Stability-based model selection for clustering
Clustering

- Clustering is the art of finding groups in data (Kaufman and Rousseeuw, 1990).
- What is a cluster?
  - Group of objects separated from other clusters
- Most clustering algorithms do not give information whether a partition is meaningful.
Silhouettes

The Silhouette coefficient is defined for each example

\[ s(x) = \frac{b(x) - a(x)}{\max(b(x), a(x))} \]

Sort \( s(x) \) and group by cluster:

Figures generated using the scikit-learn silhouette method
Silhouettes

The Silhouette coefficient is defined for each example

\[ s(x) = \frac{b(x) - a(x)}{\max(b(x), a(x))} \]

Sort \( s(x) \) and group by cluster:
Hierarchical clustering

Algorithm outline:

Start with each data point in a separate cluster

At each step merge the closest pair of clusters
Stability under sampling

Premise: a clustering algorithm has captured some of the structure in a dataset if clustering solutions over different subsamples are similar.
The main idea:

**Perturb** the data (by sampling or adding noise)

**Compare** pairs of clusterings (either to a reference or among pairs of perturbed clusterings)
Comparing clusterings

It’s not easy because the label of a cluster is arbitrary!

One possible solution:

Characterize a clustering using a matrix $C$

$$C_{ij} = \begin{cases} 
1 & \text{if } x_i \text{ and } x_j \text{ belong to the same cluster and } i \neq j, \\
0 & \text{otherwise.}
\end{cases}$$

Given two clusterings, compute their dot product:

$$\sum_{i,j} C_{ij}^{(1)} C_{ij}^{(2)}$$

Normalize it to be between 0 and 1.
Generating the distribution of cluster stability

**Input:** $X$ \{a dataset\}, $k_{\text{max}}$ \{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: $\text{cluster}(X, k)$; a similarity measure between labels: $s(L_1, L_2)$

```plaintext
1: $f = 0.8$
2: **for** $k = 2$ to $k_{\text{max}}$ **do**
3:     **for** $i = 1$ to num\_subsamples **do**
4:         $\text{sub}_1 = \text{subsamp}(X, f)$ \{a sub-sample with a fraction $f$ of the data\}
5:         $\text{sub}_2 = \text{subsamp}(X, f)$
6:         $L_1 = \text{cluster}(\text{sub}_1, k)$
7:         $L_2 = \text{cluster}(\text{sub}_2, k)$
8:         Intersect$ = \text{sub}_1 \cap \text{sub}_2$
9:         $S(i, k) = s(L_1(\text{Intersect}), L_2(\text{Intersect}))$ \{Compute the similarity on the points common to both subsamples\}
10:     **end for**
11: **end for**
```

Figure 2: The Model explorer algorithm.

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. The advantage of using a hierarchical clustering method is that the same
Data

A subset of yeast genes that belong to five categories; expression measured in 79 different conditions:
Data

Here are the next 3 principal components:
Stability as a function of number of clusters

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

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. We used the MYGD functional annotation to choose the 5 functional classes that were most learnable by SVMs, and that were noted by Eisen et al. to cluster well. 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)
When there is no structure

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 $P(s_k > ¥)$, where $s_k$ is the random variable that denotes the similarity between partitions into $k$ clusters, and $¥$ is a constant. A value of $¥ = 0.9$ 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. They are among the methods tested by Milligan and Cooper. 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 208 points distributed randomly on the unit cube
Using stability to select how many PCs

Stability for varying number of principal components:
Using stability to select how many PCs

Dendrogram using all PCs vs two PCs:
Using stability to select how many PCs

Let’s put some information on the dendrogram:

- Typical nearest neighbor distance
- Smallest inter-cluster distance

1. Use stability to select how many PCs.
2. Place some information on the dendrogram.

1. Typical nearest neighbor distance.
2. Smallest inter-cluster distance.
Conclusions

Properties of the method:

- Can be used with any clustering method
- Makes no assumptions about the data
- Can detect lack of structure
- A principled approach that seems to work well
- Computationally expensive
- Can be used to select other “parameters” of the clustering

Things I have not covered here:

- Stability of individual clusters
References

Sources:

Code: