A Bayesian Approach to Cluster Validation

Hoyt A. Koepke
Department of Computer Science
University of British Columbia
Vancouver, BC
hoytak@cs.ubc.ca

Bertrand Clarke
Department of Statistics
University of British Columbia
Vancouver, BC
riffraff@stat.ubc.ca

Abstract

In this paper, we propose a novel approach to validating clusterings. We treat a given clustering as a baseline and define a collection of perturbations of it that give possibly different assignment of points to clusters. If these are indexed by a hyperparameter, integrating with respect to a prior gives an averaged assignment matrix. This matrix can be visualized as a heat map, allowing clusterings and their stability properties to be readily seen. The difference between an averaged assignment matrix and the baseline gives a measure of the stability of the baseline. This approach motivates a general and computationally fast algorithm for evaluating the stability of distance-based and exponential-model type clusterings, including k-means. In addition, these criteria can be used to choose the optimal number of clusters. Our method compares favorably with data based perturbation procedures, such as subsampling, in some conditions such as small sample size. In addition, there is evidence that our method performs better relative to subsampling methods on some problems.

1 Introduction

Validating the solution to unsupervised learning problems, of which clustering is the most common example, is an increasingly important problem in many emerging fields. Once a candidate clustering is found, it is important to assess how well it reflects the natural structure of the data. The field of cluster validation addresses this by providing techniques for examining the stability of a clustering. While numerous algorithms exist to find clusters in many types of data, and these algorithms play an important role in many areas of research (see (Jain, Murty, & Flynn 1999) for a review of clustering algorithms), relatively few methods exist to evaluate the adequacy or stability of the resulting clustering. In this paper, we propose and demonstrate a novel approach to this problem that permits reliable and efficient methods to assess the stability of a clustering.

We define the stability of a clustering as the resistance of the clusters to perturbations; a more stable clustering is able to withstand more severe perturbations without significant changes in the point-to-cluster assignment. Previous approaches to cluster stability have focused on perturbing the data set (e.g. by adding noise to the data or by subsampling). These, while successful, have several inherent downsides, such as the label matching problem, which impede analysis of the results. In contrast, our approach is to introduce the perturbations into the clustering process itself, which allows for ways around some of these problems.

This also follows the Bayesian methodology of conditioning on the data but expressing uncertainty in the model. By perturbing parameters in the clustering function, such as the decision boundaries used for point to cluster assignment, we are expressing uncertainty about the criteria and measurements that the clustering function uses to partition the data.

While our approach is motivated in part by Bayesian theory, we show that it can yield results comparable to data-perturbation approaches at a fraction of the computational cost. Furthermore, it allows us to quickly compute a visual display of the ways clusters exchange points under perturbation, something computationally difficult, or even ill-defined, with data perturbation methods. Finally, there are numerous ways of introducing perturbations into the clustering process in a sensible way, which allows for new research into cluster stability assessment.

In our approach, we modify the clustering function to take a hyperparameter that quantitatively introduces some type of perturbation, then integrate over a prior on the hyperparameter to obtain a averaged assignment matrix. This averaged membership matrix can then be plotted using a heat map-like plot to show how points move under perturbation, providing an intuitive picture of the stability of the clustering. Furthermore, by integrating over a similarity-based index, we can naturally incorporate many well-studied scalar stability indices. To our knowledge, this approach is unique.

In section 2, we formally describe the abstract framework of our approach. In section 3, we apply it to clustering procedures based on exponential family models, including those produced by k-means. In section 4, we look at practical clustering investigation, including a heat map visualization of the averaged assignment matrix and scalar stability indices. The heat map is a novel tool for assessing the behavior of clusterings under perturbation, and we discuss using it to determine the number of clusters and demonstrate it on real data. Finally, in section 5, we present the relative performance of a scalar stability index based on our approach. The results show that our approach is comparable to existing methods under many conditions and performs markedly better under some.

1.1 Relation to previous work

Related work on measuring stability usually involves perturbing the data (Hennig 2004), (Giurcaneanu & Tabus 2004), but the method we propose relies on perturbing the clustering function itself. Methods based on subsampling (Abul *et al.* 2003), (Ben-Hur, Elisseeff, & Guyon 2002), (Smolkin & Ghosh 2003) or resampling (Lange *et al.* 2002), (Roth *et al.* 2002), (Lange *et al.* 2004), (Moller & Radke 2006) the data have yielded promising results. Other approaches include using prediction strength (Tibshirani & Walther 2005) or measuring replication and consistency across cross-validation (Breckenridge 1989), (Breckenridge 2000). Much research has also focused on using various stability criteria for determining the number of clusters, with some success (Ben-Hur, Elisseeff, & Guyon 2002). For a more comprehensive description of clustering validation, we refer the reader to (Lange *et al.* 2004), (Jiang, Tang, & Zhang 2004) or (Halkidi, Batistakis, & Vazirgiannis 2001).

Many of these data-perturbation approaches can be seen as a special case of perturbing the clustering function. For example, weighting or turning data points on and off corresponds to sub-sampling; adding random vectors to the data points corresponds to adding noise. However, we show that introducing the perturbation later in the clustering process can have advantages, namely computational efficiency and having correctly matched labels across clusterings.

Research has also focused on developing scalar measures comparing two clusterings, among them the Hubert-Arabie Adjusted Rand index (Hubert & Arabie 1985) (see (Steinley 2004) for a recent analysis) and a relatively recent one proposed by Meilă called Variation of Information (Meila 2003), (Meila 2007), both of which we revisit later. Beyond these, we refer the reader to (Maulik & Bandyopadhyay 2002), (Bezdek & Pal 1998), or (Meila 2007) for comparisons and descriptions of various stability indices. As mentioned, our method naturally incorporates such indices (see section 4.2).

Using stability indices to choose the appropriate number of clusters, often by selecting the most stable among several test cases, is also popular (Vetrov 2006). (Maulik & Bandyopadhyay 2002) and (Bezdek & Pal 1998) compare indices partly by how well they predict this. However, it is not clear how often these indices are used in practice for model selection, perhaps because interpreting the results can be difficult (Tibshirani & Walther 2005).

We suggest that one inherent downside to using scalar stability indices in practice for model selection (and perhaps other tasks) is the fact that the behavior of the clustering – i.e. *why* a particular index indicates that one clustering is more stable than another – is, in part, hidden from from the user. While an overall or per-cluster assessment of the stability of a clustering can be quite useful, providing additional relevant and accurate information gives the user more confidence in the procedure and the stability of the clustering as a whole. Our method attempts to provide such information by indicating not only *how* stable the clusters are, but also *why* by showing the behavior of the clustering under perturbation. We hope that our method provides an intuitive way to assess cluster stability and behavior in a variety of applications and a starting point from which to develop further tools.

2 Framework

The abstract framework we propose can, at a high level, be applied universally to almost any clustering function. Our definitions make no assumptions about how the hyperparameter affects the clustering function; the details will vary based on the clustering algorithm and what type of stability one wishes to assess.

