Spectral Clustering with Two Views

Virginia R. de Sa DESA@UCSD.EDU
Department of Cognitive Science, 0515
University of California, San Diego
9500 Gilman Dr.
La Jolla, CA 92093-0515

Abstract

In this paper we develop an algorithm for spectral clustering in the multi-view setting where there are two independent subsets of dimensions, each of which could be used for clustering (or classification). The canonical examples of this are simultaneous input from two sensory modalitites, where input from each sensory modality is considered a view, as well as web pages where the text on the page is considered one view and text on links to the page another view. Our spectral clustering algorithm creates a bipartite graph and is based on the "minimizing-disagreement" idea. We show a simple artifically generated problem to illustrate when we expect it to perform well and then apply it to a web page clustering problem. We show that it performs better than clustering in the joint space and clustering in the individual spaces when some patterns have both views and others have just one view.

Spectral clustering is a very successful idea for clustering patterns. The idea is to form a pairwise affinity matrix A between all pairs of patterns, normalize it, and compute eigenvectors of this normalized affinity matrix (graph Laplacian)L. It can be shown that the second eigenvector of the normalized graph Laplacian is a relaxation of a binary vector solution that minimizes the normalized cut on a graph (Shi & Malik, 1998; J.Shi & Malik, ; Meila & Shi, 2001; Ng et al., 2001). Spectral clustering has the advantage of performing well with non-Gaussian clusters as well as being easily implementable. It is also non-iterative with no local minima. The Ng,Jordan,Weiss(Ng et al., 2001) (NJW) generalization to multiclass clustering (which we will build on) is summarized below for data patterns x_i to be clustered in to k clusters.

- Form the affinity matrix $A(i,j) = exp(-||x_i - x_j||^2/2\sigma^2)$
- Set the diagonal entries $A(i,i) = 0$
- Compute the normalized graph Laplacian as $L = D^{-.5}AD^{-.5}$ where D is a diagonal matrix with $D(i,i) = \sum_j A(i,j)$
- Compute top k eigenvectors of L and place as colums in a matrix X
- Form Y from X by normalizing the rows of X
- Run kmeans to cluster the row vectors of Y
- pattern x_i is assigned to cluster α iff row i of Y is assigned to cluster α

In this paper we develop an algorithm for spectral clustering in the multi-view setting where there are two independent subsets of dimensions, each of which could be used for clustering (or classification). The canonical examples of this are multi-sensory input from two modalities where input from each sensory modality is considered a view as well as web pages where the text on the page is considered one view and text on links to the page another view. Also computer vision applications with multiple conditionally independent sensor or feature vectors can be viewed in this way.

1. Algorithm Development

Our spectral multi-view algorithm is based on ideas originally developed for the (non-spectral) Minimizing-Disagreement algorithm (de Sa, 1994a; de Sa & Ballard, 1998). The idea behind the Minimizing-Disagreement (M-D) algorithm is that two (or more)

Appearing in *Proceedings of the Workshop on Learning with Multiple Views*, 22^{nd} ICML, Bonn, Germany, 2005.

networks receiving data from different views, but with no explicit supervisory label, should cluster the data in each view so as to minimize the disagreement between the clusterings. The Minimizing-Disagreement algorithm was described intuitively using the following diagram shown in Figure 1. In the figure imagine that there are two classes of objects, with densities given by the thick curve and the thin curve and that this marginal density is the same in each one-dimensional view. The scatter plots on the left of the figure show two possible scenarios for how the "views" may be related. In the top case, the views are conditionally independent. Given that a "thick/dark" object is present, the particular pattern in each view is independent. On the right, the same data is represented in a different format. In this case the values in view 1 are represented along one line and the values in view 2 along another line. Lines are joined between a pair if those values occurred together. The minimizing disagreement algorithm wants to find a cut from top to bottom that crosses the fewest lines – within the pattern space (subject to some kind of balance constraint to prevent trivial solutions with empty or near empty clusters). Disagreement is minimized for the dashed line shown. Here we transform this intuitive idea for 1-D views to a general algorithm on a weighted bipartite graph.

The difficulty in transforming this intuitive idea into a general algorithm for a M-D spectral algorithm is that in describing it as making a cut from top to bottom, we assume that we have a neighborhood relationship within each top set and bottom set, that is not explicitly represented. That is we assume that points drawn in a line next to each other are similar points in the same view. Treating the points as nodes in a graph and applying a graph cut algorithm, would lose that information.

One solution would be to simply connect co-occurring values **and** also join nearest neighbors (or join neighbors according to a similarity measure) in each view. This, however, raises the tricky issue of how to encode the relative strengths of the pairing weights with the within-view affinity weights.

