A stability based method for discovering structure in clustered data

Asa Ben-Hur*, Andre Elisseeff† and Isabelle Guyon*
BioWulf Technologies LLC
*2030 Addison st. Suite 102 Berkeley, CA 94704
†305 Broadway (9th Floor) New-York, NY 10007

Abstract

We present a method for visually and quantitatively assessing the presence of structure in clustered data. The method exploits measurements of the stability of clustering solutions obtained by perturbing the data set. Stability is characterized by the distribution of pairwise similarities between clusterings obtained from sub samples of the data. High pairwise similarities indicate a stable clustering pattern. The method can be used with any clustering algorithm; it provides a means of rationally defining an optimum number of clusters, and can also detect the lack of structure in data. We show results on artificial and microarray data using a hierarchical clustering algorithm.

1 Introduction

Clustering is widely used in exploratory analysis of biological data. With the advent of new biological assays such as DNA microarrays that allow the simultaneous recording of tens of thousands of variables, it has become more important than ever to have powerful tools for data visualization and analysis. Clustering, and particularly hierarchical clustering, play an important role in this process. [1,2,3]

Clustering provides a way of validating the quality of the data by verifying that groups form according to the prior knowledge one has about sample categories. It also provides means of discovering new natural groupings. [4] Yet there is no generally agreed upon definition of what is a "natural grouping." In this paper we propose a method of detecting the presence of clusters in data that can serve as the basis of such a definition. It can be combined with any clustering algorithm, but proves to be particularly useful in conjunction with hierarchical clustering algorithms.

The method we propose in this paper is based on the stability of clustering with respect to perturbations such as sub-sampling or the addition of noise. Stability can be considered an important property of a clustering solution, since data, and gene expression data in particular, is noisy. Thus we suggest stability as a means for defining meaningful partitions. The idea of using stability to evaluate clustering solutions is not new. In the context of hierarchical clustering, some authors have considered the stability of the whole hierarchy. [5] However, our experience indicates that in most real world cases the complete dendrogram is rarely stable. The stability of partitions has also been addressed. [6,7,8] In this model, a figure of merit is assigned to a partition

of the data according to average similarity of the partition to a set of partitions obtained by clustering a perturbed dataset. The optimal number of clusters (or other parameter employed by the algorithm) is then determined by the maximum value of the average similarity. But we observed in several practical instances that considering the average, rather than the complete distribution was insufficient. The distribution can be used both as a tool to visually probe the structure in the data, and to provide a criterion for choosing an optimal partition of the data: plotting the distribution for various numbers of clusters reveals a transition between a distribution of similarities that is concentrated near 1 (most solutions highly similar) to a wider distribution. In the examples we studied, the value of the number of clusters at which this transition occurs agrees with the intuitive choice of the number of clusters. We have developed a heuristic for comparing partitions across different levels of the dendrogram that make this transition more pronounced. The method is useful not only in choosing the number of clusters, but also as a general tool for making choices regarding other components of the clustering algorithm. We have applied it in choosing the type of normalization and the number of leading principal components. [9]

Many methods for selecting an optimum number of clusters can be found in the literature. In this paper we report results that show that our method performs well when compared with some of the more successful methods reported in recent surveys. [10,11] This may be explained by the fact that our method does not make assumptions about the distribution of the data or about cluster shape as most other methods; [11,10] only our method and the gap statistic can detect the absence of structure. Our method has advantages over information-theoretic criteria based on compression efficiency considerations and over related Bayesian criteria[12] in that they are model free, and work with any clustering algorithm. Some clustering algorithms have been claimed to generate only meaningful partitions, so do not require our method for this purpose. [4,13] We also mention the method of Yeung *et al.*[14] for assessing the relative merit of different clustering solutions. They tested their method on microarray data; however, they do not give a way of selecting an optimal number of clusters, so no direct comparison can be made.

The paper is organized as follows: in Section 2 we introduce the dot product between partitions and express several similarity measures in terms of this dot product. In Section 3 we present our practical algorithm. Section 4 is devoted to experimental results of using the algorithm. This is followed by a discussion and conclusions.

2 Clustering similarity measures

In this section we present several similarity measures between partitions found in the literature, [15,7] and express them with the help of a dot product. We begin by reviewing our notation. Let $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, and $\mathbf{x}_i \in \mathbb{R}^d$ be the dataset to be clustered.

