Comparison of Spectral Clustering Methods

Deepak Verma
Department of CSE
University of Washington
Seattle, WA 98195-2350
deepak@cs.washington.edu

Marina Meilă
Department of Statistics
University of Washington
Seattle, WA 98195-4322
mmp@stat.washington.edu

Abstract

We take apart, combine and compare on real and artificial data the features of the four best-known spectral clustering algorithms. We find that the algorithms behave more similarly then expected, especially if the data are near a case called *perfect*, where three of the algorithms are equivalent.

1 Introduction

Spectral clustering methods have recently grown in popularity. Several new algorithms [1, 2, 3, 4] and applications [2] have been published. In this paper we aim to put the existing algorithms in perspective, by comparing them to each other both theoretically and experimentally. As far as we know, this is the first comparison dedicated to spectral algorithms for general purpose clustering; [5] did a similar comparison between spectral algorithms for image segmentation. We will try to answer the following questions: how different are the various spectral algorithms in theory and in practice? Do the difference between algorithms recommend them for specific applications? What features of the algorithms are most important for their success? In order to answer the last question fairly, from the four basic algorithms that were selected for comparison, we created a set of about 30 by interchanging their components (typically, by pairing the first stage of one algorithm with the second stage of several other algorithms). From this enlarged set, 19 combinations were filtered into this paper.

The next section presents the four basic algorithms that we selected from the literature. Then, in section 3 we analyze their behavior under perfect conditions. This theoretical comparison will unveil some strong similarities between apparently different algorithms. In section 5 we undertake a comparison in experiments, to establish (1) how the ideal performance degrades under noise conditions, and (2) if there are significant differences on real data sets. Section 6 discusses the findings and 7 concludes the paper.

2 Algorithms

This section briefly explains the various spectral algorithms that we compared. First let us introduce some notation. The set of data points to be clustered will be denoted by I, with $|I| = n$. For each pair of points $i, j \in I$ a similarity $S_{ij} = S_{ji} \geq 0$ is given. The similarities S_{ij} can be viewed as weights on the *undirected* edges ij of a graph G over I. The matrix $S = [S_{ij}]$ plays the role of a "real-valued" adjacency matrix for G. Let $d_i = \sum_{j \in I} S_{ij}$ be called the *degree* of node i, and D be the diagonal matrix with d_i

as its diagonal. P denote the corresponding stochastic matrix $P = D^{-1}S$. A clustering $\mathcal{C} = \{C_1, C_2, \ldots C_K\}$ is a partitioning of I into the nonempty mutually disjoint subsets $C_1, \ldots C_K$. In the graph theoretical paradigm a clustering represents a *multiway cut* in the graph G.

All the algorithms use eigen vectors of a matrix (derived from S) to map the original data points into the K dimensional vectors $\{\gamma_1, \gamma_2, \ldots, \gamma_n\}$ of the *spectral domain* R^K. These vectors are then clustered in standard clustering algorithms for Euclidean spaces.

There are two kinds of algorithms that we considered. The *multiway* algorithms which directly split the points into K clusters and the *recursive bipartitioning* algorithms which recursive split the points into two partitions until K partitions are obtained. Amongst these we picked the two best known algorithms in each category and compared the four of them. Here are the four algorithms

The Meila-Shi (Multicut) algorithm: This algorithm was suggested in [4].

```
1: Compute the stochastic matrix $P = D^{-1}S$.
2: Compute $v^1, v^2, \ldots, v^K$ the eigenvectors of $P$ corresponding to the $K$ largest eigenvalues.
3: Form the matrix $V_{n \times k} = [v^1, v^2, \ldots, v^K]$ with these vectors as columns.
4: Cluster the rows of $V = [\gamma_1, \gamma_2, \ldots, \gamma_n]^T$ as points in a $K$-dimensional space.
```

The Ng,Jordan,Weiss (NJW) algorithm:[3]The general framework is same as above except for the following changes: a) In step 1 The Laplacian $L = D^{-\frac{1}{2}}SD^{-\frac{1}{2}}$ is used instead of P. Its top K eigen vectors form the $n \times K$ matrix U. b) The rows U are normalized to have unit norm. The resulting matrix is denoted by Y.