Instead, our solution is to draw reduced weight co-occurrence relationships between neighbors of an observed pair of patterns (weighted by a unimodal function such as a Gaussian). We call our algorithm **sM-D** Each input in each view is represented by a node in the graph. The strength of the weight between two nodes in different views depends on the number of multi-view patterns (which we can think of as co-occuring pairs of patterns) that are sufficiently close (in both views) (with a fall off in weight as the distances grow). This representation has the semantics that we believe there is noise in the actual patterns that occur or alternatively that we wish to consider the pairings simultaneously at multiple scales.

More specifically, let us define $x_i^{(v)}$ as view v of the ith pattern. We will construct a graph node for each view of each pattern and define $n_{(i,v)}$ to represent the node for view v of the ith pattern. Now consider the pattern $x_1^{(1)} = [1\ \ 2\ \ 1]'$ (where throughout this paper ' denotes the transpose operator) and the pattern $x_2^{(1)} = [1\ \ 2\ \ 1]' + \vec{\epsilon}'$. These two patterns should probably be considered identical for small $\vec{\epsilon}$. This means that $x_1^{(2)}$ the co-occurring pattern for $x_1^{(1)}$ should probably also be linked with $x_2^{(1)}$. The Gaussian weighting allows us to do this in a smooth way for increasing $\vec{\epsilon}$. To compute the total weight between node $n_{(i,1)}$ and $n_{(j,2)}$ we sum over all observed pattern co-occurrences (k=1 to p): the product of (the (Gaussian weighted) distance between $x_i^{(1)}$ (the pattern represented by $n_{(i,1)}$) and $x_k^{(1)}$ and the same same term for the relationship between the $x_j^{(2)}$ and $x_k^{(2)}$. That is

$$w_{ij} = \sum_p e^{-\frac{||(x_i^{(1)} - x_k^{(1)})||^2}{2\sigma_1^2}} e^{-\frac{||(x_j^{(2)} - x_k^{(2)})||^2}{2\sigma_2^2}} \quad (1)$$

$$= [A_{v1} \times A_{v2}]_{ij} \quad (2)$$

where A_{v1} is the affinity matrix for the view 1 patterns and A_{v2} the affinity matrix for just the view 2 patterns. $A_{v1}(i,j) = e^{\frac{-||(x_i^{(1)} - x_j^{(1)})||^2}{2\sigma_1^2}}$. Note that the product between the Gaussian weighted distances within each view is just the Gaussian weighted normalized distance between the two concatenated patterns (when considered as multi-view patterns).

Then we take the $p \times p$ matrix of w's and put it in a large $2p \times 2p$ matrix of the form

$$A_{sM-D} = \begin{bmatrix} 0_{p\times p} & W \\ W' & 0_{p\times p} \end{bmatrix}$$

where $0_{p\times p}$ represents a $p \times p$ matrix of zeros (and we will drop the subscript from here on for clarity). This matrix could then be considered an affinity matrix (for a bipartite graph) and given to the spectral clustering algorithm of (Ng et al., 2001). However note that the next step is to compute eigenvectors of the matrix

$$D^{-.5} A_{sM-D} D^{-.5}$$

where D is a diagonal matrix with $D(i,i) = \sum_j A_{sM-D}(i,j)$ (row sums of A_{sM-D}) which is equal

to (where D_{row} (D_{col}) is the diagonal matrix with diagonal entries equal to the row (column) sums of W)

$$\begin{bmatrix} D_{row}^{-.5} & 0 \\ 0 & D_{col}^{-.5} \end{bmatrix} \begin{bmatrix} 0 & W \\ W' & 0 \end{bmatrix} \begin{bmatrix} D_{row}^{-.5} & 0 \\ 0 & D_{col}^{-.5} \end{bmatrix}$$

but that matrix has the same eigenvectors as the matrix

$$\begin{bmatrix} D_{row}^{-.5} W D_{col}^{-1} W' D_{row}^{-.5} & 0 \\ 0 & D_{col}^{-.5} W' D_{row}^{-1} W D_{col}^{-.5} \end{bmatrix}$$

which has conjoined eigenvectors of each of the blocks $D_{row}^{-.5} W D_{col}^{-1} W' D_{row}^{-.5}$ and $D_{col}^{-.5} W' D_{row}^{-1} W D_{col}^{-.5}$ and these parts can be found efficiently together by computing the SVD of the matrix $L_W = D_{row}^{-.5} W D_{col}^{-.5}$. This trick is used in the co-clustering literature (Dhillon, 2001; Zha et al., 2001), but there the affinity submatrix W is derived simply from the term document matrix (or equivalent) not derived as a product of affinity matrices from different views [1]. The final clustering/segmentation is obtained from the top eigenvectors. There are several slightly different ways to cluster the values of this eigenvector. We use the prescription of Ng, Jordan and Weiss from the first page where Y is obtained as follows.

```
Av1 =exp(-distmatview1/(2*sigsq1));
Av2 =exp(-distmatview2/(2*sigsq2));
W=Av1*Av2;
Dtop=(sum(W'));
Dbot=(sum(W));
Lw=diag(Dtop.^(-.5))*W*diag(Dbot.^(-.5));
[U,S,V]=svds(Lw)
X=[U(:,1:numclusts);V(:,1:numclusts)];
Xsq=X.*X;
divmat=repmat(sqrt(sum(Xsq')'),1,numclusts);
Y=X./divmat;
```