A *labeling* $\mathcal{L}$ is a partition of X into k subsets $S_1, \ldots, S_k$. We use the following representation of a labeling by a matrix C with components:

$$C_{ij} = \begin{cases} 1 & \text{if } \mathbf{x}_i \text{ and } \mathbf{x}_j \text{ belong to the same cluster and } i \neq j\,, \\ 0 & \text{otherwise}\,. \end{cases} \tag{1}$$

Let labelings $\mathcal{L}_1$ and $\mathcal{L}_2$ have matrix representations $C^{(1)}$ and $C^{(2)}$, respectively. We define the dot product

$$\langle \mathcal{L}_1, \mathcal{L}_2 \rangle = \left\langle C^{(1)}, C^{(2)} \right\rangle = \sum_{i,j} C^{(1)}_{ij} C^{(2)}_{ij}\,. \tag{2}$$

This dot product computes the number of pairs of points clustered together, and can also be interpreted as the number of common edges in graphs represented by $C^{(1)}$ and $C^{(2)}$, and we note that it can be computed in $O(k_1 k_2 n)$.

As a dot product, $\langle \mathcal{L}_1, \mathcal{L}_2 \rangle$ satisfies the Cauchy-Schwartz inequality: $\langle \mathcal{L}_1, \mathcal{L}_2 \rangle \leq \sqrt{\langle \mathcal{L}_1, \mathcal{L}_1 \rangle \langle \mathcal{L}_2, \mathcal{L}_2 \rangle}$, and thus can be normalized into a correlation or cosine similarity measure:

$$cor(\mathcal{L}_1, \mathcal{L}_2) = \frac{\langle \mathcal{L}_1, \mathcal{L}_2 \rangle}{\sqrt{\langle \mathcal{L}_1, \mathcal{L}_1 \rangle \langle \mathcal{L}_2, \mathcal{L}_2 \rangle}}\,. \tag{3}$$

This similarity measure was introduced by Fowlkes and Mallows. [7] Next, we show that two commonly used similarity measures can be expressed in terms of the dot product defined above. Given two matrices $C^{(1)}, C^{(2)}$ with 0-1 entries, let N_{ij} for $i, j \in \{0, 1\}$ be the number of entries on which $C^{(1)}$ and $C^{(2)}$ have values i and j, respectively. The *matching coefficient*[15] is defined as the fraction of entries on which the two matrices agree:

$$M(\mathcal{L}_1, \mathcal{L}_2) = \frac{N_{00} + N_{11}}{N_{00} + N_{01} + N_{10} + N_{11}}\,. \tag{4}$$

The *Jaccard coefficient* is a similar ratio when "negative" matches are ignored:

$$J(\mathcal{L}_1, \mathcal{L}_2) = \frac{N_{11}}{N_{01} + N_{10} + N_{11}}\,. \tag{5}$$

The matching coefficient often varies over a smaller range than the Jaccard coefficient since the N_{00} term is usually a dominant factor. These similarity measures can be expressed in terms of the labeling dot product and the associated norm:

$$\begin{aligned} J(\mathcal{L}_1, \mathcal{L}_2) &= \frac{\left\langle C^{(1)}, C^{(2)} \right\rangle}{\left\langle C^{(1)}, C^{(1)} \right\rangle + \left\langle C^{(2)}, C^{(2)} \right\rangle - \left\langle C^{(1)}, C^{(2)} \right\rangle} \\ M(\mathcal{L}_1, \mathcal{L}_2) &= 1 - \frac{1}{n^2} \| C^{(1)} - C^{(2)} \|^2 \end{aligned}$$

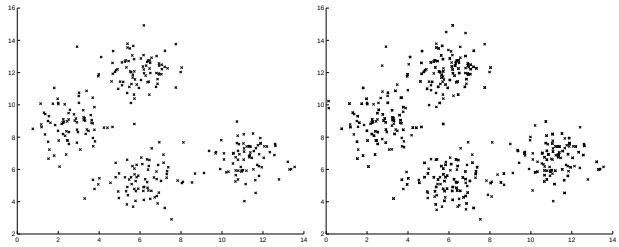

Figure 1: Two 250 point sub-samples of a 400 point Gaussian mixture.

