# Clustering by weighted cuts in directed graphs

Marina Meilă
Department of Statistics
University of Washington
mmp@stat.washington.edu

William Pentney
Dept. of Computer Science & Engineering
University of Washington
bill@cs.washington.edu

**Abstract**

In this paper we formulate spectral clustering in directed graphs as an optimization problem, the objective being a weighted cut in the directed graph. This objective extends several popular criteria like the normalized cut and the averaged cut to asymmetric affinity data. We show that this problem can be relaxed to a Rayleigh quotient problem for a symmetric matrix obtained from the original affinities and therefore a large body of the results and algorithms developed for spectral clustering of symmetric data immediately extends to asymmetric cuts.

## 1 Introduction

Spectral methods for clustering pairwise data use eigenvalues/vectors of a matrix (usually a transformation of an *affinity matrix* $A = [A_{ij}]$) in order to assign data points to clusters. Insofar, most published spectral algorithms operate on symmetric matrices. However, there are important cases when data points have pairwise relationships that are not symmetric. Such data will be called *link data*. Hyperlinked domains like the web are a foremost example of link data. If the affinity between pages $i$ and $j$ is conveyed by the presence of a link, then the resulting $A_{ij}$ is asymmetric. Even when more complex affinity functions are constructed, by e.g taking into account the anchor text, the similarity of the anchor text to the linked page, etc., the resulting affinity is usually asymmetric, reflecting the directed structure of the web graph. Similar to hyperlinked domains are citation networks, where a link from document $i$ to $j$ exists if $i$ contains a reference to $j$. Citation networks are obviously asymmetric. Data sets describing social networks, economic transactions, internet communications, and alignment scores between biological sequences are also often asymmetric [12].

The commonly used approach for spectral clustering link data is to obtain a symmetric matrix $\tilde{A}$ from the original $A$ and then to apply spectral clustering techniques to $\tilde{A}$. Typical transformations used in the literature include $\tilde{A} = A + A^T$, $\tilde{A} = A^T A$ , and $\tilde{A} = \begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix}$. [1] These methods have the disadvantage that in many cases a clustering that is present in the original asymmetric $A$ becomes partially or completely invisible after symmetrization, as demonstrated in [8] and figure 1.

Thus, we propose to attack clustering of link data directly from the original asymmetric affinity matrix $A$. Some previous approaches exist. The work of [8] uses the random walks perspective to define the clustering algorithm. Zhou et al. [21] construct a symmetric matrix from $A$ as the Laplacian of an operator representing a weighted sum of co-reference scores, for which an intuitive motivation is given. In a later paper, [19] the authors introduce another method based on random walks, which is a generalization to directed graphs of the Shi and Malik normalized cuts method [10].

Here we formulate clustering as an optimization problem, the objective being the minimization of a weighted cut in the directed graph. We show that this problem is equivalent to a Rayleigh quotient problem [13] for a symmetric matrix obtained from $A$ and therefore a large body of the results and algorithms developed for spectral clustering of symmetric affinities can be used for it.

## 2 The Generalized Weighted Cut

In the following, $A$ will be a typically asymmetric $n \times n$ matrix with real, non-negative elements. The assumption is that $A_{ij}$ represents the affinity of point $i$ for point $j$ with $i, j \in \{1, 2, \dots n\}$. Therefore, in a graph representation, $A_{ij}$ would represent the weight on the directed edge $\vec{ij}$. The indices $k, k'$ will be used to index subsets of $[n] = \{1, \dots, n\}$ in a partition; we will call these subsets *clusters*. The indices $i, j$ will index elements of $[n]$.

We denote by $D_i = \sum_{j \in [n]} A_{ij}$ the *out-degree* of node $i \in [n]$ and by $D$ the diagonal matrix with $D_i$, $i \in [n]$ on the diagonal. The out-degrees $D_i$ can be

[1] The last two amount to the same spectral problem.

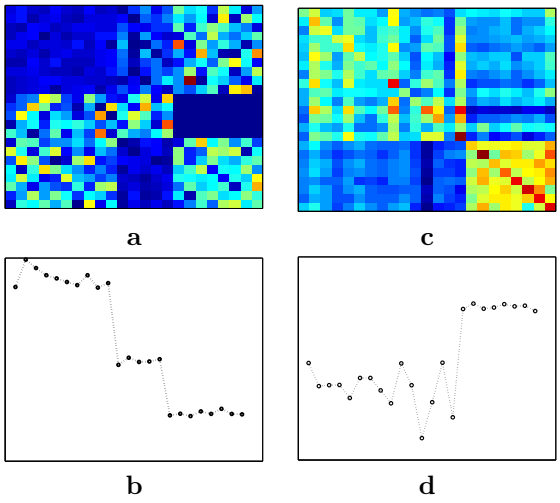



Figure 1: The loss of information by symmetrization: An asymmetric matrix $A$ **(a)**. The first eigenvector of the matrix $H(B)$ obtained from $A$ as described in section 3 is plotted in **(b)** vs the data points, demonstrating that there are 3 clusters in the data. The transition matrix $\tilde{P}$ obtained from the symmetrized affinity $\tilde{A} = A^T A$ **(c)**. The clustering in $A$ is partly effaced in $\tilde{P}$. The eigenvectors of $\tilde{P}$ are not piecewise constant **(d)**. (Here only the second eigenvector is plotted, but all of them have been examined).

thought of as "weights" associated to the graph nodes; such an interpretation is central to the symmetric case [4, 18]. Here, in addition to the weights $D_i$, we assume that the user may provide two other sets of positive weights for the nodes: the *volume weights* denoted by $T_i$ and the *row weights* $T_i'$ (the associated diagonal matrices being denoted $T$ and $T'$ respectively). The meaning of these names will become apparent shortly.

