14

Resampling Methods for Unsupervised Learning from Sample Data

Ulrich Möller
Leibniz Institute for Natural Product Research and Infection Biology – Hans Knöll Institute, Germany

1. Introduction

Two important tasks of machine learning are the *statistical learning* from sample data (SL) and the *unsupervised learning* from unlabelled data (UL) (Hastie et al., 2001; Theodoridis & Koutroumbas, 2006). The synthesis of the two parts – the *unsupervised statistical learning* (USL) – is frequently used in the cyclic process of inductive and deductive scientific inference. This applies especially to those fields of science where promising, testable hypotheses are unlikely to be obtained based on manual work, the use of human senses or intuition. Instead, huge and complex experimental data have to be analyzed by using machine learning (USL) methods to generate valuable hypotheses. A typical example is the field of functional genomics (Kell & Oliver, 2004).

When machine learning methods are used for the generation of hypotheses, human intelligence is replaced by artificial intelligence and the proper functioning of this type of 'intelligence' has to be validated. This chapter is focused on the validation of *cluster analysis* which is an important element of USL.

It is assumed that the data set is a sample from a mixture population which is statistically modeled as a mixture distribution. Cluster analysis is used to 'learn' the number and characteristics of the components of the mixture distribution (Hastie et al., 2001). For this purpose, similar elements of the sample are assigned to groups (clusters).

Ideally, a cluster represents all of the elements drawn from one population of the mixture. However, clustering results often contain errors due to lacking robustness of the algorithms. Rather different partitions may result even for samples with small differences. That is, the obtained clusters have a random character. In this case, the generalization from clusters of a sample to the underlying populations is inappropriate. If a hypothesis derived from such clustering results is used to design an experiment, the outcome of this experiment will hardly lead to a model with a high predictive power. Thus, a new study has to be performed to find a better hypothesis. Even a single cycle of hypothesis generation and hypothesis testing can be time-consuming and expensive (e.g., a gene expression study in cancer research, with 200 patients, lasts more than a year and costs more than 100.000 dollars). Therefore, it is desirable to increase the efficiency and effectiveness of the scientific progress by using suitable validation tools.

An approach for the statistical validation of clustering results is data resampling (Lunneborg, 2000). It can be seen as a special Monte Carlo method that is, as a method for

finding solutions to statistical problems by simulation (Borgelt & Kruse, 2006). The choice of a *suitable* resampling method for any cluster validation task is not trivial. On the one hand, such a method is expected to simulate random samples that have the same structure that underlies the original sample – even though the true structure is unknown. On the other hand, it is undesired that the method introduces any additional structure into the simulated data, because this kind of error can not be recognized from the clustering results in the absence of the ground truth.

Once, clustering results (partitions) have been generated for a set of resamples, three steps are usually performed. i) The stability of the partitions under the influence of resampling is calculated. When desired, stability scores can be obtained also for single clusters and individual assignments of data points to clusters. ii) A *consensus partition* is determined that best possible represents the characteristics which are common to the resample partitions. iii) The number of clusters is estimated, typically based on the maximization of a partition stability score. For methods that can be used to perform the steps i) to iii) see, for example, (Strehl & Gosh, 2002; Topchy et al., 2005; Fred & Jain, 2006 and Ayad & Kamel, 2008).

Resampling-based cluster validation is not yet common standard. In many software tools for cluster analysis, resampling methods are missing. Some new methods were published only recently. The choice of the appropriate resampling technique depends on the data properties, the goal and constraints of the study and on the clustering methods used. The purpose of this contribution is to review available techniques, to summarize existing benchmark results and to give recommendations for the selection and use of the methods. Furthermore, a new method called *nearest neighbor resampling* is presented.

In statistics resampling schemes are subdivided into parametric and non-parametric methods. The use of parametric methods for cluster validation will be briefly characterized in section 2. In section 3 non-parametric methods will be reviewed. Section 4 is a summary of benchmarking tests of different resampling techniques. Section 5 refers to results of the new resampling method previously described in section 3.5. Finally, section 6 contains a discussion of the described methods and conclusions for their future application.

2. Parametric resampling

Parametric resampling is also known as *parametric bootstrapping*. Methods of this type are used to fit a parametric model to the data. That is, the hypothesis is made that the data follow a theoretical distribution and certain parameters of this distribution (mean, variance etc.) are estimated. Then resample data sets are sampled from the distribution with the parameter values set to the obtained estimates. In cluster analysis a mixture distribution $P = \Sigma_i \varepsilon_i P_i$ is assumed, where P_i, $i = 1,\ldots,C$, are the C distributions generating C "true" clusters respectively, and ε_i is the probability that a sample point from P_i is drawn.

In principle, this approach has attractive properties. Examples for the validation of clustering results obtained from gene expression data are contained in (McLachlan & Khan, 2004). However, there exist also arguments against the use of parametric resampling for cluster validation. One argument concerns the lack of justification for the (more or less arbitrary) selection of a particular theoretical distribution as a model for real data with an unknown distribution (Yu, 2003; Lunneborg, 2000). Hennig (2007) argued that parametric bootstrapping does not suggest itself for the aim of cluster validation, because parametric methods discover structures generated by the assumed model much better than patterns in real data for which the model does not hold. This could lead to overoptimistic assessments

of the stability of clustering structures. If the original sample has clearly more dimensions than data points, model fitting may be impaired by the "curse of dimensionality". Further arguments can be found in (Tseng & Wong, 2005). In the sequel, we consider non-parametric resampling methods that may be used in cluster analysis.