Suppose $\mathscr{C}(K, x)$ is a clustering function that partitions a set of n data $x = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$ into a set $\{C_1, C_2, ..., C_K\}$ of K clusters based on some structural feature of the data. Our approach is to create a new clustering function $\mathscr{C}^\star(K, x, \boldsymbol{\lambda})$ by modifying $\mathscr{C}(K, x)$ to take a hyperparameter $\boldsymbol{\lambda}$, where the role of $\boldsymbol{\lambda}$ is (informally) to perturb some relevant aspect of the clustering. We then define a prior distribution $\pi(\boldsymbol{\lambda})$ for $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}$ over the hyperparameter.

Without loss of generality, let $\mathscr{C}(K, x)$ return an $n \times K$ assignment matrix $\mathbf{A} = [a_{ij}]$ such that a_{ij} equals 1 if $\mathbf{x}_i$ is in cluster C_j and 0 otherwise. Likewise, suppose that, for a given $\boldsymbol{\lambda}$, $\mathscr{C}^\star(K, x, \boldsymbol{\lambda})$ returns a similarly defined matrix $\mathbf{A}^\star(\boldsymbol{\lambda}) = [a^\star_{ij}(\boldsymbol{\lambda})]$, where we explicitly denote the dependence on $\boldsymbol{\lambda}$. We treat x and K as constants since variations in x can be incorporated into the perturbation indexed by $\boldsymbol{\lambda}$ and multiple K's can be handled by allowing empty clusters.

Integrating $a^\star_{ij}(\boldsymbol{\lambda})$ over λ_j with respect to $\pi(\boldsymbol{\lambda})$ gives us an $n \times K$ matrix $\boldsymbol{\Phi}$ that expresses the average membership of $\mathbf{x}_i$ in C_j under perturbation. (Here we assume that the class labels are matched up; in section 2.1 we discuss relaxing this condition.) Formally, $\boldsymbol{\Phi}$ can be expressed as a Riemann-Stieltjes integral (Kestelman 1960), which handles both continuous and discrete $\boldsymbol{\Lambda}$:

$$\boldsymbol{\Phi} = [\phi_{ij}] = \int_{\boldsymbol{\Lambda}} \mathbf{A}^\star(\boldsymbol{\lambda}) \mathrm{d}\Pi(\boldsymbol{\lambda}) \tag{1}$$

The integration spreads the binary membership matrix $\mathbf{A}^\star(\boldsymbol{\lambda})$ across the clusters; thus ϕ_{ij} can take on any value between 0 and 1. Since $\sum_j \phi_{ij} = 1$, each row of $\boldsymbol{\Phi}$ can be interpreted as a probability vector, with ϕ_{ij} indicating the probability that datum $\mathbf{x}_i$ belongs to cluster j given the prior.

We can define the averaged matching matrix $\mathbf{M} = [m_{jj'}]$ in terms of $\boldsymbol{\Phi}$:

$$\mathbf{M} = \mathbf{A}^T \boldsymbol{\Phi} \Leftrightarrow m_{jj'} = \sum\nolimits_i a_{ij}\phi_{ij'} = \sum\nolimits_{i:a_{ij}=1} \phi_{ij'} \tag{2}$$

In this matrix, $m_{jj'}$ represents the total point-mass (each point having a mass of 1) in the baseline cluster C_j that moves to cluster C'_j under perturbation. Based on the probabilistic interpretation of $\boldsymbol{\Phi}$, $m_{jj'}/|C_j|$ (normalizing $\mathbf{M}$ across rows) is the probability of a point in the baseline cluster j belonging to cluster j' under perturbation. Likewise, $m_{jj'}/n$ is the probability that a randomly-selected point belongs to cluster j in the baseline clustering and to cluster j' under perturbation.

2.1 Perturbations and label matching

The clustering function above can be described as a two stage process, as illustrated in Figure 1; formally, $\mathscr{C}(K,x) = \mathscr{C}_P(\mathscr{C}_S(K,x))$. The first stage, $\mathscr{C}_S$, processes the data and outputs information (statistics) about the clustering (e.g. centroids in the case of k-means). The second stage, $\mathscr{C}_P$, uses this information to partition the data points into clusters. We put no constraints on the form of the information (e.g. it could be an assignment, making the latter partitioning step trivial), so this description is sufficiently general.

If the perturbation is introduced before or into $\mathscr{C}_S$, i.e. $\mathscr{C}^\star(K,x,\boldsymbol{\lambda}) = \mathscr{C}_P(\mathscr{C}_S^\star(K,x,\boldsymbol{\lambda}))$, the cluster labels may not be correctly matched between runs. As a trivial example, two runs of random-start k-means may produce identical clusterings, but the points associated with one centroid in the first run may be associated with a differently labeled centroid in the second.

The label matching problem is inherent in the data perturbation methods and has received some attention. Following (Breckenridge 2000) and (Lange *et al.* 2004), we can permute the labels to maximize the similarity between the baseline clustering and the perturbed clustering. Formally, we can introduce a $K \times K$ permutation matrix $\mathbf{P}(\boldsymbol{\lambda})$ to express this. Equation (1) then becomes:

$$\boldsymbol{\Phi} = [\phi_{ij}] = \int_{\boldsymbol{\Lambda}} \mathbf{A}^\star(\boldsymbol{\lambda})\mathbf{P}(\boldsymbol{\lambda})\mathrm{d}\Pi(\boldsymbol{\lambda}) \tag{3}$$

where

$$\mathbf{P}(\boldsymbol{\lambda}) = \underset{\mathbf{Q}\in\boldsymbol{\mathcal{P}}}{\operatorname{argmax}}\ \operatorname{trace}\left[\mathbf{A}^T\mathbf{A}^\star(\boldsymbol{\lambda})\mathbf{Q}\right] \tag{4}$$

The definition for equation (2) which follows remains unchanged. This corresponds to solving a bipartite graph matching problem between the two sets of clusters labels, where the edge weights are proportional to the number of shared points. Finding the optimal $\mathbf{P}(\boldsymbol{\lambda})$ can be done in $O(K^3)$ time using the Hungarian method (Kuhn 1955) or network flow simplex (Chvtal 1983). While this is tractable when testing a small set of $\boldsymbol{\lambda}$, it still imposes a significant problem.

One way to avoid this problem is to rely on scalar stability indices such as those mentioned in section 1.1, most of which are, by intention, invariant to label permutations. We discuss using these indices within our framework in section 4.2; however, as discussed, these provide a limited summary of cluster stability.

Our framework provides an alternative way around the problem by introducing the perturbation and hyperparameter into the $\mathscr{C}_P$ step, so $\mathscr{C}^\star(K,x,\boldsymbol{\lambda}) = \mathscr{C}_P^\star(\mathscr{C}_S(K,x),\boldsymbol{\lambda})$. Because the cluster statistics are already calculated, perturbations will not mix up the labels. This avoids the label matching problem completely, something not possible with data-perturbation methods.

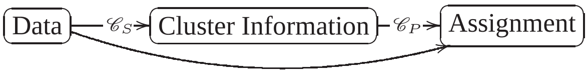

Figure 1: The clustering process.

