Statistica Sinica **15**(2005), 19-33

A RAPID METHOD FOR THE COMPARISON OF CLUSTER ANALYSES

Cavan Reilly, Changchun Wang and Mark Rutherford

University of Minnesota

Abstract: Cluster analysis has become a very popular tool for the exploration of high dimensional data. Dozens of algorithms have been proposed, each with its own merits and shortcomings. It is not known to what extent various methods give the same results, nor is it even clear how to measure how similar is the output of two distinct algorithms. Here we propose a statistic that is designed to measure the "correlation" between two clustering methods when applied to a particular data set. In contrast to the Rank index, the most common statistic useed for this purpose, the method is very fast. We provide an algorithm that approximates the statistic and demonstrate two of its possible uses. Finally, we use this statistic to understand the clustering in a data set in the context that motivated this work: analysis of a gene expression experiment.

Key words and phrases: Cluster analysis, Cohen's kappa, Metropolis algorithm, microarray, traveling salesman problem.

1. Introduction

Cluster analysis (also known as unsupervised learning) is an exploratory technique that aims to uncover groups of units that have similar values on a set of variables. One of the most basic problems when attempting a cluster analysis is how to define a cluster. There are scores of different algorithms available, and these differ in the manner in which they define a cluster (in addition to the technical details of how a solution is actually found). The most useful way to think about different clustering algorithms is in terms of the shape of the implied clusters: some algorithms look for spherical clusters, while others look for ellipsoidal clusters (see, e.g., Banfield and Raftery (1993) or Fraley and Raftery (2002) for a recent review). Unfortunately, it is often very hard to make an argument for a particular algorithm a priori since it is difficult to defend a choice for cluster shapes, especially since cluster analysis is typically used in an exploratory context.

An alternative to trying to choose the "right" clustering method is to apply many of the available algorithms to a data set, then examine to what extent these different methods concur. Moreover, given the interpretation of algorithms in terms of the shape of the implied clusters, such comparisons can be enlightening. Finally, if many methods largely agree except one, the researcher will know

that that method is likely to be an "outlier" and should treat the results of such a method with caution. Here we propose a method for comparing a set of cluster algorithms as applied to a specific data set, our goal is to produce a matrix analogous to a correlation matrix that displays the similarities between the methods. While one could apply a cluster algorithm to the output of the cluster algorithms (and thereby cluster the cluster analyses), we do not favor this since one still has to determine what criterion should be used in the second clustering.

The feature common to all clustering algorithms that we exploit for the purposes of our comparison is that, if a number of clusters is specified, then the methods partition the units into clusters. The full comparison between a set of algorithms is made by allowing the number of clusters to vary, thus the user need only specify a range for the number of clusters, not an actual value (since there are only a finite number of clusters possible, one can consider every possible number of clusters). To gain insight into the relations between the methods, one can either examine similarity for every given number of clusters or average over all numbers of clusters. As we will see, for the examples we have considered thus far, the extent of agreement persists over a wide range of the number of clusters, hence comparing algorithms for just a few choices of the number of clusters appears adequate.

There are a number of methods available for comparing partitions, but these are usually computationally intensive. The most popular method is Hubert and Arabie's modified Rank index (Hubert and Arabie (1985)). This method determines how often two partitions classify two units as being in the same partition. Since one must consider all pairs in a data set, if there are n units in the data set, then $\binom{n}{2}$ comparisons must be made, thus the method is an $O(n^2)$ computation. Hence, for large n, computing the Rank index can be time consuming, especially if we compare many clustering methods and allow the number of clusters to vary over a wide range. The method we present requires only the solution of an $O(n)$ problem.

For comparing specific types of cluster analyses, other methods are available. For example, if one restricts attention to hierarchical clustering algorithms, then every method produces a dendrogram, hence we can compare the clustering methods by the use of metrics for dendrograms. The literature on metrics for dendrograms has largely been concerned with unrooted trees since comparing phylogenetic trees is the primary application of these methods, although one could adapt those techniques to rooted trees (see, e.g., Robinson (1971), Waterman and Smith (1978), Critchlow, Pearl and Qian (1996)). The advantage of the method proposed here, as opposed to methods based on metrics for dendrograms, is our method's ability to compare non-hierarchical methods. This is especially important for the application to large data sets (e.g., microarrays) due to the use of self-organizing maps (the default clustering method used by many in microarray analysis) and the K-means algorithm (which is very useful for clustering large

data sets). In addition, our method also has the practical advantage of being relatively fast compared to these methods: the computational complexity for our method depends on the number of clusters, whereas the computational complexity for dendrogram metrics depends on the number of items clustered (DasGupta et al. (1997)). This observation is especially important since our method, like dendrogram metrics, requires the solution of an NP-complete problem.