3. Non-parametric resampling

3.1 Sampling from a sample

Several methods can be referred to as re-sampling in the literal sense according to the common (non-statistical) definition of the word *sample* [1]. In such methods the data points of a resample are drawn from the set of data points contained in the original sample.

Bootstrapping. The non-parametric version of bootstrapping is usually described as "drawing with replacement". That is, each bootstrap sample is obtained by drawing N data points randomly and with replacement from the original sample, where N is the number of data points in this sample. If the population size N_P is finite and relatively small compared to N, (i.e., $N_P / N < 20$), another procedure is conventionally used (see Lunneborg, 2000). This procedure guarantees that the empirical distribution of the union of all bootstrap resamples agrees accurately with the empirical distribution of the original sample. In any case some original data points are likely represented more than once in a bootstrap sample, while accordingly, other original points are missing in the resample.

It has been shown that for increasing values of N, the percentage of original data which are not contained in a bootstrap sample converges to about 37%. If this information loss is considered to be too large for an adequate recognition of the data structure, the bootstrap scheme could be applied to M randomly selected points of the sample X ($M < N$), while the resample is completed by the $N–M$ points of X not used for the bootstrapping. This modification would allow to control the degree of information loss associated with the bootstrap scheme (Möller & Radke, 2006a). Moreover, this resampling version could be performed by using random numbers M_r for the generation of $r = 1, 2, \ldots$ bootstrap samples with reasonable boundaries of the interval from which the values M_r are drawn. This selection could make the results less depending on the heuristic choice of parameter M.

Subsampling. The original data set X is used to draw random subsets $Y_r \subset X$, $r = 1, 2, \ldots$ The size of a drawn subset, $S = \text{card}(Y_r)$, is a control parameter. Usually, S is fixed for all subsamples to be used in an application. If S is not much smaller than the original sample size N, clustering results of different subsamples may be very similar and not informative. The choice of S clearly smaller than N can be recommended if the information retained on average in a subsample is sufficient to obtain reasonable estimates of the unknown underlying distribution. Resampling-based clustering methods have been introduced including the subsampling of 70% (Tseng & Wong, 2005), 80% (Monti et al., 2003) and 90% (Fred & Jain, 2006) of the data. It may not be easy to select an optimal subsample size in a particular application. To avoid an inappropriate choice for this parameter, the subsample size could be varied from subsample to subsample. For example, the subsample size is uniformly drawn from an interval that represents 75-90% of the size of the original sample.

[1] A sample of things is a number of them that are chosen at random out of a larger group and then used to test ideas or to provide information about the whole group (Collins Cobuild Dictionary, 1987).

An alternative way, without an explicit specification of the subsampling size, would be to generate a bootstrap sample and to discard the identically replicated points (Hennig, 2007).
Subdivision. The original sample X is split into two disjoint subsets $Y \cup Z = X$. Clustering is used to generate the partitions π_Y and π_Z, from Y and Z, respectively. In addition, a classifier C_Y is build from the subset Y and the label set π_Y. Then C_Y is applied to the subset Z providing the partition π_{YZ}. Finally, the predictability of π_Z based on π_{YZ} is assessed. For the success of this strategy it has to be ensured that in general each subset Y and Z contain sufficient information about the underlying distribution necessary to infer a reasonable model from the data. Dudoit and Fridlyand (2003) presented an example, where the 'training' set Y and the 'test' set Z consist respectively of 2/3 and 1/3 of the original sample.

3.2 Jittering

Real data samples contain random measurement errors. Even if the same objects were observed multiple times under the same experimental conditions, the data are likely to be different. These differences can be simulated by generating copies of the original sample and adding random values to each of these data sets. The normal distribution with zero mean is traditionally used for this purpose. If estimates of the measurement error exist, these information can be utilized to define the parameters of the error distribution. Otherwise, heuristic rules can be applied.
Hennig (2007) defined such a resampling scheme as follows. 1) For all p dimensions of the original sample data $X = (x_1,\ldots, x_N)$, compute the N–1 differences d_{ij} between neighboring data values in dimension p: for $i = 1,\ldots, N$–1, $j = 1,\ldots, p$, d_{ij} is the difference between the (i+1)-th and the i-th order statistic of the j-th dimension. For $j = 1,\ldots, p$, let q_j be the empirical quantile of the d_{ij}, where q is a tuning constant. 2) Draw noise e_n, $n = 1,\ldots, N$, independent and identically distributed from a normal distribution with a zero mean and a diagonal matrix as covariance matrix with diagonal elements $\sigma_1^2 = q_1^2,\ldots, \sigma_p^2 = q_p^2$ and compute the resample points $y_n = x_n + e_n$ for $n = 1,\ldots, N$. (For an example see section 4).