We used the generalized eigen system $Sx = \lambda Dx$ for NJW and Multicut instead of the formulations used in [3, 2] for better numerical stability.

The Shi and Malik (SM) algorithm: [2] isa recursive bipartitioning algorithm.

```
1: Compute the stochastic matrix $P = D^{-1}S$
2: Calculate $v^2$, the eigenvector corresponding to the second largest eigenvalue.
3: Sort elements of $v^2$ in increasing order. Now partition this *ordered* list into two parts such that the *normalized cut* ([2]) over the two partition is minimized.
4: Recursively partition the two clusters till $K$ clusters are produced.
```

The Kannan, Vempala and Vetta (KVV) algorithm: ([1]), Algorithm II [1]) is similar to SM but based on conductance. The differences are: (1)in step 3. the ordered list it split so as to minimize the conductance instead of normalized cut; (2) in step 4 the next cluster chosen is also based on conductance ([1]). We implemented two slightly different versions based on how the P is calculated for a subcluster called `kvv_add` and `kvv_mult`.

3 Theoretical Results

Here we compare the algorithms with respect to what we call their *perfect points*, values of S for which these spectral methods are supposed to perform well. We will show that, even though apparently the four algorithms are different, some of them are very similar near the perfect points.

The perfect S When each of the clusters is reduced to a point ($\gamma_i = \gamma_j \; \forall i, j \in C_s \; \forall s$) by the spectral mapping we say that S is **perfect** for the respective algorithm. A vector $v = [v_1, v_2, \ldots, v_n]^T$ is **piecewise constant** (*PC*) w.r.t a clustering Δ iff $v_i = v_j$ whenever i, j are in the same cluster.

A matrix S is called **block diagonal** (BD) w.r.t. a clustering Δ iff $S_{ij} = 0$ whenever i and

[1] We tried " Algorithm I" as well but it did not give good performance. "Algorithm III" assumes a special form of the affinity matrix that prevents it to be used in our comparisons.

j belong to different clusters. It can be easily shown ([1, 2, 4, 3]) that block diagonal S is perfect for all the methods.

Block stochastic P Let P be a stochastic matrix. P is **block stochastic** [4] (BS) w.r.t. a clustering $\Delta = \{C_1, \dots C_K\}$ iff for all $s, s' = 1, \dots K$ the sums $P_{is} = \sum_{j \in C_s} P_{ij}$ are equal for all $i \in A_{s'}$ and the matrix $R = [P_{ss'}]$ (with $P_{ss'} = \sum_{j \in A_{s'}} P_{ij}, i \in A_s$) is non-singular. A block stochastic P is guaranteed ([4]) to have some K eigenvectors PC w.r.t. Δ. In the rest of the paper we will assume that the PC vectors of P are always the top K eigenvectors. So from now on we will say that a matrix P has K **piecewise constant vectors** (PCE), or is block-stochastic w.r.t. Δ when its top K eigenvectors are piecewise constant. Thus, if a P has PCE, the corresponding S is perfect for the Multicut algorithm. In this subsection we examine the behavior of other algorithms on S, assuming that P is BS; U, V, Y have the same meaning as in Section 2.

Proposition 1 *Let $A_{n \times k}$ be a real matrix and $D_{n \times n}$ be a diagonal matrix such that $A^T DA = I_{k \times k}$ and that A has atmost k unique rows. Then A is guaranteed to have* exactly *k unique* orthogonal *rows. (proved in [6]).*

Consider a BS P and the corresponding V. Since $SV = \Lambda DV, V^T DV = I$. So the above proposition implies that whenever P is perfect for Multicut , all the clusters in the spectral domain are unique and orthogonal (but not orthonormal).

