A Resampling Approach to Cluster Validation

Volker Roth, Tilman Lange, Mikio Braun, and Joachim Buhmann
University of Bonn, Institute of Computer Science III,
Römerstrasse 164, D-53117 Bonn, Germany,
{*roth, lange, braunm, jb*}@*cs.uni-bonn.de*

Abstract. The concept of *cluster stability* is introduced as a means for assessing the validity of data partitionings found by clustering algorithms. It allows us to explicitly quantify the quality of a clustering solution, without being dependent on external information. The principle of maximizing the cluster stability can be interpreted as choosing the most *self-consistent* data partitioning. We present an empirical estimator for the theoretically derived stability index, based on imitating independent sample-sets by way of resampling. Experiments on both toy-examples and real-world problems effectively demonstrate that the proposed validation principle is highly suited for model selection.

Keywords. Unsupervised learning, cluster validation, model selection

1 Introduction

Unsupervised learning or clustering aims at extracting hidden structure in a data set. This goal requires to find an answer to some fundamental questions: (i) what structure are we interested in? (ii) what kind of clustering model provides a suitable method for extracting the desired structure? (iii) what is the "correct" number of clusters? The first question can be viewed as a problem definition, the last two ones are usually referred to as the problem of cluster validation. In general, validating clustering solutions means evaluating results of cluster analysis in a quantitative and objective fashion. Such an evaluation can be based on two types of criteria: (i) External criteria: a clustering solution is matched to a priori information, i.e. external information that is not contained in the dataset. (ii) Internal criteria: the quality measure is exclusively based on the data themselves. Internal criteria can roughly be subdivided into those methods which assess the fit between the data and the expected structure, and techniques which focus on the stability of the solution.

In this work we present a resampling-based method for cluster validation. According to the above taxonomy, it is an internal criterion which requires no prior information. The central ingredient of the validation method is the notion of *cluster stability*. Stability measures the variability of solutions, which were built on two independent samples from the same data source. The stability concept has a clear theoretical interpretation as choosing the most self-consistent data partitioning. Clustering is treated as the problem of identifying the *hypothetical* supervisor, that explains the data in the most consistent way. Among all possible labeling of the data (i.e among all possible supervisors), we select the one which provides us with the smallest expected number of misclassifications on a test set, given a fixed grouping algorithm.

Seen from a technical viewpoint, our resampling method refines related approaches initially described in [1] and later generalized in different ways in

[4] and [5]. From a conceptual viewpoint, however, it extends these heuristic approaches by providing a clear theoretical background for model selection in unsupervised clustering problems.

2 Stability and risk minimization

Applying the stability concept to unsupervised clustering problems is only meaningful, if we think about clustering as partitioning the *whole object space*. This view of the problem requires the user to specify a predictive inference mechanism for data-grouping methods. While this may be viewed as a very strong requirement, we believe that otherwise any general concept of purely data-dependent (internal-) cluster validity would be inherently ill-defined.

Contrary to classification problems, algorithms for clustering data cannot be stated in a purely discriminative fashion. The missing labeling information requires an additional modeling step: given a set of physical measurements (or *features*) describing the objects to be partitioned, we must specify how these features relate to the structure we are interested in. This is usually done by specifying a grouping *principle*, such as inter-cluster compactness criteria or intra-cluster separation measures. Based on such a principle, different grouping algorithms can be formulated. In spite of the fact that the stability framework also applies to this general setup, in the following we will restrict ourselves to the more intuitive situation in which we have already specified an algorithm. The open question within this framework is the "correct" number of clusters, a problem which is usually called the *model order selection problem*.

We may formalize this by considering potential supervisors, each of which labels the n objects contained in a training set by a different number $2 \leq k < n$ of clusters (we do not consider the degenerate case $k = 1$ which of course always has perfect stability). A grouping algorithm α assigns labels to objects, and in this sense it defines a potential supervisor. For this supervisor, α can then be viewed as a classifier with zero empirical risk. More precisely, we have the following situation: a data generator produces a series of i.i.d. sets of n objects

$$O_1^n, \ldots, O_l^n, \ldots \tag{1}$$

If the set O_i^n is presented to a fixed clustering algorithm α, we obtain a set of hypothetical labels

