Linearized Cluster Assignment via Spectral Ordering

Chris Ding CHQDING@LBL.GOV
Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720

Xiaofeng He XHE@LBL.GOV
Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720

Abstract

Spectral clustering uses eigenvectors of the Laplacian of the similarity matrix. They are most conveniently applied to 2-way clustering problems. When applying to multi-way clustering, either the 2-way spectral clustering is recursively applied or an embedding to spectral space is done and some other methods are used to cluster the points. Here we propose and study a K-way cluster assignment method. The method transforms the problem to find valleys and peaks of a 1-D quantity called cluster crossing, which measures the symmetric cluster overlap across a cut point along a linear ordering of the data points. The method can either determine K clusters in one shot or recursively split a current cluster into several smaller ones. We show that a linear ordering based on a distance sensitive objective has a continuous solution which is the eigenvector of the Laplacian, showing the close relationship between clustering and ordering. The method relies on the connectivity matrix constructed as the truncated spectral expansion of the similarity matrix, useful for revealing cluster structure. The method is applied to newsgroups to illustrate introduced concepts; experiments show it outperforms the recursive 2-way clustering and the standard K-means clustering.

1. Introduction

In recent years spectral clustering emerges as solid approach for data clustering. Spectral clustering includes a class of clustering methods (Bach & Jordan, 2003; Chan et al., 1994; Ding et al., 2002; Hagen & Kahng, 1992; Meila & Xu, 2003; Ng et al., 2001; Shi & Malik, 2000; Yu & Shi, 2003) that use eigenvectors of the Laplacian of the symmetric matrix $W = (w_{ij})$ containing the pairwise similarity between data objects i, j. Spectral clustering has well-motivated clustering objective functions and many interesting and useful properties can be proved.

Spectral clustering is most conveniently applied to 2-way clustering problem using a single eigenvector. When applying to multi-way (K-way) clustering, there are two main approaches: (1) the 2-way spectral clustering is recursively applied or (2) an embedding to spectral space using several eigenvectors is first done and some other methods, such as K-means (Ng et al., 2001; Zha et al., 2002; Bach & Jordan, 2003), are used to cluster the points. These cluster assignment methods are indirect (except perhaps a recent study (Yu & Shi, 2003)).

2. Linearized cluster assignment

Here we propose and study a direct K-way cluster assignment method. The method transforms the problem to one of finding valleys and peaks of a 1-D quantity called cluster crossing, which measures the cluster overlap across a cut point along linear ordering of data objects. In other words, the method linearizes the clustering assignment problem.

The linearized assignment algorithm depends crucially on an algorithm for ordering objects based on a pairwise similarity metric. The ordering is such that adjacent objects are similar while objects far away along the ordering are dis-similar. We show that for such an ordering objective function the inverse index permutation has a continuous (relaxed) solution which is the eigenvector of the Laplacian of the similarity matrix. This spectral ordering approach has been previously considered for reduction of the envelope of a sparse

Appearing in *Proceedings of the 21st International Conference on Machine Learning*, Banff, Canada, 2004.

symmetric matrix. (Barnard et al., 1993). Our contributions are (1) providing a clear ordering objective function and a new derivation, and (2) introducing a modification that significantly improves the ordering. This is discussed in §3.

The actual linearization is performed via the cluster crossing, the sum of similarities symmetrically across a cut point along the linear ordering. Computationally, this is the sum along anti-diagonal direction on W within a pre-specified bandwidth. Details are discussed in §4.

If the clusters in a dataset are well-separated, i.e., the similarity matrix W is nearly disconnected, the clustering crossing along the spectral ordering can easily detect the clusters. For datasets where clusters moderately or strongly overlap, cluster crossing directly computed from the similarity matrix W provides weak signals for revealing cluster structure. The connectivity matrix (Ding et al., 2002) provides sharper cluster structure and is adoped in the linearized assignment algorithm. This is briefly discussed in §5.

In summary, the linearized assignment algorithm depends on three techniques: (i) an ordering of the data objects, (ii) cluster crossing, (iii) the connectivity matrix. These are discussed in the following sections.

3. Distance sensitive ordering

Given n objects and the similarities between them $W = (w_{ij})$, the objective of ordering is to insure that (i) adjacent objects are similar (ii) the larger the distance between the objects, the less similar the two objects are.