While comparing different clustering outputs may appear to be a simple problem, there are complications awaiting the unwary. To get a better grasp on the nature of the problem, consider Figure 1. There we see the results of two cluster analyses: the points reflect measurements in two dimensions and the numerals in the figure represent the "cluster number" associated with each unit. By cluster number we mean the symbol that the algorithm associates with a group, these symbols partition the units. If we define the equivalence relation "being in the same cluster", then two cluster algorithms could be equivalent in terms of this relation, yet assign individual points different cluster numbers due to the arbitrary way in which cluster numbers are assigned by the algorithm. Figure 1 shows a typical example: while the two methods largely agree, there is no connection between the cluster numbers used by the two methods. With only several clusters in two or three dimensions, one can graphically compare the results or examine tables of the cluster numbers to understand the extent of agreement, but if we apply 20 different methods to cluster 12,000 different units with measurements in a 10-dimensional space into 30 clusters, comparing the methods in such simple ways becomes unwieldy. One could use Pearson's χ^2 statistic as a measure of agreement here since it is invariant with respect to the ordering of the categories, but this statistic does not have a simple interpretation as a measure of agreement (see Krieger and Green (1999) for more on the use of Pearson's χ^2 in this fashion).

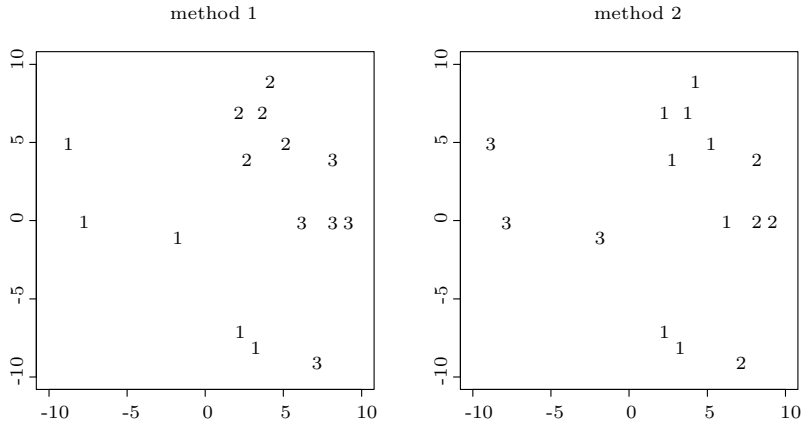

Figure 1. Example of two different cluster algorithms producing different partitions of the units based on measurements in two dimensions.

2. Measuring Agreement

For a fixed number of clusters, any clustering algorithm assigns every unit to a cluster. If we are to compare two methods, then we must measure to what extent the methods agree with respect to cluster assignment for each unit. This problem is very similar to the problem of measuring inter-rater agreement, a problem that has received much attention in the statistical literature. The important difference between the problems is that the categories to which raters assign patients are fixed and meaningful, whereas the categories used by clustering algorithms have no meaning except to define a partition among the units. Setting this difference aside for now, we note that since there is no ordering to the cluster groups, the standard measure of inter-rater agreement is Cohen's κ statistic (Cohen (1960)). Motivated by this practice, we will measure the similarity between two methods with this statistic. Since we can interpret κ as an intraclass correlation (see Fleiss (1975)), our basis for comparing cluster output has a familiar interpretation. Since cluster algorithms only define equivalence relations, we need to extend κ to this situation.

2.1. Cohen's κ coefficient

Cohen's κ measures pairwise agreement among a set of raters making categorical judgments, correcting for expected chance agreement. A general expression for κ is

$$\kappa = \frac{P_o - P_e}{1 - P_e},$$

where P_o is the observed probability of agreement between the two raters and P_e is the expected probability of agreement under the assumption of independent rating by the two raters. As is obvious from the definition, κ must be less than or equal to 1 and its lower bound depends on P_e, but will be less than zero. If the raters devise the ratings independently of one another, then κ has a mean value of zero (conditional on the marginal distributions of the raters).