3 Application to exponential family models

In this section, we examine introducing perturbations into the assignment stage of the clustering function, $\mathscr{C}_P$, when $\mathscr{C}_S$ produces exponential family models (Barndorff-Nielsen 1978) that can be parameterized in the form $\exp[-d_{ij} - g_j]$, where d_{ij} is a distance metric and g_j are other model parameters. Ignoring ties, which in practice can be broken randomly, a point i is assigned to cluster j if $\exp[-d_{ij} - g_j] \geq \exp[-d_{i\ell} - g_\ell]\ \forall\,\ell$. In the case of clusters defined by centroids (e.g. k-means), $d_{ij} = \|\mathbf{x}_i - \boldsymbol{\mu}_j\|_2$ (assuming an L2 norm) and $g_j = \text{const}$ gives the correct assignment. For a weighted mixture of multivariate Gaussians with hard assignment, $d_{ij} = \frac{1}{2}(\mathbf{x}_i - \boldsymbol{\mu}_j)^T\boldsymbol{\Sigma}_j^{-1}(\mathbf{x}_i - \boldsymbol{\mu}_j)$ and $g_j = \frac{1}{2}\log(w_j/|2\pi\boldsymbol{\Sigma}_j|)$.

We tested two ways of introducing perturbations into such models. The first was to scale the distance to each cluster center d_{ij} by a hyperparameter so $d_{ij} \to d_{ij}\lambda_j$. The second approach is to add the hyperparameter so $d_{ij} \to d_{ij} + \lambda_j$. Because the distance metrics determine the assignment of points to clusters, we suggest these are reasonable ways to introduce the perturbation. Another feature of this approach is that the computational cost is independent of the dimension of the data if the d's and g's are already calculated.

For the multiplicative case, the averaged assignment under perturbation is then given by:

$$\phi_{ij} = \int_S \pi(\boldsymbol{\lambda})\mathrm{d}\boldsymbol{\lambda} = \int_S \prod_\ell \pi_\ell\left(\lambda_\ell\right)\mathrm{d}\lambda_\ell \tag{5}$$

where $S = \{\boldsymbol{\lambda} : \lambda_j d_{ij} + g_j \leq \lambda_\ell d_{i\ell} + g_\ell\}$ is the region of integration and we assume independence between the hyperparameters, allowing us to factor the prior. The corresponding equation for the additive case is similar, differing only in that the region of integration S becomes $\{\boldsymbol{\lambda} : d_{ij} + g_j + \lambda_j \leq d_{i\ell} + g_\ell + \lambda_\ell\}$.

To illustrate this, suppose the g terms are constant and suppose $\mathbf{x}_i$ is assigned to cluster j in the unperturbed assignment. If we hold all the hyperparameters fixed except the one associated with cluster j, then, as we increase λ_j, the maximum value of d_{ij} such that $\mathbf{x}_i$ stays in cluster j decreases. If d_{ij} is small relative to $d_{i\ell}, \ell \neq j$, it stays assigned to C_j for higher values of λ_j; if d_{ij} is roughly the same, it moves to another cluster more quickly. If we let λ_j have an exponential distribution, so $\pi_\ell(\lambda_\ell) = ae^{-a\lambda_\ell}\mathbf{1}_{\{\lambda_\ell \geq 0\}}$, equation (5) can be integrated directly, giving

$$\phi_{ij} = \frac{d_{ij}^{-1}}{d_{ij}^{-1} + \sum_{\ell\neq j} d_{i\ell}^{-1}}. \tag{6}$$

This perturbation-induced soft assignment gives, informally, a measure of the competition between the cluster centers; tighter competition implies a less stable solution.

The other multiplicative prior we evaluate in this paper is a Gamma distribution with the shape parameter α set to 2. In this case, evaluating the integral becomes more tedious, and we derive an algorithm in appendix (A) to calculate $\mathbf{\Phi}$ analytically. The same technique can be used to derive equation (6) for non-constant g's.

Additive perturbation allows us to focus on the boundary regions where the distances terms are close to each other. In this paper, we use $\pi_\ell(\lambda_\ell) = ae^{-a\lambda_\ell}\mathbf{1}_{\{\lambda_\ell \geq 0\}}$ as the prior distribution for additive perturbations as well. In this case, ϕ_{ij} can be evaluated analytically; the derivation follows the technique in appendix (A) mentioned above. Let $\Delta_{i\ell} = d_{i\ell^\dagger} + g_{\ell^\dagger}$ for $\ell = 1, ..., K$, with the † indicating a permutation of $1, ..., K$ chosen so $\Delta_{i1} \leq \Delta_{i2} \leq \cdots \leq \Delta_{iK}$. Then

$$\phi_{ij^\dagger} = \frac{1}{j}\gamma_{ij} - \sum\nolimits_{\ell=j+1}^{K} \frac{\gamma_{i\ell}}{\ell(\ell-1)} \qquad (7)$$

where

$$\gamma_{i\ell} = \exp\left[-a\left(\ell\Delta_{i\ell} - \sum\nolimits_{m=1}^{\ell}\Delta_{im}\right)\right]. \qquad (8)$$

In this case, the scaling a parameter controls the strength of the perturbation, whereas a drops out in the multiplicative case above. In equations (7) and (8), as a increases, the perturbation becomes negligible and the averaged assignment matrix approaches the 0-1 baseline assignment. As a decreases, the differences between points relative to the perturbation becomes negligible, causing ϕ_{ij} to approach $1/K$. We found this method to be particularly useful in discovering interactions between clusters, which we discuss in the next section.

4 Assessing clusterings using $\mathbf{\Phi}$

Once a clustering is found, one should ask how well it reflects the natural structure of the data before trying to interpret it in the context of the experiment. Ultimately, techniques for cluster validation need to aid in answering two overlapping questions regarding quantity and quality. First, does the number of proposed clusters accurately reflect the data (Maulik & Bandyopadhyay 2002)? Second, how representative of the modes of the underlying distribution is the clustering? For example, one might want to know how well the data supports two clusters being distinct or whether all the significant modes in the data are represented by a cluster. In this section, we describe two ways of using our method to help answer these questions.

4.1 Visualizing and interpreting $\mathbf{\Phi}$

We present here a simple way to intuitively visualize the behavior of a clustering, i.e. the way clusters give and take points under perturbation, by plotting a rearranged form of $\mathbf{\Phi}$ as a heat map. This heat map requires the cluster labels to be matched up; thus it is computationally difficult using data perturbation methods or by introducing the perturbation before the cluster statistics are calculated, but it feasible when the perturbation is introduced after the cluster statistics are generated.