$$Y^n := \{\alpha_i(O_i^n)\}, \tag{2}$$

where the variable Y takes values in the range of $\{1, \ldots, k\}$. In the following we will always denote by α_i a fixed algorithm trained on the sample O_i^n. Despite the clustering algorithm is considered fixed, the actual partition of the object space still depends on the training set. The principle of cluster stability estimates the degree of this dependence with respect to differences between partitions. Presenting a second set of objects O_j^n to the algorithm, we define (in-)stability as the empirical risk of the rule α_j on the test-set O_i^n, assuming the true labels are those defined in (2). Note that this is a measure of the generalization ability of α_j, since both sets O_i^n and O_j^n are drawn *independently* from the same data source. Denoting with $\mathbf{1}\{\cdot\}$ an indicator function, our instability functional is defined as the empirical probability of false predictions (y_l are the ones defined in equation (2))

$$d''\left[\alpha_i(O_i^n), \alpha_j(O_i^n)\right] = \frac{1}{n}\sum_{l=1}^{n} \mathbf{1}\{y_l \neq \alpha_j(o_l)\}, \text{ with } O_i^n = \{o_1, \ldots, o_n\}. \tag{3}$$

It is important, that the setting is asymmetric in the sense that the rule α_j is used in a predictive sense, whereas α_i is evaluated on its training set. If we would have tested both rules on a *third* set, it is possible to measure high stability in overfitting situations where both rules cannot generalize at all: consider for example an algorithm that only memorizes the objects seen in the training set, and always predicts label i for unseen objects. The same problem occurs, if we measure the stability based on a *common fraction* of the learning sets. The latter also explains, why stability measures are only meaningful with respect to generalization ability, if the grouping algorithm can be used in a predictive sense.

Selecting a grouping solution with high stability corresponds to identifying a hypothetical supervisor, for which the chosen learning algorithm provides on average a small risk on a test set of size n (we averaged over many tuples of learning- and training sets). Since by assumption all solutions have zero empirical training risk, the principle of favoring stable solutions can be viewed as selecting the most self-consistent labeling.

Due to the inherent permutation symmetry of clustering algorithms, however, the following problem arises in equation (3): even if exactly the same objects are grouped together, the labels may be arbitrarily permuted. To overcome this problem, we consider the loss incurred for all permutations π of the set of predicted labels $\alpha_j(O_i^n)$, and choose the one with smallest risk:

$$d_i'(\alpha_i, \alpha_j) := \min_{\pi} d'' \left[\alpha_i(O_i^n), \pi(\alpha_j(O_i^n))\right]. \tag{4}$$

Fortunately, there is no need to explicitly examine all $k!$ possible matches. We can rather use the *Hungarian method* for solving *minimum weighted perfect bipartite matching* problems with computational complexity of $\mathcal{O}(k^3)$ ([6], p. 248).

It holds that $d_i'(\alpha_i, \alpha_j) \leq 1 - 1/k$. To see this let α, β be two labelings, and let π be drawn uniformly from the set of all permutations of k indices. Then, straightforward calculations show that $E_\pi d''(\alpha, \pi(\beta)) = 1 - 1/k$ and therefore $\min_\pi d''(\alpha, \pi(\beta)) \leq 1 - 1/k$. This bound is tight: Let ρ be the *random predictor* which assigns labels uniformly at random. It can be shown that the instability of ρ converges to $1 - 1/k$ as $n \to \infty$.

In the non-asymptotic case, we cannot neglect the influence of the minimization over permutations. The estimator is biased towards higher stability. However, we can easily estimate the random stability by sampling. In order to use the (in)stability value for selecting the "correct" number of clusters k, the dependency of $d_i'(\rho_i, \rho_j)$ on k requires us to normalize the measurements. Taking expectations with respect to different pairs of sets (O_i^n, O_j^n), the final estimator that ensures fair comparisons between different values of k then reads:

$$\hat{d}(\alpha) = \frac{E_{i,j}[d_i'(\alpha_i, \alpha_j)]}{E_{i,j}[d_i'(\rho_i, \rho_j)]} \tag{5}$$

In practice, different pairs of sets (O_i^n, O_j^n) are hardly available (note that the size of the individual sets n should be large), but we may emulate these sets by iteratively splitting the total set of objects into two disjoint halves.