While many other measures of agreement have been proposed, the important distinction between these measures concerns accounting for expected agreement (see Fleiss (1975)). Accounting for expected agreement has certain implications in the context of comparing cluster analyses. When outliers are present, cluster algorithms often make these points clusters with a single observation. When this happens we can find $\kappa \approx 0$ since $P_o \approx P_e$. For example, suppose we compare two methods for clustering $n + 2$ points and suppose we allow two clusters. Furthermore, suppose both methods agree except they each find a different outlier and make the outlier a cluster. Then $P_o = n/(n + 2)$ and $P_e = ((n + 1)/(n + 2))^2 + (n + 2)^{-2}$, so $P_o - P_e = -2(n + 2)^{-2}$ and $\kappa = -(n + 1)^{-1}$. Thus, for large n, $\kappa \approx 0$ (Section 4 provides details on this calculation). In contrast, a method

that did not account for chance agreement would find the methods agree very well (since the methods do agree for $n/(n+2)$ of the cases). This is a strength of κ, since detecting differences in what is considered an outlier provides a sensitive tool for discriminating between methods.

3. The $\kappa_{\max}$ statistic

We propose to consider all possible mappings of cluster numbers from one method to those of another, and to use the mapping that makes κ as large as possible. We choose the mapping that maximizes κ (as opposed to the average κ for instance) because we want to conclude that cluster algorithms are in perfect agreement if they define the same equivalence relation. We refer to this statistic as $\kappa_{\max}$.

Since we are comparing two methods that use the same number of clusters, the mapping of cluster numbers from one method to those of another is a permutation. Hence we propose to measure agreement between cluster algorithms by maximizing κ over the set of permuted cluster numbers, where we only permute the cluster numbers from one method. We present the results for comparing several methods in a $\kappa_{\max}$ matrix, a symmetric matrix with 1 along the diagonal and off-diagonal elements given by the $\kappa_{\max}$ between each of the two methods.

If two methods under comparison use different numbers of clusters (or, if one wants to compare the clusters obtained using the same method but different numbers of clusters), then one can still use κ. The idea here is to suppose that the clustering with the fewer number of clusters actually has as many clusters as the other method, but assigns no units to these clusters.

4. Calculating the $\kappa_{\max}$ statistic

When the number of clusters is small, we can quickly find the $\kappa_{\max}$ statistic by enumeration. As a simple example, consider Figure 1 again. The first step is to find the contingency table that compares the two methods using the cluster numbers assigned by the algorithm. Here this table is

	1	2	3
1	2	0	3
2	5	0	0
3	1	4	0

.

To find $\kappa_{\max}$, consider permuting the columns of this table. The optimal permutation will in general depend on our optimality criterion unless the methods are in perfect agreement, but here it is easy to see that the optimal permutation is given by

	3	1	2
1	3	2	0
2	0	5	0
3	0	1	4

.

For this example we find $P_o = 3/15+5/15+4/15 = 12/15$ and $P_e = (1/3)(3/15)+(1/3)(8/15)+(1/3)(4/15) = 1/3$, hence $\kappa_{\max} = 0.7$. This indicates that there is fair agreement, as evident from the plot. Moreover, by simulating this statistic (conditional on the table margins) under the assumption that the clusters are formed independently by the two methods, we find the mean of this statistic is 0.2 and we would observe such a large value for the statistic less than 1 in 1,000 times.

As the number of clusters increases, we are maximizing over a set that grows exponentially in size, hence enumeration becomes impractical. In most applications the number of clusters is not very large, thus enumeration is usually adequate. Despite this, we would like a method that works for any number of clusters. Optimizing a function over a set of permutations is a common problem in numerical analysis: it is equivalent to the famous traveling salesman problem. While no explicit solution exists (short of enumeration), good solutions can often be found by use of simulated annealing (see Press et al. (1992) for a good treatment along with code for implementation). An S-plus function that finds the $\kappa_{\max}$ matrix for the difficult case of many clusters (more than five) can be obtained at http://www.biostat.umn.edu/~cavanr.

4.1. The null distribution of the $\kappa_{\max}$ statistic

While the statistic proposed here is primarily a tool for exploratory data analysis, proper interpretation requires understanding of the distribution of this statistic when the clustering methods assign points to clusters independently of one another. Since the distribution of $\kappa_{\max}$ will depend on the margins of the table that compares outputs, in practice the most straightforward way to determine the null distribution is to simulate it conditional on the table margins, as in the example from the previous section.

For large sample sizes, $\kappa_{\max}$ is approximately zero by the Law of Large Numbers under the assumption of independent cluster assignment. The rate at which the statistic approaches zero can be computed for the case of two clusters, and illustrated for other cases with simulations. Simulations suggest that $\kappa_{\max}$ goes to zero at the same rate as in the case of two clusters.

First, suppose there are two clusters and write the table as

	1	2	
1	x_1	x_2	n_{11}
2	x_3	x_4	n_{12}
	n_{21}	n_{22}	n

.

