A General Purpose Computer-Assisted Clustering Methodology: Supplemental Notes

Justin Grimmer* Gary King†

December 2, 2010

*Assistant Professor, Department of Political Science, Stanford University; Encina Hall West 616 Serra St., Palo Alto, CA, 94305.(617)710-6803

†Albert J. Weatherhead III University Professor, Institute for Quantitative Social Science, 1737 Cambridge Street, Harvard University, Cambridge MA 02138; http://GKing.harvard.edu, king@harvard.edu, (617) 495-2027.

1 List of Clustering Methods

We summarize here the types of different clustering algorithms included in our applications and software. Existing algorithms are most often described as either statistical and algorithmic. The statistical models are primarily mixture models, including a large variety of finite mixture models (Fraley and Raftery, 2002; Banerjee et al., 2005; Quinn et al., 2006), infinite mixture models based on the Dirichlet process prior (Blei and Jordan, 2006), and mixture models (Blei, Ng and Jordan, 2003). The algorithmic approaches include methods that partition the documents directly, those that create a hierarchy of clusterings, and those which add an additional step to the clustering procedure. The methods include some which identify an exemplar document for each cluster (Kaufman and Rousseeuw, 1990; Frey and Dueck, 2007) and those which do not (Schrodt and Gerner, 1997; Shi and Malik, 2000; Ng, Jordan and Weiss, 2002; von Luxburg, 2007). The hierarchical methods can be further sub-divided into agglomerative (Hastie, Tibshirani and Friedman, 2001), divisive (Kaufman and Rousseeuw, 1990), and other hybrid methods (Gan, Ma and Wu, 2007). To use in our program, we obtain a flat partition of the documents from hierachical clustering methods. A final group includes methods which group words and documents together simulatenously (Dhillon, 2003) and those which embed the documents into lower dimensional space and then cluster (Kohonen, 2001). Some methods implicitly define a distance metric among documents but, for those that do not, we include many ways to measure the similarity between pairs documents, which is an input to a subset of the clustering methods used here. These include standard measures of distance (Manhattan, Euclidean), angular based measures of similarity (cosine), and many others.

Our software is written modularly so that new approaches can easily be included.

Table 1: Clustering Methods and Distance Metrics Available in Our Program

Method Name	Metric/Estimation/Tuning Parameter Varied	Citation
k-means	Manhattan	Forgy (1965)
k-means	Euclidean	
k-means	Minkowksi (p=4)	
k-means	Maximum	
k-means	Canberra	
k-means	Cosine	
k-means	Correlation	

Method Name	Metric Name	Citation
k-means	Binary	
k-means	Spearman's Ranked-Correlation	
k-means	Kendall	
k-means	Random Forest Distance	
Fuzzy k-means	Manhattan	Gath and Geva (1989)
Fuzzy k-means	Euclidean	
Fuzzy k-means	Minkowksi (p=4)	
Fuzzy k-means	Maximum	
Fuzzy k-means	Canberra	
Fuzzy k-means	Cosine	
Fuzzy k-means	Correlation	
Fuzzy k-means	Binary	
Fuzzy k-means	Spearman's Ranked-Correlation	
Fuzzy k-means	Kendall	
Fuzzy k-means	Random Forest Distance	
Trimmed k-means	Manhattan	Cuesta-Albertos, Gorda
Trimmed k-means	Euclidean	
Trimmed k-means	Minkowksi (p=4)	
Trimmed k-means	Maximum	
Trimmed k-means	Canberra	
Trimmed k-means	Cosine	
Trimmed k-means	Correlation	
Trimmed k-means	Binary	
Trimmed k-means	Spearman's Ranked-Correlation	
Trimmed k-means	Kendall	
Trimmed k-means	Random Forest Distance	
k Harmonic-means	Manhattan	Zhang, Hsu and Dayal
k Harmonic-means	Euclidean	
k Harmonic-means	Minkowksi (p=4)	
k Harmonic-means	Maximum	
k Harmonic-means	Canberra	
k Harmonic-means	Cosine	
k Harmonic-means	Correlation	
k Harmonic-means	Binary	
k Harmonic-means	Spearman's Ranked-Correlation	
k Harmonic-means	Kendall	
k Harmonic-means	Random Forest Distance	
k-medoids	Manhattan	Kaufman and Rousseeu
k-medoids	Euclidean	
k-medoids	Minkowksi (p=4)	
k-medoids	Maximum	
k-medoids	Canberra	
k-medoids	Cosine	
k-medoids	Correlation	

Method Name	Metric Name	Citation
k-medoids	Binary	
k-medoids	Spearman's Ranked-Correlation	
k-medoids	Kendall	
k-medoids	Random Forest Distance	
Affinity Propagation	Manhattan	Frey and Dueck (2007)
Affinity Propagation	Euclidean	
Affinity Propagation	Minkowksi (p=4)	
Affinity Propagation	Maximum	
Affinity Propagation	Canberra	
Affinity Propagation	Cosine	
Affinity Propagation	Correlation	
Affinity Propagation	Binary	
Affinity Propagation	Spearman's Ranked-Correlation	
Affinity Propagation	Kendall	
Affinity Propagation	Random Forest Distance	
Affinity Propagation	Encoding Metrics	
Maximum Entropy Clustering	Euclidean (Varying β values)	Karayiannis (1994)
Agglomerative Hierarchical	Manhattan (Link = Ward)	McQuitty (1966)
Agglomerative Hierarchical	Euclidean (Link = Ward)	
Agglomerative Hierarchical	Minkowski (Link =Ward)	
Agglomerative Hierarchical	Maximum (Link =Ward)	
Agglomerative Hierarchical	Canberra (Link =Ward)	
Agglomerative Hierarchical	Cosine (Link =Ward)	
Agglomerative Hierarchical	Correlation (Link =Ward)	
Agglomerative Hierarchical	Binary (Link =Ward)	
Agglomerative Hierarchical	Spearman Ranked-Correlation (Link =Ward)	
Agglomerative Hierarchical	Kendall (Link =Ward)	
Agglomerative Hierarchical	Random Forest Distance (Link =Ward)	
Agglomerative Hierarchical	Manhattan (Link = Single)	
Agglomerative Hierarchical	Euclidean (Link = Single)	
Agglomerative Hierarchical	Minkowski (Link =Single)	
Agglomerative Hierarchical	Maximum (Link =Single)	
Agglomerative Hierarchical	Canberra (Link =Single)	
Agglomerative Hierarchical	Cosine (Link =Single)	
Agglomerative Hierarchical	Correlation (Link =Single)	
Agglomerative Hierarchical	Binary (Link =Single)	
Agglomerative Hierarchical	Spearman Ranked-Correlation (Link =Single)	
Agglomerative Hierarchical	Kendall (Link =Single)	
Agglomerative Hierarchical	Random Forest Distance (Link =Single)	
Agglomerative Hierarchical	Manhattan (Link = Complete)	
Agglomerative Hierarchical	Euclidean (Link = Complete)	
Agglomerative Hierarchical	Minkowski (Link =Complete)	
Agglomerative Hierarchical	Maximum (Link =Complete)	
Agglomerative Hierarchical	Canberra (Link =Complete)	