3.3 Combination of bootstrapping and jittering

When using the (non-parametric) bootstrapping scheme, about one third of the resample points will be identical replicates of original sample points. Each group of such identical points could be seen as a mini-cluster. The occurrence of these artificial clusters, generated by a statistical analysis tool, may induce inappropriate models of the true data structure. In particular, when clustering the resample data, the artificially replicated data points may be misinterpreted as true clusters (Monti et al., 2003). Moreover, for some implementations of clustering and multidimensional scaling methods the identical bootstrap replicates may cause numerical problems. Hennig (2007) proposed the combination of bootstrapping and jittering as a way to avoid or to reduce these problems.

3.4 Perturbation

Data sets for applications of statistical machine learning are usually generated with a precision that is high enough to measure intra-population variability. Therefore, any data point of a sample is likely to be different from any data point of another (disjoint) sample – even if the measurement error was zero. This type of inter-sample differences is not realistically simulated when using the above non-parametric methods. (*Sampling from a sample* provides highly overlapping data sets that all consist of random selections from the

same set of original points, while *jittering* leads to data sets that simulate differences comparable to those caused by measurement errors.) Another resampling strategy may be desired for a better (non-parametric) simulation of inter-sample differences due to intra-population variability. Estimates of intra-population variability that could be used for such a simulation are usually unavailable prior to cluster analysis. Under these circumstances, a simple simulation is the addition of random values onto the data. Here this approach is called 'perturbation'.

Let $X \in \Re^{N \times p}$ be the original p-dimensional sample consisting of N data points. Then for r = 1, 2, …, resample r is obtained as follows. $Y_r = X + \xi_r$, where $\xi_r \in \Re^{N \times p}$ is a sample of size N from a p-dimensional distribution. The parameters of this distribution, such as variance, can be specified based on an estimate obtained from the original sample. For example, the random variable ξ may be selected to have a normal distribution with zero mean vector and $c \cdot \sigma \in \Re^p$, where c denotes a constant, $\sigma = (\sigma_1, \ldots, \sigma_p)$ is an empirical estimate of the variability of the data. Bittner et al. (1999), chose c = 0.15 and σ being the median standard deviation of the entire sample. Möller and Radke (2006a) used several values of c equal to 0.01, 0.05 and 0.1, where σ represented the standard deviation from the grand mean of the data.

Perturbation and *jittering* are conceptually similar resampling techniques. However, their implementation may differ quantitatively in the values of statistical parameters used to simulate intra-population variability and measurement error based on external knowledge, estimates or assumptions.

3.5 Nearest neighbor resampling

The *perturbation* technique has two shortcomings in a cluster validation study. First, the method will induce inappropriate inter-resample differences if the true intra-population variability *differs* between several populations of the mixture population. The reason is that the random values used to perturb every data point are drawn from the *same* distribution. Therefore, the data points originally drawn from some populations are perturbed too strongly or too weakly or both types of error may occur simultaneously. Second, even if the intra-population variability is constant across all populations within the mixture, it is difficult to adjust the parameter(s) of the distribution used for drawing the random values. An overestimation of the proper perturbation strength would have the consequence that true data structures which are present in the original sample may not be retained in any resample. Otherwise, an underestimation of the perturbation strength would lead to very similar resamples and spurious, high cluster stability. To avoid false interpretations of a perturbation-based clustering study, it may be appropriate to repeat the analysis with different values of the perturbation strength (e.g., Möller & Radke, 2006a). A non-parametric resampling approach where the choice of the perturbation strength is less critical is nearest neighbor resampling (NNR).

The idea behind NNR can be explained as follows. A high intra-population variability is characterized by a wide distribution and a low probability of drawing a point from the respective part of the hyperspace. Accordingly, the distances between sample points in this part of the hyperspace are high. For low intra-population variability the opposite is true. Clearly, if two or more populations of a mixture population have overlapping distributions, the total probability is increased and sample points will have decreased inter-point distances compared to those obtained from any single population. The relationship between population variability and inter-point distances can be utilized to simulate random samples,

where the advantages of a perturbation approach are utilized and knowledge, estimates or assumptions about the distributions of existing populations are not required.

Here we consider the following strategy for NNR. 1) For each original sample point x_n, n = 1,…, N, an estimate of the inter-point distances in the neighborhood of x_n is obtained. This neighborhood is defined by the k nearest neighbors of x_n according to a user-selected metric. 2) The direction vector for the perturbation of x_{nr} with respect to x_n is selected. 3) Resample point x_{nr} is generated by adding a random vector to x_n with the direction as selected in step two and the vector length being a function of the estimated inter-point distances in the neighborhood of x_n. The rationale underlying the choice of a k-NN approach is the same as in supervised learning. Most of the k nearest neighbors of data point x_n are assumed to belong to the same class (population) as x_n. Therefore, the neighboring points of x_n are assumed to provide an estimate of intra-population variability. Below, two versions of NNR are described.