Then $\kappa_{\max}$ is the maximum of $(x_2/n + x_3/n - n_{11}n_{22}/n^2 - n_{12}n_{21}/n^2)/(1 - n_{11}n_{22}/n^2 - n_{12}n_{21}/n^2)$ and $(x_1/n + x_4/n - n_{21}n_{11}/n^2 - n_{22}n_{12}/n^2)/(1 - n_{21}n_{11}/n^2 - n_{22}n_{12}/n^2)$. To simplify computations, suppose $n_{ij} = n/2$ for $i, j = 1, 2$. Then some algebra indicates that $\mathrm{E}\kappa_{\max} = 4/n\mathrm{E}[\max\{n/2 - X, X\}] - 1$, where $X \sim \mathrm{Bin}(n/2, 1/2)$. If n is divisible by 4, then for $k = n/4 + 1, \ldots, n/2$, $P\{\max\{n/2 - X, X\} = k\} = P\{X = k\} + P\{X = n/2 - k\} = 2^{-n/2+1}\binom{n/2}{k}$, and for $k = n/4$, $P\{\max\{n/2 - X, X\} = k\} = P\{X = n/4\} = 2^{-n/2}\binom{n/2}{n/4}$. Then

$$\begin{aligned}
\mathrm{E}[\max\{n/2 - X, X\}] &= \frac{n}{4}2^{-\frac{n}{2}}\binom{\frac{n}{2}}{\frac{n}{4}} + \sum_{k=\frac{n}{4}+1}^{\frac{n}{2}} k2^{-\frac{n}{2}+1}\binom{\frac{n}{2}}{k} \\
&= 2^{-\frac{n}{2}-2}n\binom{\frac{n}{2}}{\frac{n}{4}} + 2^{-\frac{n}{2}+1}\frac{n}{2}2^{\frac{n}{2}-2} \\
&= 2^{-\frac{n}{2}-2}n\binom{\frac{n}{2}}{\frac{n}{4}} + \frac{n}{4},
\end{aligned}$$

since

$$\sum_{k=\frac{n}{4}+1}^{\frac{n}{2}} k\binom{\frac{n}{2}}{k} = \frac{n}{2}2^{\frac{n}{2}-2}.$$

Therefore $\mathrm{E}\kappa_{\max} = 2^{-n/2}\binom{n/2}{n/4}$, and for large n, $\mathrm{E}\kappa_{\max} \approx (2\sqrt{n\pi})^{-1}$.

It is difficult to extend the above arguments to the case where there are more than two clusters, hence we conducted some simulations to investigate the behavior of the $\kappa_{\max}$ statistic. We suppose the marginal totals are all equal and cluster assignments are made independently by two methods. Figure 2 shows the expected values for 5, 10 and 15 clusters (100 data sets were simulated for each sample size and number of clusters). The expected value goes to zero fairly rapidly. In fact, if we plot the inverse of the square of the mean values from these simulations against the sample size we see that the expected value of $\kappa_{\max}$ seems to be $O(n^{-1/2})$, as in the two cluster case above. Indeed, if we regress the inverse of the square of the mean values from the simulations on the sample size for the numbers of clusters considered here, the resulting R^2 values are 0.9973, 0.9998 and 0.9989. We speculate that the number of clusters enters the expectation as an exponent. Evidence in favor of this is provided by noting that a regression of the logarithm of the mean values from the simulations on the logarithm of the

sample size and the number of clusters yields an $R^2 = 0.993$ with all variables being significant and no evidence for interactions.

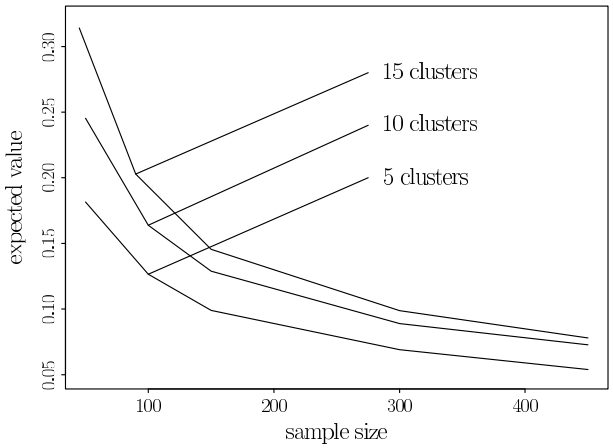

Figure 2. The expected value of the $\kappa_{\max}$ statistic when the cluster assignments are independent as a function of the sample size.

5. Some Applications of $\kappa_{\max}$

5.1. Sensitivity of K-means to the initial values

One application is to the K-means algorithm for cluster analysis. It is common to use several different starting points when one uses K-means clustering due to sensitivity of the algorithm to initial values (see, e.g., Johnson and Wichern (1992)), but there is little guidance as to how one should assess this sensitivity.