The ordering is defined by the index permutation $\pi(1, 2, \cdots, n) = (\pi_1, \cdots, \pi_n)$. For a vector $\mathbf{x} = (x_1, \cdots, x_n)^T$, the permuted vector is $\pi(\mathbf{x}) = (x_{\pi_1}, \cdots, x_{\pi_n})^T$. The permuted similarity matrix is $(\pi W \pi^T)_{ij} = w_{\pi_i,\pi_j}$. Let $J_\ell(\pi) = \sum_{i=1}^{n-\ell} w_{\pi_i,\pi_{i+\ell}}$ represent the pairwise similarities between objects with fixed distance ℓ on the permuted order. We define the global ordering objective as

$$\min_\pi J(\pi),\ J(\pi) = \sum_{\ell=1}^{n-1} \ell^2 J_\ell(\pi) = \sum_{\ell=1}^{n-1} \ell^2 \sum_{i=1}^{n-\ell} w_{\pi_i,\pi_{i+\ell}} \tag{1}$$

Here larger distance similarities are minimized more heavily than smaller distances, to ensure that the *larger* the distance between a pair of objects, the *less* similar these two objects.

Let us compute the optimal π. First, let $j = i + \ell$ or $\ell = |i - j|$, $J(\pi)$ can be rewritten as

$$J(\pi) = \frac{1}{2}\sum_{i,j}(i-j)^2 w_{\pi_i,\pi_j} = \frac{1}{2}\sum_{\pi_i,\pi_j}(i-j)^2 w_{\pi_i,\pi_j}$$

Replacing π_i by i in the summation and noting that index i is permuted to π_i^{-1}, where π^{-1} is the inverse permutation, we obtain

$$J(\pi) = \frac{1}{2}\sum_{i,j}(\pi_i^{-1} - \pi_j^{-1})^2 w_{ij}$$

$$= \frac{n^2}{8}\sum_{i,j}\left(\frac{\pi_i^{-1} - (n+1)/2}{n/2} - \frac{\pi_j^{-1} - (n+1)/2}{n/2}\right)^2 w_{ij}.$$

For simplicity, we define the shifted and rescaled inverse index permutation

$$q_i \equiv \frac{\pi_i^{-1} - (n+1)/2}{n/2} \in \{\frac{1-n}{n}, \frac{3-n}{n}, \cdots, \frac{n-1}{n}\}, \tag{2}$$

which satisfies

$$\sum_i q_i = 0, \quad \sum_i q_i^2 = 1. \tag{3}$$

where $\mathbf{q}$ is further scaled by $q_i \to (n^3/12 - n/3)^{-1/2} q_i$ which does not change the permutation. Note that

$$\sum_{ij}(q_i - q_j)^2 w_{ij} = \sum_{ij}(q_i^2 + q_j^2 - 2q_i q_j) w_{ij} = 2\mathbf{q}^T(D - W)\mathbf{q}$$

where D is a diagonal matrix with each diagonal element being the sum of the corresponding row ($d_i = \sum_j w_{ij}$). Therefore, we need to minimize $\mathbf{q}^T(D-W)\mathbf{q}$ for q_i taking those discrete values of Eq.(2), subject to the constraints in Eq.(3). Using a Lagrangian multiplier for the second constraint in Eq.(3), minimization of $J(\pi)$ becomes

$$\min_{\mathbf{q}} \tilde{J}_1, \quad \tilde{J}_1 = \frac{\mathbf{q}^T(D-W)\mathbf{q}}{\mathbf{q}^T\mathbf{q}} \tag{4}$$

Finding the optimal solution for the discrete values of $\mathbf{q}$ is a combinatorial optimization problem, and is likely to have no polynomial-time optimal algorithms. However a *continuous* solution for $\mathbf{q}$ can be computed. We *relax* the restriction that q_i must take discrete values of Eq.(2) in $[-1, 1]$, and let q_i take continuous values in $[-1, 1]$. With this, $\tilde{J}_1$ can be minimized by solving an eigenvalue problem. It is well-known that $\mathbf{q}$ is an eigenvector of the equation

$$(D - W)\mathbf{q} = \zeta\mathbf{q}. \tag{5}$$

Clearly $\mathbf{q}_0 = \mathbf{1} = (1, \cdots, 1)^T$ is an eigenvector with $\zeta_0 = 0$. All other eigenvector are orthogonal to $\mathbf{q}_0$, i.e.,

the first constraint in Eq.(3) is also satisfied. Therefore $\mathbf{q}_1$ is the desired continuous solution of the distance sensitive ordering.

We note that earlier work on sparse matrix envelope reduction (Barnard et al., 1993) based on different motivation, reaches the same eigenvector solution. Our contribution here is to introduce the distance sensitive objective function $J(\pi)$, and provide the detailed derivation using shifted inverse permutation vector π^{-1} to show that the solution is $\mathbf{q}_1$.