Nearest neighbor resampling 1 (NNR1). (Möller & Radke, 2006b)

0. Let $X = (x_1,\ldots, x_N) \subset \Re^p$ be the original sample. Chose $k \geq 2$ and a metric for calculating the distance between elements of X.
1. For each sample point x_n, n = 1,…, N, determine Y_n, that is, the set containing x_n and its k nearest neighbors. Calculate d_n, the mean of the distances between each member and the center (mean) of the set Y_n.
2. For each resample r, r = 1, 2, …, and each sample point x_n, n = 1,…, N, perform the following steps.
3. Chose a random direction vector ξ_{nr} in the p-dimensional data space (i.e., ξ_{nr} is a p-dimensional random variable uniformly distributed over the hyper-rectangle $[-1, 1]^p$).
4. Rescale the direction vector ξ_{nr} to have the vector length equal to d_n (calculated in step 1).
5. Generate point n of resample r: $x_{nr} = x_n + \xi_{nr}$.

The fixed point-wise perturbation strength (d_n) has been selected to ensure an effective perturbation of each sample point (i.e., to avoid spurious high cluster stability). The method NNR1 can be used to simulate random samples from an unknown mixture population with different intra-population variability and a diagonal matrix as covariance matrix of each population. However, the latter assumption may be too strong for a number of real data sets. For example, the NNR1 method may simulate resample clusters with a hyper-globular shape also in cases where the corresponding clusters in the original sample have a hyper-ellipsoidal shape. (This is a consequence of the fixed perturbation strength in conjunction with the uniformly distributed direction vector.)

Therefore, the user should have other choices for calculating the amount and direction of the perturbation. Experiments have shown that the unintentional generation of artificial outliers by the resampling method may prevent reasonable clustering results of the resamples, while the original sample may have been clustered appropriately. For example, in some cases the fuzzy C-means (FCM) clustering algorithm provided 'missing clusters' for NNR1-type resamples, but not for the original sample (data not shown). Missing clusters were introduced in (Möller, 2007) as being inappropriate clustering results of the FCM. As a conclusion, another method, NNR2, was developed for the analysis of high-dimensional data sets. In NNR2, a data point can be 'shifted' only towards and not beyond one of its nearest neighbors (i.e., into a region of the feature space that actually contains some data).

Furthermore, the mean-based estimate d_n in step 1 of the NNR1 method could be biased if the neighbors of x_n contain outliers or if they contain data points which have been drawn from a population different than the one from which x_n has been drawn. This source of bias can be reduced or avoided by using a robust estimate of the typical inter-point distance such as the median.

Nearest neighbor resampling 2 (NNR2).

0. Let $X = (x_1,\ldots, x_N) \subset \Re^p$ be the original sample. Chose $k \geq 2$, two constants $c_1 \geq 0$ and $c_2 > c_1$ for a data-specific calibration of the perturbation strength and a metric for calculating the distance between elements of X.
1. For each sample point x_n, $n = 1,\ldots, N$, determine the k nearest neighbors of x_n and calculate d_n, the median distance between all pairs of these k neighbors.
2. For each resample r, $r = 1, 2, \ldots$, and each sample point x_n, $n = 1,\ldots, N$, perform the following steps.
3. Chose one of the k nearest neighbors of x_n at random. This data point is denoted by x_m. The direction vector from x_n to x_m is used as the direction vector ξ_{nr} for the perturbation of the sample point x_n to generate the resample point x_{nr}.
4. Draw the value c_{nr} from the uniform distribution over the interval $[c_1, c_2]$. Calculate the distance d_{nm} between the sample points x_n and x_m. If d_{nm} is larger than d_n, set the amount of perturbation $|\xi_{nr}| = c_{nr} \cdot d_n$, otherwise set $|\xi_{nr}| = c_{nr} \cdot d_{nm}$, where $|.|$ denotes the vector length. Briefly, $|\xi_{nr}| = c_{nr} \cdot \min(d_n, d_{nm})$.
5. Generate point n of resample r: $x_{nr} = x_n + \xi_{nr}$.

The NNR2 method restricts the positions of simulated (resample) points to the set of points that lie on the lines interconnecting an original sample point and its k nearest neighbors. Real samples are not constrained in this way. However, the application of this constraint leads to the simulation of resample points that cover only those regions of the feature space which are actually occupied by observed data. NNR2 has two advantages in cluster validation studies. Artificial outliers and resulting biases of resample clusterings can be largely avoided. More importantly, there may be data structures which are recognized from a clustering of the original sample, but are no longer separable after a perturbation like that in section 3.4 or that induced by the NNR1 method. The constrained perturbation by the NNR2 method is likely to simulate samples in which such (weakly separable) structures are preserved.

NNR2-type perturbation can be calibrated by adjusting the parameters k, c_1 and c_2. A higher maximal perturbation strength is achieved by increasing the values of k and/or c_2. When choosing $c_2 = 1$ the maximum amount of perturbation for each point equals the median distance between the k nearest neighbors of the respective point. The minimum amount of perturbation of each point can be adjusted by choosing $c_1 > 0$.

3.6 Outlier simulation

Real data sets may contain outliers – even though the data has been processed by a method for the detection and removal of outliers. Therefore, it is desirable to know how robust the result of a clustering algorithm is with respect to the presence of outliers. This knowledge can then be used to select a robust result among a number of candidate results obtained by different clustering algorithms or the same algorithm with different settings of a control parameter (especially, the number of clusters).