This is a result of the observation that $N_{11} = \left\langle C^{(1)}, C^{(2)} \right\rangle$, $N_{01} = \left\langle 1_n - C^{(1)}, C^{(2)} \right\rangle$, $N_{10} = \left\langle C^{(1)}, 1_n - C^{(2)} \right\rangle$, $N_{00} = \left\langle 1_n - C^{(1)}, 1_n - C^{(2)} \right\rangle$, where 1_n is an $n \times n$ matrix with entries equal to 1. The above expression for the Jaccard coefficient shows that it is close to the correlation similarity measure, as we have observed in practice.

3 The model explorer algorithm

When one looks at two sub-samples of a cloud of data points, with a sampling ratio f (fraction of points sampled) not much smaller than 1 (say $f > 0.5$), one usually observes the same general structure (Figure 1). Thus it is reasonable to postulate that a partition into k clusters has captured the "inherent" structure in a dataset if partitions into k clusters obtained from running the clustering algorithm with different sub-samples are similar, i.e. close in structure according to one of the similarity measures introduced in the previous section. "Inherent" structure is thus structure that is stable with respect to sub-sampling. We cast this reasoning into the problem of finding the optimal number of clusters for a given dataset and clustering algorithm: look for the largest k such that partitions into k clusters are stable. Note that rather than choosing just the number of clusters, one can extend the scope of the search for a set of variables where structure is most apparent, i.e. stable. This is performed elsewhere.[9]

We consider a generic clustering algorithm that receives as input a dataset (or similarity/dissimilarity matrix) and a parameter k that controls either directly or indirectly the number of clusters that the algorithm produces. This input convention is applicable to hierarchical clustering algorithms: given k, cut the tree so that k clusters are produced. We want to characterize the stability for each value of k. This is accomplished by clustering sub-samples of the data, and then computing the similarity between pairs of sub-samples according to similarity between the labels of the points common to both sub-samples. The result is a distribution of similarities for each k. The algorithm is presented in Figure 2.

The distribution of the similarities is then compared for different values of k

Input: X {a dataset}, $k_{\max}$ {maximum number of clusters}, num_subsamples {number of subsamples}
Output: $S(i,k)$ {list of similarities for each k and each pair of sub-samples }
Require: A clustering algorithm: cluster(X,k); a similarity measure between labels: $s(L_1, L_2)$

$f = 0.8$
for $k = 2$ to $k_{\max}$ **do**
  for $i = 1$ to num_subsamples **do**
    sub_1 =subsamp(X, f) {a sub-sample with a fraction f of the data}
    sub_2 =subsamp(X, f)
    L_1 =cluster(sub_1, k)
    L_2 =cluster(sub_2, k)
    Intersect= $sub_1 \cap sub_2$
    $S(i,k) = s(L_1(Intersect), L_2(Intersect))$ {Compute the similarity on the points common to both subsamples}
  end for
end for

Figure 2: The Model explorer algorithm.

(Figure 3). In our numerical experiments (Section 4) we found that, indeed, when the structure in the data is captured by a partition into k clusters, many sub-samples have similar clustering, and the distribution of similarities is concentrated close to 1.

Remark 3.1 For the trivial case $k = 1$, all clusterings are the same, so there is no need for any computation in this case. In addition, the value of f should not be too low; otherwise not all clusters are represented in a sub-sample. In our experiments the shape of the distribution of similarities did not depend very much on the specific value of f.

4 Experiments

In this section we describe experiments on artificial and real data. We chose to use data where the number of clusters is apparent, so that one can be convinced of the performance of the algorithm. In all the experiments we show the distribution of the correlation score; equivalent results were obtained using other scores as well. The sampling ratio, f, was 0.8 and the number of pairs of solutions compared for each k was 100. As a clustering algorithm we use the average-link hierarchical clustering algorithm.[15] The advantage of using a hierarchical clustering method is that the same

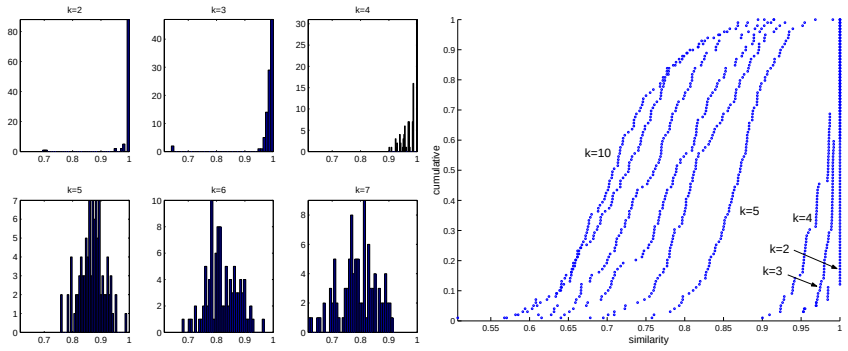

Figure 3: Left: histogram of the correlation similarity measure; right: overlay of the cumulative distributions for increasing values of k.