Now we make a crucial modification on the above solution which (a) improves the quality of the solution, and (b) makes a direct connection to the scaled PCA and connectivity matrix in §5. The modification is made on the constraints in Eq.(3); we weight each point i with the degree d_i, the column sum of the similarity matrix W. In graph theory (Chung, 1997), d_i is called the **volume** of node i. The new constraints are

$$\sum_i q_i d_i = 0, \quad \sum_i q_i^2 d_i = 1. \tag{6}$$

(more discussion later). With these constraints, the minimization problem of $J(\pi)$ becomes

$$\min_{\mathbf{q}} \tilde{J}_2, \quad \tilde{J}_2 = \frac{\mathbf{q}^T(D-W)\mathbf{q}}{\mathbf{q}^T D\mathbf{q}}. \tag{7}$$

Relaxing q_i to continuous values in $(-1, 1)$. the solution for $\mathbf{q}$ satisfies the eigenvalue equation

$$(D-W)\mathbf{q} = \zeta D\mathbf{q}. \tag{8}$$

Let $\mathbf{q} = D^{-1/2}\mathbf{z}$. Substituting it into Eq.(8), we obtain

$$D^{-1/2}WD^{-1/2}\mathbf{z} = \lambda\mathbf{z}, \ \lambda = 1-\zeta. \tag{9}$$

This is a standard eigenvalue equation. Thus the eigenvectors $\mathbf{z}_k$ and $\mathbf{q}_k$ have the orthogonality relation

$$\mathbf{z}_k^T\mathbf{z}_\ell = \mathbf{q}_k^T D\mathbf{q}_\ell = \delta_{k\ell} = \begin{cases} 1 & \text{if} \quad k=\ell \\ 0 & \text{if} \quad k\neq\ell \end{cases} \tag{10}$$

The trivial eigenvector is $\mathbf{q}_0 = \mathbf{e}$ with $\zeta_0 = 0$. Thus the constraints in Eq.(6) are automatically satisfied.

Since $(D-W)$ is semipositive definite, we have

$$\zeta_k \geq 0, \quad \lambda_k = 1-\zeta_k \leq 1. \tag{11}$$

In distance-sensitive ordering, we seek $\mathbf{q}_k$ with the smallest ζ_k, or the largest λ_k. The desired solution for the permutation π_i^{-1} is $\mathbf{q}_1(i)$ (the i^{th} element of $\mathbf{q}_1$), subject to the rescaling and a constant shift condition according to Eq.(2). Note that

$$\mathbf{q}_1(i) < \mathbf{q}_1(j) \implies \pi_i^{-1} < \pi_j^{-1}. \tag{12}$$

Thus π^{-1} can be uniquely recovered from $\mathbf{q}_1$. A simple implementation to recover the permutations is to sort the elements of $\mathbf{q}_1$ in increasing order. This sorting induces the desired index permutation π.

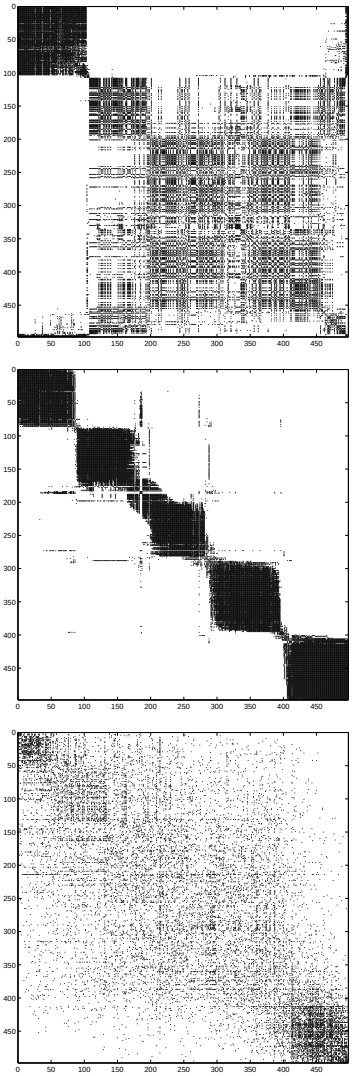

Figure 1. The connectivity matrix of 5-newsgroup (see §5) is displayed using the $\tilde{J}_1$ ordering of Eq.(4) (top), and using the $\tilde{J}_2$ ordering of Eq.(7) (middle). The original cosine similarity of the 5 newsgroups is shown in bottom.