Note that computing the SVD of the matrix $L_W = D_{row}^{-.5} W D_{col}^{-.5}$, gives two sets of eigenvectors, those of $L_W L_W'$ and those of $L_W' L_W$. The algorithm above concatenates these to form the matrix Y (as one would get if performing spectral clustering on the large matrix A_{sM-D}). This thus provides clusters for each view of each pattern. To get a cluster for the multi-view pattern, when both views are approximately equally reliable, the top p rows of the Y matrix can be averaged with the bottom p rows before the k-means step. If one view is significantly more reliable than the other, one can just use the Y entries corresponding to the more reliable view (The eigenvectors of $L_W L_W'$ reveal the clustering for the view 1 segments and the eigenvectors of $L_W' L_W$ for the view 2 segments). .

[1] It is possible to combine these ideas and use multiple views, each (or one) of which is a co-clustering

For comparison, we consider the patterns to be in the joint space given by the inputs in the two views. We call this algorithm **JOINT** In this case, we can simply use the standard spectral clustering algorithm to determine clusters. Note that in this case

$$\begin{aligned} A_{JOINT}(i,j) &= e^{-\frac{||(x_i - x_j)||^2}{2\sigma^2}} \\ &= e^{-\frac{||(x_i^{(1)} - x_j^{(1)})||^2 + ||x_i^{(2)} - x_j^{(2)})||^2}{2\sigma^2}} \\ &= A_{v1}(i,j)^{\frac{\sigma_1^2}{\sigma^2}} \cdot A_{v2}(i,j)^{\frac{\sigma_2^2}{\sigma^2}} \end{aligned}$$

Thus the affinity matrix for clustering in the joint space can be obtained by a componentwise product (Hadamard product or .* in Matlab) of the affinity matrices for the individual modalities. [As shown above, a person who ignored the multi-view structure of the data would use one σ^2 for all dimensions, however to give this algorithm the best chance we allowed the use of different σ_1^2 and σ_2^2.] In other words, we actually used $A_{JOINT}(i,j) = A_{v1}(i,j) \cdot A_{v2}(i,j)$

We also compare our algorithm to one where the affinity matrices of the two individual modalities are added. This idea is mentioned in (Joachims, 2003) for the semi-supervised case. We call this algorithm **SUM**. case $A_{SUM}(i,j) = A_{v1}(i,j) + A_{v2}(i,j)$.

2. Theoretical Comparison of Algorithms

As discussed in (Ng et al., 2001), the simplest case for spectral clustering algorithms, is when the affinity matrix is block diagonal. One can easily see that the following statements are true.

Statement 1: For consistent block diagonal A_{v1} and A_{v2}, all 3 algorithms preserve block diagonal form.

Statement 2: If the affinity matrix in one view is block diagonal but random in the other then only **JOINT** results in a block diagonal affinity matrix.

When is the **sM-D** algorithm better than the **JOINT** algorithm? Figure 2 shows a simple example that shows that clustering in the joint space and M-D style algorithms are not identical. The datapoints are numbered for the purposes of discussion. Consider in particular the membership of the circled datapoint (4). The **sM-D** algorithm would cluster it with datapoints 1, 2 and 3. The **JOINT** algorithm is much more likely (over a wider range of parameters and noise levels) to cluster datapoint 4 with datapoints 5,6,7 and 8. To quantify this effect, we constructed an affinity matrix for each view from the example in Figure 2 and

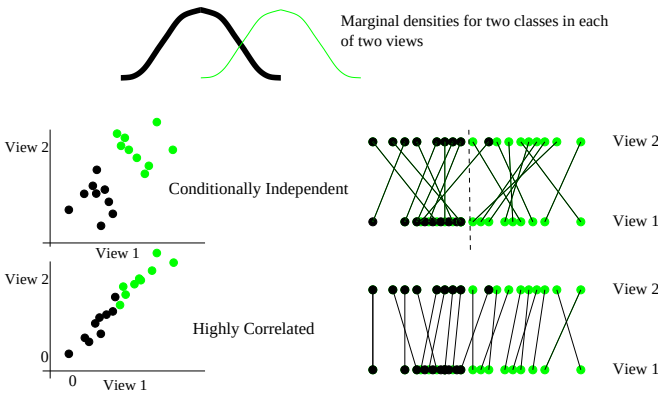


Figure 1. **A** Consider two classes of patterns with two 1-D views. The top of the figure represents the density for the two pattern classes (bold and unbold) in View 1. Assume the marginal densities in View 2 are similar. An example scatterplot is shown on the left of the figure. On the right,the same data is presented in a different format. Here lines are joined between co-occurring patterns in the two imaginary 1-D views/modalities (as shown at top). The M-D algorithm wants to find a partition that crosses the fewest lines. Two cases are shown for when the views are conditionally independent or highly correlated. In the conditionally independent case, there is a clear non-trivial optimal cut. In the correlated case,there are many equally good cuts and the M-D algorithm will not perform well in this case.