Method Name	Metric Name	Citation
Agglomerative Hierarchical	Cosine (Link =Complete)	
Agglomerative Hierarchical	Correlation (Link =Complete)	
Agglomerative Hierarchical	Binary (Link =Complete)	
Agglomerative Hierarchical	Spearman Ranked-Correlation (Link =Complete)	
Agglomerative Hierarchical	Kendall (Link =Complete)	
Agglomerative Hierarchical	Random Forest Distance (Link =Complete)	
Agglomerative Hierarchical	Manhattan (Link = Average)	
Agglomerative Hierarchical	Euclidean (Link = Average)	
Agglomerative Hierarchical	Minkowski (Link =Average)	
Agglomerative Hierarchical	Maximum (Link =Average)	
Agglomerative Hierarchical	Canberra (Link =Average)	
Agglomerative Hierarchical	Cosine (Link =Average)	
Agglomerative Hierarchical	Correlation (Link =Average)	
Agglomerative Hierarchical	Binary (Link =Average)	
Agglomerative Hierarchical	Spearman Ranked-Correlation (Link =Average)	
Agglomerative Hierarchical	Kendall (Link =Average)	
Agglomerative Hierarchical	Random Forest Distance (Link =Average)	
Agglomerative Hierarchical	Manhattan (Link = McQuitty)	
Agglomerative Hierarchical	Euclidean (Link = McQuitty)	
Agglomerative Hierarchical	Minkowski (Link =McQuitty)	
Agglomerative Hierarchical	Maximum (Link =McQuitty)	
Agglomerative Hierarchical	Canberra (Link =McQuitty)	
Agglomerative Hierarchical	Cosine (Link =McQuitty)	
Agglomerative Hierarchical	Correlation (Link =McQuitty)	
Agglomerative Hierarchical	Binary (Link =McQuitty)	
Agglomerative Hierarchical	Spearman Ranked-Correlation (Link =McQuitty)	
Agglomerative Hierarchical	Kendall (Link =McQuitty)	
Agglomerative Hierarchical	Random Forest Distance (Link =McQuitty)	
Agglomerative Hierarchical	Manhattan (Link = Median)	
Agglomerative Hierarchical	Euclidean (Link = Median)	
Agglomerative Hierarchical	Minkowski (Link =Median)	
Agglomerative Hierarchical	Maximum (Link =Median)	
Agglomerative Hierarchical	Canberra (Link =Median)	
Agglomerative Hierarchical	Cosine (Link =Median)	
Agglomerative Hierarchical	Correlation (Link =Median)	
Agglomerative Hierarchical	Binary (Link =Median)	
Agglomerative Hierarchical	Spearman Ranked-Correlation (Link =Median)	
Agglomerative Hierarchical	Kendall (Link =Median)	
Agglomerative Hierarchical	Random Forest Distance (Link =Median)	
Agglomerative Hierarchical	Manhattan (Link = Centroid)	
Agglomerative Hierarchical	Euclidean (Link = Centroid)	
Agglomerative Hierarchical	Minkowski (Link =Centroid)	
Agglomerative Hierarchical	Maximum (Link =Centroid)	
Agglomerative Hierarchical	Canberra (Link =Centroid)	

Method Name	Metric Name	Citation
Agglomerative Hierarchical	Cosine (Link =Centroid)	
Agglomerative Hierarchical	Correlation (Link =Centroid)	
Agglomerative Hierarchical	Binary (Link =Centroid)	
Agglomerative Hierarchical	Spearman Ranked-Correlation (Link =Centroid)	
Agglomerative Hierarchical	Kendall (Link =Centroid)	
Agglomerative Hierarchical	Random Forest Distance (Link =Centroid)	
Model-Based Hierarchical		Fraley (1998)
Proximus	Manhattan	Koyuturk, Graham and
Proximus	Euclidean	
Proximus	Minkowksi (p=4)	
Proximus	Maximum	
Proximus	Canberra	
Proximus	Cosine	
Proximus	Correlation	
Proximus	Binary	
Proximus	Spearman's Ranked-Correlation	
Proximus	Kendall	
Proximus	Random Forest Distance	
ROCK	Manhattan	Guha, Rastogi and Shi
ROCK	Euclidean	
ROCK	Minkowksi (p=4)	
ROCK	Maximum	
ROCK	Canberra	
ROCK	Cosine	
ROCK	Correlation	
ROCK	Binary	
ROCK	Spearman's Ranked-Correlation	
ROCK	Kendall	
ROCK	Random Forest Distance	
Divisive Hierarchical	Manhattan	Kaufman and Rousseeu
Divisive Hierarchical	Euclidean	
Divisive Hierarchical	Minkowksi (p=4)	
Divisive Hierarchical	Maximum	
Divisive Hierarchical	Canberra	
Divisive Hierarchical	Cosine	
Divisive Hierarchical	Correlation	
Divisive Hierarchical	Binary	
Divisive Hierarchical	Spearman's Ranked-Correlation	
Divisive Hierarchical	Kendall	
Divisive Hierarchical	Random Forest Distance	
DISMEA	Manhattan	Gan, Ma and Wu (2007
DISMEA	Euclidean	
DISMEA	Minkowksi (p=4)	
DISMEA	Maximum	

Method Name	Metric Name	Citation
DISMEA	Canberra	
DISMEA	Cosine	
DISMEA	Correlation	
DISMEA	Binary	
DISMEA	Spearman's Ranked-Correlation	
DISMEA	Kendall	
DISMEA	Random Forest Distance	
Fuzzy	Manhattan	**?**)
Fuzzy	Euclidean	
Fuzzy	Minkowksi (p=4)	
Fuzzy	Maximum	
Fuzzy	Canberra	
Fuzzy	Cosine	
Fuzzy	Correlation	
Fuzzy	Binary	
Fuzzy	Spearman's Ranked-Correlation	
Fuzzy	Kendall	
Fuzzy	Random Forest Distance	
QTCLust	Varying Radii	Heyer, Kruglyak and Y
Self-Organizing Map	Hexagon	Kohonen (2001)
Self-Organizing Map	Square	
Self-Organizing Tree	Euclidean	Brock et al. (2008)
Self-Organizing Tree	Correlation	
Unnormalized Spectral	Manhattan	von Luxburg (2007); Sc
Unnormalized Spectral	Euclidean	
Unnormalized Spectral	Euclidean	
Unnormalized Spectral	Minkowski (p=4)	
Unnormalized Spectral	Maximum	
Unnormalized Spectral	Canberra	
Unnormalized Spectral	Cosine	
Unnormalized Spectral	Correlation	
Unnormalized Spectral	Binary	
Unnormalized Spectral	Spearman's Ranked-Correlation	
Unnormalized Spectral	Kendall	
Unnormalized Spectral	Random Forest Distance	
Meila and Shi Spectral	Manhattan	Meila and Shi (2001)
Meila and Shi Spectral	Euclidean	
Meila and Shi Spectral	Minkowski (p=4)	
Meila and Shi Spectral	Maximum	
Meila and Shi Spectral	Canberra	
Meila and Shi Spectral	Cosine	
Meila and Shi Spectral	Correlation	
Meila and Shi Spectral	Binary	
Meila and Shi Spectral	Spearman's Ranked-Correlation	