set of trees can be used for all values of k, by looking at different levels of the tree each time. To tackle the problem of outliers, we cut the tree such that there are k clusters, each of them not a singleton (thus the total number of clusters can be higher than k). This is extended to consider partitions that contain k clusters, each of them larger than some threshold. This helps enhance the stability in the case of a good value of k, and de-stabilizes clustering solutions for higher k, making the transition from highly similar solutions to a wide distribution of similarities more pronounced.

We begin with the data depicted in Figure 1, which is a mixture of four Gaussians. The histogram of the score for varying values of k is plotted in figure 3. We make several observations regarding the histogram. At $k = 2$ it is concentrated at 1, since almost all the runs discriminated between the two upper and two lower clusters. At $k = 3$ most runs separate the two lower clusters, and at $k = 4$ most runs found the "correct" clustering which is reflected in the distribution of scores still concentrated near 1. For $k > 4$ there is no longer one preferred solution, as is seen by the wide spectrum of similarities. We remark that if the clusters were well separated, or the clusters arranged more symmetrically, there would not have been a preferred way of clustering into 2 or 3 clusters as is the case here; in that case the similarity for $k = 2, 3$ would have been low, and increased for $k = 4$. In such cases one often observes a bimodal distribution of similarities.

The next dataset we considered was the yeast DNA microarray data of Eisen *et al.*[1] We used the MYGD functional annotation to choose the 5 functional classes that were most learnable by SVMs,[16] and that were noted by Eisen *et al.* to cluster well. [1] We looked at the genes that belong uniquely to these 5 functional classes. This gave a dataset with 208 genes and 79 features (experiments) in the following classes: (1)

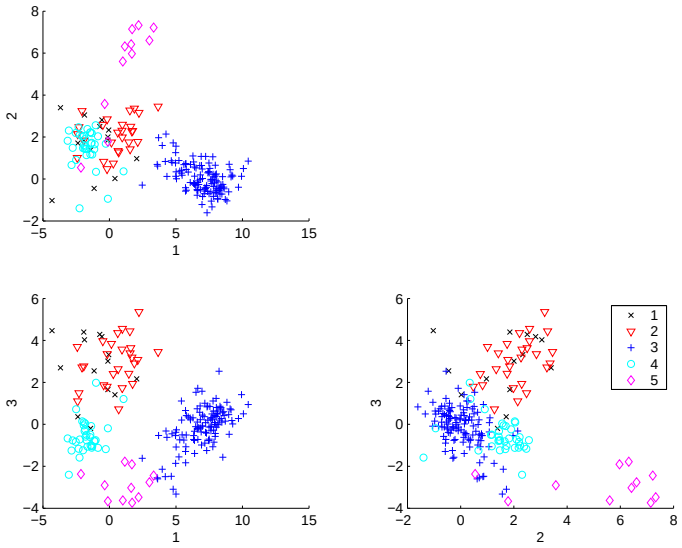

Figure 4: First three principal components of the yeast microarray data. The legend identifies the symbols that represent each functional class. Class number corresponds to the numbers given in the listing of the classes in the text.

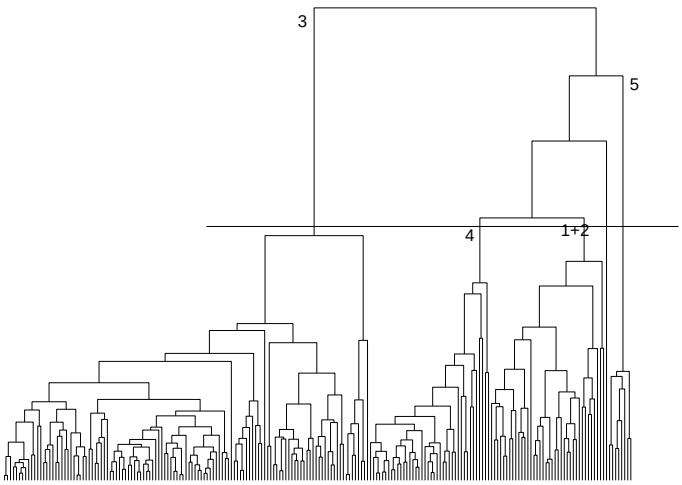

Figure 5: Dendrogram for yeast microarray data. Numbers indicate the functional class represented by each cluster. The horizontal line represents the lowest level at which partitions are still highly stable.