In a directed graph one can have a node $i$ with outdegree $D_i = 0$ but with incoming links (indegree $> 0$). Therefore we will not assume $D_i > 0$, but we can assume without loss of generality (w.l.o.g.) that no node has both indegree and outdegree 0, as it would be completely isolated and could be removed from further consideration for the purpose of grouping. We will also assume $T_i, T_i' > 0$ as these are user-defined weights. W.l.o.g we can assume that $\sum_i D_i = 1$.

Let $\mathcal{C} = \{C_1, \ldots, C_K\}$ be a clustering of $[n]$. Then, we define the *cluster degrees* $D_k$ and the *cluster volume weights* $T_k$, $k = 1, \ldots K$ by

$$D_k = \sum_{i \in C_k} D_i, \quad T_k = \sum_{i \in C_k} T_i \tag{2.1}$$

We can now introduce the *generalized weighted cut* ($WCut$) associated with $\mathcal{C}$ by

$$WCut(\mathcal{C}) = \sum_{k=1}^{K} \sum_{k' \neq k} \frac{Cut(C_k, C_{k'})}{T_k} \tag{2.2}$$

with

$$Cut(C_k, C_{k'}) = \sum_{i \in k} \sum_{j \in k'} T_i' A_{ij} \tag{2.3}$$

The intuition behind this definition is similar to the motivation for the normalized cut [11, 5] in the symmetric $A$ case: we want to find a cut of low weight in the graph, but one which creates clusters of balanced sizes. Here the volume weights $T_k$ stand for the "sizes" of the clusters, and the row weights $T_i'$ play the role of a row normalization for the matrix $A$. They were introduced to make the new criterion more flexible, as seen below.

**2.1 Examples of weighted cuts** We now show that the new, generalized criterion allows us to study a variety of existing and new criteria for clustering in graphs in a unified manner.

We shall start with the better understood setting of a symmetric affinity matrix $A$, which is a special case for us. In this case, if one takes

$$P = D^{-1} A \tag{2.4}$$

to be the transition matrix of a (reversible) Markov chain then the $D_i$'s are equal to the stationary probabilities $\pi_i$. The $WCut$ will recover the multiway version of the normalized cut $MNCut$ of [7, 18]

$$MNCut(\mathcal{C}) = \sum_{k=1}^{K} \sum_{k' \neq k} \frac{Cut(C_k, C_{k'})}{D_k}$$

with

$$Cut(C_k, C_{k'}) = \sum_{i \in k} \sum_{j \in k'} A_{ij}$$

by taking $A \leftarrow P$, $T' = T \leftarrow D$. This example motivates the names for $T$ and $T'$: $T_i'$ is the weighting of row $i$ in $Cut(C_k, C_{k'})$, while $T_i$ is the node "volume" used to balance the cluster sizes. This point of view also highlights the double role played by the $D_i$ values in the undirected case. They represent both the nodes' stationary probabilities and the row normalization constants that turn a symmetric affinity matrix $A$ into a transition matrix $P$, which will, for symmetric $A$, always represent the transition matrix for a reversible Markov chain. The duality breaks for irreversible chains (aka asymmetric $A$); there the stationary distribution

$\pi = [\pi_i]_i$, represented by the left eigenvector of $P$, is usually different from the (out)degrees vector $[D_i]_i$. [2]

We will return now to the general case of an asymmetric affinity matrix $A$. Formally, one can define the *MNCut* in the same way as above. However, since the stationary distribution is not given by the node outdegrees $D_i$, it may be more sensible to normalize the rows by the $D_i$'s in order to obtain transition probabilities, while weighting the nodes according to their stationary distribution $\pi_i \neq D_i$. This can be done by taking

$$T_i = \pi_i,\ T'_i = \pi_i/D_i,\ A \leftarrow A. \tag{2.5}$$

One can show that the resulting weighted cut sums the conditional probabilities of leaving each cluster when the chain is at the stationary distribution $\pi$, which coincides with one of the interpretations of the normalized cut in undirected graphs [7]. (The proof is left as an exercise to the reader).

If $\pi$ is replaced by any other distribution $\pi'$ (by setting $T_i = \pi'_i,\ T'_i = \pi'_i/D_i$, the corresponding *WCut* will represent the sum of evasion probabilities under $\pi'$. For instance, by taking $\pi'$ to be the uniform distribution, we get the evasion probabilities for a chain that is restarted from the uniform at each step. For

$$T_i = T'_i = 1/n \tag{2.6}$$

we obtain the so called *average cut*, where the cut sizes are normalized by the number of elements in each cluster.

The weighted cuts approach gives us more flexibility. While using a (stationary) distribution to weigh the graph nodes may be meaningful sometimes, we will show that this is not always so. User-selected weights not directly interpretable as distributions can be meaningful too, as it is the case in the average cut case, when the clusters sizes equal the number of nodes in a cluster. We argue that for some problems, the nodes' outdegrees themselves can be used as weights. This would be a "direct authority" model, where a node's importance is equal to the number of its outgoing links. Many have noticed that in the case of social networks (like citations or the web) it is the incoming links that generate authority; therefore, from now on, unless otherwise specified, we will assume that any matrix representing citations, web links or the such will have been transposed.

