Cluster Stability Analysis using Sub-sampling

Reda Alhajj, Osman Abul
Department of Computer Science,
University of Calgary, Calgary, Alberta, Canada.
{abul, alhajj}@cpsc.ucalgary.ca

Faruk Polat
Department of Computer Engineering,
Middle East Technical University, Ankara, Turkey
polat@ceng.metu.edu.tr

Abstract

Cluster stability research is involved with the validity of clusters generated by a clustering algorithm. In other words, it answers whether generated clusters are true clusters or due to chance. Estimating true numbers of clusters is related to this problem, since often the cluster validity is based on this estimate. In the literature, there are a number of methods available for both purposes. In most of the cases, assessing validity turns out to be determining the best parameter of clustering algorithm. The confidence estimation is addressed in relatively less number of research papers. In those, confidence is given in terms of the proportion of cases clustering together. Our motivation is making confidence estimation about the clusters itself, i.e. not specifically addressing specific cases. Here we propose three meta-methods from this perspective for cluster stability problem. To the best of the our knowledge, these methods are novel. The methods are all based on sub-sampling of the dataset. The methods are general and can be used with evaluation of clustering generated by wide range of clustering algorithms available.

The first method, first makes a clustering using given clustering algorithm and cluster count. Next, it randomly samples from the labelled clusters, then it builds a supervised classifier on the selected subset, the induced classifier evaluates the non-selected portion. Random sub-sampling and evaluation steps are repeated many times, finally the overall accuracy gives the stability of the clustering. To find the best stable clustering for the given algorithm, overall steps are repeated for all possible number of clusters and best stable clustering is chosen for confidence estimation. Instead of random sub-sampling, 10-fold cross-validation is also employed.

The second method is based on the subset selection of original clusters. First of all given clustering algorithm finds clusters. For each subset of these clusters, an algorithm that estimates the true number of clusters is used. The argument here is that, if initial clustering is stable, then for each subset of it we expect number of clusters estimated is the same as cardinality of selected subset. The above single step is for assessing the reliability of cluster itself. If the reliability of randomized algorithm like k-means is the concern, the overall steps are repeated for averaging. The confidence is computed as the ratio of correct estimations. It may be the case that, clustering has given large number of clusters (e.g. say 20 clusters). In this case, trying all subsets become computationaly-intractable so we resort to subset sampling instead.

The third method uses the idea that if a cluster is stable, further clustering the cases in the cluster will reveal one cluster. For each of the clusters, an estimator algorithm is run and expected to give that there is one cluster. The whole step is repeated many times with sub-sampling of dataset, i.e. a bootstrapping approach. Confidence is computed similar to the second method. Bootstrapping approach is employed for confidence estimation. The second and third method can also be used for selecting the best number of clusters in the sense that give highest confidence.

1 Introduction

The word "clustering" (a.k.a. unsupervised classification) refers to methods of grouping objects based on some similarity measure between them. The algorithms for clustering can be classified into four classes, *Partitional*, *Hierarchical*, *Density-based* and *Grid-based*, [Halkidi01]. For each of class there are subclasses and different approaches, e.g. conceptual, fuzzy, self-organizing maps etc. The clustering task consists of all the steps of clustering problem and can be divided into five steps (last two is optional) [Jain99].

1. Pattern representation
2. Pattern proximity measure definition
3. Clustering
4. Data abstraction
5. Cluster validity analysis

In the present paper we only consider the last step which is somehow related to other steps. Given a dataset, all applicable clustering algorithms produce a clustering depending on their parameters. Usually it is the case that, different algorithms even the same algorithm with distinct parameters generate different clusterings. *Cluster validity analysis* refers how to assess the confidence in the resulting clusters.

For a few dimensional datasets, the clustering result can be visualized and clusters can be validated by human experts. But, for large dimensions it becomes nearly impossible, so other methods that are automatic are needed. *Compactness* (i.e. members of each cluster should closer to each other) and *separation* (i.e. the clusters should be widely spaced) are the main criteria for evaluation of clustering results [Halkidi01]. Based on these criteria a number of indices are proposed for evaluating clusters and selection of best cluster numbers.