Given the averaged assignment matrix and the baseline assignment, we construct the heat map as follows. We construct an index mapping on the rows (i's) of $\mathbf{\Phi}$ that rearranges the rows of $\mathbf{A}$ so that the nonzero elements are all in contiguous blocks along the columns and these arranged in descending order along the rows. Within the blocks, we permute the rows so the elements in the block $\mathbf{\Phi}$ in descending order. Thus the blocks along the "diagonal" denote the stability of points relative to their assigned clusters under perturbation, and the "off-diagonal" blocks represent membership in other clusters under perturbation. Each row shows

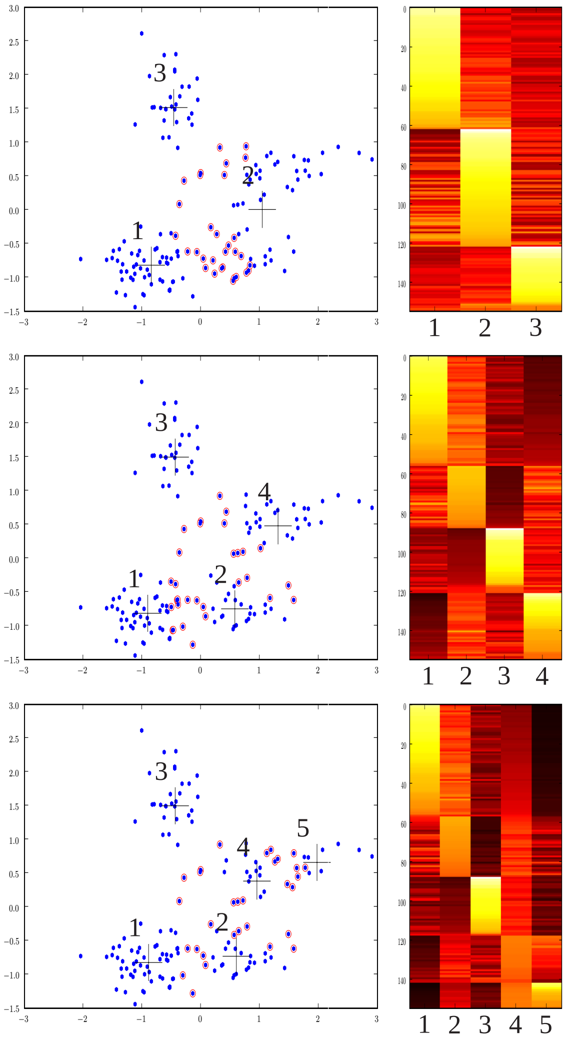

Figure 2: Use of the heat map for investigation the structure of the data. We drew 150 data points from a mixture of four Normals. The mixing weights were (0.2, 0.25, 0.35, 0.2) and the centers were at (-2, 5), (2, 1.5), (-3, -3), and (1,-3). The covariances were all different and each had nonzero correlation. Using k-means, the data were clustered with three (top), four (middle), and five (bottom) centroids. Setting $a = 0.07$ in the additive perturbation equation (7), we calculated the averaged assignment matrices for the three cases. These lead to the heat maps on the right. White represents stability; black represents instability. These heat maps allow us to deduce that the correct number of clusters is four. Additionally, the least stable 20% of the points, as measured by $\phi_{ij} - \max_{\ell \neq j} \phi_{i\ell}$, are circled in red and track the boundaries between the clusters.

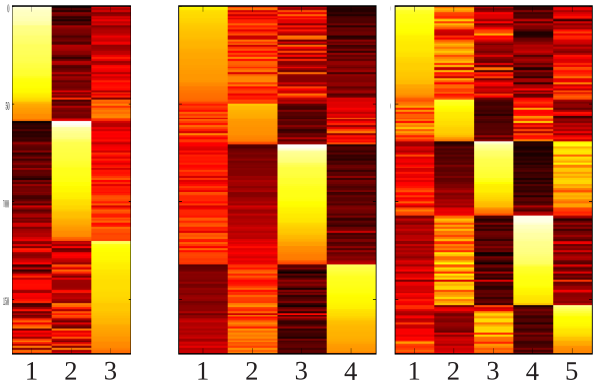

Figure 3: The behavior of clusterings in the wine data set with three (left), four (middle), and five (right) clusters under additive perturbation with a $\mathcal{E}xp(0.012)$ prior. White indicates stability relative to a cluster and black indicates little or no assignment to a cluster under perturbation.

how the corresponding point behaves when the cluster is perturbed; the point mass in unstable points will move away from the cluster block and into other clusters, as indicated by the red and orange colors to the side of such blocks. Dark red or black indicates separation between clusters. This allows us to visualize clearly how clusters exchange points when perturbed. Note that all comparisons must be done keeping in mind that only the rows are normalized; comparisons within columns are not on the same scale and, though suggestive, are not formally justified.

While much of the previous work on cluster analysis has focused on using stability indices to select the number of clusters, (see related work in section 1.1), the averaged assignment matrix provides a more intuitive window into the behavior of the clustering under perturbation. Indeed the heat map of $\mathbf{\Phi}$ indicates not only which points are unstable relative to their assigned clusters, but which clusters and points cause the instability. With too many clusterings, we should see mutual instability between some of the clusters, indicating they could be merged; with too few, we should see a lack of separation between the clusters.

An example of this using a simple 2-d case is shown in Figure 4. On the top, there isn't significant separation between any of the clusters as denoted by the orange and red throughout the diagonal blocks. In the four cluster case, cluster pairs (1,2) and (2,4) still exchange some point mass, but there's significant separation between clusters as shown by the darker squares. With five clusters, we still have significant separation between some of the clusters, but clusters 4 and 5 exchange significant point mass under perturbation, indicating mutual instability. Thus we conclude the correct number of clusters is four.

Figure 3 illustrates the stability heat map on the wine data set, analyzed by (Aeberhard, Coomans, & de Vel 1992). The data set contains the results of a chemical analysis of 178 wines grown in the same region in Italy but derived from three different cultivars. The features in the data set are the quantities of 13 constituents found in each of the three types of wines. The dataset is a good way to test clustering algorithms as the three different classes are reasonably well separated. To demonstrate our heat map, we normalized the data and clustered it into three, four, and five clusters using k-means, perturbed the assignment using the additive exponential method, and plotted the result.

There is some instability with three clusters, but in the four cluster plot, clusters 1 and 2 exchange a lot of point mass and are less stable overall than the others, indicating mutual instability. The effect is more apparent with five clusters, where clusters pairs (1, 2) and (3, 5) are mutually unstable, signifying too many clusters. Thus we conclude that, of these test cases, three clusters best fits the data (which is correct).

4.2 Scalar stability indices

Within our framework, we can naturally incorporate many of the scalar stability indices referred in section 1.1. Suppose $\text{sim}\left(\boldsymbol{\mathcal{C}}_1, \boldsymbol{\mathcal{C}}_2\right)$ is an index that compares two clusterings $\boldsymbol{\mathcal{C}}_1$ and $\boldsymbol{\mathcal{C}}_2$ and returns a scalar assessment of their similarity (see section section 1.1). We can then integrate over the similarity index to obtain an averaged scalar stability index:

$$\text{sim}^{\star}\left(\mathscr{C}(K,x)\right) = \int_{\mathbf{\Lambda}} \text{sim}\left(\mathscr{C}(K,x), \mathscr{C}^{\star}(K,x,\boldsymbol{\lambda})\right) \mathrm{d}\Pi(\boldsymbol{\lambda}) \quad (9)$$

We present here two scalar assessments of cluster stability. The first is the Hubert-Arabie Adjusted Rand index and the second is the Variation of Information index, both described in section 1.1. These can be expressed in a form that, based on the probabilistic nature of $\mathbf{\Phi}$, allows us to bypass the integration in equation (9) given $\mathbf{\Phi}$ and $\mathbf{\Lambda}$.