[2]The discussion above omits the following fact: A *reversible* Markov chain, if it is *reducible*, i.e. if $A$ is block diagonal, will have multiple stationary distributions. However, the distribution defined by $\pi_i = D_i$ will always be one of them. So, simplifying matters only slightly, we will refer to it here as *the stationary distribution*.

Instead of an affinity matrix, we can have a transition matrix $P^*$ of a user-defined random walk on the graph, with stationary distribution $\pi^*$. One such random walk, called the *teleporting random walk* jumps according to $P = D^{-1}A$ a fraction $\alpha$ of the time, and jumps with uniform probability to any node other than the current node a fraction $1-\alpha$ of time. This amount to replacing the original $P$ with

$$P_\alpha = \alpha P + (1-\alpha)P_u \tag{2.7}$$

where $P_u$ represents the transition matrix of the "uniform"random walk, $(P_u)_{ij} = (1-\delta_{ij})/(n-1)$. This technique is similar to that used by the PageRank algorithm adopted by popular search engines such as Google [1]. Variants of such random walks like backwards random walks, or two-step backward and forward random walks have been used by [3, 19]. The weighted cut becomes the criterion of [19] for

$$T = T' = \Pi^*,\ A \leftarrow P^* \tag{2.8}$$

where $\Pi^*$ denotes the matrix having $\pi^*_i$, the stationary distribution of the teleporting random walk, on its diagonal. This is proved in section 4.

A few other remarks need to be made about asymmetric matrices $A$. First, it is possible for some node to have $D_i = 0$ (no outgoing links). One can avoid divisions by zero in 2.4 by e.g setting $A_{ii} = D_i = 1$. This way the output-less node is turned into a *sink*.

Second, for some random walks with all $D_i > 0$, in the stationary distribution we can have some $\pi_i = 0$. This case is illustrated in figure 3 in which all but the last node, a sink, have $\pi_i = 0$. This situation can be redressed by switching to a teleporting random walk (which also solves the first problem), but we will see that this solution is not the only one, and is not always the best one. Others [3] have reported great sensitivity of the eigenvectors of interest to the value $\alpha$ appearing in (2.7) and our experiments have confirmed this.

In summary, the $T$ weights may represent a user defined weighting of the nodes, that can correspond to the stationary distribution of a random walk on the graph, to a uniform distribution, to the indegrees or outdegrees, or to a monothone function thereof. The matrix $A$ is either the original affinity matrix, or a preprocessed form, like the transition matrix of a user defined random walk over the graph. The row weights $T'$ are a row normalization of $A$.

Table 1 summarizes these examples. Finally we shall note that the authors are aware the above could be rewritten without using $T'$ at all; we preferred this slightly redundant formulation because it helps underscore the bifurcation of the symmetric normalized

Table 1: A summary of the various instantiations of the *WCut* described in this paper. $\Pi$ denotes the diagonal matrix with a (stationary) distribution over $[n]$ the diagonal.

| | |
|---|---|
| $T = D$, $T' = D$, $A \leftarrow D^{-1}A$ | Normalized Cut for symmetric matrix, [11, 7] |
| $T = \Pi$, $T' = \Pi D^{-1}$, $A \leftarrow A$ | evasion probability in 1 step under $\Pi$, [19] |
| $T = \Pi^*$, $T' = \Pi^*$, $A \leftarrow P^*$ | cf. above for modified r.w., e.g. [19] with teleporting |
| $T = \mathbf{1}$, $T' = D^{-1}$, $A \leftarrow A$ | evasion prob. under uniform distribution |
| $T = T' = \mathbf{1}$, $A \leftarrow A$ | average cut |
| $T = D$, $T' = \mathbf{1}$, $A \leftarrow A$ | *WNCut* (introduced here) |

cut normalizations in the more general situation of asymmetric matrices.

## 3 A spectral bound for *WCut*

Given a weighted directed graph described by matrix $A$ and additional volume and row weights $T, T'$ we want to find a clustering $\mathcal{C}^*$ such that

$$WCut(\mathcal{C}^*) = \min_{\mathcal{C}} WCut(\mathcal{C}) \tag{3.9}$$

We will show that this discrete problem can be relaxed to an eigenvector/value problem for a symmetric matrix. Then, spectral clustering algorithms designed for symmetric matrices like [7] can be used to find the optimal clustering under the same conditions as in the symmetric case.

In the following we assume w.l.o.g $T'_i$ is a constant and drop it for the simplicity of the exposition. If this is not the case, one can simply replace $A$ by $T'A$ and $D$ by $T'D$. We represent a clustering $\mathcal{C} = \{C_1, \ldots, C_K\}$ by an $n \times K$ matrix $X$, where $x_{:k}$, the $k$-th column of $X$ is the indicator vector of cluster $C_k$. The weighted cut $WCut(\mathcal{C})$ can be expressed successively as

$$WCut(\mathcal{C}) = \tag{3.10}$$

$$= \sum_{k=1}^{K} 1/T_k \sum_{i \in C_k} (D_i - \sum_{j \in C_k} A_{ij}) \tag{3.11}$$

$$= \sum_{k=1}^{K} \frac{x_{:k}^T (D - A) x_{:k}}{x_{:k}^T T x_{:k}} \tag{3.12}$$

$$= \sum_{k=1}^{K} \tilde{y}_{:k}^T B \tilde{y}_{:k} \tag{3.13}$$

$$\text{with } B = T^{-1/2}(D - A)T^{-1/2} \tag{3.14}$$

$$\text{and } \tilde{y}_{:k} = T^{1/2} x_{:k} / \sqrt{T_k} \tag{3.15}$$

For $A$ symmetric, the matrix $D - A$ represents the *unnormalized Laplacian* of $A$. In general, $D - A$ is an asymmetric matrix with non-negative diagonal satisfying $(D - A)\mathbf{1} = 0$ (where $\mathbf{1}$ represents the column vector of all ones). The matrix $Y = [\tilde{y}_{:k}]_{k=1}^K$ has orthonormal columns.

For any matrix $B$, the *Hermitian part*[3] of $B$ is defined as $H(B) = \frac{1}{2}(B + B^T)$. It is easy to see that $H(B)$ is always a symmetric matrix, and it has non-negative elements whenever $B$ has non-negative elements. We say that a vector $v$ is *piecewise constant* (p.c.) w.r.t a clustering $\mathcal{C}$ iff $v_i = v_j$ whenever points $i, j$ are in the same cluster. We are now ready to prove the following.

PROPOSITION 3.1. **(The asymmetric multicut lemma)** *For any clustering $\mathcal{C}$ of $[n]$, $WCut(\mathcal{C}) \geq \sum_{k=1}^{K} \lambda_k(H(B))$ where the eigenvalues are counted in increasing order (the smallest eigenvalue first). Moreover, let $Y$ be the $n \times K$ matrix formed with the eigenvectors of $H(B)$ corresponding to $\lambda_1(H(B)), \ldots \lambda_K(H(B))$. Then, the bound is attained iff $T^{-1/2}Y$ has piecewise constant columns.*

*Proof.* Let $X$ be the indicator matrix of $\mathcal{C}$, and $\tilde{y}_{:k}$ be defined as in (3.15). We successively relax the problem

$$MNCut(X) = \sum_{k=1}^{K} \tilde{y}_{:k}^T B \tilde{y}_{:k} \tag{3.16}$$

$$\geq \min_{z_k \in \mathbf{R}^n \text{ orthon}} \sum_{k=1}^{K} z_k^T B z_k \tag{3.17}$$

$$\geq \min_{z_k \in \mathbf{C}^n \text{ orthon}} \mathrm{Re} \sum_{k=1}^{K} z_k^* B z_k \tag{3.18}$$

$$= \min_{z_k \in \mathbf{C}^n \text{orthon}} \sum_{k=1}^{K} z_k^* H(B) z_k \tag{3.19}$$

$$= \sum_{k=1}^{K} \lambda_k(H(B)) \tag{3.20}$$

[3] In the case of complex $B$, which we won't need to consider here, $B^T$ is replaced by $B^*$ the transpose-conjugate of $B$.

The vectors $z_k$ that achieve the minimum in (3.20) are the first $K$ eigenvectors of $H(B)$. Because they are real, the second inequality above is in fact an equality.

The second part of the lemma is proved as follows. Let $X$ be a clustering for which the bound is attained and $Y \in \mathbf{R}^{n\times K}$ the matrix formed by the first eigenvectors of $H(B)$; denote $\hat{T} = \text{diag}\{T_k,\ k = 1,\dots K\}$. Hence, the orthonormal columns of $T^{1/2}X\hat{T}^{-1/2}$ lie in the subspace spanned by $Y$. In other words $T^{1/2}X\hat{T}^{-1/2} = YU$ with $U$ a $K \times K$ unitary matrix. Therefore, $X(\hat{T}^{-1/2}U^{-1}) = T^{-1/2}Y$. $X$ has p.c. columns and multiplication to the right preserves this property, which proves one direction of the iff. Now assume that $Z = T^{-1/2}Y$ has p.c columns w.r.t some clustering given by $X$. Then $Z = X\hat{Z}$ with $\hat{Z}$ the $K \times K$ matrix containing the distinct rows of $Z$. We need to show that the columns of $T^{1/2}X\hat{T}^{-1/2}$ are in the space spanned by $Y$. As $Y = T^{1/2}X\hat{Z}^{-1}$, it is sufficient to show that $\hat{Z}^{-1}\hat{T}^{-1/2}$ is a unitary matrix, or equivalently, that $(\hat{Z}^{-1}\hat{T}^{-1/2})^{-1} = \hat{T}^{1/2}\hat{Z}$ is unitary. $(\hat{T}^{1/2}\hat{Z})^T\hat{T}^{1/2}\hat{Z} = \hat{Z}^T\hat{T}\hat{Z} = \hat{Z}^T(X^TTX)\hat{Z} = Z^TS, = Y^TM^{-1/2}AT^{-1/2}Y = Y^TY = I$.

## 4 Minimizing *WCut* by spectral clustering

The proposition immediately suggests a spectral algorithm for minimizing the *WCut*. Essentially, the algorithm is a simple modification of the spectral clustering algorithm of [7] used for symmetric $A$. The only major difference is the first step, where $H(B)$ plays the role of the normalized Laplacian $I - D^{-1/2}AD^{-1/2}$. A minor difference is that usually in symmetric clustering algorithms the subtraction from $I$ is omitted and thus the largest eigenvectors are those of interest. Of course, many other variants of BestWCut can be obtained from other existing clustering algorithms that minimize normalized cuts.