The $\kappa_{\max}$ statistic provides a reasonable method for measuring the reproducibility of the clustering. To demonstrate, we simulated 1,000 observations from a ten component 8-variate normal mixture model, randomly selected six sets of starting points distributed within the convex hull of the data, then used the K-means algorithm (starting from each set of starting points) to cluster the observations. We chose new starting points and repeated the procedure for various numbers of clusters. We used S-plus to perform these computations, and the K-means algorithm converged (see Hartigan and Wong (1979)) in all cases.

Figure 3 displays the $\kappa_{\max}$ matrix using a gray scale image plot to represent the values in the matrix (there is one row and column for each set of starting points). We see that the initial values lead to solutions that largely agree for any number of clusters, but when we use the correct number of clusters (ten) the mean $\kappa_{\max}$ reaches its maximal value (0.89, with next largest 0.85) and the standard deviation of the $\kappa_{\max}$ is at its smallest (0.009, with next smallest 0.013). This suggests that use of the incorrect number of clusters leads to sensitivity of

the solution with respect to the starting values, which is not surprising. Finally, note that even when we specify the correct number of clusters there appears to be two different solutions even though the algorithm converges in all cases. Perhaps the $\kappa_{\max}$ matrix could be used to devise more stringent convergence criteria for the K-means algorithm, analogous to the use of multiple chains in Markov chain Monte Carlo simulation.

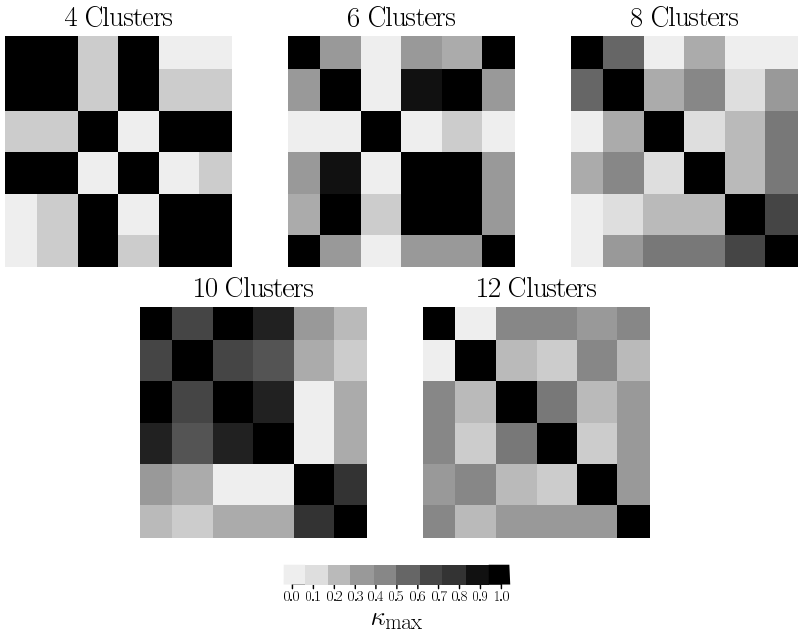

Figure 3. The $\kappa_{\max}$ matrix comparing six different starting points using the K-means algorithm for five different numbers of clusters.

5.2. Comparing cluster algorithms

Another application of the $\kappa_{\max}$ statistic is in assessing the relative merits of a variety of clustering algorithms under varying circumstances. By simulating data sets and calculating the $\kappa_{\max}$ statistic, one has a method of gauging a new cluster algorithm against existing algorithms.

As an illustration, we generated data sets with 100 observations using one of four, 6-component 8-variate normal mixture models. We first simulated a set of weights for the mixture components (these were uniform), then we simulated the centers for each component (using six independent $\mathrm{N}_8(0, I)$ deviates or six independent $\mathrm{N}_8(0, 16I)$ deviates, where I is the 8 by 8 identity matrix). We used two correlation matrices (one with independence and one with two groups of variables that are highly correlated within groups but basically independent

across groups) to obtain distinct covariance matrices (all standard deviations are 1).

Given a data set, we applied one of six cluster algorithms to the data (all are available in S-plus): hierarchical clustering (with complete linkage and simple linkage), model based clustering (with the trace criterion, the determinant criterion and with no constraints), and K-means clustering. For each set of six cluster outputs and the true cluster identities we then calculated the $\kappa_{\max}$ matrix. We repeated the entire process 100 times and found the mean $\kappa_{\max}$ matrix.