In Figure 1, we show a matrix where $\tilde{J}_1$ ordering is compared to $\tilde{J}_2$ ordering. Clearly, $\tilde{J}_2$ ordering provides a better distance-sensitive ordering. The values of the initial ordering objective $J(\pi)$ are

$$\begin{aligned} J(\pi)/\langle J\rangle &= 0.846, \quad \text{using } \tilde{J}_1 \\ J(\pi)/\langle J\rangle &= 0.584, \quad \text{using } \tilde{J}_2 \\ J(\pi)/\langle J\rangle &= 0.949, \quad \text{using random ordering} \end{aligned} \tag{13}$$

where $\langle J\rangle = (\sum_{ij} w_{ij}/n^2)\sum_{ij}(i-j)^2$ is the expected

mean value for $J(\pi)$. Another way to measure the effects of ordering is to use the bandwidth and envelope of a symmetric sparse matrix C. The bandwidth $b(i)$ at row i is the largest distance between the diagonal element and any nonzero element in row i. The bandwidth of the entire matrix is the largest of $b(i)$ and the envelope is the sum of $b(i)$. For the $\tilde{J}_1$ ordering of C, bandwidth $= 495$, envelope $= 156,240$. For the $\tilde{J}_2$ ordering of C, bandwidth $= 320$, envelope $= 48,650$. Clearly, $\tilde{J}_2$ ordering is better.

$\tilde{J}_2$ ordering uses the weighted constraints of Eq.(6) while $\tilde{J}_1$ uses the unweighted constraints of Eq.(3). To understand why the weighting leads to better ordering, first observe that objects with large d_i will get smaller $|q_i|$ to balance out the equation. Now we rewrite Eq.(2) as $\pi_i^{-1} = q_i - (n+1)/2$ ignoring the overall scaling factor. Since $\pi^{-1} = \{1, \cdots, n\}$, $(n+1)/2$ is in the middle. Smaller $|q_i|$ indicates π_i^{-1} is near the middle, thus objects with large d_i are more likely to be permuted towards the middle using the weighted constraints of Eq.(6). This is favorable, since objects with large d_i are more likely to have more edges; and moving these objects towards middle decreases the distances among these similar objects, therefore improves $J(\pi)$.

Connection to spectral clustering

Note that eigenvector of Eq.(5) is used in the *Ratio cut* spectral clustering (Hagen & Kahng, 1992) and eigenvector of Eq.(8) is used in the *normalized cut* (Shi & Malik, 2000) and min-max cut (Ding et al., 2001) spectral clustering.

In deriving 2-way spectral clustering, only the signs of the cluster indicator vector are useful and all objects in a cluster have the same magnitude. This indicator vector is then relaxed into the eigenvector. In our derivation of spectral ordering, both the sign and magnitude of the scaled and shifted permutation vector are useful, see Eq.(2). Since an eigenvector has both sign and magnitude, the relaxation of the permutation vector is therefore better quality approximation than the relaxation of cluster indicator.

From this analysis, we believe the better reason for the success of spectral clustering is due to the ordering, instead of relaxing the discrete cluster indicators. In fact, this ordering perspective is used in actual implementation of spectral clustering (Hagen & Kahng, 1992; Shi & Malik, 2000): one first sort $\mathbf{q}_1$ to provide a linear ordering, then along this ordering, search for the cut that optimizes the cluster objective function. Thus our ordering analysis provides a deeper understanding of spectral clustering.

Our results indicates that $\tilde{J}_2$ ordering is better than $\tilde{J}_1$ ordering. This is due to the weighting of d_i, the node degree, in Eq.(6). Similar motivation is used in *normalized cut*. Let s_{12} be the cut between two subgraphs C_1, C_2. All three graph clustering objective function can be written as $J = s_{12}/a_1 + s_{12}/a_2$. For *Ratio cut*, $a_k = \sum_{i \in C_k} 1$. For *normalized cut*, $a_k = \sum_{i \in C_k} d_i$. This weighting of the subgraph volume improves upon the simple weighting of the subgraph size in ratio cut. For min-max cut, $a_k = \sum_{i,j \in C_k} w_{ij}$, the sum of edge weights inside C_k.

That $\tilde{J}_2$ ordering is better than $\tilde{J}_1$ ordering implies that *normalized cut* and *min-max cut* in general provides a better clustering than ratio cut. This fact is observed in experiments (Shi & Malik, 2000; Ding et al., 2001).

It is sometimes happens that there is a symmetry among several nodes in the graph, i.e., $G(W)G^T = W$, where G is a permutation specifying an element in the invariant symmetry group. In this case, an eigenvector of $W, D - W, D^{-1/2}WD^{-1/2}$ will have several nodes with the same value. If this happens, neither the ordering nor clustering problems can be uniquely determined. This is not necessarily a weakness of the spectral methods, although it become obvious from the perspective of the eigenvector. In practice, this happens rarely for weighted graphs.

4. Cluster crossing and assignment

We start with cluster overlap. Given two clusters C_k, C_l, the cluster overlap can be defined as the sum of pairwise associations between two clusters,

$$s_{kl} = \sum_{i \in C_k, j \in C_l} w_{ij}. \tag{14}$$

In spectral clustering, s_{kl} are minimized.