Algorithm 4.1. (Algorithm BestWCut)

Input Affinity matrix $A$, weights $T$, $T'$, number of clusters $K$

1. $A \leftarrow T'A$
2. $D_i \leftarrow \sum_{j=1}^{n} A_{ij}$, $D = \text{diag}\{D_i\}_i$
3. $H(B) \leftarrow \frac{1}{2}T^{-1/2}(2D - A - A^T)T^{-1/2}$
4. Compute $Y$ the $n \times K$ matrix with orthonormal columns containing the $K$ smallest eigenvectors of $H(B)$
5. Cluster the rows of $X = T^{-1/2}Y$ as points in $\mathbf{R}^K$. (**Variant** Normalize the rows of $Y$ to have length 1, then cluster them as points in $\mathbf{R}^K$.)

The proof that this algorithm will minimize the *WCut* if the columns of $X$ are almost p.c. is a direct consequence of the analog proofs in the symmetric case (see e.g [4, 18]).

To summarize the above results: minimizing the generalized weighted cut in directed graphs is in general NP-hard, because it includes the symmetric case which is proved to be hard. However, in special cases, when the integrality gap between the discrete problem (3.9) and the relaxed problem (3.20) is sufficiently small, the optimum can be obtained by solving an eigenproblem for a symmetric matrix. Note that there is only one relaxation between the original asymmetric discrete optimization problem and the final symmetric eigenproblem, which corresponds to the integrality constraint on the entries of $X$. Thus, remarkably, for a large family of weighted cuts represented by *WCut*, the asymmetric version of problem (3.9) and the symmetric version are qualitatively the same and they both involve the eigenvectors of a symmetric matrix.

## 5 Relationship with other work

For symmetric $A$, which corresponds to a reversible Markov chain, the data are mapped by $X$ the eigenvectors of the transition matrix $P = D^{-1}A$. Therefore, when $X$ has p.c columns, or near this case, the result can be interpreted in two equivalent ways. On one hand, the Markov chain can be aggregated according to clustering $\mathcal{C}^*$ without loss of information. The points in each cluster $C_k$ have the same transition probabilities to other clusters [7]. On the other, the same $\mathcal{C}^*$ can be shown to optimize the *MNCut* of $A$ [4]. If $A$ is asymmetric, the problem bifurcates: what was one clustering criterion with two interpretations becomes two distinct criteria. One can either (1) cluster by the possibly complex eigenvectors of $P$ (an approach taken by [8]) in which case the goal is to find clusters of points that behave in the same way, or (2) cluster by the eigenvectors of $H(B)$ in which case the objective is the minimization of a weighted cut.

For what asymmetric matrices are cases (1) and (2) the same? As the formulation (1) does not take $T, T'$ as parameters, we assume we are in the case of the (asymmetric) normalized cut.

Proposition 5.1. *Let $A$ be an asymmetric real matrix, $D_i > 0$ for all $i \in [n]$, $T' = I$ and $T = D$. Let $Y \in \mathbf{R}^{n\times K}$ contain the first (orthonormal) eigenvectors of $H(B)$. If there is a unitary completion of $Y$ to $U = [Y\ Y_2]$ so that*

$$U^TBU = \begin{bmatrix} B_1 & 0 \\ 0 & B_2 \end{bmatrix} \tag{5.21}$$

*with $B_1 \in \mathbf{R}^{K\times K}, B_2 \in \mathbf{R}^{(n-K)\times(n-K)}$ and $\max|\lambda(B_1)| < \min|\lambda(B_2)|$, then the algorithm of [8]*

*and the* BestWCut *algorithm produce the same spectral mapping.*

*Proof.* If $Y, B$ satisfy (5.21) then $\operatorname{span}(Y)$ is an invariant subspace of $B$ [13], the eigenvalues of $B_1$ are the smallest magnitude eigenvalues of $B = I - D^{-1/2}AD^{-1/2}$, and the subspace $\operatorname{span} Y$ is the $K$-th principal subspace of $B$ (counting eigenvalues by increasing magnitude), i.e $Y = V\hat{U}$ where the columns of $V$ are the eigenvectors of $B$ and $\hat{U}$ is a $K \times K$ unitary matrix. The algorithm of [8] maps the data into the subspace $D^{-1/2}V$ while BestWCut maps them into $T^{-1/2}Y = D^{-1/2}Y$, hence the spectral mapping is the same up to a unitary transformation. In addition, when (5.21) is satisfied, the eigenvalues of $B_1$ are not defective[4].

As a consequence, the two criteria are equivalent when the relevant eigenvectors $Y$ of the symmetrized $H$ define an invariant subspace for both $B$ and $B^T$. In other words, although $B$ and hence $A$ is an asymmetric matrix, it contains a "symmetric core" corresponding to its highest eigenvalues.

We now show that the *WCut* is a generalization to $K$ way partitions of the criterion proposed by [21].

Proposition 5.2. *For $T = T' \leftarrow \Pi$, $A \leftarrow P$, with $P$ a stochastic matrix having the unique stationary distribution $\pi$, $\Pi = \operatorname{diag}\{\pi\}$,*

$$H(B) = I - (\Pi^{1/2}P\Pi^{-1/2} + \Pi^{-1/2}P\Pi^{1/2})/2 \tag{5.22}$$

The proof is elementary. In [20] clustering is done by the e-vector corresponding to the second largest e-value of $(\Pi^{1/2}P\Pi^{-1/2} + \Pi^{-1/2}P\Pi^{1/2})/2$ which is identical with $I - H(B)$ for $H(B)$ as in (5.22). Our algorithm embeds the nodes in the space spanned by the eigenvectors of the $K = 2$ smallest e-values of the latter, hence the two spectral embeddings are the same.