It can be easily shown that Y in the NJW algorithm can also be obtained by normalizing rows of V (from Multicut) to have unit length (because $U = D^{-\frac{1}{2}}V$). By the above proposition, we have that the rows of Y are also perfectly clustered with the clusters orthogonal to each. other and hence BS P is perfect for NJW as well. This generalizes the result in [3] where this property was shown for block diagonal case and shows that a perfect S for Multicut is also perfect for NJW . This shows that NJW will work well in a much wider range of cases then previously believed. The reverse can also be shown though the proof is slightly more involved [6]. It follows that:

Theorem 2 *Whenever the S is perfect for Multicut it is perfect for NJW and vice versa.*

In other words, Multicut and NJW are equivalent when S is perfect, although they use different spectral mappings; NJW maps the clusters to orthonormal vectors, while Multicut maps them to orthogonal vectors of different lengths. Away from the perfect S the behavior of the two algorithms may differ. We will investigate this in the experiments section.

Recursive Algorithms We have shown the equivalence of for NJW and Multicut block stochastic P. Now we examine the behavior of the recursive algorithms in the same situation.

Take the case when we are splitting the points $I' \subseteq I$ into two clusters. (At the top level or in any of the recursive steps). Let the P' that we have (*for I'*) be block stochastic w.r.t Δ'. In that case the second eigen vector would be PC w.r.t. to the Δ so when we reorder the points based on this eigen vector we have the points in the right *order*. That is, if we chose the right point to partition , none of the clusters in Δ would be split. Also the P'', P''' for these two split cluster be block stochastic (w.r.t to the two parts of Δ) setting the optimal stage for the recursive sub steps.

The criteria that the two algorithms use for splitting (minimum Ncut or conductance) do not ensure that this optimal point of partition is chosen, unless P is block diagonal (then Ncut and conductance are zero at only these positions). So, for the block diagonal case, all algorithms are equivalent, but we cannot say what happens in the general BS case. As it will turn out from the experiments, one of the algorithms, SM , behaves exactly like the multiway algorithms, while KVV behaves differently.

The above remarks also suggest an alternative to the SM and KVV algorithms in which one chooses the partition based on largest *difference* in the sorted eigenvector. We explored this in our experiments.

4 Datasets

Artificial Block Stochastic Dataset: The first dataset is an artificial S such that the corresponding P is block stochastic. There are five true clusters of sizes $10, 20, 30, 20, 20$. The purpose of this dataset is to demonstrate the stability of spectral algorithms to noise on a case where linkage algorithms do not perform so well and spectral do. We made this example difficult on purpose: (1) the degree D_i have a very high range (max D_i/min D_i 2000), (2) the smallest degrees are in the smallest cluster, making this easy to “lose” in noise, (3) the probability mass in the off-diagonal blocks of P is large. This example will thus test the spectral algorithms under extreme conditions. More about this S is available in [6].

Gene Expression Data: The second dataset is a gene expression dataset [7] for which we had true clustering and clustering results from a gaussian mixture model base algorithms ([8]). It consists of fluctuation of gene expression data over two cellcycles. The objective is to cluster these corresponding to the five phases of the cell cycle. There are two kinds of pre processing done leading to two different datasets. We call the first dataset cellcycle (Log normalized) and the second cellcycle-std (Standardized). We followed [8] and computed similarity as the correlation coefficients between the gene expression levels of the different genes adding 1 to make the similarity matrix positive.

NIST Handwritten Digits: We used the data set and preprocessing of [9] obtaining an 8x8 matrix of integer in the range 0..16 for each digit. We further down sampled the dataset to 100 elements per digit giving a total of thousand 64 dimensional points and 10 clusters. We call this dataset digit1000. We took aside from this set the data for the digits that were easier to distinguish (digits 0,2,4,6,7) creating another dataset called digitFive1000. The similarity for these data was computed as the *AffinityMatrix* with $\sigma = 10$.