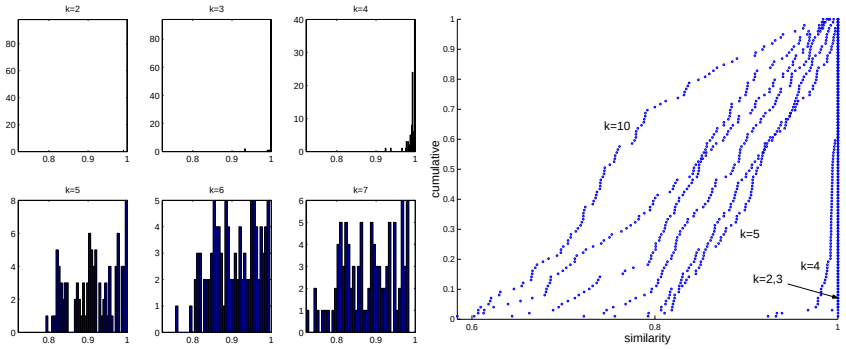

Figure 6: Left: histogram of the correlation similarity measure for the yeast gene expression data for increasing values of k. For $k = 2$ all similarities were equal to 1. Right: overlay of the cumulative distribution functions.

Tricarboxylic acid cycle or Krebs cycle (14 genes), (2) Respiration chain complexes (27 genes), (3) Cytoplasmaticribosomal proteins (121 genes), (4) proteasomes (35 genes), and (5) Histones (11 genes). The first three principal components were then extracted (see Figure 4). One can clearly see four clusters in the data; these agree well with the MYGD classes, with classes 1 and 2 strongly overlapping. The distribution and histogram of scores is given in Figure 6. We observe the same behavior as in the Gaussian mixture data. Between $k = 4$ and $k = 5$ there is a transition from a distribution that has a large component near 1, to a wide distribution that is similar to the distribution on random data (see below). Since there was a singleton cluster, the total number of clusters is 5. The clusters agree well with the protein classes that were assigned to the genes in the MYGD database, with the exception that clusters 1 and 2 were clustered together. The way the dendrogram was cut to produce the partition is illustrated in Figure 5. Looking at the dendrogram one might think that further splitting of cluster 3 is justified. However, the length of the edge in the dendrogram can be misleading: in the case of average linkage the length of the edge is proportional to the average distance between clusters, and since cluster 3 is large, a long edge does not necessarily imply a well separated sub-cluster. At high levels of the hierarchy long edges generally result simply because the clusters become larger, even if the data contains no structure. When clusters 1 and 2 are assigned the same label, the similarity between the clustering and the known classification is 0.97. We note that principal component analysis (PCA) not only allows visualization of the data, it enhances cluster structure reflected in the stability and also improves the agreement of the clustering with the MYGD classes. [9]

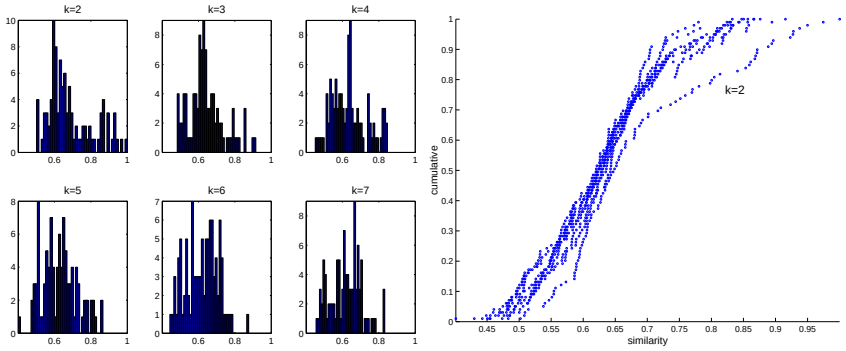

Figure 7: Left: histogram of the correlation score for 208 points uniformly distributed on the unit cube; right: overlay of the cumulative distributions of the correlation score.