Extending the asymptotic results of [17] to the asymmetric case is also possible. We only note here that the spectrum of the matrix $H(B)$ and of the corresponding limit operator (under similar assumptions to the ones in [17]) is not generally contained in $[0, 2]$; its essential spectrum does not reduce to a single point either. Therefore the asymptotic convergence of the eigenvectors of interest is not assured in all cases.

Finally, let us note that although we have shown that minimizing weighted cuts in directed graphs by spectral methods amounts to symmetric spectral clustering on a "symmetrized" matrix $H$, the symmetrization obtained differs from the naive approaches $S = \frac{1}{2}(A + A^T)$ (except when the in- and out-degrees are equal) and $S = AA^T$.

[4] I.e the corresponding eigenspaces are of maximal dimension [13].

## 6 Experiments

We ran experiments comparing our approach with several other clustering techniques applicable to asymmetric affinity data. All techniques, including ours, consist of two steps: first the affinity data are mapped to a low dimensional vector space, then the mapped data are clustered in that space. The clustering was done in all cases by both $k$-means and single link methods; we report results for whichever method of clustering provided better results with that embedding for the experiment. The mappings for each method are listed in the table in table 2.

We measure clustering performance in two metrics: the *classification error (CE)*, described in [16], and the *variation in information (VI)*, described in [6]. A clustering that coincides with the "correct clustering" will result in both CE and VI being 0.

**Synthetic Data.** We first tested our algorithm by using synthetic data sets generated by creating a set $S$ of data points with a predefined clustering $C$, generating random weights between (0,1) for each directed edge between clusters, and added small random edges between points in different clusters for noise.

We ran experiments on twenty test data sets of 400 points and six embedded clusters, all generated according to the above scheme. An image of one such affinity matrix generated can be seen in Figure 2. The mean and variance of the CE over all runs may be seen in Table 4, while the mean and variance of the VI over all runs may be seen in Table 4. We see here that the *WNCut* algorithm outperforms other algorithms with regard to clustering error and VI, and *WACut* outperforms many previous existing techniques. The symmetrized versions of these techniques perform admirably, but with a slight decrease in performance, demonstrating the value of maintaining the asymmetry of the linkage in the graph.

**Biological Sequence Data.** For our next experiment, we ran the algorithm on a more difficult data set to cluster: a subset of biological sequence data from the Structural Classification of Proteins (SCOP) database, version 1.67 [9]. As mentioned in [8], some algorithms for measuring the similarity of biological sequence data, such as the Smith-Waterman algorithm [12], produce asymmetric affinities between the sequences compared.

The SCOP database contains roughly 24,000 proteins, divided into seven classes. Each class is further divided into folds, and each fold divided into superfamilies. We took proteins from the five largest folds from one class of the database - a total of 960 sequences -

Table 2: The algorithms used for testing.

| | |
|---|---|
| $WNCut$ | The BestWCut algorithm with $T = D$ (*weighted normalized cut*) |
| $WACut$ | The BestWCut algorithm run with $T = \mathbf{1}$, (*weighted average cut*) |
| $WNCut(A + A^T)$ | $WNCut$ algorithm with symmetrized matrix $A + A^T$ as input |
| $WACut(A + A^T)$ | $WACut$ algorithm with symmetrized matrix $A + A^T$ as input |
| $WNCut(AA^T)$ | $WNCut$ algorithm with symmetrized matrix $AA^T$ as input |
| $WACut(AA^T)$ | $WACut$ algorithm with symmetrized matrix $AA^T$ as input |
| SVD | Top left $K$ singular vectors of $A$ |
| MDS | Classical multidimensional scaling |
| Isomap | Isomap algorithm [14] with $l = 5$ neighbors |
| Zhou et al [21] | Algorithm of Zhou et al (BestWCut with $T = \pi$, the stationary distribution of teleporting random walk with $\eta = 0.85$ |
| RHC | Recursive hierarchical clustering algorithm of [8] |

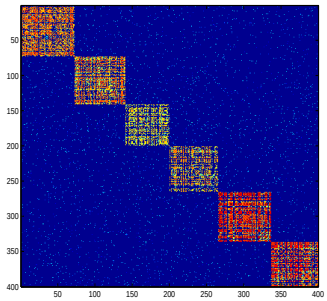



Figure 2: Synthetic affinity matrix with 400 points and six clusters.

Table 3: Mean and variance of clustering error on clusterings of 20 synthetic affinity matrices w/400 points.

| Method | Mean CE | Var CE |
|---|---|---|
| BestWCut($WNCut$) | 0.03 | 0.00 |
| BestWCut($WACut$) | 0.14 | 0.02 |
| BestWCut($WNCut$,$A + A^T$) | 0.05 | 0.01 |
| BestWCut($WACut$,$A + A^T$) | 0.20 | 0.01 |
| BestWCut($WNCut$,$AA^T$) | 0.12 | 0.02 |
| BestWCut($WACut$,$AA^T$) | 0.14 | 0.02 |
| SVD | 0.31 | 0.01 |
| MDS | 0.67 | 0.00 |
| Isomap | 0.20 | 0.01 |
| Zhou et al | 0.31 | 0.00 |
| RHC | 0.34 | 0.00 |

Table 4: Mean and variance of VI on clusterings of 20 synthetic affinity matrices w/400 points.