For the investigation of cluster stability with respect to outliers Hennig (2007) proposed the replacement of a subset of data points by noise, where "noise points should be allowed to lie far away from the bulk (or bulks) of the data, but it may also be interesting to have noise points in between the clusters, possibly weakening their separation". The author cited Donoho's and Huber's concept of the finite sample replacement breakdown point as a related methodological basis.

Replacing points by noise. Choose M, the number of data points to be replaced by noise, where $1 \leq M < N$ with N being the size of the original sample X. Select a noise distribution and replace M elements of X by points drawn from the noise distribution. For example, the uniform distribution on a hyperrectangle $[-c, c]_p \subset \Re^p$, $C > 1$, may be used, where X had been transformed before the replacement to have a zero mean vector and the identity matrix as covariance matrix.

Addition of noise points. The replacement of original points by noise causes a loss of information which may impair the modeling of the data structure based on a resample clustering. Therefore, an alternative method is proposed here. The M points drawn from the noise distribution could also be added to the data set (i.e., without eliminating any original point). The artificial increase of the resample size in comparison to the original sample size may be less problematic for the purpose of cluster validation than it could be for other resampling applications. It is also possible to find a balance between the artificial increase of the resample size and the information loss: M_R points are replaced, while M_A points are added, where $M_R + M_A = M$. Reasonable choices for M_R and M_A may have to be sought experimentally by the user.

3.7 Feature resampling

Data randomization schemes can also be applied to the set of features used to characterize the population. Such methods will be subsumed below under the term 'feature resampling'. Two of the subsequently described methods (feature subsampling and leave one feature out) leave the information about one or more features unused when generating a resample. These methods may be useful if the number of features p is larger than the number of data points N, where the N points in the p-dimensional coordinate system actually span a data space with less than p (i.e., at most N–1) dimensions. An example is the clustering of biological tissues based on gene expression data, where often $40 \leq N \leq 300$ and $p \geq 1000$ (cf. Monti et al., 2003). In such cases the clustering may become more effective (because redundant information are eliminated) and the computational effort of the clustering would decrease (owing to the dimension reduction).

Feature subsampling. For $r = 1, 2, \ldots$, select a subset of s_r features randomly from the entire set of p features ($1 \leq s_r < p$). Resample r is obtained by extracting the data of the original sample for the selected features only. The value of s_r can be fixed for generating all resamples (e.g., Smolkin & Gosh, 2003). Alternatively, s_r can be a random variable. Yu et al. (2007) defined the value of s_r to be uniformly distributed over the integer range between $0.75p$ and $0.85p$.

Feature multiscale bootstrapping. There exists a version of bootstrapping which is similar to feature subsampling with variable subsampling size. In this method bootstrap resamples of a variable size $M \leq N$ are drawn from the original sample. This method has been applied to the set of features (gene expression values) when clustering tumor samples (Suzuki & Shimodaira, 2004). An implementation of the method is available in the free statistical software R (Suzuki & Shimodaira, 2006).

Leave one feature out. Generate a set of $i = 1,\ldots, p$ resamples, where p is the number of all features. Resample i contains the original data of all features except feature i. If the number of features is large, the p resamples are relatively similar. Accordingly, a resample clustering is likely to generate p similar partitions and a cluster stability assessment of these partitions may not be informative. A cluster validation approach developed for 'leave one feature out' resamples is the 'figure of merit' (FOM), motivated by Efron's jackknife approach. The FOM quantifies how well the data clustering based on all features except feature i can predict the clustering based on only the data of feature i. For the details see (Yeung et al., 2001).

Feature mapping. Several methods exist for the mapping of a data set into a lower-dimensional space. Among these methods randomized maps suggest themselves for the application to resampling-based cluster validation due to their attractive properties. First, these projections generate random variations of the input data, where the strength of variation can be adjusted almost arbitrarily. Second, some characteristics of the data in the original space, such as the distances between points, are approximately preserved in the projected space (i.e., metric distortions are bounded according to the Johnson-Lindenstrauss theory). Third, the number of dimensions of the projected space can be slightly or considerably smaller than the number of dimensions of the original space. The dimensionality of the projected subspace in which a limited distortion can be obtained depends only on the cardinality of the data and the magnitude of the admissible distortion. For details see (Bertoni & Valentini, 2006). For potential users an implementation of some of these methods is available in the free statistical software R (Valentini, 2006).

Feature weighting. The features to be included into a resample data set can also be randomly weighted. When using continuous positive weights, the information of every feature is included at a certain degree. The lognormal distribution with the mean $\mu = -\log 2$ and the variance $\sigma^2 = 2*\log 2$ can be used for the drawing of the weights. The method can be interpreted as an alternative approach to bootstrapping. The use of the lognormal distribution can be motivated based on relationships of this distribution with the Poisson distribution and the binomial distribution, where the latter is the underlying distribution of a drawing with replacement. The authors of this method (Gana Dresen et al., 2008) called their approach *resampling based on continuous weights*.