ran spectral clustering algorithms on noisy versions of these affinity matrices for varying levels of noise and varying cross-cluster strength m.

$$A_{v1} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & m & 0 & m & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & m & 0 & m & 0 \\ 0 & m & 0 & m & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & m & 0 & m & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

$$A_{v2} = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & m & m & 0 & 0 \\ 0 & 0 & 1 & 1 & m & m & 0 & 0 \\ 0 & 0 & m & m & 1 & 1 & 0 & 0 \\ 0 & 0 & m & m & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

The cross-cluster strength m relates to the relative spacing between the two clusters with respect to the σ^2 parameter in the spectral clustering algorithm. The results are robust over a broad range of noise levels (10^{-19} to 10^{-1}). For m=0, all three algorithms correctly cluster nodes 1-4 and 5-8. However for $m \geq .05$ the **JOINT** method breaks down and groups one of nodes 4 or 5 with the wrong cluster. The **SUM** algorithm breaks down for $m \geq .81$ and the **sM-D** algorithm continues to group appropriately until $m = .92$. Figure 3 explains these results graphically as well as showing the actual (pre-noise) matrices computed W_{sM-D},A_{SUM}, and A_{JOINT}.

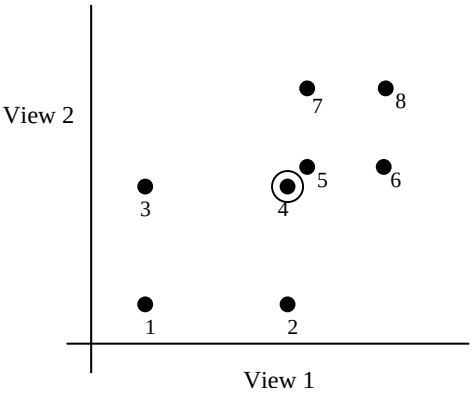


Figure 2. A simple example that would give a different solutions clustered in the joint space **JOINT**, than if the **sM-D** algorithm was used.

3. Clustering results with the course webpage dataset

This dataset consists of two views of web pages. The first view consists of text on the web page and the second view consists of text on the links to the web page (Blum & Mitchell, 1998). We use the six class (course, department, faculty, project, staff, student) version in (Bickel & Scheffer, 2004) consisting of tfidf (term frequency inverse document frequency - where a document is stored as a vector of weighted words. Tfidf weights words more if they occur more in a document and downweights words that occur often in the full dataset) vectors without stemming. Patterns were normalized within each view so that squared distances reflected the commonly used cosine similarity measure.

We use the average entropy error metric of (Bickel & Scheffer, 2004)

$$E = \sum_{i=1}^{k} \frac{m_i(-\sum_j p_{ij} \log_2(p_{ij}))}{m}$$

where p_{ij} is the proportion of cluster i that is from mixture component j, m_i is the number of patterns in class i and m is the total number of patterns. On this dataset, with this error measure, perfect agreement would result in $E = 0$, everybody in the same

class would give $E = 2.219$ (and equal size clusters with probability measurement equal to the base class probabilities also gives $E = 2.2$).

We first compared the algorithms on the full dataset. To do this we first searched for good σ_1 and σ_2 from clustering in the individual views.

We found that (with the proper normalization), the joint method worked slightly better(E=1.64) than the sum (E=1.70) and m-d version (E=1.66) (standard error estimates are provided later when 90% of the data is used). For comparison, Bickel and Scheffer report measures on the same error measure (with 6 clusters) of approximately[2] 1.73 (multi-view) and 2.03 (single view) for their mixture-of-multinomials EM algorithm and approximately 1.97 (multi-view) and 2.07 (single view) for their spherical k-Means algorithm (Bickel & Scheffer, 2004).

As mentioned, when computing the SVD of the matrix $L_W = D_{row}^{-.5} W D_{col}^{-.5}$, one gets two sets of eigenvectors, those of $L_W L_W'$ and those of $L_W' L_W$ and for equally reliable views, the Y matrices can be averaged before the k-means step. For this dataset however, view 1 is significantly more reliable than view 2 and we obtain improved performance by simply using the eigenvectors from view 1.