Method Name	Metric Name	Citation
Meila and Shi Spectral	Kendall	
Meila and Shi Spectral	Random Forest Distance	
Ng, Jordan, Weiss Spectral	Manhattan	Ng, Jordan and Weiss (
Ng, Jordan, Weiss Spectral	Euclidean	
Ng, Jordan, Weiss Spectral	Minkowski (p=4)	
Ng, Jordan, Weiss Spectral	Maximum	
Ng, Jordan, Weiss Spectral	Canberra	
Ng, Jordan, Weiss Spectral	Cosine	
Ng, Jordan, Weiss Spectral	Correlation	
Ng, Jordan, Weiss Spectral	Binary	
Ng, Jordan, Weiss Spectral	Spearman's Ranked-Correlation	
Ng, Jordan, Weiss Spectral	Kendall	
Ng, Jordan, Weiss Spectral	Random Forest Distance	
Shi-Malik Spectral	Manhattan	Shi and Malik (2000)
Shi-Malik Spectral	Euclidean	
Shi-Malik Spectral	Minkowski (p = 4)	
Shi-Malik Spectral	Maximum	
Shi-Malik Spectral	Canberra	
Shi-Malik Spectral	Cosine	
Shi-Malik Spectral	Correlation	
Shi-Malik Spectral	Binary	
Shi-Malik Spectral	Spearman's Ranked-Correlation	
Shi-Malik Spectral	Kendall	
Shi-Malik Spectral	RandomForest Distance	
Dirichlet Process, Multinomial	Variational Approximation	Blei and Jordan (2006)
Dirichlet Process, Normals	Variational Approximation	**?**)
Mixture, Multinomials	EM Algorithm	Gelman et al. (2003)
Mixture, Multinomials	Variational Approximation	
Mixture, von-Mises Fisher	EM Algorithm	Banerjee et al. (2005)
Mixture, von-Mises Fisher	Variational Approximation	
Mixture of Normals	EM-algorithm	Fraley and Raftery (200
Mixture of Normals	Variational Approximation	
Co-clustering Mutual Information	NA	Dhillon, Mallela and M
Co-clustering (SVD)	NA	Dhillon (2003)
LLAhclust	LLA	Lerman (1991)
LLAhclust	tippet	
LLAhclust	average	
LLAhclust	complete	
LLAhclust	fisher	
LLAhclust	uniform	
LLAhclust	normal	
LLAhclust	maximum	
CLUES	Euclidean	Wang, Qiu and Zamar
CLUES	Correlation	

Method Name	Metric Name	Citation
bclust	Manhattan	Leisch (1999)
bclust	Euclidean	
bclust	Minkowksi (p=4)	
bclust	Maximum	
bclust	Canberra	
bclust	Cosine	
bclust	Correlation	
bclust	Binary	
bclust	Spearman's Ranked-Correlation	
bclust	Kendall	
bclust	Random Forest Distance	
c-shell	euclidean	Rajesh (1996)
c-shell	manhattan	
Latent-Dirichlet Allocation	Variational Approximation	Blei, Ng and Jordan (20
Expressed Agenda Model	Variational Approximation	Grimmer (2010)

2 Extensions of the Clustering Space

Here we describe two methods for extending beyond the space that we constructed.

First, we consider a way of randomly sampling clusterings from the entire Bell space. When desired, a researcher could then add some of these to the original set of clusterings and rerun the same visualization. To do this, we developed a two step method of taking a uniform random draw from the set of all possible clusterings. First, sample the number of clusters K from a multinomial distribution with probability $\text{Stirling}(K, N)/\text{Bell}(N)$ where $\text{Stirling}(K, N)$ is the number of ways to partition N objects into K clusters (i.e., known as the Stirling number of the second kind). Second, conditional on K, obtain a random clustering by sampling the cluster assignment for each document i from a multinomial distribution, with probability $1/K$ for each cluster assignment. If each of the K clusters does not contain at least one document, reject it and take another draw (see Pitman, 1997).

A second approach to expanding the space beyond the existing algorithms directly extends the existing space by drawing larger concentric hulls containing the convex hull of the existing solutions. To do this, we define a Markov chain on the set of partitions, starting with a chain on the boundaries of the existing solutions. To do this, consider a clustering of the data $\boldsymbol{c}_j$. Define $\mathcal{C}(\boldsymbol{c}_j)$ as the set

of clusterings that differ by exactly by one document: a clustering $\boldsymbol{c}_j^{'} \in \mathcal{C}(\boldsymbol{c}_j)$ if and only if one document belongs to a different cluster in $\boldsymbol{c}_j^{'}$ than in $\boldsymbol{c}_j$. Our first Markov chain takes a uniform sample from this set of partitions. Therefore, if $\boldsymbol{c}_{j'} \in \mathcal{C}(\boldsymbol{c}_j)$ (and $\boldsymbol{c}_j$ is in the "interior" of the set of partitions) then $p(\boldsymbol{c}_{j'}|\boldsymbol{c}_j) = \frac{1}{NK}$ where N are the number of documents and K is the number of clusters. If $\boldsymbol{c}_{j'} \notin \mathcal{C}(\boldsymbol{c}_j)$ then $p(\boldsymbol{c}_{j'}|\boldsymbol{c}_j) = 0$. To ensure that the Markov chain proceeds outside the existing hull, we add a rejection step: For all $\boldsymbol{c}_{j'} \in \mathcal{C}(\boldsymbol{c}_j)$ $p(\boldsymbol{c}_{j'}|\boldsymbol{c}_j) = \frac{1}{NK}\mathrm{I}(\boldsymbol{c}_{j'} \notin \text{Convex Hull})$. This ensures that the algorithm explores the parts of the Bell space that are not already well described by the included clusterings. To implement this strategy, we use a three stage process applied to each clustering $\boldsymbol{c}_k$: First, we select a cluster to edit with probability $\frac{N_j}{N}$ for each cluster j in clustering $\boldsymbol{c}_k$. Conditional on selecting cluster j we select a document to move with probability $\frac{1}{N_j}$. Then, we move the document to one of the other $K-1$ clusters or to a new cluster, so the document will be sent to a new clustering with probability $\frac{1}{K}$.

3 Insights from Partisan Taunting

Examples from Lautenberg's press releases and contemporary political discourse suggests new insights into Congressional behavior. Partisan taunting creates the possibilty of *negative* credit claiming: when members of Congress undermine the opposing party's efforts to claim credit for federal funds. For example, the DCCC issued a press release accusing Mary Bono Mack (R-CA,45) of acting "hypocritically" for announcing "$40 million for two long-awaited improvement projects to I-10, even though she voted against the improvements". Partisan taunting also allows members of a party to claim credit for legislative work even when no reform actually occurred. Both Democrats and Republican caucuses regularly issue statements, blaming inaction in the Congress on the other party. For example a June 27, 2007 press release from the Senate Democratic caucus reads, "Senate Republicans blocked raising the minimum wage".