4. Results of benchmarking studies

The performance of the above resampling methods is not easily predicted based on a theoretical analysis. Therefore, empirical comparisons of different methods provide useful information for the selection of a method in future applications. This section is a summary of main results reported in five studies which included benchmarking tests of different resampling schemes in a clustering context. In the next section these results will be discussed aiming at general suggestions for the use and choice of resampling methods applied to cluster validation.

In the sequel, the term *bootstrapping* always refers to its non-parametric version. The bootstrap scheme (drawing with replacement) was always applied to the full original sample. To keep the reported information concise the following symbols will be used.

Symbols / abbreviations

N number of observations (data points) in an original sample
p number of dimensions (i.e., features used to describe the members of a population)
R number of resamples generated by using one of the resampling schemes

S subsampling size (percentage of data randomly drawn from the original data sample)
K_R number of clusters generated when clustering each resample data set
K number of clusters of a consensus partition obtained from the set of resample partitions
K_t true (known) number of classes (populations) represented by a benchmarking data set

Minaei-Bidgoli et al. (2004) compared bootstrapping and subsampling for five benchmarking data sets with $N >> p$. The number of resamples R varied from 5 to 1000 and $S \in [5\%, 75\%]$. All resample partitions were obtained by using the K-means clustering algorithm. Resampling performance was measured based on the misassignment (error) obtained for the clustering partitions in comparison to the a priori known class structure of benchmark data sets. The error rate was always calculated for a partition representing the consensus of the R resample partitions. Four different methods from the literature were used providing four consensus partitions in each case. While the generation of resample partitions was repeated for different pre-specified values of the number of clusters ($K_R \in [2, 20]$, $K_R > K_t$), each consensus partitions was calculated to have exactly the true number of clusters ($K = K_t$). The error was calculated after finding the optimal assignment between the obtained consensus clusters and the known classes. All experiments were repeated at least 10 times and average errors were reported for some of the best parameter settings of the entire procedure (resampling, resample clustering and consensus clustering).

The error rates obtained for bootstrapping and subsampling were similar. Because the results for subsampling were based on only 5 to 75% of the data sets (parameter S), the authors considered subsampling as a flexible method that can be used to reduce the computational cost in many data mining tasks.

Möller & Radke (2006) compared bootstrapping, subsampling (S = 80%) and perturbation (with three values of the perturbation strength, see section 3.4). R = 20 was fixed in all experiments. Resampling performance was measured based on the rate of false estimates of the number of clusters obtained for the set of the R resample partitions. For each data set 458 estimates of the number of clusters were obtained, resulting from the application of 12 clustering techniques and 41 cluster validity indices. The clustering methods included different hierarchical agglomeration schemes and different metrics, a so-called K-medoid clustering and two versions of fuzzy C-means clustering. Only those of the 458 results were used for the final interpretation where the correct (a priori known) number of clusters was obtained for the original sample as well as for the majority of the resamples. (These constraints were used to exclude errors due to poor original sampling, poor cluster analysis and/or poor configuration of the resampling scheme.) The following data were analyzed: five realizations of each of the stochastic models 2, 3, 4, 6 and 7 described in (Dudoit and Fridlyand, 2003), three microarray data sets with the 200 most differentially expressed genes (*Leukemia*, *CNS* and *Novartis* data described in Monti et al., 2003), the data sets *Iris*, *Liver*, *Thyroid* and *Wine* from the UCI repository (Asuncion & Newman, 2007), and a data set of functional magnetic resonance imaging data. Data sets with $N >> p$ as well as $N << p$ were included.

In general, the error rates obtained for the perturbation technique were smaller than the error rates for subsampling. Both perturbation and subsampling led to clearly smaller error rates than bootstrapping. The same ranking was obtained when considering all (about 15.000) estimates of the number of clusters without applying the mentioned constraints. The occurrence of false estimates even for a perturbation with 1% noise indicated that the small errors obtained for the perturbation scheme are not spurious results (i.e., the perturbation was effective). The authors concluded that the increased errors for subsampling and bootstrapping may have been a consequence of the information loss (i.e., 20% and about

37% of the original sample were not used for the generation of a resample in the subsampling and bootstrapping schemes, respectively). The authors further concluded that resampling schemes without this information loss are more useful in cluster validation studies, in particular, when the original samples have a small size.

Hennig (2007) compared bootstrapping, subsampling (S = 50%), the replacement of sample points by noise (M = 0.05N, c = 3 and M = 0.2N, c = 4, see section 3.6), two versions of jittering (parameter q was set respectively to the 0.1- and 0.25-quantiles of the values d_{ij}, see section 3.2), and the combination of bootstrapping and jittering (q = 0.1). R = 50 was fixed in all experiments. Resampling performance was measured based on several types of results. First, cluster stability was assessed by calculating the agreement between the partition generated from each resample and the partition obtained for the original sample (The agreement between clusters of two partitions was measured by the Jaccard index (cf. Theodoridis & Koutroumbas, 2006).) Second, for model data with true cluster memberships, it was measured how well the clustering of an original sample represented the model structure. (The Jaccard index was applied to the cluster memberships of each original sample and the true cluster memberships.) Third, the correlation between the two aforementioned types of results was calculated. Different clustering methods were used, namely, a method called normal mixture plus noise, K-means, 10% trimmed K-means and average linkage hierarchical agglomeration. 50 original samples were generated for each of two stochastic models (K_t = {3, 6}, N >> p). One model included outliers. One biological data set (N = 366, p = 306) was analyzed that was known to contain substructure – without exact knowledge about the 'true' cluster composition.