We attempt to cluster validity problem and propose three algorithms. In the first method, initial clustering results are tried to be validated by supervised classifiers. The dataset is divided into training and test sets and accuracy of classifier is evaluated on the test set. Since, test set is also generated by the same distribution we expect high accuracy if the initial clustering is a good one. This method computes confidence in the generalization capability of initial clustering. In the second method, the fact that if the initial clustering is a good one then each of its subsets should be good ones. This can be considered the confidence estimation of initial clustering. The third method is similar to second method and takes the dual approach that for each generated clusters it should not tend to break itself on perturbations. In other words each cluster is expected to be stable itself. By repeating this process a number of times on subsamples we get a confidence estimation.

The paper is organized as follows. In the section 2, some background and recent work are given on the cluster validity. Section 3 presents our three methods for cluster validity analysis. Experimental results are presented in Section 4. Finally, we conclude in Section 5.

2 Cluster Validity and Stability

There are basically three methods of assessment of validity, *internal*, *external* and *relative*, [Jain99], [Halkidi01], [Fridlyand01]. *Internal indices* measure how the clustering result reflects the structure inherent in the dataset. Here only the inherent features of the dataset is used for measurement, i.e. no external information is consulted. As inherent features usually between and within sum of square matrices are used. There are a number of indices available, including *silhouette, gap, gapPC*[Fridlyand01]. These indices also define how to select the best number of clusters.

In external assessment of validity there is a known priori structure, and an *external index* is computed using this structure and generated structure. These indices define a measure of degree of match between these two structures. The indices are usually defined on contingency tables of two partitions. Entry, n_{ij}, in the row i and column j of this table is number of patterns that belong to cluster i in the priori partition and cluster j in the generated partition. These indices include *Jaccard*, *Rand* and *FM*. The *FM* measure is used in *Clest* algorithm and given below, [Fridlyand01].

$$FM = \frac{(1/2)(Z-n)}{\sqrt{\sum_{i=1}^{R}\binom{n_{i.}}{2}\sum_{j=1}^{C}\binom{n_{.j}}{2}}} \tag{1}$$

where $n = \sum_{i=1}^{R}\sum_{j=1}^{C} n_{ij}$, $Z = \sum_{i=1}^{R}\sum_{j=1}^{C} n_{ij}^2$, $n_{i.} = \sum_{j=1}^{C} n_{ij}$ and $n_{.j} = \sum_{i=1}^{R} n_{ij}$. R and C is the number of clusters of priori and generated clusters, respectively.

Relative assessment compares two structures and measures their relative merit. The idea is to run the clustering algorithm for possible number of parameters (e.g. for each possible number of clusters) and identify the clustering scheme that best fits the dataset. Recent work on cluster validity research is concentrated on a kind of *relative indices* called *cluster stability*, [Roth02], [Ben-Hur02], [Ben-Hur03], [Kerr00], [Levine01], [Fridlyand01], [Zhang00]. Cluster stability exploits the fact that when multiple data sources are sampled from the same distribution, the clustering algorithms should behave in the same way and produce similar structures.

In [Roth02], a supervised predictor is built on each clustered resampling of original dataset and their match with the original clustering labelling is used as a measure of stability or degree of match. They show that selection of supervised predictor makes difference but measured validity is still valid for other choices. They define an instability measure for taking the game-theoretic approach. The number of clusters minimizing this instability measure is used as best cluster number.

[Roth02] presents an algorithm for estimating the true number of clusters. For each cluster count, dataset is resampled twice and clustered using the same generic clustering algorithm. Similarity between these two clustering is measured using the either *Jaccard coefficient* or *matching coefficient*. The resampling and similarity computations are repeated many times for each number of clusters for confidence estimation. The averaged values are used as measures of stability of clustering generated by the given clustering algorithm. The histograms and cumulative distributions are generated and plotted for selecting best cluster number. Smallest stable cluster count is estimated as the correct number of clusters. The selection is obvious in cumulative distributions diagram and they have also given a measure for automating this process. The algorithm has a nice property that if there is no gap between similarities across all cluster counts, it is said that dataset does not contain clusters, i.e. cluster count is 1.