The main advantage of our algorithm is that it can allow us to combine sources of information with different numbers of views. To see this, remember that the affinity submatrix W is in terms of how similar pairs are to co-occurring pairs. Thus a single view pattern $x_i^{(1)}$ from view 1 does not contribute to the library of paired occurrences but can still be related to patterns $x_j^{(2)}$ in view 2 according to how similar the pair $(x_i^{(1)}, x_j^{(2)})$ is to the set of co-occurring patterns. Thus we can construct a full bipartite affinity matrix between patterns from view 1 and those from view 2 using equation 2 where p sums over only the paired patterns. This results in a matrix multiplication of the form $A_{v1} \times A_{v2}$ where this time A_{v1} is $(p+m) \times p$ dimensional and A_{v2} is $p \times (p+n)$ dimensional where there are p co-occurring (multi-view) patterns and m patterns with only view 1 and n patterns with only view 2 (see Figure 4). Note that the bottom right quadrant of the resulting W matrix computes the affinity between an unpaired view 1 pattern and an unpaired view 2 pattern according to the sum of the affinities between this pair $(x_{p+i}^{(1)}, x_{p+j}^{(2)})$ and each of the set of observed pairs $\{(x_1^{(1)}, x_1^{(2)}), ...(x_p^{(1)}, x_p^{(2)})\}$. The affinity between two pairs of patterns is the product between the affinity between each view of each pattern.

In this case we use the eigenvectors of $L_W L_W'$ to find the clusters for both the paired and view 1 data and must use the eigenvectors of $L_W' L_W$ to find the clusters for the data that only has view 2.

For comparison, we consider two other alternatives for clustering data that consists of some multi-view patterns and some single view patterns.

Alternative A using JOINT: cluster only the p patterns consisting of $x_i^{(1)}$ and $x_j^{(2)}$ concatenated in the joint space. Spectral clustering will give clusters for these patterns. To report clusters for the $m+n$ unpaired patterns, report the cluster of the nearest same view paired pattern of the pattern.

Alternative B: cluster the patterns from each view separately. In this case the pairing information is lost.

Results for different values of p are reported in Tables 1 thru 3. Table 1 shows that there is a very slight but significant performance advantage for the multi-view patterns using Alternative A when 2084 (90%) of the patterns have both views, but that Alternatives B and our sM-D method perform significantly better on the patterns that only have values for view 1 and our sM-D method performs significantly better than both alternatives for patterns that only have values for view 2. When only 1158 (50%) of the patterns are provided with two views, the sM-D algorithm performs significantly better in all categories. Table 3 shows how the sM-D algorithm varies for different numbers of paired patterns. (The slight improvement in clustering performance (with increased variance) for the paired view data in the 50% paired case is likely due to an increased chance of not including inappropriate pairs in the paired dataset. Performance decreases with non independent sources of information have been observed with the non-spectral M-D algorithm. If leaving out some data vectors increases the independence between views, we would expect improved performance.) Performance for the single view data is seen to decrease gradually with less paired training data.

One value of an algorithm that can train with multi-view data and report data for single-view data would be when the single-view data arrive at a later time. We are working on using the Nystrom approximation (Charless Fowlkes & Malik, 2004) for such out of sample estimates. This would allow us to train with paired data and provide cluster labels for later unpaired data.

[2] estimated from their graph

Table 3. Average Entropy for **sM-D** for varying amount of two-view data. (See Table 1 for an explanation of terms)

	2084 (90%)	**1621 (70%)**	**1158 (50%)**	**694 (30%)**	**231 (10%)**
both views	$1.68 \pm .003$	$1.66 \pm .006$	$1.64 \pm .01$	$1.68 \pm .01$	$1.76 \pm .03$
View 1 only	$1.63 \pm .02$	$1.66 \pm .01$	$1.66 \pm .006$	$1.67 \pm .01$	$1.73 \pm .02$
View 2 only	$1.83 \pm .02$	$1.91 \pm .01$	$1.95 \pm .006$	$1.97 \pm .01$	$2.00 \pm .01$

Table 1. Average Entropy where 2084 (90%) of the Patterns have both views. Alt. is an abbreviation for Alternative. All values are given $\pm$ 1 standard error of the mean over 10 runs. The both view line refers to the error for patterns that had two views, View 1 only refers to errors on patterns that consisted of only view 1 and View 2 only refers to errors on patterns that consisted of View 2 only. All errors are using the average entropy error measure

	Alt. A	**Alt B**	**sM-D**
both views	$1.66 \pm .003$	$1.68 \pm .002$	$1.68 \pm .003$
View 1 only	$1.83 \pm .02$	$1.64 \pm .02$	$1.63 \pm .02$
View 2 only	$1.95 \pm .02$	$2.04 \pm .003$	$1.83 \pm .02$

Table 2. Average Entropy where 1158 (50%) of the Patterns have both views. (See Table 1 for an explanation of terms)

	Alt. A	**Alt. B**	**sM-D**
both views	$1.67 \pm .01$	$1.69 \pm .002$	$1.64 \pm .01$
View 1 only	$1.90 \pm .02$	$1.68 \pm .006$	$1.66 \pm .006$
View 2 only	$2.04 \pm .006$	$2.04 \pm .003$	$1.95 \pm .006$

4. Discussion

We have shown that spectral clustering is competitive in the webpage domain and have introduced a novel multi-view spectral clustering algorithm. While it performs slightly worse than properly normalized joint spectral clustering in the full webpage domain, the difference is small and the sM-D algorithm has the major advantage that it allows single view patterns to benefit from the paired dataset. This allows one to incorporate all available information to form the best clusters when there is lots of single-view data to be clustered.