Due to the choice of the analysis design, three types of results were distinguished. 1) partitions of original samples with a fairly good representation of the model structure and a stable clustering of the resample data that corresponded to this model structure, 2) partitions of original samples with a relatively poor representation of the model structure and an unstable clustering of the resample data and 3) partitions of original samples with a relatively poor representation of the model structure and, nevertheless, a stable clustering of the resample data. The results of the types 1 and 2 are desirable, because they permit appropriate conclusions about the performance of clustering of unknown data based on resample cluster stability scores. Results of type 3 are problematic. If the original sample does not adequately represent the true population structure, also the clustering of this sample may not represent the true structure. Even though it is desirable to obtain an indication of the poor modeling result, namely, an *unstable* clustering for the resample data. Otherwise, this kind of inappropriate modeling cannot be distinguished from proper clustering models when the true population is unknown.

Based on all results, subsampling was considered as being the best method, followed by the combination of bootstrapping/jittering and bootstrapping alone. The replacement of data points by noise was also useful in a number of case, including some cases where the other methods did not perform well (i.e., they provided a number of type-3 results). Jittering showed generally a poor performance (i.e., a relatively large fraction of type-3 results for most of the data sets and clustering algorithms). The author concluded that a good strategy in practice can be the use of *one* of the schemes bootstrapping, bootstrapping/jittering and subsampling together with *one* scheme for replacing data by noise.

Gana Dresen et al. (2008) compared bootstrapping and feature weighting. R = 1000 was fixed in all experiments. Resampling performance was measured based on the stability of branches of cluster trees (dendrograms) obtained from hierarchical agglomerative clustering of the resample data sets. Furthermore, a majority consensus tree was generated from the resample

cluster trees and this consensus tree was compared with the cluster tree obtained from the original sample (based on the Rand index; cf. (Theodoridis & Koutroumbas, 2006)). For the comparison, gene expression data from 24 chromosomes (p = 8 to 648 probe sets) of N = 20 tumor patients were used. For a subset of the data, knowledge about actual clustering structure was available. A data set containing p = 7 features of N = 22 primates was also analyzed. In addition, it was investigated how well groups of simulated differentially expressed genes can be robustly detected based on bootstrapping and feature weighting.
In a number of cases bootstrapping and feature weighting showed comparable performance. However, in several cases bootstrapping led to inappropriate consensus cluster trees. That is, the structure was inappropriate, many spurious singleton clusters were obtained and especially the false clusters proved to be stable under the bootstrap procedure. The authors concluded that resampling with continuous weights is strongly recommended because it performed at least as well as bootstrapping and in some cases it surpassed bootstrapping. In particular, feature weighting was more appropriate than bootstrapping to cluster small size samples.
Möller and Radke (2006b) reported results of estimating the number of clusters based on two different approaches, denoted here by A and B. In approach A (Monti et al., 2003) resampling is performed by subsampling (S = 80%). In approach B (Möller & Radke, 2006b) nearest neighbor resampling (NNR1) was used. Approach B led to better results than A on high-dimensional gene expression benchmark data ($N << p$). In particular, a fairly good recovery of known tumor classes was possible based on just R = 10 nearest neighbor resamples in approach B, while approach A led to similar or worse results based on R = 200 or R = 500 subsamples (with R depending on the clustering algorithm). These results indicated the usefulness of nearest neighbor resampling; however, the performance differences may partly be attributable to the different methods selected in the approaches A and B, respectively, for clustering and for estimating the number of clusters.

5. Results of nearest neighbor resampling

Results of a direct benchmarking of NNR and other resampling methods are currently not available. However, several cluster validation results based on NNR have been obtained. Ulbrich (2006) used the NNR1 algorithm to identify robust and prognostic gene expression patterns by clustering of tumor patients. Guthke et al. (2007) performed clustering to find co-expression patterns of genes for the subsequent utilization in systems biology. They showed that the NNR1-based cluster stability analysis can be used to complement and confirm the results of a different quality assessment, namely the vote of so-called cluster validity indices (Bezdek and Pal, 1998).
The use of the NNR2 method has provided strong indications that (estrogen receptor positive) breast cancer can be robustly subdivided into three, perhaps four, classes which are represented by different prognostic gene expression profiles. This result has been consistently obtained for gene expression data and survival time data generated in four different studies based on two different DNA microarray platforms and including the data from more than 700 tumor patients (Iffert, 2007).
In combination with methods presented by Fred and Jain (2006), the NNR2 algorithm was recently applied to the gene expression benchmark data sets of known tumor classes published by Monti et al. (2003). In several cases the obtained class recovery scores were higher than those obtained by Monti et al. based on subsampling and those obtained by Yu et al. (2007) who analyzed the same data based on feature subsampling (Möller, 2008). However, the cluster analysis methods used in these studies were also different.