5 Experimental Setup

Algorithms: To exactly distinguish between the effects of the various components of the spectral algorithms we implemented a whole range of algorithms containing most of the variations of the algorithms mentioned above, plus a few new ones.

The list of all the algorithms implemented is shown in table 1. In the spectral domain the similarity matrix S^γ (if required) is obtained by computing the *AffinityMatrix* with $\sigma = 0.2$ ($S_{ij} = \exp(-||x_i - x_j||^2/2\sigma^2)$)This choice was straight forward in case of NJW algorithm as points lie on a unit sphere. We used the same value for Multicut as well.

The anchor ([10]),ward ([11]),kmeans refer to the respective algorithms applied after the spectral mappings. In kmeans we performed kmeans with 5 runs of initializing with orthogonal centers and 20 runs initialized with random centers (and chose the one with minimum distortion). This seems to give much better results than using the simple kmeans alone.

We also had the intuition that the spectral methods might be more effective in clustering points *after* mapping S^γ itself into another spectral domain. To explore that possibility we implemented the *double spectral* methods like ang_mcut_ward in which the first we map the points in the ang spectral domain and then those points in the mcut spectral domain, finally grouping them using the ward method.

Performance evaluations To evaluate an algorithm we compared the clustering it produced to the “true” clustering using the Variation of Information (VI) metric. ([12]). We chose VI because its good properties, in particular its scale invariance properties which make our

Table 1: List of Algorithms: The first word denotes the spectral mapping used and the next the postprocessing. We would use `ang` to the denote the NJW algorithm and `mcut` to denote Multicut algorithm.

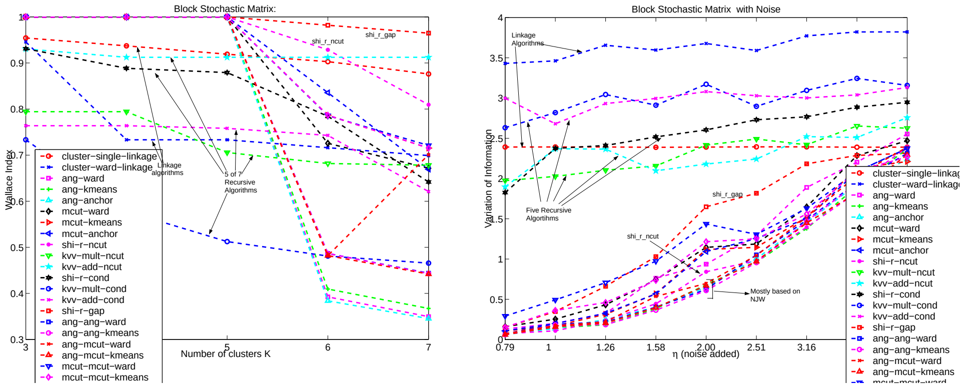
Linkage	`single_linkage`	`ward_linkage`	
Recursive	`shi_r_ncut`	`kvv_mult_ncut`	`kvv_add_ncut`
	`shi_r_cond`	`kvv_mult_cond`	`kvv_add_cond`
	`shi_r_gap`		
Multiway	`ang_ward`	`ang_kmeans`	`ang_anchor`
	`mcut_ward`	`mcut_kmeans`	`mcut_anchor`
Doubly Spectral	`ang_ang_ward`	`ang_ang_kmeans`	`ang_mcut_ward`
	`ang_mcut_kmeans`	`mcut_mcut_ward`	`mcut_mcut_kmeans`

comparisons more meaningful. In the extended paper we used also another metric, arriving to similar results.

The experiments with the artificial S had two goals: (1) to confirm the theoretical predictions for the perfect S, and (2) to study the algorithms' behavior in noise around the perfect S. The noise added was symmetric, independent, $noise_{ij} \sim \text{uniform}(0,1) \times \eta \times \sqrt{D_i \times D_j}/n$ with the magnitude $\log_{10} \eta = -0.1, 0, \ldots 0.7$. We ran all clustering allgorithms 10 times for each level of the noise, with K from 3 to 7.