Let $p_j = \left|C_j\right|/n$ denote the probability that a datum $\mathbf{x}_i$ is assigned to cluster j in the baseline clustering and cluster j' under perturbation, and let $p_{jj'} = m_{jj'}$ and $p'_{j'} = \left[\sum_i \phi_{ij'}\right]/n$ be as defined in section 2. Then, the integration allows us to use the asymptotic form of the Hubert-Arabie Adjusted Rand index (Meila 2007), which becomes:

$$\begin{aligned} \mathcal{AR}^{\star}(\mathscr{C}(K,x)) &= \int_{\mathbf{\Lambda}} \mathcal{AR}(\mathscr{C}(K,x), \mathscr{C}^{\star}(K,x,\boldsymbol{\lambda})) \mathrm{d}\Pi(\boldsymbol{\lambda}) \\ &= \frac{\sum_j \sum_{j'} {p_{jj'}}^2 - \left(\sum_j {p_j}^2\right)\left(\sum_{j'} {p'_{j'}}^2\right)}{\frac{1}{2}\left[\left(\sum_j {p_j}^2\right) + \left(\sum_{j'} {p'_{j'}}^2\right)\right] - \left(\sum_j {p_j}^2\right)\left(\sum_{j'} {p'_{j'}}^2\right)} \end{aligned} \quad (10)$$

Meilă's Variation of Information is:

$$\begin{aligned} \mathcal{VI}^{\star}(\mathscr{C}(K,x)) = &- \sum_j p_j \log p_j - \sum_{j'} p'_{j'} \log p'_{j'} \\ &- 2 \sum_j \sum_{j'} p_{jj'} \log \frac{p_{jj'}}{p_j p'_{j'}} \end{aligned} \quad (11)$$

where all summations are from 1 to K. $\mathcal{AR}(\boldsymbol{\mathcal{C}}_1, \boldsymbol{\mathcal{C}}_2)$ ranges between 0 and 1, with 1 indicating stability, whereas $\mathcal{VI}(\boldsymbol{\mathcal{C}}_1, \boldsymbol{\mathcal{C}}_2)$ is a metric, with 0 indicating a perfect match. Note that these indices are independent of the ordering on the cluster labels; thus they provide a way around the label matching problem (see section 2.1). For the derivations and interpretation of equations (10) and (11), we refer the reader to (Hubert & Arabie 1985) and (Meila 2003).

5 Verification and testing

In our tests, we compare a scalar stability index based on multiplicative perturbation using an exponential prior and a gamma prior with one based on data perturbation using subsampling. In general, our method performs reasonably well. In higher dimensions, there is evidence that it outperforms data perturbation approaches, particularly when the size of the clusters is small.

We found that the Variation of Information index performed slightly worse than the Hubert-Arabie Adjusted Rand index, but that may be because it is more difficult to heuristically compare across different numbers of clusters. Also, while the additive method is best at creating meaningful heat maps, we have not yet found a way to automatically tune the free parameter a to allow us to meaningfully compare the resulting stability indices across different k; thus we leave that method out of the comparison as well.

To generate the synthetic data, we sampled 100 points from an equally weighted mixture of k_{true} $\mathcal{N}\left(\mu, \sigma^2_{\text{within}}\mathbf{I}\right)$ Gaussians, with $\mu \overset{\text{iid}}{\sim} \mathcal{N}\left(0, \sigma^2_{\text{between}}\mathbf{I}\right)$, and the cluster size at 5 or above. The free parameters are the dimension, the number of clusters k_{true}, and the ratio of within cluster variance (σ^2_{within}) to between cluster variance ($\sigma^2_{\text{between}}$). A higher $\sigma^2_{\text{within}}/\sigma^2_{\text{between}}$ ratio means the points in the clusters are more spread out relative to the cluster centers and the problem becomes more difficult.

Each test consists of running k-means on a sampled data set with $k = 2, ..., 10$ centroids. We then perturb the resulting cluster models using one of the methods described in section 3 and use equation (10) to generate a measure of stability. The estimate of k_{true}, $\hat{k}$, is the most stable.

We also compare our method against data perturbation methods using subsampling (see section 1.1). The subsampling method compares, using a stability index, the clustering on the full data against clusterings on data where 30%, 40%, 50%, 60%, and 70% of the original points have been discarded at random. We then use the median of the final list of stability indices in estimating k_{true}.

For each given dimension and $\sigma^2_{\text{within}}/\sigma^2_{\text{between}}$, we calculate $\hat{k}$ on 750 distributions for each of $k_{\text{true}} = 2, ..., 10$. We then plotted the mean absolute deviation of $\hat{k}$ from k_{true} as a function of $\sigma^2_{\text{within}}/\sigma^2_{\text{between}}$, as shown in Figure 4, for 15, 100, and 500 dimensions. In all three cases, our $\mathcal{G}a(2, \beta)$ method outperformed the subsampling based method until the problem becomes significantly difficult.

The multiplicative perturbation with the $\mathcal{G}amma(2, \beta)$ prior and using $\mathcal{AR}^\star$ generally performed the best. We suspect that the exponential prior performs worse than the $\mathcal{G}amma(2, \beta)$ prior because the former puts more weight on multiplicative perturbations close to zero which may falsely convey stability when all the distances are compared. The data perturbation approach still tended to be better on the hardest problems, but only when the points within the clusters were spread out enough to overlap significantly. At that point, it becomes questionable whether k_{true} accurately reflects the structure of the data and the accuracy of the test is debatable.