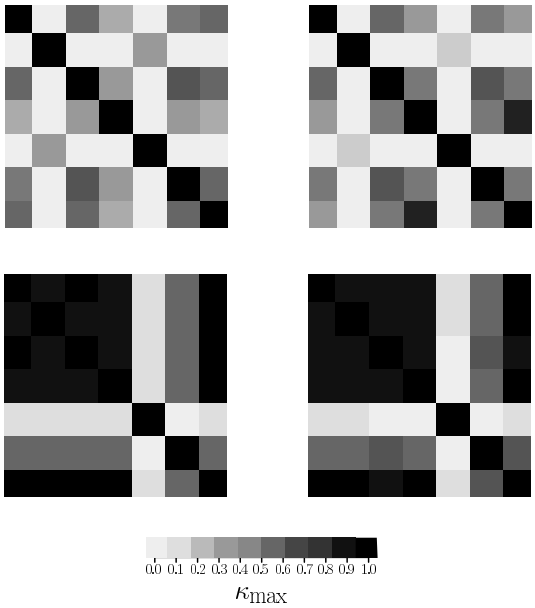

Figure 4. The $\kappa_{\max}$ matrix comparing six different methods and the true cluster membership (last row and column) from a 2×2 factorial simulation experiment. The top row has clusters that are closer together than in the bottom row. The left column has spherically symmetric clusters, while the right row has ellipsoidal clusters.

Figure 4 displays the mean $\kappa_{\max}$ matrix for each of the four scenarios considered here. The first six rows and columns represent the methods (in the order listed above) and the last row and column represents the true cluster membership. As expected, when the standard deviation of the centers is larger (shown in the bottom row), all methods do better except model based clustering with no constraints, which appeared to never do well. In addition, K-means differed from the other methods and did not perform as well. When there is correlation within clusters, the determinant criterion becomes more similar to the group of three methods that seek spherically symmetric clusters (i.e., complete linkage,

the trace criterion and K-means), but it outperforms all of the other methods. This finding is expected since we know that the determinant criterion is supposed to do well in this context (see Banfield and Raftery (1993)).

In summary, we find that simple methods seeking hyperspherical clusters, such as Ward's method (which uses the trace criterion) and K-means, typically do quite well (and usually agree with one another), while methods that seek to find clusters of arbitrary size and shape (e.g., the model based, unconstrained method) do poorly. Moreover, these conclusions seem to hold even when there is substantial correlation among measurements within a mixture component, despite the fact that these more general methods allow for such correlations and therefore should do better.

6. Application to a cDNA Microarray Experiment

6.1. Cluster analysis in microarray experiments

DNA microarrays allow comprehensive surveys of gene expression. While it is customary to think of the relationships between transcript levels across conditions or time as being related thorough some high dimensional system of biochemical equations, it is not clear that estimation of the parameters of such systems is possible using microarrays because of the many mechanisms of post-transcriptional regulation, in addition to limitations of the technology. Despite these obstacles, a system of equations would have implications for more global properties of transcript expression. For instance, although we probably cannot accurately estimate equilibrium constants for the biochemical reactions implied by such a system from microarray data, a system may imply that certain groups of transcripts will tend to be induced or repressed as a whole. While even the specification of such systems is largely impossible for most cellular systems, we can start to empirically construct the broad outlines of these systems by examining which transcripts have similar gene expression across a variety of conditions.

Based on this perspective, researchers have explored the use of many different clustering methods for microarray analysis. In the application to microarrays, units are usually transcripts which are grouped on the basis of their gene expression across various circumstances. Eisen et al. (1998) were the first to apply cluster analysis to microarray data. They used the pair-wise average-linkage criterion and implemented the method with an agglomerative algorithm. As an alternative, the divisive, hierarchical clustering method was applied to gene expression data by Alon et al. (1999). Additionally, to partition expression data into groups, self-organizing maps (Toronen et al. (1999)), K-means clustering and the quality cluster algorithm of Heyer et al. (1999) have been explored. The notion of two-dimensional clustering was used by Alon et al. (1999) and Perou et al. (1999), in which not only the genes but also the arrays are organized by

clustering. Getz et al. (2000) presented a coupled two-way clustering approach to gene microarray data analysis. One could apply the methods for comparing cluster algorithms presented here to these two way clustering methods by applying the method separately to the two clusterings.