The experiments on real data were repeated 5 times for each K, with different random initializations for the postprocessing stage (were it applied); K ranged ±2 around the true value for each data set.

6 Results and Discussion

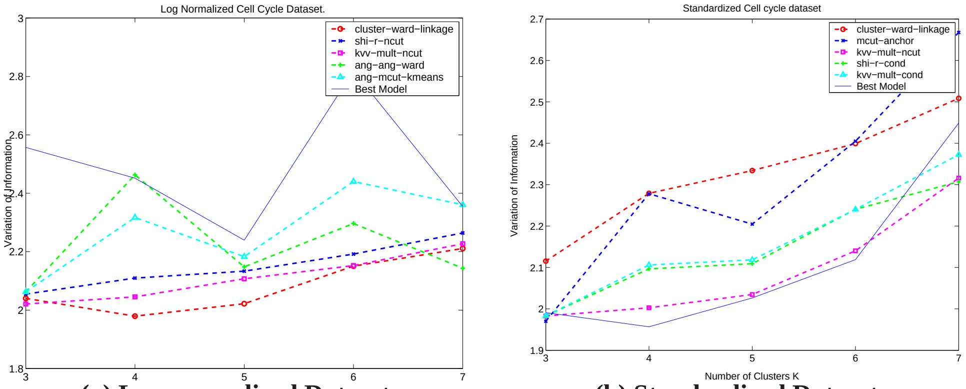


(a) Varying K. No noise **(b) True K. Noise added.**

Figure 1: **Block Stochastic Dataset**. a) Performance on Wallace index of spectral algorithms on Block Stochastic Datasets. b) Performance on Variation of Information in presence of noise

The results of of various algorithms on the block stochastic matrix are presented in figure 1. The first graph (a) shows the Wallace index ([13]) of clustering produce w.r.t to the "true" clustering given various K as input and no noise. The wallace index would have a value of 1 in case the clustering produced does not split the true clusters. i.e. two points which were in the same cluster in the true clustering are in the same cluster in the clustering produced. The graph illustrates a lot of points that in accordance with the theoretical predictions. First of all, the multiway spectral methods, irrespective of the post processing step perform perfectly when $K \leq K_{true} = 5$. This is to be expected as all the points in a single cluster are mapped to the same point in the spectral domain. Also most of the recursive spectral algorithms end up splitting some clusters or another except for `shi_r_gap` (which in fact performs best) and `shi_r_ncut`. The reason for this is that conductance or `kvv` based methods are not able to find the optimal point to partition. Another important thing to note is that the multiway spectral algorithms degrade must more steeply if $K > K_{true}$ is used. Eigenvectors corresponding to i^{th} largest eigenvalues are no longer guaranteed to be PC if $i > K_{true}$. This means that those spectral dimensions is essentially "random" w.r.t Δ. The recursive algorithms on the other hand use only the second largest eigenvector which are PC w.r.t. Δ and so the first few cuts are lot more stable leading to better results.

The behavior of algorithms in presence of noise is quite similar. Figure 1(b) shows the how the performance degrades as noise is added. ($K = K_{true} = 5$). As expected the multiway algorithms perform the best. Like above the conductance based algorithm perform the worst. Another observation to make is that spectral algorithms with NJW as the first stage tend to degrade slightly less then those with Multicut as the first stage. This was hinted in [3] and is a result of mapping the points on a unit sphere which gets rid of radial variation.
.