3 The Algorithm in Practice

With the theoretical derivation of the stability concept in the last section, a practical algorithm can now be outlined as follows:

1. Split the object set of size $2n$ into two sets of equal size, O_1^n and O_2^n.
2. Present the first dataset to the algorithm. The result is the mapping α_1 of each of the objects in O_1^n to one of k clusters.
3. Apply α to the second set O_2^n. Use α_2 to predict the cluster membership of all objects contained in the first set.
4. Set O_1^n now has two different labelings. Find the correct permutation of labels by using the Hungarian method for *minimum weighted perfect bipartite matching*. The costs for identifying labels i and j are the number of miss-classifications with respect to the labels $Y^n = \alpha_1(O_1^n)$ (these are assumed correct).
5. Normalize with respect to the random stability.
6. Iterate the whole procedure from step 1 to 5, average over assignment costs and compute the expected (in-)stability value.
7. Iterate the whole procedure for each k to be tested.

4 Experiments

In a first **toy example**, we demonstrate the model selection performance of the proposed stability estimator for synthetically generated datasets. We have drawn data samples from 3 different sources forming concentric rings in the 2-dimensional plain. A suitable clustering method for line-shaped data of this kind is *Path-based Clustering*, [2]. Figure 1 shows the estimated (in-)stability curve for this grouping model. We can clearly identify the solution for $k = 3$ as the most stable partition with an estimated risk of $\hat{d}(\alpha) < 0.1\%$. This model significantly outperforms all other choices of k.

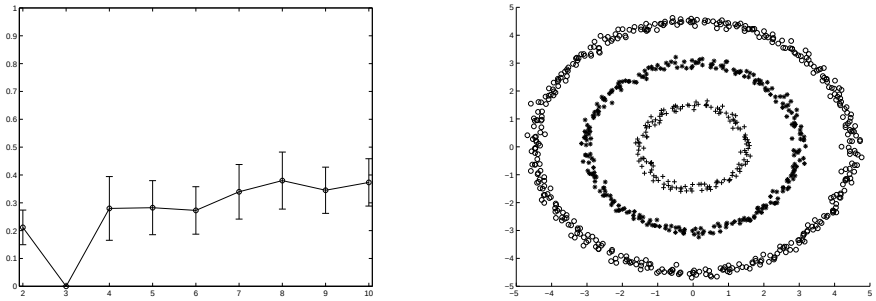

Fig. 1. Averaged instability curve and standard deviations for the 3 rings toy-example (left panel). The most stable solution with $k = 3$ (right panel).

The **Iris Dataset** is perhaps the best known database to be found in the pattern recognition literature [3]. The data report four characteristics (sepal width, sepal length, pedal width and pedal length) of three species of Iris flower. Each class contains 50 instances. One class is linearly separable from the other two, the latter are not linearly separable from each other. These observations are important because we use k-means for clustering, which generates linear class boundaries. Thus, we expect a two-cluster solution to two be very stable on different subsets. Estimating three clusters should also be

possible with high stability. Our experiments nicely coincide with these expectations: we have randomly split the data 30 times in two subsets, applied a k-means clustering algorithm (optimized by deterministic annealing), and measured the matching costs between the two partitions. The resulting instability curve is depicted in figure 2. The fact that the most stable number of clusters is two and not three reflects from our design decision to link cluster validity to stability only and to discard improvements of model fit by more complex clustering solutions. Two clusters yield a larger distortion for the Iris data than three clusters but the stability is lower for three clusters since linear separability is violated.

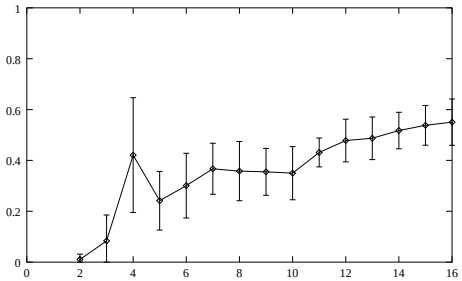

Fig. 2. Estimated instability for the Iris dataset vs number of classes, and standard deviations. The solution for $k = 2$ is extremely stable (risk $\hat{d}(\alpha) \approx 0.1\%$, the one for $k = 3$ still has a relatively low estimated risk $\hat{d}(\alpha) \approx 8\%$.