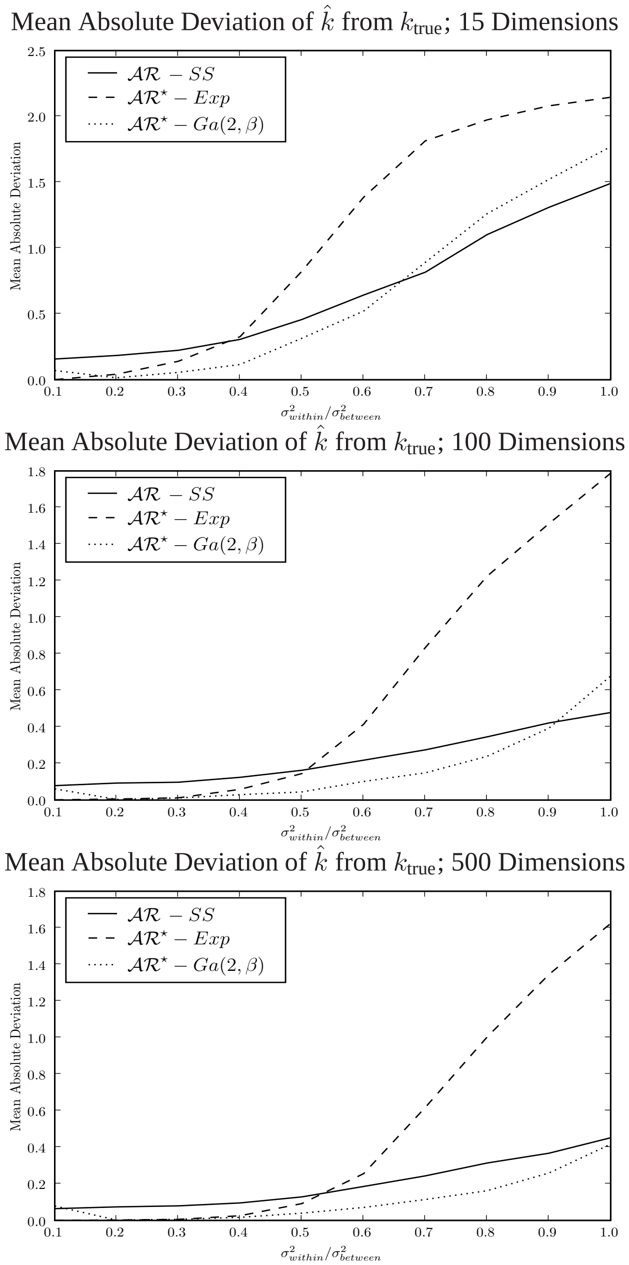

Figure 4: The mean absolute deviation of the three methods described in the text as a function of increasing $\sigma^2_{\text{within}}/\sigma^2_{\text{between}}$. $\mathcal{AR} - SS$ is the subsampling method and $\mathcal{AR}^\star - Exp$ and $\mathcal{AR}^\star - Ga(2, \beta)$ use multiplicative perturbation with an exponential prior and a $\mathcal{G}a(\alpha = 2, \beta)$ prior, respectively. Our methods, particularly $\mathcal{G}a(2, \beta)$, outperform the subsampling methods until $\sigma^2_{\text{within}}/\sigma^2_{\text{between}}$ becomes significantly large and the clusters tend to overlap.

Note that the subsampling method never achieves 100% accuracy, even on the easiest problems. This is likely because of the small number of data points in many of the clusters; subsampling throws away information on each trial and by doing so may miss smaller clusters. In these cases, our method has an advantage as it considers all the available

$\mathcal{AR}$ using data subsampling.

$\hat{k} \longrightarrow$ $k_{\text{true}} \downarrow$	2	3	4	5	6	7	8	9	10
3	.08	.43	.09	.16	.09	.06	.04	.02	.02
4	.04	.17	.29	.26	.12	.06	.02	.02	.01
5	.05	.10	.24	.35	.10	.08	.03	.02	.02
6	.05	.12	.19	.16	.29	.10	.06	.02	.02
7	.05	.11	.17	.19	.23	.09	.04	.06	.05
8	.08	.07	.20	.16	.17	.16	.06	.07	.02
9	.04	.14	.16	.17	.16	.13	.05	.09	.06

$\mathcal{AR}^{\star}$ with Exponential Prior.

$\hat{k} \longrightarrow$ $k_{\text{true}} \downarrow$	2	3	4	5	6	7	8	9	10
3	.06	.34	.43	.09	.04	.01	.01	.01	.01
4	.06	.13	.42	.26	.06	.03	.01	.01	.02
5	.04	.05	.11	.24	.44	.06	.03	.02	.01
6	.06	.06	.08	.37	.22	.10	.05	.03	.03
7	.05	.03	.04	.07	.12	.24	.20	.15	.12
8	.11	.05	.05	.06	.09	.12	.33	.13	.08
9	.14	.05	.04	.05	.06	.09	.29	.13	.14

$\mathcal{AR}^{\star}$ with a $\mathcal{G}amma(\alpha = 2, \beta)$ prior.

$\hat{k} \longrightarrow$ $k_{\text{true}} \downarrow$	2	3	4	5	6	7	8	9	10
3	.04	.36	.41	.09	.04	.02	.01	.01	.01
4	.02	.14	.43	.27	.06	.03	.02	.02	.01
5	.03	.03	.11	.27	.43	.07	.03	.02	.01
6	.05	.05	.08	.39	.23	.09	.05	.03	.03
7	.03	.02	.02	.05	.13	.23	.21	.17	.12
8	.10	.05	.04	.05	.09	.12	.35	.12	.08
9	.09	.05	.04	.05	.05	.09	.34	.15	.15

Table 1: The performance of three methods on 500 dimensional data drawn from a hierarchical Gaussian mixture model described in the text. For a given $k_{\text{true}} \in \{3, ..., 9\}$, we generate 750 distributions and estimate the number of clusters on each. The entries within a row given the resulting distribution of $\hat{k}$. Larger entries close to the diagonal indicate better performance. In this test, our methods compare favorably to the data perturbation methods. When the number of clusters is small, so each cluster is large, data perturbation is comparable to ours. As the number of clusters grows, so the average number of points in a cluster decreases, our method begins to greatly outperform data perturbation methods, as shown by the elements close to the diagonal in the resulting tables.

information.

In higher dimensions, the ratio of any two random distance measures approaches 1 (Hinneburg, Aggarwal, & Keim 2000), and this is often referred to as the "curse of dimensionality," as it makes clustering and similar procedures much more difficult. If the clusters are not well separated, spurious clusterings cause all three methods to break down. The two cases we looked at in which all three methods perform well enough to be compared are when the clusters are uniformly Gaussian, as in our generated data, and when the clusters are separated enough to make them well defined.

In our test of the latter case, our approach significantly outperforms data subsampling. We generated clusters using a hierarchical model, where distribution is drawn from a mixture of $\mathcal{N}\left(\mu, \operatorname{diag}\left(\sigma^2{}_1, \sigma^2{}_2, ..., \sigma^2{}_n\right)\right)$, where $\sigma_i^2 \overset{\text{iid}}{\sim} \mathcal{G}a(2\tau_k, 1/2)$ and $\tau_k^2 \overset{\text{iid}}{\sim} \mathcal{G}a(0.2, 1/2)$. This causes significant variance in the size and shape of the clusters, which is what one would expect in real data. We chose these parameters to maximize the average variance of the clusters while still keeping the results comparable. Increasing the average variance beyond this causes the error rate of all the methods to fall apart and perform little better than random. Table 1 shows the performance of our method as a function of $\hat{k}$ and k_{true}.

While the above tests, by themselves, are insufficient to establish our method for wide use on real data, they show that it holds potential to match or outperform existing data perturbation methods. More work is needed to understand the ways of introducing perturbation and the conditions under which our method performs reliably, and more tests are needed on more difficult and varied data. However, it is worthwhile to note that the subsampling method requires multiple runs of the clustering algorithm, making it more computationally expensive. Furthermore, the above tests use only the scalar stability indices; the heat map gives additional information which we ignore here for the purpose of comparison. In assessing clusterings on real data, we suggest that the heat map is far more useful than scalar stability indices.

6 Conclusion

In this paper, we have proposed a novel and promising framework based on perturbing the clustering function from which to develop new methods for cluster validations. We introduce a new tool for investigating the behavior of clusterings under perturbations in the form of a heat map plot of the averaged assignment matrix and show how existing stability indices can be incorporated into our framework. We propose three specific methods and demonstrate that they compare favorably against existing methods under some conditions.