The spectral Minimizing-Disagreement algorithm was motivated by the earlier Minimizing-Disagreement algorithm(de Sa, 1994a; de Sa & Ballard, 1998) and we believe that of the different ways of spectral clustering with multiple views, sM-D best incorporates the idea of minimizing the disagreement of the outputs of two classifiers (clusterers). In the appendix we reproduce an argument from (de Sa, 1994b; de Sa & Ballard, 1998) that motivates, in the 1-D case, the minimizing-disagreement approach as an approximation to minimizing misclassifications.

The spectral implementation of the Minimizing-Disagreement idea shares many of the advantages and disadvantages of other spectral techniques. It does not work as well for multi-class classifications as for binary. It is quick to implement and run (with sparse matrices) and has a guaranteed global optimum which is related by a relaxation to the desired optimum.

Putting the algorithm in the framework of graph partitioning should allow easier comparison and combination with results from clustering in the joint space. Also it should be straightforward to modify the algorithm to incorporate some labeled data so that the algorithm can be used in a semi-supervised way. We are currently exploring these avenues.

Appendix: Minimizing Disagreement as an Approximation to Minimizing Misclassifications

The M-D algorithm to minimize the disagreement corresponds to the LVQ2.1 algorithm(Kohonen, 1990) except that the "label" for each view's pattern is the hypothesized output of the other view. To understand how making use of this label, through minimizing the disagreement between the two outputs, relates to the true goal of minimizing misclassifications in each view, consider the conditionally independent (within a class) version of the 2-view example illustrated in Figure 5. In the supervised case (Figure 5A) the availability of the actual labels allows sampling of the actual marginal distributions. For each view, the number of misclassifications can be minimized by setting the boundaries for each view at the crossing points of their marginal distributions.

However in the self-supervised system, the labels are not available. Instead we are given the output of the

other view. Consider the system from the point of view of view 2. Its patterns are labeled according to the outputs of view 1. This labels the patterns in Class A as shown in Figure 5B. Thus from the actual Class A patterns, the second view sees the "labeled" distributions shown. Letting a be the fraction of Class A patterns that are misclassified by view 1, the resulting distributions of the real Class A patterns seen by view 2 are $(1-a)P(C_A)p(x_2|C_A)$ and $(a)P(C_A)p(x_2|C_A)$.

Similarly Figure 5C shows View 2's view of the patterns from class B (given View 1's current border). Letting b be the fraction of Class B patterns misclassified by view 1, the distributions are given by $(1-b)P(C_B)p(x_2|C_B)$ and $(b)P(C_B)p(x_2|C_B)$. Combining the effects on both classes results in the "labeled" distributions shown in Figure 5D. The "apparent Class A" distribution is given by $(1-a)P(C_A)p(x_2|C_A) + (b)P(C_B)p(x_2|C_B)$ and the "apparent Class B" distribution by $(a)P(C_A)p(x_2|C_A)+(1-b)P(C_B)p(x_2|C_B)$. The crossing point of these two distributions occurs at the value of x_2 for which $(1-2a)P(C_A)p(x_2|C_A) = (1-2b)P(C_B)p(x_2|C_B)$. Comparing this with the crossing point of the actual distributions that occurs at x_2 satisfying $P(C_A)p(x_2|C_A) = P(C_B)p(x_2|C_B)$ reveals that if the proportion of Class A patterns misclassified by view 1 is the same as the proportion of Class B patterns misclassified by view 1 (i.e. $a = b$) the crossing points of the distributions will be identical. This is true even though the approximated distributions will be discrepant for all cases where there are any misclassified patterns ($a > 0$ OR $b > 0$). If $a \approx b$, the crossing point will be close.

Simultaneously the second view is labeling the patterns to the first view. At each iteration of the algorithm both borders move according to the samples from the "apparent" marginal distributions.

Acknowledgments

Many thanks to Ulf Brefeld, Tobias Scheffer, Steffen Bickel, and Anjum Gupta for kindly sending their processed datasets and to Marina Meila and Deepak Verma for providing a great library of spectral clustering code at http://www.cs.washington.edu/homes/deepak/spectral/library.tgz. Finally, warm thanks to Patrick Gallagher, Jochen Triesch, Serge Belongie and three anonymous reviewers for helpful comments. This work is supported by NSF CAREER grant 0133996.