6.2. Sources of experimental data

As an illustration of the use of the $\kappa_{\max}$ statistic for data analysis, we apply the technique to an experiment aimed at understanding the etiology of porcine reproductive and respiratory syndrome virus (PRRSV). Gene expression in porcine alveolar macrophages infected by one of three PRRSV strains was measured at 4 hours and 24 hours post-infection by using spotted cDNA microarrays containing 139 genes and expressed sequence tags from swine and PRRSV. RNA from the virus infected samples was labeled during reverse transcription with the red-fluorescent dye Cy5 and mixed with a mock infected sample, labeled with the green-fluorescent dye Cy3, from the same time point. For the analyses presented here, the data take the form of a 139 by 6 matrix of average $\log_2$ expression ratios, with a row for each gene and a column for each experimental condition (from two different time points infected by three strains of virus relative to the control).

6.3. Clustering algorithms to be compared

For this data set, we considered nine different cluster algorithms. We distinguish these by the criterion used to define the clusters, and whether the method allows for Poisson noise distributed over the six dimensional space of measurements (Poisson noise allows for outliers). Following the nomenclature of Banfield and Raftery (1993), we used the following methods: trace with noise, determinant with noise, determinant without noise, spherical with noise, S* with noise, unconstrained with noise, unconstrained without noise, the centroid method of Sneath and Sokal (1973) without noise, and K-means without noise. We allowed the number of clusters to vary from 3 to 12. The trace criterion (also known as the sum of squares criterion) looks for hyperspherical clusters of the same size. In contrast, the determinant criterion (Friedman and Rubin (1967)) favors ellipsoidal clusters with the same size pointing in the same direction. The spherical criterion seeks hyperspherical clusters of varying sizes. The criterion S* is appropriate for clusters that have the same size and orientation but have different shapes, while the unconstrained criterion (Scott and Symons (1971)) allows clusters to have different orientations, shapes, and sizes. Both the centroid and K-means algorithms are heuristic methods that seek hyperspherical clusters, but

use different methods for finding the clusters. Once again, all cluster analysis was conducted using S-plus.

6.4. Results for the PRRSV microarray experiment

Figure 5 shows the $\kappa_{\max}$ for ten different numbers of clusters (the rows and columns represent the cluster algorithms and are in the order listed above). First note that the extent of agreement does not vary as we vary the number of clusters. Note that there are two groups of cluster algorithms that agree amongst themselves but differ across groups. One group consists of the determinant without noise, the spherical criterion, the S* criterion, the centroid method and the K-means algorithm. The other group consists of the two versions of the unconstrained criterion (with and without noise). There is very little agreement across these two groups. These results are sensible since the first group seeks largely spherical clusters (some allow ellipsoidal clusters) while the second group has no constraints. The trace criterion with noise differs from all other cluster methods, as does the determinant criterion with noise.

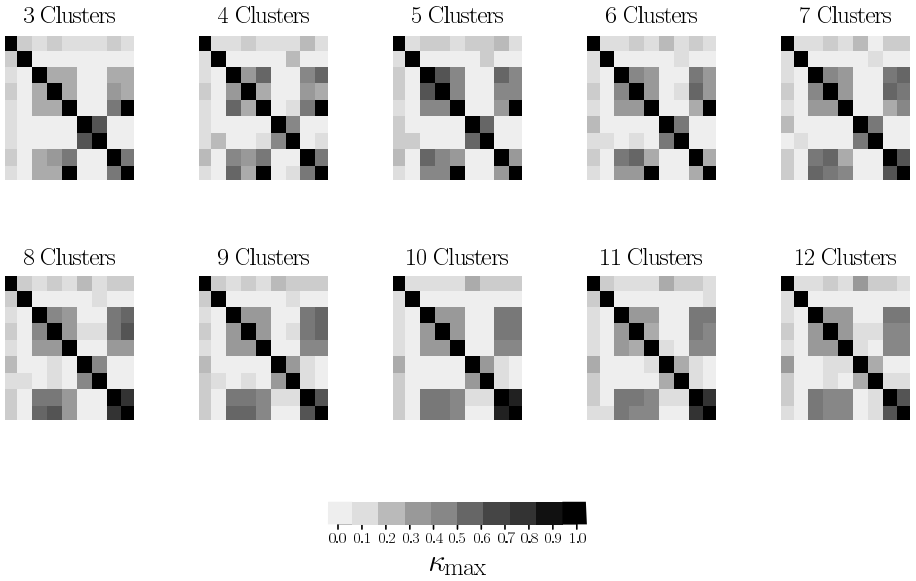

Figure 5. The $\kappa_{\max}$ matrix comparing the nine methods used to analyze the gene expression data set using ten different numbers of clusters.

