (a) data set comprising three classes, (b) partitioning based on minimizing NCut.

4. Synthetic Example

To illustrate the incorporation of soft must-link and cannot-link constraints into NCuts, we compare our proposed algorithm with Yu-Shi NCuts [18] and Eriksson et al. NCuts [6] using the synthetic example introduced in Maji et al. [10]. Figure 1a shows data randomly generated in three sets, S_1, S_2, and S_3, with 100 data points in each set. After constructing a graph with edge weights between points $\mathbf{p}_i$ and $\mathbf{p}_j$ given by $w_{i,j}=\exp\left(-\left\|\mathbf{p}_i-\mathbf{p}_j\right\|^2/2\sigma^2\right)$ with $\sigma=2$, we performed NCuts, which correctly partitions the points into three separate clusters as shown in Fig. 1b.

Next, as in [10], we attempt to group S_1 and S_3 together by adding various must-link and cannot-link constraints as shown in Fig. 2. In the left three columns of Fig. 2, must-link constraints are created between each pair of points circled in red; in the right three columns, must-link constraints are created between all pairs of points circled in the same color, and cannot-link constraints are created between points circled in red and those circled in green. In the third and sixth columns, erroneous constraints are added by "accidentally" selecting two points in S_2 to be grouped with those in S_1 and S_3.

In the left columns, where only must-link constraints are provided, we see the results of Yu-Shi NCuts [18] and SS-NCuts. In the right columns, where two groups of constraints are provided, we separately pair subsets of red and green circled points to form must-link constraints, and we form cannot-link constraints are between each red/green point pair. In these columns, we see the results of Eriksson et al. NCuts [6] and SSNCuts (Yu-Shi NCuts is not applicable in the presence of cannot-link constraints).

Note that in situations where there are no erroneous manually-labeled points, Yu-Shi NCuts, Eriksson et al. NCuts, and SSNCuts all nicely separate the data into the desired clusters. However, in the presence of even a single erroneous manually-labeled point, the hard-constrained versions of NCuts fail; whereas, SSNCuts is able to suc-