| Method | Mean VI | Var VI |
|---|---|---|
| BestWCut($WNCut$) | 0.06 | 0.01 |
| BestWCut($WACut$) | 0.22 | 0.03 |
| BestWCut($WNCut$,$A + A^T$) | 0.08 | 0.03 |
| BestWCut($WACut$,$A + A^T$) | 0.33 | 0.04 |
| BestWCut($WNCut$,$AA^T$) | 0.19 | 0.05 |
| BestWCut($WACut$,$AA^T$) | 0.23 | 0.05 |
| SVD | 0.51 | 0.04 |
| MDS | 1.98 | 0.01 |
| Isomap | 0.38 | 0.04 |
| Zhou et al | 0.83 | 0.04 |
| RHC | 1.60 | 0.13 |

Table 5: Performance of BestWCut algorithms vs. other techniques on SCOP and WebKB data.

| | SCOP | | WebKB | |
|---|---|---|---|---|
| Method | CE | VI | CE | VI |
| BestWCut($WNCut$) | 0.41 | 1.13 | 0.002 | 0.03 |
| BestWCut($WACut$) | 0.34 | 1.08 | 0.06 | 0.13 |
| BestWCut($WNCut$,$A + A^T$) | 0.52 | 1.28 | 0.01 | 0.10 |
| BestWCut($WACut$,$A + A^T$) | 0.45 | 1.37 | 0.68 | 2.56 |
| BestWCut($WNCut$,$AA^T$) | 0.49 | 1.48 | 0.57 | 1.67 |
| BestWCut($WACut$,$AA^T$) | 0.47 | 1.42 | 0.69 | 2.50 |
| SVD | 0.41 | 1.16 | 0.34 | 1.47 |
| MDS | 0.50 | 0.96 | 0.37 | 1.36 |
| Isomap | 0.69 | 2.50 | 0.47 | 1.63 |
| Zhou et al | 0.53 | 1.67 | 0.66 | 3.22 |
| RHC | 0.48 | 1.48 | 0.61 | 1.70 |

and attempted to cluster them according to their similarities as measured by the Smith-Waterman algorithm. This data is somewhat more difficult to cluster owing to, among other issues, the presence of subclusters within the clusters; algorithms are susceptible to dividing up clusters according to their superfamilies rather than the folds. We then compared the clusterings to the true clustering of the proteins by fold. The results may be seen in Table 5. The BestWCut algorithms are competitive with or outperform the other techniques tested.

**Web Graph Data.** For the next experiment, we used the adjacency matrix $A$ produced by the link graph of a set of web pages pulled from a crawl of four universities' computer science departments (Washington, Wisconsin, Texas, and Cornell); the crawl was collected for the WebKB project [2] Some pages from other sites in the original WebKB crawl were removed. In the matrix $A$, $A_{ij} = 1$ if page $i$ has a link to page $j$ and 0 otherwise. The test data set contains 3,946 pages and only 8,308 links. Most, but not all, of the links in the graph are between two pages in the same department, rather than pages in different departments, and the graph is very sparse; for this reason, clustering the data can represent a challenge. Since within-domain links are far more frequent than between-domain links, we used the university affiliation of pages as a natural true clustering for the data, and compared this to the clusterings generated by the algorithms. These results may also be seen in Table 5. We see good results from the formulations of the $WNCut$ algorithm relative to the alternative techniques with regard to both metrics; the CE is considerably lower on both. We see that symmetrization does provide benefit on this particular problem, however, as $WNCut$ on the symmetric matrix $A+A^T$ performs quite well.

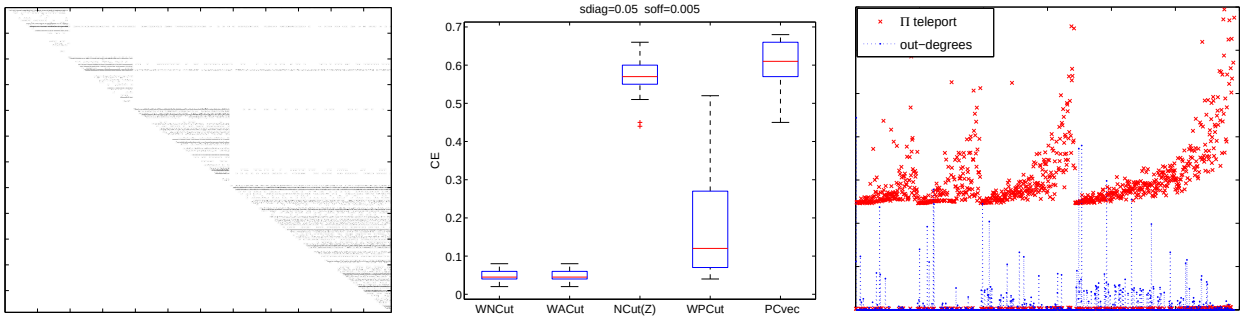
The last set of experiments highlights some of the properties of asymmetric matrices with respect to clustering. For this purpose, we constructed a highly asymmetric $A$ shown in figure 3. This has dimension 600, $K = 4$ clusters, and a distribution of $0, 1$ entries, in which the 1's are sparse and only strictly above the diagonal. This emulates the (transpose of) a matrix of citations. The probabilities of links ("citations") in the diagonal and off-diagonal blocks is 0.05 and respectively 0.005. The density of the rows is highly uneven. We made the lower right element a 1, for purposes of normalization. This matrix has a stationary distribution with $\pi_i = 0$ for all $i < n$. We have compared on it all the existing algorithms for asymmetric matrices, and the clustering results are in figure 3, b. We see that weighting the nodes by the outdegrees $D$ ($WNCut$) is optimal, followed closely by the average cut. The product method of [21] ($NCut(Z)$) and the method based on piece-wise constantness of the eigenvectors [8] ($PCvec$) perform poorly; $WPCut$ is the method of [19], and differs from the optimal ones only by the weighting of the nodes, which is done with the $\pi$ of a teleporting random walk with $\alpha = 0.01$[5]. In figure 3 it is shown why. Because the skewness of the link structure, the stationary distribution of the teleporting random walk puts its weights on the *last* entries in each cluster, almost independently of the link structure inside the cluster. However, as it turns out, for this problem the highly connected nodes can almost perfectly define the cluster structure (as shown by the low error rates obtained), so having $T = D$ is reasonable. We stress once again that the "outdegrees" in this artificial problem are representing the indegrees in a real citation matrix.