Looking further into the relations between the methods illustrates how an interplay between the agreement measures and the clustering output can be used to further understand the nature of the clustering in the data. As noted in Section 2.1, algorithms often make a single item its own cluster if that item appears

to be an outlier, and by allowing for noise we explicitly allow for the presence of outliers. Here several methods are identifying outliers as clusters. These methods use different criteria for clusters, hence they identify different points as outliers, and so they strongly disagree. This is the explanation for the discrepant effects of allowing for noise when using the determinant criterion compared to the unconstrained criterion. The unconstrained version is not as sensitive as the determinant criterion to the assumption of Poisson noise, because allowing for noise does not alter the sort of clusters the unconstrained criterion seeks, while this assumption does have a large impact on the determinant criterion. These considerations suggest to us that the use of any of the first group of methods identified in the previous paragraph results in a clustering that is robust, while the other methods seem less useful.

Acknowledgement

Thanks to NIH grant #1P30-CA79458-01.

References

Alon, U., Barkai, N., Notterman, D., Gish, K., Ybarra, S., Mack, D. and Levine, A. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Nat. Acad. Sci.* **96**, 6745-6750.

Banfield, J. and Raftery, A. (1993). Model-based Gaussian and non-Gaussian clustering. *Biometrics.* **49**, 803-822.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement* 196037-196046.

Critchlow, D., Pearl, D. and Qian, C. (1996). The triples distance for rooted bifurcating phylogenetic trees. *Systems Biology* **45**, 323-334.

DasGupta, B., He, X., Jiang, T., Li, M., Tromp, J. and Zhang, L. (1997). On distances between phylogenetic trees. in *Proc. 8th ACM-SIAM Symposium on Discrete Algorithms*, 427-436.

Eisen, M., Spellman, P., Brown, P. and Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proc. Nat. Acad. Sci.* **95**, 14863-14868.

Fleiss, J. (1975). Measuring agreement between two judges on the presence or absence of a trait. *Biometrics* **31**, 651-659.

Fraley, C. and Raftery, A. (2002). Model-based clustering, discriminant analysis and density estimation. *J. Amer. Statist. Assoc.* **97**, 611-631.

Friedman, H. and Rubin, J. (1967). On some Invariant Criteria for Grouping Data. *J. Amer. Statist. Assoc.* **62**, 1159-1178.

Getz, G., Levine, E. and Domany, E. (2000). Coupled two-way clustering analysis of gene microarray data. *Proc. Nat. Acad. Sci.* **97**, 12079-12084.

Hartigan, J. and Wong, M. (1979). A k-means clustering algorithm. *Appl. Statist.* **28**, 100-108.

Heyer, L., Kruglyak, S. and Yooseph, S. (1999). Exploring expression data: identification and analysis of co-expressed genes. *Genome Research* **9**, 1106-1115.

Johnson, R. and Wichern, D. (1992). *Applied Multivariate Statistical Analysis.* Prentice Hall, Englewood Cliffs, New Jersey.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of 5th Berkeley Symposium* **1**, 281-297.

Perou, C., Jeffrey, S., van de, R., Rees, C., Eisen, M., Ross, D., Pergamenschikov, A., Williams, C., Zhu, S., Lee, J., Lashkari, D., Shalon, D., Brown, P. and Botstein, D. (1999). Distinctive gene expression patterns in human mammary epithelial cells and breast cancer. *Proc. Nat. Acad. Sci.* **96**, 9212-9217.

Press, W., Teukolsky, S., Vetterling, W. and Flannery, B. (1993). *Numerical Recipes in C : The Art of Scientific Computing*. Cambridge University Press, Cambridge.

Robinson, D. (1971). Comparison of labeled trees with valency three. *J. Combinatorial Theory, Series B* **11**, 105-119.

Scott, A. and Symons, M. (1971). Clustering methods based on likelihood ratio criteria. *Biometrics* **27**, 387-397.

Sneath, P. and Sokal, R. (1973). *Numerical Taxonomy* W. H. Freeman, San Francisco.

Toronen, P., Kolehmainen, M., Wong, G. and Castren, E. (1999). Analysis of gene expression data using self-organizing maps. *FEBS Letters* **451**, 142-6.

Waterman, M. and Smith, T. (1978). On the similarity of dendrograms. *J. Theoret. Biology* **73**, 789-800.

Division of Biostatistics, University of Minnesota, A448 Mayo Bldg., MMC 303, 420 Delaware St. SE, Minneapolis, MN 55455-0378.

E-mail: cavanr@biostat.umn.edu

Division of Biostatistics, Department of Veterinary PathoBiology, University of Minnesota.

E-mail: wang0307@umn.edu

Department of Veterinary PathoBiology, University of Minnesota.

E-mail: ruthe003@umn.edu

(Received February 2003; accepted September 2004)