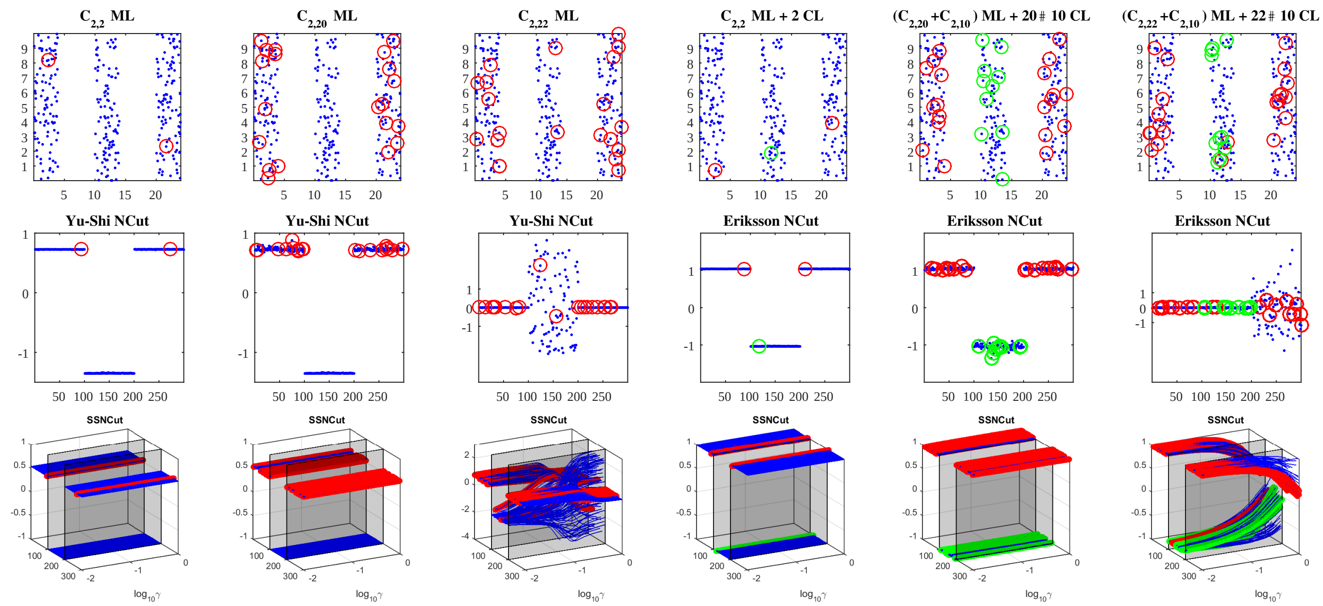


Figure 2: Comparison of our SSNCut algorithms to Yu-Shi NCuts [18] and Eriksson et al. NCuts [6] for various constraints shown by red and green circled points. In all situations where Yu-Shi and Eriksson NCuts can be used, SSNCuts successfully separates S_2 from S_1 and S_3 for a range of values of γ. In addition, in situations where Yu-Shi and Eriksson NCuts fail due to the presence of erroneous manual labels, SSNCuts succeeds as γ and $ga\tilde{m}ma$ are decreased.

cessfully separate the data as the constraint weights are decreased γ. This suggests that even if hard constraints become impossible to satisfy, soft constraints can be effective.

5. Qualitative Segmentation Results

To illustrate how SSNCuts performs on real imagery in comparison and contrast with hard-constrained versions of NCuts, we follow the evaluation strategy of Maji et al. [10] and present qualitative results on images from the PASCAL VOC dataset [7]. As done in [10], we will refrain from performing a quantitative validation using segmentation benchmarks or wrapping the various NCuts algorithms inside foreground extraction systems like GrabCuts [12].

Figure 3 shows a variety of images taken from the PASCAL VOC database. For each image, we use a paintbrush tool to manually identify regions of the image that should belong to the foreground object (orange) and background (blue). To make the subsequent segmentation algorithms more computationally efficient, we first perform an oversegmentation of the image into superpixels using Simple Linear Iterative Clustering (SLIC) [1] as implemented in the VLFeat toolbox [15], and we construct a graph having each superpixel as a vertex.