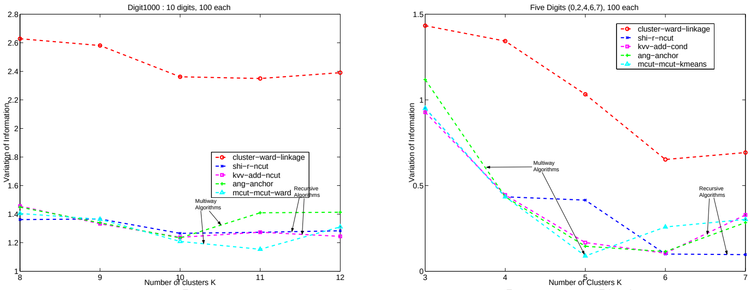


(a) Log normalized Dataset **(b) Standardized Dataset**

Figure 2: **Cellcycle Dataset.** a) Log normalized dataset. The vectors are make more gaussian by taking the log of the gene expression. b) Standardized dataset. Each gene expression is normalized to have mean 0 and variance 1.

Now lets us take a look at the first real dataset: The yeast cellcycle. The graphs in Figure 2 show the VI w.r.t. the true clustering as different K is given input to the algorithm. To illustrate the performance of different classes of algorithms (Linkage,Recursive,Multiway), we include the best algorithm in each class, followed by the next best two in any class. The sixth line denotes the *best* performance of *any* model based algorithm on that particular K. As we can see in Figure 2(a) the spectral algorithms are quite competitive to the model based algorithms. However it is very much dependent on the pre processing step as in Figure 2(b) the model based algorithms are better. Another important thing to note is that recursive spectral algorithms are performing better that multiway algorithms. We think that when the structure in the dataset is not that much then the multiway algorithms would tend to perform not so well. In essence however the results we obtained indicate the various spectral algorithms are quite similar to each other and there is no clear winner. Other experiments further vindicate this fact as other datasets have different particular algorithms as winners. We have just presented part of results above which seem to be more or less consistent over different datasets.

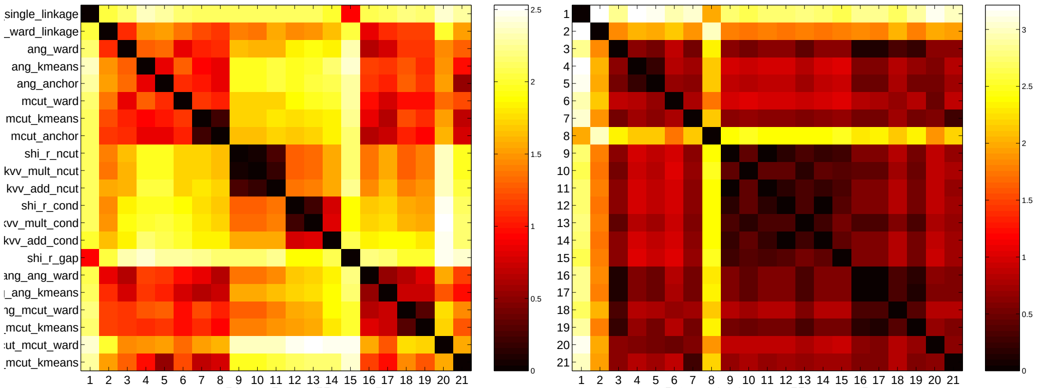
The results on the two digits dataset are shown in Figure 3. One thing to note immediately is that the linkage algorithms perform the worst and there is no significant difference between the multiway and recursive algorithms. The multiway algorithms perform the best when $K = K_{true}$ and more notably degrade much faster for higher K . In case of the five digits



(a) 10 digits **(b) Five digits**

Figure 3: **Digits Dataset.** a) Shows the results on all the digits b) Shows the results on five easily separable digits (0,2,4,6,7).

example the Multiway example are much more successful in figuring out the right clusters and all recursive algorithm are not (except `kvv_add_cond`).