Cluster overlap involves all $|C_k| * |C_l|$ pairwise similarities. We define cluster crossing as the sum of a small fraction of the pairwise similarities. This is aided by linear ordering data points. Given a linear order o of all objects, at each site i of the order, we can sum over w_{ij} within a window size $2m+1$ across the site i,

$$\rho(i) = \sum_{j=1}^{m} w_{o(i-j),o(i+j)}$$

This corresponds to sum along the anti-diagonal directions in the similarity matrix W with a bandwidth m. There are $2n-1$ anti-diagonals in a matrix, among them $\rho(i)$ are n full-step anti-diagonals. We also utilize

the other $n-1$ half-step anti-diagonals, i.e.,

$$\rho(i \pm 1/2) = \sum_{j=1}^{m} w_{o(i-j),o(i+j\pm 1)}.$$

The final crossing is the weighted average

$$\tilde{\rho}(i) = \rho(i+1/2)/4 + \rho(i)/2 + \rho(i-1/2)/4.$$

(When i is close to the two ends, i.e., $n-i \leq m$ or $i \leq m$, the sum should be properly weighted to reflect the fact that the number of similarities in the sum is less than the normal case.)

Clearly, cluster crossing $\tilde{\rho}(i)$ should have a minimum at the cluster boundary between C_k, C_l. As i moves away from the boundary, $\tilde{\rho}(i)$ increases. This form the basis of the linearized cluster assignment. This approach works for $K > 2$ as well as for $K = 2$; In essence, it reformulate a problem of K-way clustering with pairwise similarities into a K-way clustering problem in 1-dimension.

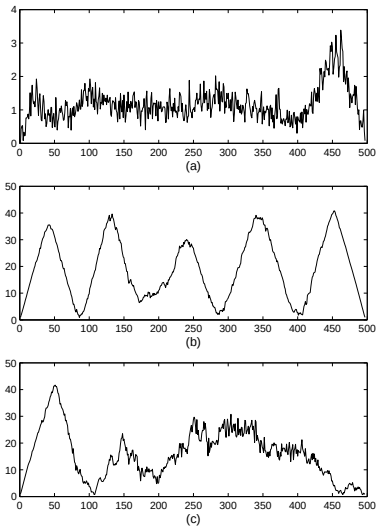

Figure 2. Crossing curves for dataset A. Top: computed based on the similarity matrix W shown in Fig.1(bottom). Middle: computed based on the connectivity matrix C using $\tilde{J}_2$ ordering shown in Fig.1(middle). Bottom: computed based on C using $\tilde{J}_1$ ordering shown in Fig.1(top).

To illustrates the basic ideas in this approach. we compute the crossing of the matrix C shown in Figure 1(bottom). The crossing curves are shown in Figure 2. For matrix C using $\tilde{J}_2$ ordering, the crossing (middle panel in Figure 2) exhibit clearly the five-cluster structure.

In cluster crossing curve, the valleys are more important than the peaks. This is because between two consecutive valleys, there could be several overlapping clusters so that the peaks are not as pronounced as the valleys. Using the valleys, we can clearly separate sets of clusters (composite cluster).

This suggest a divide-and-conquer approach, i.e., recursively apply the algorithm on each set of clusters, until the total number of cluster reach the pre-specified K, or some other criteria are met (like the top-down divisive clustering approach). For example in Figure 1 (middle), the 2nd and 3rd clusters overlap slightly, and the corresponding valley point in the crossing curve (middle panel in Figure 2) is not as low as others (although unambiguously clear). We might consider these two cluster as one composite cluster. Thus we cut the crossing into 4 clusters at present round and cut the composite cluster in next round.

5. Scaled principal components and connectivity matrix

Given a symmetric similarity matrix W, one can obtain a spectral decomposition to get the principal component analysis (PCA), a widely used technique in multivariate statistics. In scaled PCA proposed in (Ding et al., 2002), one performs the following spectral decomposition

$$W = D^{1/2}(D^{-1/2}WD^{-1/2})D^{1/2}$$

$$= D^{1/2}(\sum_{k=1}^{K} \mathbf{z}_k \hat{\lambda}_k \mathbf{z}_k^T)D^{1/2}$$

here the spectral decomposition is performed on the scaled matrix $\widehat{W} = D^{-1/2}WD^{-1/2}$. Clearly, the eigenvectors $\mathbf{z}_k$ are governed by Eq.(9). The magnitudes of all eigenvalues are less than 1, as in Eq.(11). The connectivity matrix C is obtained by truncating the PCA expansion at K terms and setting the eigenvalues to unity, $\lambda_k = 1$, as

$$C = D^{1/2} \sum_{k=1}^{K} \mathbf{z}_k \mathbf{z}_k^T D^{1/2} = D \sum_{k=1}^{K} \mathbf{q}_k \mathbf{q}_k^T D \qquad (15)$$

where $\mathbf{q}_k = D^{-1/2}\mathbf{z}_k$ is called *scaled principal components* due to its similarity to the usual PCA. $\mathbf{q}$ is governed by Eq.(8), and is closely related to spectral clustering. It is shown via a perturbation analysis that C has a so-called self-aggregation property that connectivities (matrix elements in C) between different clusters are suppressed while connectivities within clusters are enhanced. Thus C is useful for revealing cluster structure.