Partisan taunting is also an important element of position taking, allowing members of Congress to juxtapose their own position against the other party's. Senator Lautenberg used this strategy in a press release when he "filed an amendment to rename the 'Tax Reconciliation Act of 2005,' to reflect the true impact the legislation will have on the nation if allowed to pass. Senator Lautenberg's amendment would change the name of the measure to 'More Tax Breaks for the Rich and More

Debt for Our Grandchildren Deficit Expansion Reconciliation Act of 2006.' The Republican bill would provide more tax cuts to the wealthiest Americans while saddling our grandchildren with additional debt."

Partisan taunting also overlaps the category of advertising, which occurs in Lautenberg's press release when he "Expresses Shock Over President Bush's Mock Search for Weapons of Mass Destruction". While devoid of policy content, this statement allows Lautenberg to appear as a sober statesman next to a juvenille administration joke.

4 Technical Details

4.1 Defining The Distance Between Clusterings

Each cluster method j ($j = 1, \ldots, J$) produces a partition (or "clustering") of the documents with K_j clusters assumed (or estimated). Denote by c_{ikj} an indicator of whether (or the extent to which) document i is assigned to cluster k under method j. For "hard" cluster algorithms (those that assign a document to only one cluster), $c_{ikj} \in \{0, 1\}$; for "soft" methods, $c_{ikj} \in [0, 1]$; and for both $\sum_{k=1}^{K_j} c_{ikj} = 1$ for all k and j. The K_j-vector denoting document i's cluster membership from method j is given by $\boldsymbol{c}_{ij}$ and is an element of the $K_j - 1$ dimensional simplex. Then we characterize a full clustering for method j with the $N \times K_j$ matrix $\boldsymbol{c}_j$.

Our distance metric builds on entropy, a function H that maps from the proportion of documents in each category to a measure of information in the documents. For clustering $\boldsymbol{c}_j$, define its entropy as (Mackay, 2003; Shannon, 1949),

$$\begin{aligned} H(\boldsymbol{c}_j) &= -\sum_{k=1}^{K}\sum_{i=1}^{N} \frac{c_{ijk}}{N} \log\left(\sum_{i=1}^{N} \frac{c_{ijk}}{N}\right) \\ &= -\sum_{k=1}^{K} p_j(k) \log p_j(k) = H(p_j(1), p_j(2), \ldots, p_j(K)) = H(\boldsymbol{p}_j) \end{aligned}$$

define the proportion of documents assigned to the k^{th} category as $\sum_{i=1}^{N} \frac{c_{ijk}}{N} = p_j(k)$ and denote as $\boldsymbol{p}_j = (p_j(1), \ldots, p_j(K))$ the vector describing the proportion of documents assigned to each category.

We now develop a measure of distance between clusterings based upon a (rescaled) measure of pairwise disagreements. Denote by $d(\boldsymbol{c}_j, \boldsymbol{c}_{j'})$ our candidate measure of the distance between two

clusterings. Define pair($\boldsymbol{c}_j, \boldsymbol{c}_{j'}$) as the number of documents in the same cluster in $\boldsymbol{c}_j$ but not in $\boldsymbol{c}_{j'}$ plus the pairs of documents in $\boldsymbol{c}_{j'}$ not in $\boldsymbol{c}_j$. The pair function is more *refined* than higher order functions: it includes the information in all higher order subsets, such as triples, quadruples, etc. This is well-known, but we offer a simple proof by contradiction here. Suppose, by way of contradiction, that clustering $\boldsymbol{c}_j$ and $\boldsymbol{c}_z$ agree on all pairs, but disagree on some larger subset m. This implies there exists a group of documents $c_{1j}, \ldots, c_{mj}$ grouped in the same cluster in $\boldsymbol{c}_j$ but not grouped in $\boldsymbol{c}_z$. But for this to be true, then there must be at least m pair differences between the two clusterings, contradicting our assumption that there are no pairwise disagreements. Note that the converse is not true: Two clusterings could agree about all subsets of size $m > 2$ but disagree about the pairs of documents that belong together.

We use three assumptions to derive the properties of our distance metric. First, we assume that the distance metric should be based upon the number of pairwise disagreements (encoded in the pair function). We extract two properties of our metric directly from this assumption. First, denote the maximum possible distance between clusterings as that which produces the maximum number of pairwise disagreements about the cluster in which the two documents belong. Denote $c(1, N)$ as the clustering where all N documents are placed into one cluster and $\boldsymbol{c}(N, N)$ the clustering where all N documents are placed into N individual clusters. Then the maximum possible pairwise disagreements is between $\boldsymbol{c}(1, N)$ and $\boldsymbol{c}(N, N)$. (Note that $c(1, N)$ implies $\binom{N}{2}$ pairs, while $c(N, N)$ implies 0 pairs, implying $\binom{N}{2}$ disagreements, the largest possible disagreement.) In addition, for each clustering $\boldsymbol{c}_j$,

$$\text{pair}(\boldsymbol{c}(1, N), \boldsymbol{c}(N, N)) \quad = \quad \text{pair}(\boldsymbol{c}(1, N), \boldsymbol{c}_j) + \text{pair}(\boldsymbol{c}(N, N), \boldsymbol{c}_j). \tag{4.1}$$

Our second property extracted from the focus on pairwise disagreements ensures that partitions with smaller distances are actually more similar — have fewer pairwise disagreements — than other partitions with larger distances. Define the *meet* between two clusterings $\boldsymbol{c}_j$ and $\boldsymbol{c}_k$ as a new (compromise) clustering, denoted $\boldsymbol{c}_j \times \boldsymbol{c}_k$, which assigns pairs of documents to the same cluster if both of the component clusterings agree they belong in the same cluster. If the two clusterings disagree, then the pair of documents are not assigned to the same cluster. A general property of a

meet is that it lies "between" two clusterings, or for any clusterings $\boldsymbol{c}_z$ and $\boldsymbol{c}_m$,

$$\text{pair}(\boldsymbol{c}_z, \boldsymbol{c}_m) \quad = \quad \text{pair}(\boldsymbol{c}_z \times \boldsymbol{c}_m, \boldsymbol{c}_z) + \text{pair}(\boldsymbol{c}_z \times \boldsymbol{c}_m, \boldsymbol{c}_m). \tag{4.2}$$

Using the pair function and an additional assumption — invariance to the number of documents included in the clustering — we define a third property of our metric: how the shared information changes as the number of clusters change. Consider the case where we *refine* a clustering $\boldsymbol{c}_j$ by dividing documents in cluster c_{jk} among a set of newly articulated clusters, $\boldsymbol{c}^{'}(n_{jk})$, and where the new clustering is $\boldsymbol{c}_j^{'}$. (If we restrict attention to the n_{jk} documents originally in cluster k in clustering j, then c_{jk} is the clustering that assigns all n_{jk} documents to the same cluster, so we write it as $\boldsymbol{c}(1, n_{jk})$.) A property of the pair function is that,

$$\text{pair}(\boldsymbol{c}_j, \boldsymbol{c}_j^{'}) \quad = \quad \sum_{k=1}^{K_j} \text{pair}(\boldsymbol{c}(1, n_{jk}), \boldsymbol{c}^{'}(n_{jk})) \tag{4.3}$$

Using Equation 4.3, we apply the invariance assumption to rescale the pair function. Therefore, we require the distance between $\boldsymbol{c}_j$ and $\boldsymbol{c}_j^{'}$ to be $d(\boldsymbol{c}_j, \boldsymbol{c}_j^{'}) = \sum_{k=1}^{K} \frac{n_{jk}}{n} d(\boldsymbol{c}(1, n_{jk}), \boldsymbol{c}_{jk}^{'})$.