Future work includes developing the theory around optimally choosing the type of perturbation and the prior, which would likely give better results than those presented here. Also, stability data from our method could be used by a clustering algorithm to improve accuracy. Finally, the technique could naturally be extended to other types of clusterings, such as graph clusterings, or other unsupervised learning problems.

7 Acknowledgments

The authors would like to thank Kevin Murphy for his helpful feedback along the way and Marius Muja for his work with the clustering procedure.

A Multiplicative perturbations under a $\mathcal{G}amma(\alpha = 2, \beta)$ prior

Calculating equation (5) for multiplicative perturbations and a $\mathcal{G}amma(\alpha=2,\beta)$ prior is reasonably straightforward albeit messy. We outline the key steps of the derivation here but omit the tedious algebra. First, let the g's be in sorted order, so $g_1 \leq g_2 \leq \cdots \leq g_K$. In practice, this can be accomplished by mapping the indices before and remapping them afterwards. The derivation for equation (7) is identical, except that the prior is different, the $g_\cdot$'s become $d_\cdot + g\cdot$, and d's become 1. We can rewrite equation (5) (keeping the prior general for convenience) as:

$$\phi_{ij}=\int\prod_{\ell}\pi(\lambda_\ell)\delta((\lambda_\ell d_{i\ell}+g_\ell)-(\lambda_j d_{ij}+g_j)-t_\ell)\mathrm{d}t_\ell \mathrm{d}\boldsymbol{\lambda} \quad (12)$$

where $\delta(\cdot)$ is the Dirac delta function, defined such that $\int_a^b \delta(x-t)\mathrm{d}t$ equals one if $a \leq x \leq b$ and zero otherwise, so $\mathbf{1}_{\{a\leq b\}} = \int_0^\infty \delta(b-a-t)\mathrm{d}t$, and $f(x) = \int f(t)\delta(x-t)\mathrm{d}t$ (Arfken, Weber, & Ruby 1996). Using this,

$$\phi_{ij}=\int\pi(\lambda_j)\prod_{\ell\neq j}\int\pi\left(\frac{1}{d_{i\ell}}(t_\ell+d_{ij}\lambda_j+g_j-g_\ell)\right)\frac{1}{d_{i\ell}}\mathrm{d}t_\ell \mathrm{d}\lambda_j \quad (13)$$

We set the shape parameter α to 2 to make the derivation tractable while still preserving the desired shape. Because of the comparison, the final result is independent of the scale parameter β, so we set $\beta = 1$ for convenience. Defining $\xi_{ij\ell}(\lambda_j) = (d_{ij}\lambda_j + g_j - g_\ell)/d_{i\ell}$ for convenience, we have

$$\begin{aligned}\phi_{ij} &= \iint \lambda_j e^{-\lambda_j}\prod_{\ell\neq j}[t_\ell + \xi_{ij\ell}(\lambda_j)]\,e^{-t_\ell-\xi_{ij\ell}(\lambda_j)} \\ &\qquad \times\ \mathbb{I}\left[t_\ell + \xi_{ij\ell}(\lambda_j) \geq 0\right]\mathrm{d}t_\ell \mathrm{d}\lambda_j \qquad (14)\\ &= \int \lambda_j e^{-\lambda_j}\prod_{\ell\neq j}\Big[(1+\xi_{ij\ell}(\lambda_j))e^{-\xi_{ij\ell}(\lambda_j)}\Big]^{\mathbb{I}[\xi_{ij\ell}(\lambda_j)\geq 0]}\mathrm{d}\lambda_j\end{aligned}$$

To evaluate the integral, we can use the division points at $\xi_{ij\ell}(\lambda_j) = 0 \Leftrightarrow \lambda_j = (g_\ell - g_j)/d_{ij}$ to break it up into at most $K+1$ discrete regions and integrate each separately. Let $h_{ij}^{\ell} = (g_\ell - g_j)/d_{ij}$ denote the division points (recall that $g_:$ is in sorted order), and for convenience, let $h_{ij}^{K+1} = \infty$. Note that $h_{ij}^{j} = 0$, so we only need to integrate between the division points $h_{ij}^{j}, h_{ij}^{j+1}, ..., h_{ij}^{K+1}$ to calculate ϕ_{ij}:

$$\phi_{ij} = \sum_{m=j}^{K}\int_{h_{ij}^m}^{h_{ij}^{m+1}}\lambda_j e^{-\lambda_j}\prod_{\substack{\ell=1\\ \ell\neq j}}^{m}(1+\xi_{ij\ell}(\lambda_j))e^{-\xi_{ij\ell}(\lambda_j)}\mathrm{d}\lambda_j \quad (15)$$

We use $\gamma_{ij} = d_{ij}\lambda_j + g_j$ to transform the variable of integration, yielding:

$$\begin{aligned}\phi_{ij} &= \sum_{m=j}^{K}\int_{g_m}^{g_{m+1}}\prod_{\substack{\ell=1\\ \ell\neq j}}^{m}\left[1-\frac{g_\ell}{d_{i\ell}}+\frac{\gamma_{ij}}{d_{i\ell}}\right]\left[\frac{\gamma_{ij}-g_j}{d_{ij}}\right] \\ &\qquad \times\exp\left[-\sum_{\ell=1}^{m}\frac{\gamma_{ij}-g_\ell}{d_{i\ell}}\right]\frac{1}{d_{ij}}\mathrm{d}\gamma_{ij}. \\ &= \frac{1}{d_{ij}}\sum_{m=j}^{K}\int_{g_m}^{g_{m+1}}\left[P_{im}(\gamma_{ij}) - P_{im}(\gamma_{ij})\frac{d_{ij}}{d_{ij}-g_j+\gamma_{ij}}\right] \\ &\qquad \times\exp\left[-\sum_{\ell=1}^{m}\frac{\gamma_{ij}-g_\ell}{d_{i\ell}}\right]\mathrm{d}\gamma_{ij} \qquad (16)\end{aligned}$$

where $P_{im}(y)$ is a polynomial, equal to $P_{im}(y) = \prod_{\ell=1}^{m}\left[1-g_\ell/d_{i\ell}+y/d_{i\ell}\right] = \sum_{k=0}^{m}p_k^{im}\,y^k$. We can evaluate the integral by first calculating the coefficients p_k^{im} using recursion. We then need to integrate the difference of two polynomials times an exponential, so the result is in the same form:

$$\begin{aligned}&\int\sum_{k=0}^{m}\left(p_k^{im}-\frac{d_{ij}p_k^{im}}{d_{ij}-g_j+y}\right)y^k e^{-y/D_{im}}\mathrm{d}y \\ &\qquad = \sum_{k=0}^{m}\left(q_k^{im}-r_k^{ijm}\right)y^k e^{-y/D_{im}} \qquad (17)\end{aligned}$$

where $D_{im} = \left[\sum_{\ell=1}^{m}d_{i\ell}^{-1}\right]^{-1}$. We can calculate q_k^{im} and integrating successively lower powers of γ_{ij}, resulting in another recurrence. Likewise, the division and integration on r_k^{ijm} can be expressed as a recurrence. Finally, we have