Connectivity matrix approach involves a noise reduction procedure. The probability that two objects i, j

belong to the same cluster is $p_{ij} = C_{ij}/C_{ii}^{1/2}C_{jj}^{1/2}$. To reduce noise one set

$$C_{ij} = 0 \quad \text{if} \quad p_{ij} < \beta, \tag{16}$$

where $\beta = 0.8$. For a range of problems, $\beta = 0.5 \sim 0.9$ leads to very similar results.

As an illustration, the original similarity matrix (based on 5 newsgroups in §6) is shown in Figure 1(bottom) using $\tilde{J}_2$ ordering, where cluster structure is not apparent. Connectivity matrix (shown in Figure 1) constructed based on this similarity matrix has clear cluster structure.

6. Complete assignment algorithm

Prespecify K as the numnber of clusters, and set bandwidth $m = n/K$ (or the expected largest cluster size). The complete algorithm is as follows: (1) Compute connective matrix C; (2) Compute the $\tilde{J}_2$ ordering of C; (3) Compute the crossing of C based on $\tilde{J}_2$ ordering. (4) Locate valley points in the crossing curve. Assign each region sandwiched between two valley points or ends to one composite cluster. (5) If the total number of current composite cluster is less than K, recursively apply the algorithm to the largest to further split it.

Note that this recursive clustering algorithm differs from the usual recursive 2-way clustering in that, a current composite cluster is partitioned into several clusters depending on the crossing curve, not restricted to 2 clusters. It is possible that all K clusters are identified using one crossing curve as in §4. Thus the total number of recursion is less than or equal to K-1, which is required by the usual recursive 2-way clustering. In step (5), the choice of next cluster to split is based on the largest (size) cluster. This simple choice is more oriented towards cluster balance. More refined choices for cluster split is discussed in (Ding & He, 2002).

The main advantage of this approach is that clusters are formed consistently. In 2-way recursive clustering, each current cluster is formed via a certain clustering objective function which is correctly motivated for only true clusters, not for composite clusters. For example in normalized cut, the cluster objective function for K-way clustering can not be recursively constructed from the 2-way clustering objective. Therefore the 2-way recursive procedure is only a heuristic for K-way clustering. In our linearized assignment, cluster are assign based on the criteria that the connectivity between clusters are small, which is valid for both true clusters and composite clusters.

7. Experiments

The linearized cluster assignment method is applied to Internet newsgroup articles. A 20-newsgroup dataset is from www.cs.cmu.edu/afs/cs/project/theo-11/www/naive-bayes.html. Word-document matrix is first constructed. 1000 words are selected according to the mutual information between words and documents in unsupervised manner. Standard `tf.idf` term weighting is used. Each document is normalized to 1.

We focus on two sets of 5-newsgroup combinations. The choice of $K = 5$ is to have some variety in the recursive steps (we avoid $K = 4, 8$). These two newsgroup combinations are listed below:

```
          A                              B
NG2:  comp.graphics            NG2:  comp.graphics
NG9:  rec.motorcycles          NG3:  comp.os.ms-windows
NG10: rec.sport.baseball       NG8:  rec.autos
NG15: sci.space                NG13: sci.electronics
NG18: talk.politics.mideast    NG19: talk.politics.misc
```

Datasets with well separated clusters are easy to handle; we are interested in clustering medium and large overlapping clusters. To measure cluster separation, we compute s_{kl} and define the symmetrically scaled cluster overlap as the *cluster separation index* between clusters C_k, C_l as $\mu_{kl} = s_{kl}/\sqrt{s_{kk}s_{ll}}$. The over-all separation is defined as

$$\mu = \frac{2}{K(K-1)} \sum_{k,l;k\neq l} \mu_{kl}.$$

Clearly $0 \leq \mu \leq 1$ and $\mu_{kk} = 1$. For a complete graph $\mu_{kl} = 1$ and $\mu = 1$. Cluster overlap s_{kl} and separation index μ_{kl} can be conveniently stored in a matrix S, where S(upper-right triangle including diagonals) = s_{kl} and S(lower-left triangle)= μ_{kl}. For datasets A, B, their overlap-separations are

$$S(A) = \begin{bmatrix} 662 & 207 & 184 & 285 & 205 \\ 0.316 & 650 & 247 & 254 & 265 \\ 0.264 & 0.357 & 732 & 256 & 281 \\ 0.422 & 0.379 & 0.361 & 691 & 310 \\ 0.258 & 0.336 & 0.336 & 0.381 & 957 \end{bmatrix}$$

$$S(B) = \begin{bmatrix} 576 & 367 & 218 & 245 & 198 \\ 0.578 & 702 & 256 & 287 & 234 \\ 0.347 & 0.371 & 682 & 273 & 286 \\ 0.470 & 0.499 & 0.482 & 472 & 229 \\ 0.261 & 0.280 & 0.347 & 0.334 & 996 \end{bmatrix}$$