The final property employs the pair function plus a scaling axiom to define the maximum distance for a fixed number of clusters K. Call the clustering that places the same number of documents into each cluster $\boldsymbol{c}(\text{uniform}, K)$ (if this clustering exists). Then the clustering with the most pairwise disagreements with $\boldsymbol{c}(\text{uniform}, K)$ is $\boldsymbol{c}(1, N)$ and so bounding on this distance bounds all smaller distances. We use a scaling assumption to require that $d(\boldsymbol{c}(\text{uniform}, K), \boldsymbol{c}(1, N)) = \log K$, i.e., that the distance between an evenly spread out clustering and a clustering that places all documents into the same category increases with the number of categories at a logarithmic rate.

Our three assumptions, and the four properties extracted from these assumptions, narrow the possible metrics to a unique choice: the *variation of information* (VI), based on the shared or conditional entropy between two clusterings Meila (2007). Further, it is a distance metric (even though we made no explicit assumptions that our distance measure be a metric). We define the VI metric by considering the distance between two arbitrary clusterings, $\boldsymbol{c}_j$ and $\boldsymbol{c}_j^{'}$. Define the proportion of documents assigned to cluster k in method j and cluster $k^{'}$ in method $j^{'}$ as $\boldsymbol{p}_{jj'}(k, k^{'}) = \sum_{i=1}^{N} c_{ikj} c_{ik'j'}/N$.

Given the joint-entropy definition of shared information between $\boldsymbol{c}_j$ and $\boldsymbol{c}_{j'}$, $H(\boldsymbol{c}_j, \boldsymbol{c}_{j'}) = -\sum_{k=1}^{K} \sum_{k'=1}^{K^{'}} \boldsymbol{p}_{jj'}(k, k^{'}) \log \boldsymbol{p}_{jj'}(k, k^{'})$, we seek to determine the amount of information cluster

$\boldsymbol{c}_j$ adds if we have already observed $\boldsymbol{c}_{j'}$. A natural way to measure this additional information is with the conditional entropy, $H(\boldsymbol{c}_j|\boldsymbol{c}_{j'}) = H(\boldsymbol{c}_j, \boldsymbol{c}_{j'}) - H(\boldsymbol{c}_{j'})$, which we make symmetric by adding together the conditional entropies: (Meila, 2007),

$$d(\boldsymbol{c}_j, \boldsymbol{c}_{j'}) \equiv VI(\boldsymbol{c}_j, \boldsymbol{c}_{j'}) \quad = \quad H(\boldsymbol{c}_j|\boldsymbol{c}_{j'}) + H(\boldsymbol{c}_{j'}|\boldsymbol{c}_j). \tag{4.4}$$

An equivalent statement of the variation of information may be more intuitive. Define the *mutual information* between clusterings $\boldsymbol{c}_j$ and $\boldsymbol{c}_{j'}$, $I(\boldsymbol{c}_j; \boldsymbol{c}_{j'})$ as

$$I(\boldsymbol{c}_j; \boldsymbol{c}_{j'}) = -\sum_{k=1}^{K}\sum_{k'=1}^{K'} p'_{jj}(k, k') \log\left(\frac{p_{jj}(k, k')}{p_j(k)p_{j'}(k')}\right) \tag{4.5}$$

4.2 Properties of the Pair Function

In this section we prove various properties of the pair function.

Lemma 1. *For all clusterings* $\boldsymbol{c}_j$, *pair*$(\boldsymbol{c}(1,N), \boldsymbol{c}(N,N)) =$ *pair*$(\boldsymbol{c}(1,N), \boldsymbol{c}_j) +$ *pair*$(\boldsymbol{c}(N,N), \boldsymbol{c}_j)$

Proof. Note that $\boldsymbol{c}(1,N)$ implies $\binom{N}{2}$ pairs of documents, so $\text{pair}(\boldsymbol{c}(1,N), \boldsymbol{c}(N,N)) = \binom{N}{2}$. Any $\boldsymbol{c}_j$ will have $g = \sum_{k=1}^{K_j}\binom{n_{jk}}{2}$ pairs of documents, where n_{jk} represents the number of documents assigned to the k^{th} cluster in clustering j. Therefore, $\text{pair}(\boldsymbol{c}(1,N), \boldsymbol{c}_j) = \binom{N}{2} - g$. If all clusterings are placed into their own clusters, then there are no pairs of clusters, so $\text{pair}(\boldsymbol{c}(N,N), \boldsymbol{c}_j) = g$. Adding these two quantities together we find that, $\text{pair}(\boldsymbol{c}(1,N), \boldsymbol{c}_j) + \text{pair}(\boldsymbol{c}(N,N), \boldsymbol{c}_j) = \binom{N}{2} = \text{pair}(\boldsymbol{c}(1,N), \boldsymbol{c}(N,N))$. So, we require for our distance metric that $d(\boldsymbol{c}(1,N), \boldsymbol{c}(N,N)) = d(\boldsymbol{c}(1,N), \boldsymbol{c}_j) + d(\boldsymbol{c}(N,N), \boldsymbol{c}_j)$ for all possible clusterings $\boldsymbol{c}_j$. □

Lemma 2. *For all clusterings* $\boldsymbol{c}_z$ *and* $\boldsymbol{c}_m$, *pair*$(\boldsymbol{c}_z, \boldsymbol{c}_m) =$ *pair*$(\boldsymbol{c}_z \times \boldsymbol{c}_m, \boldsymbol{c}_z) +$ *pair*$(\boldsymbol{c}_z \times \boldsymbol{c}_m, \boldsymbol{c}_m)$

Proof. Define $g^z = \sum_{k=1}^{K}\binom{n_{zk}}{2}$ and $g^m = \sum_{k=1}^{K'}\binom{n_{mk}}{2}$ and call the number of pairs where the two clusterings agree g^{agree}. Then $\text{pair}(\boldsymbol{c}_z, \boldsymbol{c}_m) = (g^z - g^{\text{agree}}) + (g^m - g^{\text{agree}})$. $\boldsymbol{c}_m \times \boldsymbol{c}_z$ places a pair of documents into the same cluster if and only if $\boldsymbol{c}_z$ and $\boldsymbol{c}_m$ agree that the pair belongs together, thus $\text{pair}(\boldsymbol{c}_z \times \boldsymbol{c}_m, \boldsymbol{c}_z) = g^z - g^{\text{agree}}$. By the same argument $\text{pair}(\boldsymbol{c}_z \times \boldsymbol{c}_m, \boldsymbol{c}_m) = g^m - g^{\text{agree}}$ and therefore $\text{pair}(\boldsymbol{c}_z \times \boldsymbol{c}_m, \boldsymbol{c}_m) + \text{pair}(\boldsymbol{c}_z \times \boldsymbol{c}_m, \boldsymbol{c}_z) = \text{pair}(\boldsymbol{c}_z, \boldsymbol{c}_m)$.