Another resampling based method is given in [Levine01]. In their settings full dataset is clustered first and a number of subsamples are gathered from the dataset and each of them clustered independently using the same clustering algorithm. Between original clustering and each of subsampled clusterings a figure of merit measure (degree of match in the connectivity matrix) is defined. The figure of merit is computed for each possible number of parameter sets. The plot of figure of merit measure against parameter values is used to select the best parameters.

A Gaussian finite mixture based method for estimating true number of clusters is described in [Smyth96]. The algorithm first divides dataset into training and test set. For each cluster count k a model is fitted to training set using the *Expectation Maxmization* (ML) algorithm. Resulting parameter set is evaluated on test set. These steps are repeated many times and average of them are used as estimates.

3 Methods

We denote the input dataset by T having n patterns each of them having dimension of p. So, T is effectively $n \times p$ matrix.

The algorithms can be used for different number of cluster counts and different clusterings either generated by different clustering algorithms or hand-constructed clusters. In the case of comparing different clustering algorithms we collect the confidence measure of them for all possible number of clusters. These data can be used for relative confidence estimation of clustering algorithms on the given dataset. Any clustering algorithm operating on numeric values (e.g. k-means, ORCLUS, PAM, CLARA) having the cluster count as a parameter can be used confidence estimation. For randomized algorithms like k-means confidences should be averaged on several runs. Our methods enable someone to compare a number of clustering algorithms on a given dataset based on their confidences in the stable clusterings.

The ORCLUS algorithm is proposed for high-dimensional datasets. The idea behind the algorithm is finding (potentially) different arbitrarily projected subspaces for each of the clusters. It is an iterative algorithm and starts with an initial partitions and original axis-system. In each iteration, first of all patterns are assigned to a cluster based on projected distance of them to seeds of current clustering. Then, centroid of clusters (seeds) are recomputed and the new projected subspaces are computed for each of the clusters. Following this, closer seeds are merged to obtain less number of clusters. Iteration continues until user-specified number of clustering is found and the projected subspace dimensionality of each cluster is reached to user-specified minimum.

Contrary to feature selection methods which selects dimensions in the larger eigenvalues, the algorithm selects smaller eigenvalue subspaces. The reason behind this is to reduce the variability in the projected subspace, i.e. reduce the distance within cluster. The algorithm has capabilities of

detecting outliers and scales to very large databases, for details see [Aggarwal02].

3.1 Validity estimation using supervised learning

The method validates the result of clustering with supervised classifiers. The idea behind this method is if the labels generated by cluster algorithm is valid (i.e. clusters are well-separated) the classifier using this labelling will classify them with high accuracy. To test the validity of clustering result, we train classifier on perturbed version of labelled patterns, and test it on the patterns not selected for training. For estimating confidence the subsampling is repeated many times. The average accuracy is used as a measure of confidence in the validity of clustering.

Input:T=dataset, K=number of clusters, B=number of subsampling

1. f=0.7
2. $L = Cluster(T, K)$
3. For b=1 to B do
4. $\quad L_b = subsample(L, f)$
5. $\quad C_b = Build\_Classifier(L_b)$
6. $\quad A_b = Compute\_Accuracy(C_b, L - L_b)$
7. end do
8. $A = \frac{1}{B} \sum_{b=1}^{B} A_b$

Figure 1: Validity estimation using supervised learning algorithm

The algorithm is sketched in Figure 1. In the step 2, any clustering algorithm that partitions the patterns can be used. In the step 5 of the algorithm we use the *Diagonal Linear Discriminant Analysis* (DLDA) algorithm [Dudoit01]. Authors experimented with several algorithms and DLDA is found to be one of the best in their settings and datasets. It is also employed in the *Clest* algorithm, a cluster estimation/ validation method using discriminant analysis approach [Fridlyand01].