A run on data uniformly distributed on the unit cube is shown in Figure 7. The distributions are quite similar to each other, with no change that can be interpreted as a transition from a stable clustering to an unstable one.

These examples indicate a simple way for identifying k; choose the value where there is a transition from a score distribution that is concentrated near 1 to a wider distribution. This can be quantified, e.g. by a jump in the area under the cumulative distribution function or by a jump in $P(s_k > \eta)$, where s_k is the random variable that denotes the similarity between partitions into k clusters, and η is a constant. A value of $\eta = 0.9$ would work on the set of examples considered here.

The results of our method are compared in Table 1 with a number of other methods for choosing k. We used most of the methods tested by Tibshirani *et al.* against their gap statistic method. [11] They are among the methods tested by Milligan and Cooper. [10] Jain's method uses the quotient between the in-cluster average distance and out-of-cluster average distance, averaged over all the clusters. The optimal number of clusters is chosen as the k that minimizes this quantity. The method of Calinski and Harabsz is similar, but uses a different normalization, and the squared distances. The silhouette statistic is based on comparing the average distance of the point to members of other clusters with the average distance of a point to members of its own cluster. A point is "well clustered" if it is closer on average to the members of its own cluster than to points of other clusters. The silhouette statistic is the average of the point silhouettes, and k is chosen to maximize it. The KL (Krzanowski and Lai), Hartigan, and gap statistic methods use criteria that are based on the k-dependence of a function of the within-cluster sum-squared distances. Almost all the methods were successful on the Gaussian mixture data; this is to be expected since some of the

Table 1: Number of clusters obtained by various methods for choosing the number of clusters. Subsamp denotes our method.

problem	Jain	Silhouette	KL	CH	Hartigan	gap	**subsamp**	true
4Gaus	6	4	9	4	4	4	4	4
Microarray	4	5	2	2	3	6	5	5
Random	7	9	5	2	9	1	1	1

methods are specifically constructed for such data. The microarray data proved more difficult. We note that our method gave the same results when all the variables rather than just the first three principal components were clustered, whereas the gap statistic did not give a result when all the variables were clustered. The gap statistic is based on a comparison of the within cluster sum-squared distance of the given clustering with an average obtained over random data. Perhaps the comparison with random data does not scale well to very high dimensionality. All the methods we tested, other than the gap statistic and our own method cannot detect a lack of structure: they produce a meaningful result only if it is known beforehand that the number of clusters is greater than 1. When these methods are run on data with no structure they still provide (erroneously) a result. Running these methods on sub-samples of the data can provide the information required to rule out the hypothesis of no structure: intuitively, for data with clear clusters the result is likely to remain the same, while for data with no structure the criterion is likely to be unstable, and fluctuate across sub-samples.

5 Discussion

In the set of experiments we ran, only the gap statistic method performed as well as our method. Since the gap statistic is based on a sum-squared distances criterion, it is biased toward compact clusters; our method has no such bias. Further work should include a more systematic experimental analysis to differentiate the two methods. Both methods are the most computationally expensive, requiring running the clustering algorithm a number of times. Our method can be used not only to choose the number of clusters, but also as a comparative tool that can help in choosing other aspects of the clustering such as normalization. [9] Our algorithm is most efficient with hierarchical clustering, since once a dendrogram is computed, varying the number of clusters is achieved at little additional computational expense.

The datasets analysed in this paper were chosen for illustrative purposes for having a distinct structure. One might argue that many real world datasets do not have such an obvious number of clusters. Indeed, partitions obtained on a large set of variables (e.g. thousands of genes from DNA microarrays) are usually unstable. We see that as a symptom that prior knowledge is needed to select meaningful subsets of

variables (genes) that can yield stable clusters.

Our method is related to the bootstrap and jackknife methods in its use of sampling to estimate a statistic.[17] However, in our case, sampling is used as perturbation that *generates* the statistic. (Alternatively, one can add noise to the data instead of sub-sampling.) We also note that generating pairs of clusterings can be performed in various ways: comparing pairs of clustered subsamples as done here; comparing clustered subsamples to a reference clustering; or dividing the data into two, clustering both parts, and obtaining a second clustering of each part by assigning its points to clusters according to the nearest cluster center of the other part. Stability of a classifier is a notion that was applied in supervised learning as well.[18] It was shown that stability can be used to bound the prediction error of a classifier: the more stable the classifier, the more likely it is to perform well. In future work we plan to extend this theoretical framework to the case of unsupervised learning. Finally, the notion of stability can be applied in other types of data analysis problems whose objective is to detect structure in data, e.g. extraction of gene networks, or ranking of genes according to their predictive power.

6 Conclusion

Determining the optimum number of clusters in data is an ill posed problems for which many solutions have been proposed. None of them is widely used, and the level of their performance is data dependent.[10] In this paper we propose to use the distribution of pairwise similarity between clusterings of sub-samples of a dataset as a measure of the stability of a partition. The number of clusters at which a transition from stable to unstable clustering solutions occurs can be used to choose an optimal number of clusters. In all the experiments we ran, the results coincide with the intuitive choice. Whereas most model selection methods give a result without attaching a level of confidence to it, the sharpness of the transition from stable to unstable solutions can give information on how well defined the structure in the data is, and unlike most other methods it can provide information on the lack of structure. In another study we have found it useful as a comparative tool that can help in choosing various aspects of the clustering such as the number of principal components to cluster, and which normalization to use. Thus we view our method as a general exploratory tool, and not just as a way of selecting an optimal number of clusters.

1. M. Eisen et al, "Genetics cluster analysis and display of genome-wide expression patterns" *Proc. Natl. Acad. Sci. USA* 95, 14863–14868 (1998).
2. J. Quackenbush, "Computational anaysis of microarray data" *Nature Reviews Genetics* 2(6), 418–427 (2001).
3. R. Shamir and R. Sharan, "Algorithmic approaches to clustering gene expression data" In T. Jiang, T. Smith, Y. Xu, and M.Q. Zhang, editors, *Current Topics*

in Computational Biology (MIT Press, 2001).
4. G. Getz, E. Levine, and E. Domany, "Coupled two-way clustering analysis of gene microarray data" *Proc. Natl. Acad. Sci USA* 94, 12079–12084 (2000).
5. S.P. Smith and R. Dubes, "Stability of a hierarchical clustering" *Pattern Recognition* 12, 177–187 (1980).
6. E. Levine and E. Domany, "Resampling method for unsupervised estimation of cluster validity" *Neural Computation*, to appear.
7. E.B. Fowlkes and C.L. Mallows, "A method for comparing two hierarchical clusterings" *Journal of the American Statistical Association* 78(383), 553–584 (1983).
8. M. Bittner et. al, "Molecular classification of cutaneous malignant melanoma by gene expression profiling" *Nature* 406(3), (2000).
9. A. Ben-Hur and I. Guyon, "Detecting stable clusters using principal component analysis" In *Methods in Molecular Biology* (Humana Press, to be published).
10. G.W. Milligan and M.C. Cooper, "An examination of procedures for determining the number of clusters in a data set" *Psychometrika* 50, 159–179 (1985).
11. R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a dataset via the gap statistic" *J. Royal. Statis. Soc. B*, to appear.
12. C. Fraley and A.E. Raftery, "How many clusters? which clustering method? answers via model-based cluster analysis" *Computer Journal* 41 548–588 (1998).
13. A. Ben-Hur, D. Horn, H.T. Siegelmann, and V. Vapnik, "A support vector method for hierarchical clustering" In *Advances in Neural Information Processing Systems 13* 367–373 (MIT Press, 2000).
14. K.Y. Yeung, D.R. Haynor, and W.L. Ruzzo, "Validating clustering for gene expression data" *Bioinformatics* 17(4), 309–318 (2001).
15. A.K. Jain and R.C. Dubes, *Algorithms for clustering data* (Prentice Hall, Englewood Cliffs, NJ, 1988).
16. M.P.S. Brown, W.N. Grundy, D. Lin, N. Cristianini, C. Sugnet, M. Ares, and D. Haussler, "Knowledge-based analysis of microarray gene expression data by using support vector machines" *Proc. Natl. Acad. Sci. USA* 97(1), 262–267 (2000).
17. B. Efron, *The Jackknife, the Bootstrap and Other Resampling Plans* (SIAM, Philadelphia, 1982).
18. O. Bousquet and A. Elisseeff, "Algorithmic stability and generalization performance" In *Advances in Neural Information Processing Systems 13* (MIT press, 2000).