Their average separation are $\mu(A) = 0.695, \quad \mu(B) = 0.755$. These information are useful. For example, for dataset B, NG18 (mideast) is a coherent cluster because $s_{55} = 996$ is relatively large, whereas NG13(electronics) is less coherent because $s_{44} = 472$ is relatively small. The overlap between NG2 (graphics)

and NG3(windows OS) is relatively large: $\mu_{12} = 0.578$ while the overlap between NG8 (auto) and NG13 (electronics) is also large: $\mu_{34} = 0.483$. Overall, dataset A is moderately overlapping and dataset B is strongly overlapping.

The above is based on a random sample of documents from the newsgroups. Each cluster has 100 documents. To accumulate sufficient statistics, for each newsgroup combination, we generate 5 samples and average their performance.

Dataset A

The cosine similarity matrix W among documents shown in Figure 1(bottom). The cluster structure is not clear from the similarity matrix W. The connectivity matrix C is displayed in Figure 1(top) which exhibits the cluster structure.

Clustering crossings based on W and C are shown in Figure 2. The crossing for C based on $\tilde{J}_2$ ordering (middle panel) shows clear cluster structure, whereas crossing for C based on $\tilde{J}_1$ ordering (bottom pannel) shows less clear cluster structure. Crossing for W (top pannel) show even less cluster structure.

Based on the crossing of C using $\tilde{J}_2$ ordering, local minima in the valleys are identified using a simple smoothing procedure, where the new smoothed value on each point is the average of old values on 5 nearest points. This smoothing procedure overcomes the local abrupt changes and automatically compute the more stable or consensus valley points, although the difference with un-smoothed one is often small. Data points between two valleys or ends are assigned to one cluster. Thus all points are assigned into 5 clusters in one shot.

Each of newsgroup article's cluster label is known (although not necessarily perfect). Using this, the confusion matrix $T = (t_{kl})$ for the clustering results are computed, where t_{kl} = number of points belonging to cluster k but clustered to cluster l. Based on the linearized cluster assignment results, T is computed as

$$T = \begin{bmatrix} 96 & 6 & 0 & 5 & 10 \\ 0 & 93 & 0 & 0 & 1 \\ 1 & 1 & 92 & 7 & 3 \\ 2 & 0 & 8 & 84 & 2 \\ 0 & 0 & 0 & 3 & 83 \end{bmatrix}$$

For this results, the clustering accuracy, $Q = \sum_k t_{kk}/N = 90.5\%$.

The clustering experiment is repeated for 5 different random samples. The accuracy for this linearized ordering approach is listed in Table 1. For the same dataset, the cluster accuracy using recursive 2-way spectral clustering and standard K-means are also listed in Table 1. One see that the linearized assignment outperform slightly over the recursive 2-way clustering and significantly over the K-means .

Table 1. Clustering accuracy as of different methods on the 5-newsgroup datasets. LA: linearized assignment; R2W: recursive 2-way clustering; using MinMaxCut with cluster choice for split based on largest size cluster; K-means.

Method	LA	R2W	K-means
Data A	89.0%	82.8%	75.1%
Data B	75.7%	67.2%	56.4%

Dataset B

The cosine-similarity matrix W is shown in Figure 3 (top panel, using $\tilde{J}_2$ ordering). The connectivity matrix C of this dataset is shown in Figure 4. The overlap between NG2 (computer graphics) and NG3 (Windows OS) is large; the overlap between NG8 (autos) and NG13 (electronics) is large as well. These are expected from the cluster separation indexes $S(B)$, and also can be confirmed by inspecting the cosine-similarity W shown in Figure 3 (bottom panel), using $\tilde{J}_2$ ordering based on C.

The crossing based on C is shown in Figure 5 (top panel). The lower panels show the crossing curves after four successive applications of smoothing. Based on this crossing, we can identify three composite clusters by two clear and low-lying valley points. The two large composite clusters are further clustered using the same linearized algorithm.

Repeating the experiments on 5 random samples from dataset B, the clustering accuracy is listed in Table 1. The linearized assignment outperforms the recursive 2-way spectral clustering and the standard K-means .