$$\begin{aligned}\phi_{ij} &= \frac{1}{d_{ij}}\Bigg[T_{ij,\ell=j}(g_j)e^{-G_{i,\ell=j}} \\ &\qquad + \sum_{\ell=j+1}^{K}(T_{ij\ell}(g_\ell)-T_{ij,\ell-1}(g_\ell))e^{-G_{i\ell}}\Bigg] \qquad (18)\end{aligned}$$

where

$$T_{ijm}(y) = \sum_{k=0}^{m}\left(q_k^{im}-r_k^{ijm}\right)y^k \qquad G_{i\ell} = \sum_{m=1}^{\ell}\frac{g_m-g_\ell}{d_{i\ell}} \quad (19)$$

and

$$p_k^{im}=\begin{cases}d_{im}^{-1}[p_{k-1}^{i,m-1}+(d_{im}-g_m)p_k^{i,m-1}] & 0\leq k\leq m, 1\leq m\leq K\\ 1 & m=k=0\\ 0 & \text{otherwise}\end{cases} \quad (20)$$

$$q_k^{im}=\begin{cases}D_{im}\left[(k+1)q_{k+1}^{im}+p_k^{im}\right] & 0\leq k\leq m, 1\leq m\leq K\\ 0 & \text{otherwise}\end{cases} \quad (21)$$

$$r_k^{ijm}=\begin{cases}\left[D_{im}(k+1)-d_{ij}+g_j\right]r_{k+1}^{ijm} \\ \quad +D_{im}\left[(d_{ij}-g_j)(k+2)r_{k+2}^{ijm}+d_{ij}p_{k+1}^{i,m}\right] & \\ & 0\leq k\leq m-1, 1\leq m\leq K\\ 0 & \text{otherwise}\end{cases} \quad (22)$$

If $g_1 = g_2 = \cdots = g_K$, then they can be shifted so $g_\ell = 0$ without changing the result. In this case, $G_{i\ell} = 0$, so $\phi_{ij} = T_{iji,\ell=K}(0)/d_{ij} = (q_0^{i,m=K} - r_0^{ij,m=K})/d_{ij}$.

For the general case, evaluating $T_{ijim}(\cdot)$ takes $\mathcal{O}(K)$ time per m for $\mathcal{O}(K^2)$ time per i,j pair and $\mathcal{O}(nK^3)$ overall. For $g_\ell = \mathrm{const}$, we only need $q_0^{i,m=K}$ and $r_0^{ij,m=K}$, for $\mathcal{O}(K)$ time per i,j pair and $\mathcal{O}(nK^2)$ overall. Note that calculating p_k^{im} once per i is sufficient.

References

Abul, O.; Lo, A.; Alhajj, R.; Polat, F.; and Barker, K. 2003. Cluster validity analysis using subsampling. *Systems, Man and Cybernetics, 2003. IEEE International Conference on* 2.

Aeberhard, S.; Coomans, D.; and de Vel, O. 1992. Comparison of classifiers in high dimensional settings. Technical Report 92-02, Dept. of Computer Science and Dept. Mathematix and Statistics, James Cook University of North Queensland.

Arfken, G.; Weber, H.; and Ruby, L. 1996. Mathematical Methods for Physicists. *American Journal of Physics* 64:959.

Barndorff-Nielsen, O. E. 1978. *Information and Exponential Families in Statistical Theory*.

Ben-Hur, A.; Elisseeff, A.; and Guyon, I. 2002. A stability based method for discovering structure in clustered data. *Pacific Symposium on Biocomputing* 7:6–17.

Bezdek, J., and Pal, N. 1998. Some new indexes of cluster validity. *Systems, Man and Cybernetics, Part B, IEEE Transactions on* 28(3):301–315.

Breckenridge, J. N. 1989. Replicating cluster analysis: method, consistency, and validity. *Multivariate Behavioral Research* 24(2):147–161.

Breckenridge, J. 2000. Validating cluster analysis: consistent replication and symmetry. *Multivariate Behavioral Research* 35(2):261–285.

Chvtal, V. 1983. *Linear Programming*. New York: W. H. Freeman and Company.

Giurcaneanu, C., and Tabus, I. 2004. Cluster Structure Inference Based on Clustering Stability with Applications to Microarray Data Analysis. *EURASIP Journal on Applied Signal Processing* 2004(1):64–80.

Halkidi, M.; Batistakis, Y.; and Vazirgiannis, M. 2001. On clustering validation techniques. *Journal of Intelligent Information Systems* 17(2):107–145.

Hennig, C. 2004. A general robustness and stability theory for cluster analysis.

Hinneburg, A.; Aggarwal, C. C.; and Keim, D. A. 2000. What is the nearest neighbor in high dimensional spaces? In *The VLDB Journal*, 506–515.

Hubert, L., and Arabie, P. 1985. Comparing partitions. *Journal of Classification* 2(1):193–218.

Jain, A.; Murty, M.; and Flynn, P. 1999. Data clustering: a review. *ACM Computing Surveys (CSUR)* 31(3):264–323.

Jiang, D.; Tang, C.; and Zhang, A. 2004. Cluster analysis for gene expression data: a survey. *Knowledge and Data Engineering, IEEE Transactions on* 16(11):1370–1386.

Kestelman, H. 1960. *Modern theories of integration*. Dover Publications New York.

Kuhn, H. 1955. The Hungarian Method for the Assignment Algorithm. *Naval Research Logistics Quarterly* 1(1/2):83–97.

Lange, T.; Braun, M.; Roth, V.; and Buhmann, J. 2002. Stability-based model selection. *Advances in Neural Information Processing Systems* 15.

Lange, T.; Roth, V.; Braun, M.; and Buhmann, J. 2004. Stability-based validation of clustering solutions. *Neural Computation* 16(6):1299–1323.

Maulik, U., and Bandyopadhyay, S. 2002. Performance evaluation of some clustering algorithms and validity indices. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24(12):1650–1654.

Meila, M. 2003. Comparing clusterings. *Proceedings of the Conference on Computational Learning Theory (COLT)*.

Meila, M. 2007. Comparing clusterings: an information based distance. *Journal of Multivariate Analysis* 98:873–895.

Moller, U., and Radke, D. 2006. A Cluster Validity Approach based on Nearest-Neighbor Resampling. *Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)-Volume 01* 892–895.

Roth, V.; Lange, T.; Braun, M.; and Buhmann, J. 2002. A resampling approach to cluster validation. *Statistics–COMPSTAT* 123–128.

Smolkin, M., and Ghosh, D. 2003. Cluster stability scores for microarray data in cancer studies. *BMC Bioinformatics* 4:36.

Steinley, D. 2004. Properties of the hubert-arabie adjusted rand index. *Psychol Methods* 9(3):386–96.

Tibshirani, R., and Walther, G. 2005. Cluster Validation by Prediction Strength. *Journal of Computational & Graphical Statistics* 14(3):511–528.

Vetrov, D. 2006. Evaluation of Stability of k-Means Cluster Ensembles with Respect to Random Initialization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28(11):1798–1808.