Thus, the meet provides a natural definition of the area between two clusterings, so we will require that $d(\boldsymbol{c}_z, \boldsymbol{c}_m) = d(\boldsymbol{c}_z \times \boldsymbol{c}_m, \boldsymbol{c}_z) + d(\boldsymbol{c}_z \times \boldsymbol{c}_m, \boldsymbol{c}_m)$. □

Lemma 3. *If clustering* $\boldsymbol{c}_j^{'}$ *refines* $\boldsymbol{c}_j$ *then pair*$(\boldsymbol{c}_j, \boldsymbol{c}_j^{'}) = \sum_{k=1}^{K_j}$ *pair*$(\boldsymbol{c}(1, n_{jk}), \boldsymbol{c}^{'}(n_{jk}))$

Proof. Define $K_{j'}$ as the number of clusters in the refined clustering. Apply the definition of the pair function results in $\text{pair}(\boldsymbol{c}_j, \boldsymbol{c}_j^{'}) = \sum_{k=1}^{K_j} \binom{n_{jk}}{2} - \sum_{z=1}^{K_{j'}} \binom{n_{j'z}}{2}$ (because the refinement can only break apart pairs). For each cluster k in $\boldsymbol{c}_j$, enumerate the clusters in $\boldsymbol{c}_j^{'}$ that refine k with $r = 1, \ldots, R_k$ (and note, R_k could be 1, indicating that there was no refinement). We can rewrite the pair function as $\text{pair}(\boldsymbol{c}_j, \boldsymbol{c}_j^{'}) = \sum_{k=1}^{K_j} \left(\binom{n_{jk}}{2} - \sum_{r=1}^{R_k} \binom{n_{j'r}}{2} \right) = \sum_{k=1}^{K_j} \text{pair}(\boldsymbol{c}(1, n_{jk}), \boldsymbol{c}^{'}(n_{jk}))$. □

Theorem 1 (Meila, 2007)**.** *The three assumptions imply that the distance metric is the Variation of Information, given by*

$$d(\boldsymbol{c}_j, \boldsymbol{c}_{j'}) \equiv VI(\boldsymbol{c}_j, \boldsymbol{c}_{j'}) \quad = \quad H(\boldsymbol{c}_j|\boldsymbol{c}_{j'}) + H(\boldsymbol{c}_{j'}|\boldsymbol{c}_j). \tag{4.6}$$

Proof. The four properties, derived from our three assumptions are equivalent to those stated in Meila (2007), and so the proof follows the same argument, which we present here for completeness. Applying the third and fourth properties we see that $d(\boldsymbol{c}_j, \boldsymbol{c}(N, N)) = \sum_k^{K_j} \frac{n_k}{N} d(\boldsymbol{c}(1, N_k), \boldsymbol{c}(\text{uniform}(N_k), N_k)) = \sum_k^{K_j} \frac{n_k}{N} \log n_k$ Adding and subtracting $\log N$ we have $\sum_k^{K_j} \frac{n_k}{N} \log n_k = \sum_k^{K_j} \frac{n_k}{N} (\log \frac{n_k}{N} + \log N)$, which is equal to $\log N - \text{H}(\boldsymbol{c}_j)$. By our fourth property $d(\boldsymbol{c}(1, N), \boldsymbol{c}(N, N)) = \log N$. Property 1 and this fact imply $d(\boldsymbol{c}_j, \boldsymbol{c}(1, N)) = \text{H}(\boldsymbol{c}_j)$. Now, consider two arbitrary clusterings, $\boldsymbol{c}_m$ and $\boldsymbol{c}_z$. Identify all the n_{km} observations assigned to the k^{th} cluster in method m as k_m. And collect the cluster labels for these documents in $\boldsymbol{c}_z$ in $\boldsymbol{c}_z(k_m)$. Then, $d(\boldsymbol{c}_m, \boldsymbol{c}_m \times \boldsymbol{c}_z) = \sum_{k=1}^{K_m} \frac{n_{km}}{N} d(\boldsymbol{c}(1, n_{km}), \boldsymbol{c}_z(k_m))$ and by our previous argument $\sum_{k=1}^{K_m} \frac{n_{km}}{N} d(\boldsymbol{c}(1, n_{km}), \boldsymbol{c}_z(k_m)) = \sum_{k=1}^{K_m} \frac{n_{km}}{N} H(\boldsymbol{c}_z(k_m))$ and applying properties of entropy reveals that $\sum_{k=1}^{K_m} \frac{n_{km}}{N} H(\boldsymbol{c}_z(k_m)) = H(\boldsymbol{c}_m|\boldsymbol{c}_z)$. Applying our second property then shows that $d(\boldsymbol{c}_m, \boldsymbol{c}_z) = H(\boldsymbol{c}_m|\boldsymbol{c}_z) + H(\boldsymbol{c}_z|\boldsymbol{c}_m)$ which completes the proof. □

4.3 The Sammon Multidimensional Scaling Algorithm

We define here the Sammon (1969) multidimensional scaling algorithm and show that it possesses the properties we need. Let $\boldsymbol{c}_j$ be an $N \times K_j$ matrix (for document i, $i = 1, \ldots, N$, and cluster k, $k = 1, \ldots, K_j$, characterizing clustering j), each element of which describes whether each document is (0) or is not (1) assigned to each cluster (or for soft clustering methods how a document is allocated among the clusters, but where the sum over k is still 1). For each clustering j, the goal is to define its coordinates in a new two-dimensional space $\boldsymbol{x}_j = (x_{j1}, x_{j2})$, which we collect into a $J \times 2$ matrix

$\boldsymbol{X}$. We use the Euclidean distance between two clusterings in this space, which we represent for clusterings j and $j^{'}$ as $d^{\text{euc}}(\boldsymbol{x}_j, \boldsymbol{x}_{j'})$. Our goal is to estimate the coordinates $\boldsymbol{X}^*$ that minimizes

$$\boldsymbol{X}^* \quad = \quad \texttt{argmin}_{\boldsymbol{X}} \left(\sum_{j=1}^{J} \sum_{j' \neq j} \frac{\left(d^{\text{euc}}(\boldsymbol{x}_j, \boldsymbol{x}_{j'}) - d(\boldsymbol{c}_j, \boldsymbol{c}_{j'}) \right)^2}{d(\boldsymbol{c}_j, \boldsymbol{c}_{j'})} \right). \tag{4.7}$$

Equation 4.7 encodes our goal of preserving small distances with greater accuracy than larger distances. The denominator contains the distance between two clusterings $d(\boldsymbol{c}_j, \boldsymbol{c}_{j'})$. This implies that clusterings that are small will be given additional weight in the final embedding, while large distances will receive less consideration in the scaling, just as desired.

This appendix describes two properties of the local cluster ensemble.

Avoiding Infinite Regress via Local Cluster Ensembles We first show that the local cluster ensemble avoids the infinite regress problem. To prove this, we show that our approach is approximately invariant when replacing the k-means "meta-cluster analysis" method form local cluster ensembles with any other valid clustering method, given that we employ a sufficiently large number of methods in the original set.