8. Summary

In summary, we propose and study a direct K-way cluster assignment method that linearize the clustering problem into 1-D clustering crossing curve. The method depends on an effective linear ordering provided by the spectral ordering. We prove a clear derivation of the distance sensitive ordering and show the shifted and scaled index permutation vector is relaxed into eigenvectors of the Laplacian of the similarity matrix. Our results provides a deeper insights to spectral clustering as well.

This work is supported by U.S. Department of Energy,

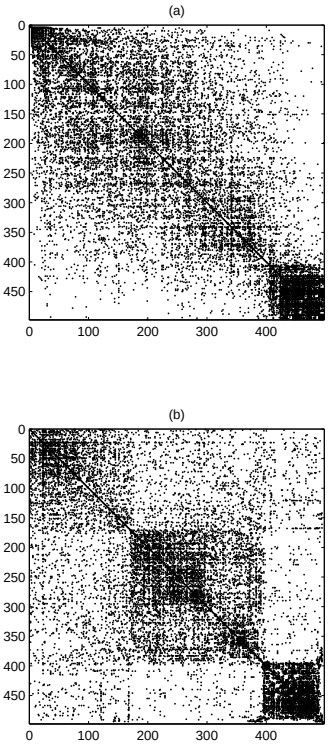

Figure 3. The cosine similarity matrix W of dataset B, displayed using (a) the $\tilde{J}_2$ ordering based on W, (b) the $\tilde{J}_2$ ordering based on the connectivity matrix C in Fig.(4).

Office of Science, Office of Laboratory Policy and Infrastructure, through an LBNL LDRD, under contract DE-AC03-76SF00098.

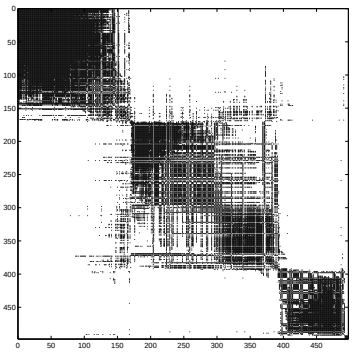

Figure 4. The connectivity matrix of dataset B.

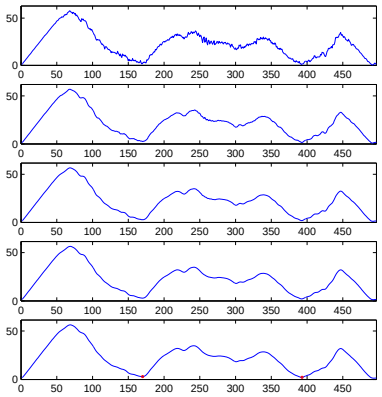

Figure 5. Crossings computed based on the connectivity matrix in Fig.(4). First curve is the crossing. Lower curves show the smoothing of the crossings. The two points in bottom curve indicate the cut points.

References

Bach, F. R., & Jordan, M. I. (2003). Learning spectral clustering. *Neural Info. Processing Systems 16 (NIPS 2003)*.

Barnard, S. T., Pothen, A., & Simon, H. D. (1993). A spectral algorithm for envelope reduction of sparse matrices. *Proc. Supercomputing '93, IEEE*, 493–502.

Chan, P., M.Schlag, & Zien, J. (1994). Spectral k-way ratio-cut partitioning and clustering. *IEEE Trans. CAD-Integrated Circuits and Systems*, *13*, 1088–1096.

Chung, F. (1997). *Spectral graph theory*. Amer. Math. Society.

Ding, C., & He, X. (2002). Cluster merge and split in hierarchical clustering. *Proc. IEEE Int'l Conf. Data Mining*, 139–146.

Ding, C., He, X., Zha, H., Gu, M., & Simon, H. (2001). A min-max cut algorithm for graph partitioning and data clustering. *Proc. IEEE Int'l Conf. Data Mining*.

Ding, C., He, X., Zha, H., & Simon, H. (2002). Unsupervised learning: self-aggregation in scaled principal component space. *Proc. 6th European Conf. Principles of Data Mining and Knowledge Discovery (PDKK 2002)*, 112–124.

Hagen, L., & Kahng, A. (1992). New spectral methods for ratio cut partitioning and clustering. *IEEE. Trans. on Computed Aided Desgin*, *11*, 1074–1085.

Meila, M., & Xu, L. (2003). Multiway cuts and spectral clustering. *U. Washington Tech Report*.

Ng, A., Jordan, M., & Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. *Proc. Neural Info. Processing Systems (NIPS 2001)*.

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

Yu, S. X., & Shi, J. (2003). Multiclass spectral clustering. *Int'l Conf. on Computer Vision*.

Zha, H., Ding, C., Gu, M., He, X., & Simon, H. (2002). Spectral relaxation for K-means clustering. *Advances in Neural Information Processing Systems 14 (NIPS'01)*, 1057–1064.