DLDA is based on *Maximum Likelihood* (ML) approach. Classifier C classifies an instance x by using the class conditional probabilities, i.e.

$$C(x) = \arg\max_k P(x|y = k) \tag{2}$$

For multivariate normal class density probabilities, i.e. $P(x|y = k) \sim N(\mu_k, \Sigma_k)$, the classifier becomes

$$C(x) = \arg\min_k \{(x - \mu_k)\Sigma_k^{-1}(x - \mu_k)' + \log|\mu_k|\} \tag{3}$$

The special case is obtained when the the class densities have the same diagonal covariance matrix. In this case, the classification formula known as DLDA discrimination rule is obtained as follows,

$$C(x) = \arg\min_k \sum_{j=1}^{p} \frac{(x_j - \mu_{kj})^2}{\sigma_j^2} \tag{4}$$

3.2 Validity using subset of clusters

The method is designed for testing the stability of each subset of clusterings. If the initial clustering is valid, then the every subset of it is expected to be valid. To test the validity of subsets, subsampling based cluster count estimation algorithms can be used. By this way, the the confidence in validity is computed based on stability of subset of original clustering.

The algorithm is given in Figure 2. In each iteration, a subset of labelling is selected randomly in step 3. For K clusters, cluster $i, 1 \leq i \leq K$ is selected with probability α, i.e. uniform selection. We set the $\alpha = 0.5$ to make the expected value of selected cluster label size $K/2$.

In the step 4 of the algorithm, a prediction-based resampling algorithm, *Clest*, [Fridlyand01] is used. In fact *Clest* is a method having several parameters and instantiations of parameters result in different algorithms. For example, actual clustering and classifier algorithms are generic. The algorithm is given in Figure 3.

Input: T=dataset, K=number of clusters, B=number of subset subsampling, kmax=maximum number of clusters

1. $L = Cluster(T, K)$
2. For b=1 to $min(B, 2^{|K|} - 1)$ do
3. $\quad L_b = patterns\ belonging\ to\ b'th\ subset\ of\ K$
4. $\quad K_b = Estimate\_ClusterCount(L_b, kmax)$
5. $\quad A_b = 1_{(K_b == number\ of\ cluster(L_b))}$
6. end do
7. $A = \frac{\sum_{b=1}^{min(B,2^{|K|}-1)} A_b}{min(B,2^{|K|}-1)}$

Figure 2: Validity using subsets of clusters algorithm

Since clustering and classifier algorithms are generic, one should select concrete algorithms. In the original algorithm authors selected *Partitioning Around Medoids* (PAM) algorithm for clustering and DLDA for classification.

3.3 Validity using cluster tendency

In this method, every cluster generated by clustering algorithm by subsampling is evaluated against null hypothesis that there is only one cluster. The motivation for this method is if a clustering algorithm produces reasonable structures for every subsamples, then every cluster is expected to be a tight structure, i.e. structure not having any further tendency of sub- clusters.

The algorithm is presented in Figure 4. In step 6, we use the *Clest* algorithm for cluster count estimation.

4 Experiments and Results

For all of the methods, we analyze the clusterings generated by well-known k-means and ORCLUS algorithms. The analysis is done for cluster counts of 2 to 10 for all the datasets. The parameter B of all three methods is set to 50. In all of the experiments the parameters of *Clest* algorithm given in Figure 3 is set as follows, $pmax = 0.05,\ dmin = 0.05,\ size\ of\ learning\ set = 2n/3,\ B = 20,\ B_0 = 20\ and\ kmax = 10$.

5 Conclusion

References

[Jain99] Jain, A.K., Murty, M.N., Flynn, P.J. *Data Clustering: A Review.* ACM Computing Surveys, Vol 31, No.3. 1999.

[Halkidi01] Halkidi, M., Batistakis, Y., Vazirgiannis, M. *On Clustering Validation Techniques.* Journal of Intelligent Information Systems Vol.17:2-3. 2001.

[Ben-Hur02] Ben-Hur, A., Elisseeff, A., Guyon, I. *A stability based method for discovering structure in clustered data.* Pacific Symposium on Biocomputing. 2002.