Suppose we employ a valid distance metric between clusterings and apply arbitrary clustering method 1 to obtain a partition of documents based upon the weighted votes for a given point. We represent this clustering with $\boldsymbol{c}_1\left(\boldsymbol{V}(\boldsymbol{w})\right)$. Now, suppose that we want to apply a second cluster method to same weighted voting matrix $\boldsymbol{c}_2\left(\boldsymbol{V}(\boldsymbol{w})\right)$. How close can we get to $\boldsymbol{c}_1\left(\boldsymbol{V}(\boldsymbol{w})\right)$ by varying the weights in $\boldsymbol{c}_2\left(\boldsymbol{V}(\boldsymbol{w})\right)$?[1] If it is close, then we are guaranteed to find the same clusterings (and therefore, make the same discoveries) using two different clustering methods.

Let $\boldsymbol{w}^*$ represent the set of weights that minimize the distance between $\boldsymbol{c}_1\left(\boldsymbol{V}(\boldsymbol{w})\right)$ and $\boldsymbol{c}_2\left(\boldsymbol{V}(\boldsymbol{w^*})\right)$, $\boldsymbol{w}^* = \texttt{argmin}_{\boldsymbol{w}'} d(\boldsymbol{c}_1\left(\boldsymbol{V}(\boldsymbol{w})\right), \boldsymbol{c}_2\left(\boldsymbol{V}(\boldsymbol{w}')\right)$. We can guarantee that, $0 \leq d(\boldsymbol{c}_1\left(\boldsymbol{V}(\boldsymbol{w})\right), \boldsymbol{c}_2\left(\boldsymbol{V}(\boldsymbol{w^*})\right) \leq \min_j d(\boldsymbol{c}_1\left(\boldsymbol{V}(\boldsymbol{w}), \boldsymbol{c}_j\right))$ or that $\boldsymbol{c}_1\left(\boldsymbol{V}(\boldsymbol{w})\right)$ and $\boldsymbol{c}_2\left(\boldsymbol{V}(\boldsymbol{w^*})\right)$ can be no farther apart $\boldsymbol{c}_1\left(\boldsymbol{V}(\boldsymbol{w})\right)$ and any of the clusterings we have already obtained. This is because we can always place all the weight on

[1] We make the assumption that the second clustering method is full range (can provide any partition) to avoid pathological counter examples. For simplicity, we also assume that when provided with a similarity matrix that is block-diagonal (diagonal blocks are zero distance, off diagonal infinite distance) the method returns the block-diagonals as the clustering. We are unaware of any existing clustering methods that violate this assumption, although theoretical examples are possible to construct. Notice, that our assumptions are different than Kleinberg (2003), avoiding well-known impossibility results.

the clustering from an existing method. If we have included all possible clusterings for a set of documents, then $d(\boldsymbol{c}_1(\boldsymbol{V}(\boldsymbol{w})), \boldsymbol{c}_2(\boldsymbol{V}(\boldsymbol{w}^*)) = 0$, because the clustering from $\boldsymbol{c}_1(\boldsymbol{V}(\boldsymbol{w}))$ is guaranteed to be present in the collection of clusterings.

This illustrates two key points about the invariance of our method to the clustering method used in creating local cluster ensembles. First, because we use a large number of clusterings to obtain many partitions of the data, any two methods used to cluster the results are likely to yield very similar insights. Second, we recognize that we cannot enumerate all possible partitions. Therefore, we restrict our attention only to those partitions that can be expressed by a combination of the collective creativity of the various academic literatures devoted to cluster analysis.

The Local Cluster Ensemble as a Relaxed Version of the "Meet" We now show that the local cluster ensemble is a *relaxed* version of the "meet," defined in Appendix 4.1: it agrees in specific cases where we would expect correspondence, and diverges to allow *local* averages. In comparison, the original version of the meet creates a cluster ensemble that gives each component method equal weight.

We first demonstrate that the meet and the local cluster ensemble agree in specific cases. Consider two clusterings $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ and denote their meet by $\boldsymbol{c}_3 = \boldsymbol{c}_1 \times \boldsymbol{c}_2$. Recall that a pair of documents will be assigned to the same cluster in $\boldsymbol{c}_3$ if (and only if) they are assigned to the same cluster in $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$. To construct the meet using a local cluster ensemble, suppose that we assign equal weight to each method $w_1 = w_2 = 0.5$ and that the local cluster ensemble assumes the same number of clusters as found in the meet, K_3. A consequence of these assumptions is that pairs of documents assigned to the same cluster in both documents will be maximally similar. The optimal solution for k-means, applied to this similarity matrix is the meet (anything else will increase the squared error in the final clustering, and therefore not be optimal). Further, it is clear that the meet of a set of clusterings provides an upper bound on the number of clusters to be found in an ensemble: using more clusters than the meet involves splitting pairs of documents that are maximally similar into *different* clusters.

We now show how the meet relates to the local cluster ensemble in general. The meet among a set of clusterings requires unanimous agreement that a pair of documents belongs to the same

cluster (the order of the pairs is irrelevant). We show this explicitly in terms of a voting matrix to compare it more directly to the local cluster ensemble. Suppose we have J clusterings and assemble the voting matrix $\boldsymbol{V}(\boldsymbol{w})$, but suppose each clustering receives equal weight $w = \frac{1}{J}$ and obtain similarity matrix $\boldsymbol{V}(\boldsymbol{w})\boldsymbol{V}(\boldsymbol{w})'$. The meet settles disputes about which documents belong in the same clusters in the most conservative way possible: requiring unanimous agreement among the clusterings that the pairs belong together.

Rather than require unanimous agreement among all clusterings to place a pair of documents in the same cluster — which would result in highly fragmented clusterings — the local cluster ensemble employs a non-unanimous voting rule; this allows for some clusterings to exert greater influence through arbitrary weights across the methods encoded in the vote matrix $\boldsymbol{V}(\boldsymbol{w})$. We then tally the total votes for each pair belonging together with $\boldsymbol{V}(\boldsymbol{w})\boldsymbol{V}(\boldsymbol{w})'$. The meta-clustering algorithm then adjudicates disputes among the clusterings about which documents belong together.

4.4 Efficiently Sampling for Cluster Quality

Here we prove that if two clusterings agree about a pair of documents — both clusterings placing the pair together in a cluster or separately in different clusters — then it does not contribute to differences in our measure of cluster quality and so resources need not be devoted to evaluating it. Our evaluation then only needs to address pairs for which clusterings disagree. Define Y as 1 if the clusterings agree about a pair and 0 if they disagree, π_a as the proportion of pairs that agree, and $1 - \pi_a = \pi_d$ as the proportion of pairs that disagree. Then,

$$\begin{aligned} \mathrm{E}[\mathrm{CQ}(\boldsymbol{c}_j - \boldsymbol{c}_{j'})] &= \mathrm{E}\left[\mathrm{E}[\mathrm{CQ}(\boldsymbol{c}_j - \boldsymbol{c}_{j'})|Y]\right] \\ &= \pi_a \underbrace{\mathrm{E}[\mathrm{CQ}(\boldsymbol{c}_j - \boldsymbol{c}_{j'})|Y=1]}_{0} + \pi_d \underbrace{\mathrm{E}[\mathrm{CQ}(\boldsymbol{c}_j - \boldsymbol{c}_{j'})|Y=0]}_{\text{Estimated by Sampling}} \end{aligned} \tag{4.8}$$

The only piece of Equation 4.8 that is unknown is the average cluster quality among the pairs where the two clusterings disagree. We can obtain an unbiased estimate of this by randomly sampling from the pairs where the two methods disagree and then obtain an unbiased estimate of the difference in cluster quality by multiplying by the proportion of pairs where there is disagreement π_d (which is easily computed from the population of pairs).

References

Banerjee, Arindam, Inderjit Dhillon, Joydeep Ghosh and Suvrit Sra. 2005. "Clustering on the Unit Hypersphere Using von Mises-Fisher Distributions." *Journal of Machine Learning* 6:1345–1382.

Blei, David M., Andrew Y. Ng and Michael I. Jordan. 2003. "Latent Dirichlet Allocation." *Journal of Machine Learning Research* 3:993–1022.

Blei, David and Michael Jordan. 2006. "Variational Inference for Dirichlet Process Mixtures." *Journal of Bayesian Analysis* 1(1):121–144.

Brock, G, V Pihur, S Datta and S Datta. 2008. "clValid: An R Package for Cluster Validation." *Journal of Statistical Software* 25(4).

Cuesta-Albertos, JA, A Gordaliza and C Matran. 1997. "Trimmed K-Means: An Attempt to Robustify Quantizers." *Annals of Statistics* 25(553-576).

Dhillon, Inderjit. 2003. "Co-clustering Documents and Words Using Bipartite Spectral Graph Partitioning." *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* pp. 89–98.

Dhillon, Inderjit, Subramanyam Mallela and Dharmendra Modha. 2003. "Information Theoretic Co-Clustering." *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 9.

Forgy, EW. 1965. "Cluster Analysis of Multivariate Data: Efficiency vs Interpretability of Classifications." *Biometrics* 21.

Fraley, C. and A.E. Raftery. 2002. "Model-based clustering, discriminant analysis, and density estimation." *Journal of the American Statistical Association* 97(458):611–631.

Fraley, Chris. 1998. "Algorithms for Model-Based Gaussian Hierarchical Clustering." *SIAM Journal of Scientific Computing* 20(1):270–281.

Frey, BJ and D Dueck. 2007. "Clustering by Passing Messages Between Data Points." *Science* 315(5814):972.

Gan, Guojun, Chaoqun Ma and Jianhong Wu. 2007. *Data Clustering: Theory, Algorithms, and Applications*. Philadelphia: Siam.

Gath, I and AB Geva. 1989. "Unsupervised Optimal Fuzzy Clustering." *IEEE Transactions On Pattern Analysis and Machine Intelligence* 11(7):773–780.

Gelman, Andrew, J.B. Carlin, H.S. Stern and D.B. Rubin. 2003. *Bayesian Data Analysis, Second Edition*. Chapman & Hall.

Grimmer, Justin. 2010. "A Bayesian Hierarchical Topic Model for Political Texts: Measuring Expressed Agendas in Senate Press Releases." *Political Analysis* .

Guha, S, R Rastogi and K Shim. 2000. "ROCK: A Robust Clustering Algorithm for Categorical Attributes." *Information Science* 25(5).

Hastie, Trevor, Robert Tibshirani and Jerome Friedman. 2001. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York: Springer.

Heyer, LJ, S Kruglyak and S Yooseph. 1999. "Exploring Expression Data: Identification and Analysis of Coexpressed Genes." *Genome Research* 9:1106–1115.

Karayiannis, NB. 1994. MECA: Maximum Entropy Clustering Algorithm. In *The 3rd IEEE International Conference on Fuzzy Systems*. pp. 630–635.

Kaufman, Leonard and Peter Rousseeuw. 1990. *Finding Groups in Data: An Introduction to Cluster Analysis*. New York: Wiley.

Kleinberg, Jon. 2003. An Impossibility Theorem for Clustering. In *Advances in Neural Information Processing Systems Proceedings of the 2002 Conference*. pp. 463–470.

Kohonen, Teuvo. 2001. *Self-Organizing Maps*. New York: Springer.

Koyuturk, M, A Graham and N Ramakrishnan. 2005. "Compression, Clustering, and Pattern Discovery in Very High-Dimensional Discrete-Attribute Data Sets." *IEEE Transactions On Knowledge and Data Engineering* 17(4).

Leisch, Friedrich. 1999. "Bagged Clustering." Working Paper 51, Adaptive Information Systems and Modelling in Economics and Management Science.

Lerman, IC. 1991. "Foundations of the Likelihood Linkage Analysis Classification Method." *Applied Stochastic Models and Data Analysis* 7:63–76.

Mackay, David. 2003. *Information Theory, Inference, and Learning Algorithms*. Cambridge: Cambridge University Press.

McQuitty, LL. 1966. "Similarity Analysis by Reciprocal Pairs for Discrete and Continuous Data." *Educational and Psychological Measurement* 26:825–831.

Meila, M and J Shi. 2001. "A Random Walks View of Spectral Segmentation." *8th International*

Workshop on Artificial Intelligence and Statistics (AISTATS) .

Meila, Marina. 2007. "Comparing Clusterings: An Information Based Distance." *Journal of Multivariate Analysis* 98(5):873–895.

Ng, Andrew, Michael Jordan and Yair Weiss. 2002. "On Spectral Clustering: Analysis and an Algorithm." *Advances in Neural Information Processing Systems 14: Proceedings of the 2002 Conference* .

Pitman, Jim. 1997. "Some Probabilistic Aspects of Set Partitions." *The American Mathematical Monthly* pp. 201–209.

Quinn, K.M., B.L. Monroe, M. Colaresi, M.H. Crespin and D.R. Radev. 2006. "How To Analyze Political Attention With Minimal Assumptions And Costs." Annual Meeting of the Society for Political Methodology.

Rajesh, Dave. 1996. "Fuzzy Shell-Clustering and Applications to Circle Detection in Digital Images." *Internationl Journal of General Systems* 16:343–355.

Sammon, John. 1969. "A Nonlinear Mapping for Data Structure Analysis." *IEEE Transactions on Computers* 18(5):401–409.

Schrodt, P.A. and D.J. Gerner. 1997. "Empirical indicators of crisis phase in the Middle East, 1979-1995." *Journal of Conflict Resolution* pp. 529–552.

Shannon, Claude E. 1949. *The Mathematical Theory of Communication.* Urbana-Champaign: University of Illinois Press.

Shi, J and J Malik. 2000. "Normalized Cuts and Image Segmentation." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(8):888–905.

von Luxburg, Ulrike. 2007. "A Tutorial on Spectral Clustering." *Statistics and Computing* 17(4):395–416.

Wang, S, W Qiu and RH Zamar. 2007. "CLUES: A Non-Parametric Clustering Method Based on Local Shrinking." *Computational Statistics & Data Analysis* 52(1):286–298.

Zhang, Bin, Meichun Hsu and Umeshwar Dayal. 1999. K-Harmonic Means: A Data Clustering Algorithm. Technical Report HPL-1999-124 HP Laboratories.