(a) Log normalized CellCycle Dataset **(b) Digits (all 10) dataset**

Figure 4: **Inter algorithm VI .**

The next graphs show the VI distance between the clusterings produced by the different clustering algorithms (for one iteration and true K). One can easily see based on this matrix the algorithms can be easily "clustered" into the the groups they belong to. That is the clustering produced by algorithms within the same group is much more similar to each other than to those in the other groups. This indicates that the clustering produced in more depended on the *type* of algorithm (multiway/recursive/doubly spectral) than the exact post processing method. One interesting to note (figure 4a) is that amongst the spectral algorithms the `shi_r_gap` seems to produce quite different clusterings. This is because while it makes a localized decision to the choose the point to partition which makes it different than other recursive algorithms. Also when the data is not BS the first few cuts made would be on the comparatively more stable second EV which makes it different than the multiway algorithms as well.

7 Conclusions

The main conclusion of our comparison is that spectral clustering algorithms are not as different as they may appear. At the perfect S, the two multiway algorithms and the `shi_gap` recursive algorithm are provably equivalent. The experiments show that `shi_ncut` also behaves exactly as the multiway algorithm. The `kvv` algorithm and the conductance based cuts obviously behave differently (and worse) in the block stochastic case, but they perform competitively on the real data. We do not know how to characterize in general the perfect case for these algorithms.

The theoretical results are validated by experiments: if P is dataset is BS, the multiway algorithms find the perfect clustering and degrade gracefully in presence of noise.

On real data, there is variability between the algorithms but we have found no clear differences. In particular, no major component or feature has emerged as clear winner. However, some lessons have been learned: (1) If the data are close to BS, then the algorithms behave more similarly to one another. One can test if this is the case by computing distortion measures of the resulting clusters. If this is small a good and stable clustering has been obtained[2] (2) Multiway methods are better as long as noise is not too high; otherwise, the recursive algorithms are more stable. Also the multiway algorithms perform worse than the recursive ones when $K > K_{true}$. (3) If the node degrees are very different, then the spectral mapping of the NJW algorithm is more robust then the one of the Meila-Shi algorithm. This difference can affect performance at node degree ratios > 100, otherwise it should not be a concern.

References

[1] Ravi Kannan, Santosh Vempala, and Adrian Vetta. On clusterings - good, bad and spectral. In *FOCS*, pages 367–377, 2000.

[2] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[3] A. Y. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 849–856, Cambridge, MA, 2002. MIT Press.

[4] Marina Meila and Jianbo Shi. A random walks view of spectral segmentation, 2001.

[5] Yair Weiss. Segmentation using eigenvectors: A unifying view. In *ICCV*, pages 975–982, 1999.

[6] Deepak Verma and Marina Meila. A comparison of spectral methods. Technical Report UW-CSE-03-05-01, Dept. of Computer Science and Engineering, University of Washington, 2003.

[7] Ka Yee Yeung. Model-based clustering and data transformations for gene expression data. *http://staff.washington.edu/kayee/model/*.

[8] K. Yeung, C. Fraley, A. Murua, A. Raftery, and W. Ruzzo. Model-based clustering and data transformations for gene expression data. Technical Report UW-CSE-01-04-02, Dept. of Computer Science and Engineering, University of Washington, 2001.

[9] C. Kaynak. Methods of combining multiple classifiers and their applications to handwritten digit recognition. Master's thesis, Institute of Graduate Studies in Science and Engineering, Bogazici University., 1995.

[10] Andrew Moore. The anchors hierarchy: Using the triangle inequality to survive high-dimensional data. In *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence*, pages 397–405. AAAI Press, 2000.

[11] J.H. Ward. Hierarchical grouping to optimize an objective function. *J. Amer. Statist. Assoc.*, pages 236 – 244, 1963.

[12] Marina Meila. Comparing clusterings. Technical Report 418, UW Statistics Department, 2002.

[13] David L. Wallace. Comment. *Journal of the American Statistical Association*, 78(383):569–576, 1983.

[14] Marina Meila and Liang Xu. Multiway cuts and spectral clustering. Technical report, University of Washington, 2003. `www.stat.washington.edu/mmp/Papers/nips03-multicut-tr.ps`.

[2] See [14] for more details.