The **yeast cell cycle dataset** [7] shows the fluctuation of expression levels of yeast genes over cell cycles. Using periodicity and correlation algorithms, (Spellman *et al.*) identified 800 genes that meet an objective minimum criterion for cell cycle regulation. By observing the time of peak expression, (Spellman *et al.*) crudely classified the 800 genes into 5 different groups.

In our experiments, we investigated both the validity of clustering solutions and their accordance with the classification proposed. From the complete dataset (available at `http://cellcycle-www.stanford.edu`), we used the 17 *cdc28* conditions for each gene, after log-transforming and normalization to zero mean and unit variance. We grouped the 17-dimensional data by minimizing the k-means cost function using a deterministic annealing strategy. The estimated instability curve over the range of $2 \leq k \leq 20$ is shown in figure 3. For each k, we averaged over 20 random splits of the data. A dominant peak occurs for $k = 5$, with an estimated misclassification risk of $\hat{d}(\alpha) \approx 19\%$.

In order to compare a 5 cluster solution with the labeling proposed by (Spellman *et al.*), we again used the bipartite matching algorithm to break the inherent permutation symmetry. The averaged agreement rate over all 5 groups is $\approx 52\%$. This may be viewed as a rather poor performance. However, the labels should not be considered as the "ground truth", but as a "crude classification with many disadvantages" as stated by the authors. Moreover, a closer view on the individual agreement rates per group shows, that at least two groups could be matched with more than 70% agreement.

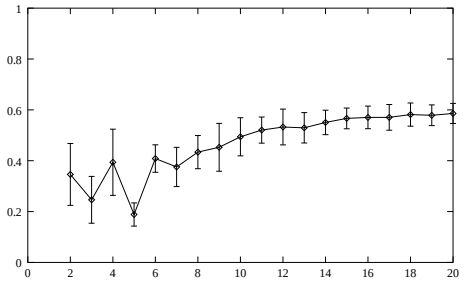

Fig. 3. Estimated instability for the Yeast Cell-Cycle dataset vs. number of classes.

5 Conclusions

We have introduced the concept of *cluster stability* as a means for solving the model order selection problem in unsupervised clustering. Given two independent object-sets from the same source, we recast grouping problems as a supervised classification task. In this scenario, the first grouping solution identifies a hypothetical supervisor, for which it provides a classifier with zero empirical risk. Presenting the second object-set to the fixed grouping algorithm, the (in-)stability functional measures the empirical probability of misclassifications with respect to the labeling by the identified supervisor. Taking expectations over tuples of object-sets, and normalizing by the stability of a random predictor, we derive a stability measure that allows us to compare solutions for different numbers of clusters in a fair and objective way. In order to estimate the cluster stability in practical applications, we introduced an empirical estimator that emulates independent samples by way of iteratively splitting the total object-set. Unless many other validation indices proposed in the literature, the estimated instability has a clear interpretation in terms of misclassification risk. The results presented in section 4 effectively demonstrate that cluster stability is a suitable measure for estimating the most self-consistent data partitioning.

References

1. J. Breckenridge. *Replicating cluster analysis: Method, consistency and validity.* Multivariate Behavioral research, 1989.
2. B. Fischer, T. Zöller, J. Buhmann. *Path Based Pairwise Data Clustering with Application to Texture Segmentation.* In: Energy Min. Meth. in Computer Vision and Pattern Recognition, LNCS 2134, 235-250, Springer, 2001.
3. R. A. Fisher. *The use of multiple measurements in taxonomic problems*, Ann. Eugen., 1936.
4. J. Fridlyand & S. Dudoit. *Applications of resampling methods to estimate the number of clusters and to improve the accuracy of a clustering method.* Stat. Berkeley Tech Report. No. 600, 2001.
5. E. Levine, E. Domany. *Resampling Method for Unsupervised Estimation of Cluster Validity.* Neural Computation 13: 2573-2593, 2001.
6. C.H. Papadimitriou & K. Steiglitz. *Combinatorial Optimization, Algorithms and Complexity*, Prentice-Hall, Englewood Cliffs, NJ, 1982.
7. P.T. Spellman, G. Sherlock, MQ. Zhang, V.R. Iyer, K. Anders, M.B. Eisen, P.O. Brown, D. Botstein, B. Futcher. *Comprehensive Identification of Cell Cycle-regulated Genes of the Yeast Saccharomyces cerevisiae by Microarray Hybridization.* Molecular Biology of the Cell 9, 3273-3297, 1998.