References

Bickel, S., & Scheffer, T. (2004). Multi-view clustering. *Proceedings of the IEEE International Conference on Data Mining.*

Blum, A., & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. *Proceedings of the Eleventh Annual Conference on Computational Learning Theory (COLT-98)* (pp. 92–100). Madison, WI.

Charless Fowlkes, Serge Belongie, F. C., & Malik, J. (2004). Spectral grouping using the nystrom method. *IEEE Transactions Pattern Analysis and Machine Intelligence, 26.*

de Sa, V. R. (1994a). Learning classification with unlabeled data. *Advances in Neural Information Processing Systems 6* (pp. 112—119). Morgan Kaufmann.

de Sa, V. R. (1994b). *Unsupervised classification learning from cross-modal environmental structure.* Doctoral dissertation, Department of Computer Science, University of Rochester. also available as TR 536 (November 1994).

de Sa, V. R., & Ballard, D. H. (1998). Category learning through multimodality sensing. *Neural Computation, 10,* 1097–1117.

Dhillon, I. S. (2001). Co-clustering documents and words using bipartite spectral graph partitioning. *KDD 2001.* San Francisco, CA.

Joachims, T. (2003). Transductive learning via spectral graph partitioning. *Proceedings of the 20th International Conference on Machine Learning (ICML 2003).*

J.Shi, & Malik, J. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 888–905.

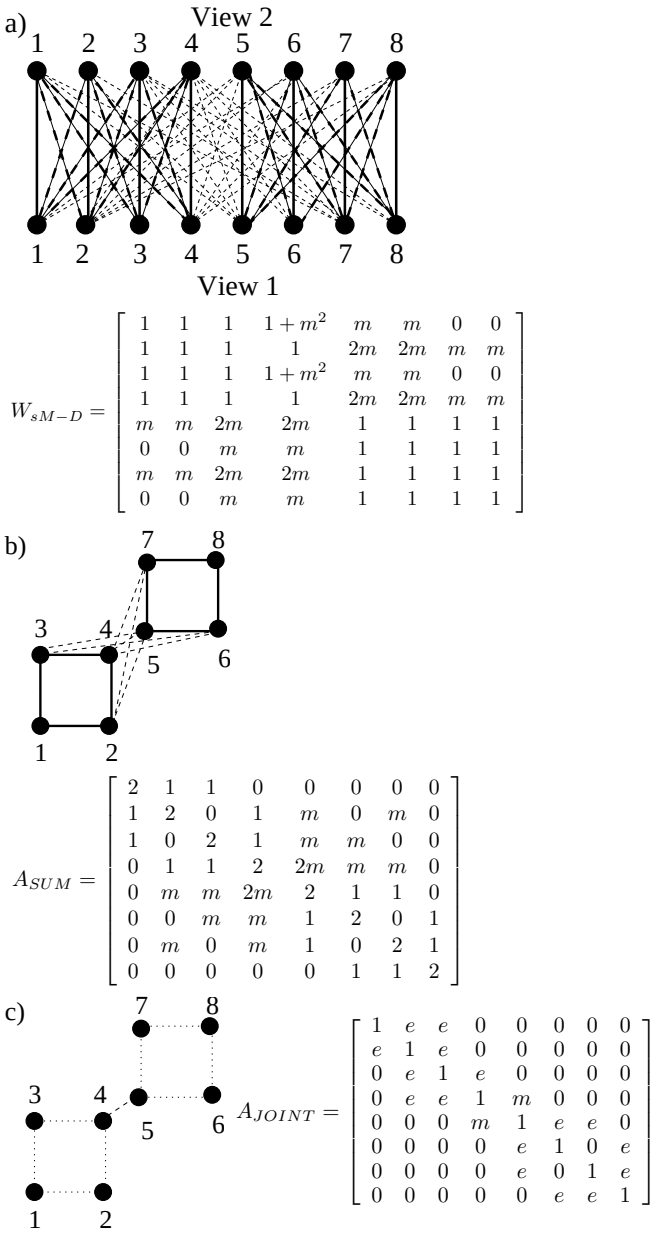
Kohonen, T. (1990). Improved versions of learning vector quantization. *IJCNN International Joint Conference on Neural Networks* (pp. I–545–I–550).