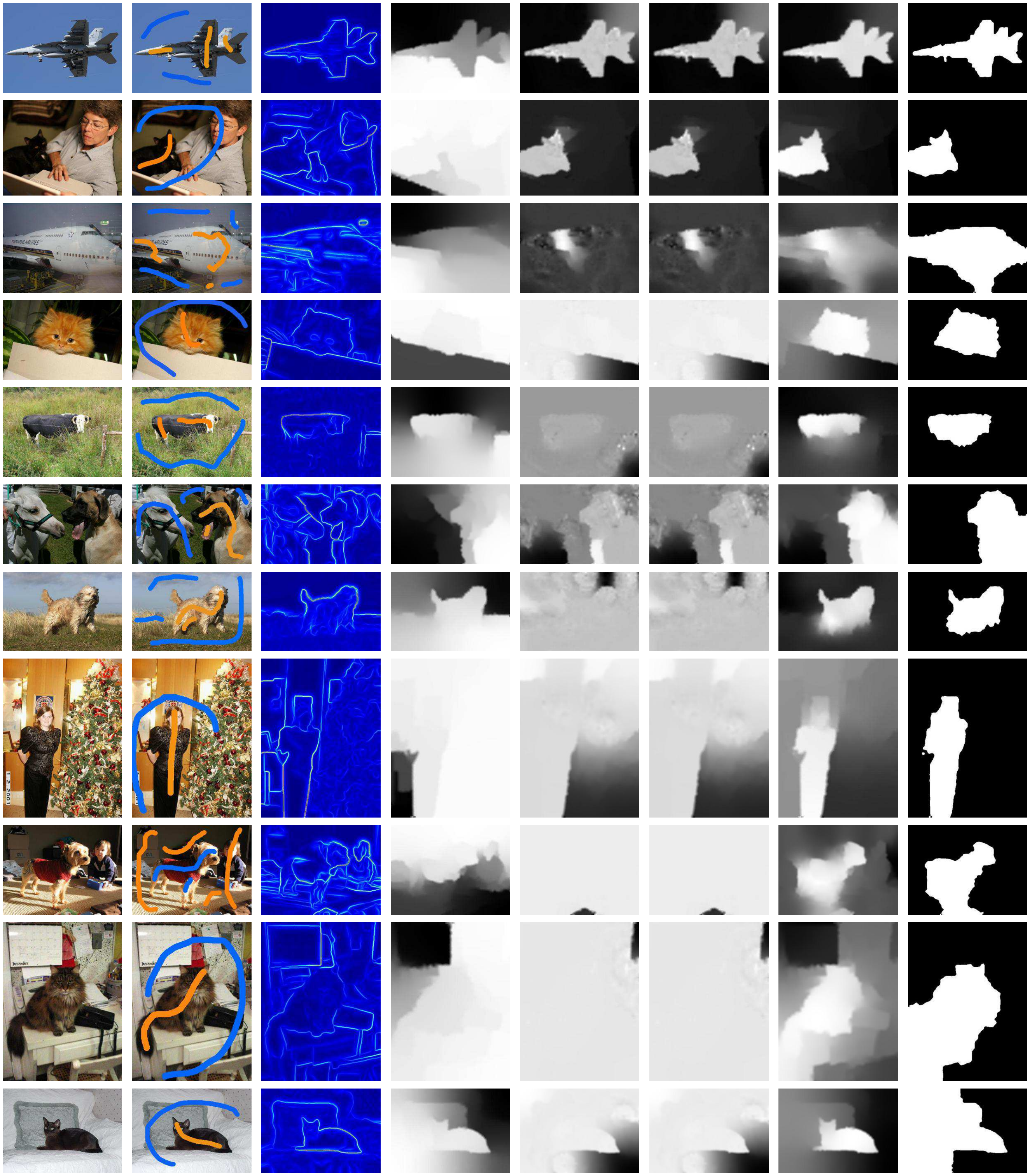
To define edge weights for the graph, we compute the *globalized probability of boundary* (gPb) [2] at each pixel, find the maximum gPb across eight orientation angles, and define the weight between superpixel i and j as:

$$W_{i,j} = \exp(-\max\{\text{gPb}(\mathbf{p}_{i,j})\}/\rho) \tag{31}$$

if $\|\mathbf{p}_{i,j}\| \leq r$ and $W_{i,j} = 0$ otherwise, where $\mathbf{p}_{i,j}$ is the line segment connecting the centroids of superpixels i and j, and ρ is a constant. We set r to be 0.1 times the maximum of the image width and height, and $\rho = 0.1$. Our choice of ρ is consistent with [2], but our choice of r is larger because of our use of superpixels to form the graph.

For each PASCAL VOC image, the generalized eigenvectors corresponding to the solutions of unconstrained, Yu-Shi, Eriksson et al., and Semi-Supervised NCuts are shown in columns 4-7. Column 8 shows a segmentation of the foreground region based on k-means clustering of the SSNCuts eigenvector. In the images in the first two rows, hard-constrained versions of NCuts are successful, and so SSNCuts shows no improvement. In the remaining rows, hard-constrained NCuts fail (in the same manner, in fact), but SSNCuts yields eigenvectors that appear to highlight the location of the foreground region. The final three rows show images for which SSNCuts yields qualitatively good eigenvectors, but for which k-means clustering yields too low of a threshold for segmentation. These results suggest that SSNCuts would likely be successful in a GrabCuts-like iterative foreground extraction system [12].

Figure 4 illustrates the influence of the weights in the soft constraints. Using one of the dog images with user-provided paintbrush strokes shown in Fig. 3, we see the



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 3: PASCAL VOC Images: (a) original, (b) original with user-provided constraints overlayed, (c) maximum gPb, (d) eigenvector from unconstrained NCuts [14], (e) eigenvector from Yu-Shi NCuts [18] with hard ML constraints, (f) eigenvector from Eriksson et al. NCuts [6] with hard ML and CL constraints, (g) eigenvector from SSNCuts, and (h) segmentation based on k-means clustering on (g).

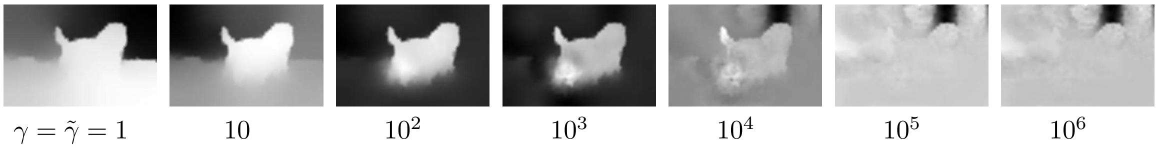


Figure 4: Result of increasing constraint weights in Semi-Supervised NCuts for the dog image from row seven of Figure 3.

influence of simultaneously increasing γ and $\tilde{\gamma}$. As the constraint weights are initially increased, the SSNCuts solution appears to successfully identify the dog in the foreground; however, as the constraint weights grow even further, SSNCuts eventually yields solutions similar to hard-constrained NCuts.

6. Conclusion

In this paper, we have extended the NCuts algorithm for image segmentation and clustering to enable soft must-link and cannot-link constraints. The constraints can be provided by an expert user, and their soft nature allows their relative influence to be varied. Through various synthetic and real-world examples, we illustrate results that are more robust than those achieved by hard constraint enforcement.

Appendix

Prototype implementations of algorithms in this paper are available at MATLAB Central (http://www.mathworks.com/matlabcentral/) under File ID #52735.

Acknowledgements

Thanks to Paul Wenger and the anonymous reviewers for helpful comments. Selene Chew was supported by RIT's Honors Summer Undergraduate Research Program.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, May 2012. 6

[2] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 33(5):898–916, May 2011. 6

[3] A. Blake, C. Rother, M. Brown, P. Perez, and P. Torr. Interactive image segmentation using an adaptive GMMRF model. In *Proc. European Conference on Computer Vision*, ECCV, pages 428–441, 2004. 1

[4] Y. Y. Boykov and M. P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In *Proc. International Conference on Computer Vision*, ICCV, pages 105–112, 2001. 1

[5] S. E. Chew and N. D. Cahill. Normalized cuts with soft must-link constraints for image segmentation and clustering. In *Proc. IEEE Western New York Image and Signal Processing Workshop*, WNYISPW, pages 6–10, November 2014. 2

[6] A. Eriksson, C. Olsson, and F. Kahl. Normalized cuts revisited: A reformulation for segmentation with linear grouping constraints. *Journal of Mathematical Imaging and Vision*, 39(1):45–61, 2011. 1, 2, 3, 4, 5, 6, 7

[7] M. Everingham, M. E. Ali Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL visual object classes challenge – a retrospective. *International Journal of Computer Vision*, 2014. 1, 6

[8] L. Grady. Random walks for image segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28(11):1768–1783, Nov 2006. 1

[9] C. Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Journal of Research of the National Bureau of Standards*, 45(4):255–282, Oct 1950. 4

[10] S. Maji, N. K. Vishnoi, and J. Malik. Biased normalized cuts. In *Proc. Computer Vision and Pattern Recognition*, CVPR, pages 2057–2064, 2011. 1, 5, 6

[11] B. N. Parlett. *The symmetric eigenvalue problem*. Society for Industrial and Applied Mathematics, 1998. 3

[12] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23:309–314, 2004. 1, 6

[13] J. Shi and J. Malik. Normalized cuts and image segmentation. In *Proc. Computer Vision and Pattern Recognition*, CVPR, pages 731–737, Jun 1997. 1

[14] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(8):888–905, Aug 2000. 1, 2, 7

[15] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/, 2008. 6

[16] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl. Constrained k-means clustering with background knowledge. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 577–584, 2001. 1, 3

[17] Z. Wu and R. Leahy. An optimal graph theoretic approach to data clustering: theory and its application to image segmentation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15(11):1101–1113, Nov 1993. 2

[18] S. X. Yu and J. Shi. Segmentation given partial grouping constraints. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(2):173–183, Feb 2004. 1, 5, 6, 7