## 7 Conclusions and Future Work

While clustering on directed graphs is recognized as important, defining principled paradigms for it is only beginning. Here we have formulated the clustering problem as optimization of a weighted cut. This framework unifies many different criteria used successfuly on undirected graphs, like the normalized cut and the averaged cut, and for directed graphs [19]. We have also highlighted the difference with a previous approach [8] based on other aspects (the piecewise constantness) of random walks in a graph.

The paper showed that the discrete optimization can be relaxed to a (continuous) symmetric eigenproblem. The result is good news for two reasons. First, many existing spectral clustering algorithms and theo-

[5] We have tried a large range of $\alpha$ values, including $\alpha$ close to 1, and these results are by far the best for this algorithm.




**a** **b** **c**

Figure 3: A sparse, strongly asymmetric matrix **(a)**. Clustering results (CE) for several algorithms, **(b)**. The outdegrees $D$ and stationary distribution $\Pi^*$ for the teleporting random walk $\alpha = 0.01$.

retical results can be applied to this problem with only minor changes. A few of them have been listed in section 4. In future work, we will develop more detailed asymptotic convergence criteria for the generalized weighted cut. Second, it is known that spectra of asymmetric matrices are often not robust to even very small perturbations, making the eigenvalues and eigenvectors of such matrices less useful for analysis [15]. By contrast, eigenvectors of symmetric matrices are both more stable and more meaningful.

The symmetric matrix obtained by us differs from the popular ways to turn an assymetric affinity $A$ into a symmetric one, and gives better experimental results. We have also shown that what used to be a single criterion for symmetric graphs, becomes two distinct criteria (the normalized cut and clustering by random walks) which are only rarely identical for link data. We believe that link data is fundamentally different than symmetric affinity data. Directed graphs allow for a much richer range of clustering objectives.

Future work will investigate the appropriateness of each criterion in clustering the different types of link data, such as social networks, proteins or relational graphs. Given the many possible versions of weighted cuts, the natural question is how to find the optimal weighting $T$ for the current data set? This question is still awaiting a final answer.

Finally, we note that our method can be easily and naturally extended to semisupervised learning, by e.g the same principle as in [21].

## References


[1] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

[2] M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery. Learning to extract knowledge from the world wide web. In *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-98)*, 1998.

[3] J. Huang, T. Zhu, and D. Schuurmans. Web communities identification from random walks. In *Joint European Conference on Machine Learning and European Conference on Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD-06)*, 2006.

[4] M. Meila and L. Xu. Multiway cuts and spectral clustering. Technical Report 442, University of Washington, May 2003.

[5] M. Meilă. The multicut lemma. Technical Report 417, University of Washington, 2002.

[6] M. Meilă. Comparing clusterings by the variation of information. In M. Warmuth and B. Schölkopf, editors, *Proceedings of the Sixteenth Annual Conference ofn Computational Learning Theory (COLT)*, volume 16. Springer, 2003.

[7] M. Meilă and J. Shi. A random walks view of spectral segmentation. In T. Jaakkola and T. Richardson, editors, *Artificial Intelligence and Statistics AISTATS*, 2001.

[8] W. Pentney and M. Meilă. Spectral clustering of biological sequence data. In M. Veloso and S. Kambhampati, editors, *Proceedings of Twentieth National Conference on Artificial Intelligence (AAAI-05)*, 2005.

[9] Structural classification of proteins database. http://scop.berkeley.edu.

[10] J. Shi and J. Malik. Normalized cuts and image segmentation. *PAMI*, 2000.

[11] J. Shi and J. Malik. Normalized cuts and image segmentation. *PAMI*, 2000.

[12] T. Smith and M. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–197, 1981.

[13] G. W. Stewart and J.-g. Sun. *Matrix perturbation theory*. Academic Press, San Diego, CA, 1990.

[14] J. B. Tenenbaum, V. D. Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[15] L. Trefethen and M. Embree. *Spectra and Pseudospec-*

*tra: The Behavior of Non-Normal Matrices and Operators*. Princeton University Press, Princeton, NJ, 2005.

[16] D. Verma and M. Meilă. A comparison of spectral clustering algorithms. TR 03-05-01, University of Washington, May 2003. (sumitted).

[17] U. von Luxburg, O. Bousquet, and M. Belkin. Limits of spectral clustering. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, number 17. MIT Press, 2005.

[18] S. X. Yu and J. Shi. Multiclass spectral clustering. In *International Conference on Computer Vision*, 2003.

[19] D. Zhou, J. Huang, and B. Scholkopf. Learning from labeled and unlabeled data on a directed graph. pages 1041–1048, 2005.

[20] D. Zhou, J. Huang, and B. Scholkopf. Learning from labeled and unlabeled data on a directed graph. pages 1041–1048, 2005.

[21] D. Zhou, B. Schölkopf, and T. Hofmann. Semi-supervised learning on directed graphs. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, number 17. MIT Press, 2005.