Meila, M., & Shi, J. (2001). Learning segmentation by random walks. *Advances in Neural Information Processing Systems 13.*

Ng, A. Y., Jordan, M. I., & Weiss, Y. (2001). On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems 14.*

Shi, J., & Malik, J. (1998). Motion segmentation and tracking using normalized cuts. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition.*

Zha, H., Ding, C., & Gu, M. (2001). Bipartite graph partitioning and data clustering. *CIKM '01.*



$$W_{sM-D} = \begin{bmatrix} 1 & 1 & 1 & 1+m^2 & m & m & 0 & 0 \\ 1 & 1 & 1 & 1 & 2m & 2m & m & m \\ 1 & 1 & 1 & 1+m^2 & m & m & 0 & 0 \\ 1 & 1 & 1 & 1 & 2m & 2m & m & m \\ m & m & 2m & 2m & 1 & 1 & 1 & 1 \\ 0 & 0 & m & m & 1 & 1 & 1 & 1 \\ m & m & 2m & 2m & 1 & 1 & 1 & 1 \\ 0 & 0 & m & m & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$A_{SUM} = \begin{bmatrix} 2 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 2 & 0 & 1 & m & 0 & m & 0 \\ 1 & 0 & 2 & 1 & m & m & 0 & 0 \\ 0 & 1 & 1 & 2 & 2m & m & m & 0 \\ 0 & m & m & 2m & 2 & 1 & 1 & 0 \\ 0 & 0 & m & m & 1 & 2 & 0 & 1 \\ 0 & m & 0 & m & 1 & 0 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 2 \end{bmatrix}$$

$$A_{JOINT} = \begin{bmatrix} 1 & e & e & 0 & 0 & 0 & 0 & 0 \\ e & 1 & e & 0 & 0 & 0 & 0 & 0 \\ 0 & e & 1 & e & 0 & 0 & 0 & 0 \\ 0 & e & e & 1 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & m & 1 & e & e & 0 \\ 0 & 0 & 0 & 0 & e & 1 & 0 & e \\ 0 & 0 & 0 & 0 & e & 0 & 1 & e \\ 0 & 0 & 0 & 0 & 0 & e & e & 1 \end{bmatrix}$$

Figure 3. The resulting graphs (and matrices) resulting from the three algorithms a) sM-D b)SUM c) JOINT. applied to the matrices A_1 and A_2 above. The light lines correspond to weights of m and 2m and the dark lines correspond to weights of 1 and $1 + m^2$. In a) the solid lines correspond to co-occurrence lines and the dashed lines, inferred relationships. In c) the faint dotted lines only arise due to noise. The e's in the matrix result only from the noise and would be different small numbers at each spot). **Each algorithm tries to find the smallest normalized cut in its graph**

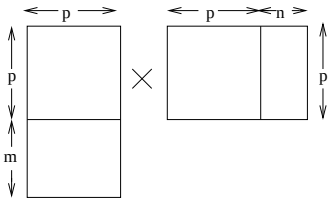


Figure 4. A graphical view of the matrix multiplication required to compute W when there are p patterns with both views, m patterns with only view 1 and n patterns with only view 2.

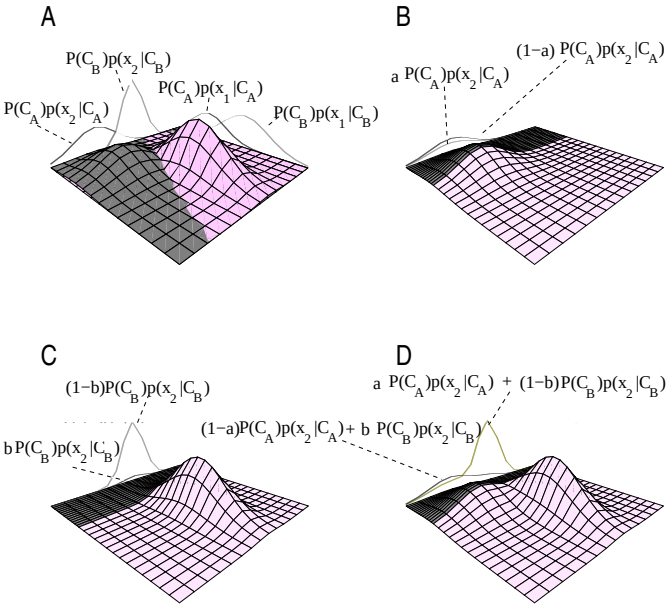


Figure 5. **An example joint and marginal distribution for a conditionally independent example problem**. (For better visualization the joint distribution is expanded vertically twice as much as the marginal distributions.) The darker gray represents patterns labeled "A", while the lighter gray are labeled "B". (A) shows the labeling for the supervised case. (B) shows the labeling of Class A patterns as seen by view 2 given the view 1 border shown. ***a*** represents the fraction of the Class A patterns that are misclassified by view 1. (C) shows the labeling of Class B patterns as seen by view 2 given the same view 1 border. ***b*** represents the fraction of the Class B patterns that are misclassified by view 1. (D) shows the total pattern distributions seen by view 2 given the labels determined by view 1. These distributions can be considered as the labeled distributions on which view 2 is performing a form of supervised learning. (However it is more complicated as view 1's border is concurrently influenced by the current position of view 2's border). See text for more details.