[Ben-Hur03] Ben-Hur, A., Guyon, I. *Detecting stable clusters using principal component analysis.* In Methods in Molecular Biology, M.J. Brownstein and A. Kohodursky (eds.) Humana press, pp. 159-182. 2003.

[Kerr00] Kerr, M.K., Churchill, G.A. *Bootstrapping Cluster Analysis: Assessing the Reliability of Conclusions from Microarray Experiments.* Proceedings of the National Academy of Sciences. 2000.

[Levine01] Levine, E., Domany, E. *Resampling Method for Unsupervised Estimation of Cluster Validity.* Neural Computation. 2001.

Input: T=dataset, K=number of clusters, B=number of runs, B_0= number of resampling, kmax=maximum number of clusters

1. $T_0 = T$
2. For k=2 to kmax do
3. For i=0 to B_0 do
4. For b=1 to B do
5. Randomly split the T_i into non-overlapping learning and test sets
6. Apply clustering algorithm P to to the learning set
7. Build a classifier using the labelled learning set
8. Apply the resulting classifier to the test set
9. Apply the clustering algorithm to the test set
10. $s_{k,i,b} = FM$ external index comparing the two sets of labels
11. end do
12. $t_{k,i} = median(s_{k,i,1}, \cdots, s_{k,i,B})$
13. T_{i+1}= Randomly generate uniform dataset in the range of T
14. end do
15. end do
16. $t_k^0 = \frac{1}{B_0} \sum_{i=1}^{B_0} t_{k,i}$
17. $p_k = \frac{\#\{i \mid t_{k,i} \geq t_{k,0},\ i=1...B_0\}}{B_0}$
18. $d_k = t_{k,0} - t_k^0$
19. $K = \{k | 2 \leq k \leq kmax\ ,\ p_k \leq pmax,\ d_k \geq dmin\}$
20. $\widehat{K} = \begin{cases} 1 & \text{if empty(K),} \\ \arg\max_{k \in K} d_k & \text{otherwise.} \end{cases}$

Figure 3: Clest algorithm

[Fridlyand01] Fridlyand, J., Dudoit, S. *Applications of resampling methods to estimate the number of clusters and to improve the accuracy of a clustering method.* University of California, Statistics Department Technical Report No:600. 2001.

[Dudoit01] Dudoit, S., Fridlyand, J., Speed, T. *Comparison of Discrimination methods for the classification of tumors using gene expression data.* Journal of American Statistical Association. 2001.

[Roth02] Roth, V., et al. *Stability-Based Model Order Selection in Clustering with Applications to Gene Expression Data.* ICANN. 2002.

[Keller00] Keller, A., et al. *Bayesian Classification of DNA Array Expression Data.* University of Washington, Computer Science Department Technical Report No: UW-CSE-2000-08-01. 2000.

[Aggarwal02] Aggarwal, C., Yu, P.S. *Redefining Clustering for High-Dimensional Apllications.* IEEE Transactions on Knowledge and Data Engineering Vol.14. 2002.

[Smyth96] Smyth, P. *Clustering using Monte Carlo Cross-Validation.* KDD'96. 1996.

[Buhmann02] Buhmann, J.M. *Learning and Data Clustering.* Handbook of Brain Theory and Neural Networks, MIT Press. 2002.

[Zhang00] Zhang, K., Zhao, H. *Assessing Reliability of gene Clusters from Gene Expression Data.* Functional Genomics. 2000.

Input: T=dataset, K=number of clusters, B=number of subsampling kmax= maximum number of clusters

1. f=0.7
2. For b=1 to B do
3. $T_b = subsample(T, f)$
4. $L_b = Cluster(T_b, K)$
5. for each cluster $c \in L_b$ do
6. $K_{b,c} = Estimate\_ClusterCount(c, kmax)$
7. $A_{b,c} = 1_{(K_{b,c}==1)}$
8. end do
9. end do
10. $A = \frac{1}{B}\sum_{b=1}^{B} \frac{\sum_{c=1}^{number\ of\ cluster(L_b)} A_{b,c}}{number\ of\ cluster(L_b)}$

Figure 4: Validity using individual clusters algorithm