6. Discussion and conclusions

Bootstrapping (drawing with replacement) is perhaps the most widely known and recommended resampling approach, because it is a standard approach for statistical inference methods (Efron & Tibshirani, 1993). If the sample size is large and the true distribution is well represented by the data, bootstrapping may also be useful for the validation of clustering results. That is, other resampling schemes may not lead to more accurate results (cf. Minaei-Bidgoli et al., 2004). Under these circumstances the user may prefer bootstrapping, because no control parameter has to be set.

However, as shown in complementary investigations (section 4), for statistical cluster validation it is recommended to prefer other methods than bootstrapping. When the sample size is large, subsampling is likely to perform as well as bootstrapping (Minaei-Bidgoli et al., 2004; Hennig, 2007) or even better (Möller & Radke, 2006a), where the clustering of subsamples requires a lower computing effort. If the clustering result is to be used as the basis for a classifier of unknown samples, the subdivision scheme (e.g., Dudoit & Fridlyand, 2003) may be the best choice, because it is focused on minimizing the prediction error, while subsampling results are commonly used for assessing cluster stability (e.g., Tseng and Wong, 2005; Fred & Jain, 2006). When the sample size was small, perturbation and resampling with continuous weights have been shown to outperform bootstrapping (Möller & Radke, 2006a; Gana Dresen et al., 2008).

If the sample size is small, a further decrease by drawing subsamples prevents the "learning" of a good model from the resample data. In this case, perturbation methods are more suitable than *sampling from a sample* (Möller & Radke, 2006a). However, the user should be aware that this type of perturbation works best only if all populations of the hypothesized mixture population have equal variability. Furthermore, this method requires an estimate or guess of the proper perturbation strength. Therefore, it may be recommended to search for stable clusters by using different values of the perturbation strength. This could increase the confidence in the validity of the obtained clusters and their completeness with respect to the true structures.

Nearest neighbor resampling (NNR) is an attractive alternative to the perturbation described in section 3.4. In the absence of prior knowledge, the parameter setting for the NNR2 method is less critical than the specification of a global perturbation strength. According to the author's knowledge, the NNR methods were described here for the first time in detail. Especially, the NNR2 method has provided promising results when clustering data with complex structures (see section 5). Therefore, based on practical experience, the author recommends the NNR approach for applications of unsupervised machine learning. Even though, more comprehensive simulations and benchmarking studies with other methods are desired to know the performance of the NNR approach in a more general context.

Feature resampling may be a way to bypass some of the problems associated with the above resampling schemes. However, the successful use of some of these techniques is limited to applications where the assumptions underlying these techniques are fulfilled. This argument applies, for example, to *feature subsampling* and *leave one feature out* which involve a loss of original information (cf. Yeung et al., 2001). *Feature mapping* (Bertoni & Valentini, 2006) appears to be a promising approach due to the combination of dimension reduction and the distance–preserving character of the mapping. It would be interesting to have empirical results indicating the relative merits of this kind of mapping in comparison to several other methods presented above. Another promising method is *resampling with*

continuous weights (Gana Dresen et al., 2008). As stated by the authors it would be interesting to investigate the performance of this method in combination with other clustering algorithms than the hierarchical ones used.

The resampling methods for the simulation of measurement errors (*jittering*) and outliers are useful if the user wants to confirm the robustness of the final clustering result with respect to these factors of influence. However, robust results of such an analysis are only a pre-condition for a good clustering model. The fact that clusters are stable under jittering and the insertion of artificial outliers must not be interpreted over-optimistically as the indication of a real mixture population.

Hennig (2007) argued that "Generally, large stability values do not necessarily indicate valid clusters, but small stability values are informative. Either they correspond to meaningless clusters (in terms of the true underlying models), or they indicate inherent instabilities in clusters or clustering methods." Following this view, any stable cluster and any good prediction based on the *subdivision* approach (section 3.1) may have to be verified by repeating the cluster analysis with an increasing amount of (random) change made to the data. One criterion for stopping these repetitions is that some clusters 'disappear' under the influence of resampling, while other clusters can still be recovered. This observation would not be expected in the absence of any true structure. Another termination criterion is fulfilled if the clustering structures 'disappear' only if the amount of random change has become clearly larger compared to the effect of the measurement error. This fact may be deducible even if the measurement error can only be roughly estimated.

An inevitable decision that has to be made by the user is the selection of the number of resamples, R. A proper value of R depends on both the structure of the investigated data and the resampling method used. In fact, compact and well separated clusters would be robustly detected based on fewer resamples than overlapping, noisy clusters. In addition, the more original sample information is utilized for generating each resample, the fewer resamples are likely to be required. For example, $R = 10,..., 30$ resamples obtained from NNR methods have been sufficient to robustly recover clustering structures of small high-dimensional samples (Ulbrich, 2006; Iffert, 2007; Möller & Radke, 2006b). In contrast, $R = 100,\ldots, 1000$ resamples have often been used for the cluster validation based on bootstrapping or subsampling (cf. section 4). If the information loss of the mapping from the original sample to the resample exceeds a data specific-threshold, the lack of information in the individual resamples may not be compensable by any increase in the number of resamples.

Computerized observation techniques in an increasing number of research areas generate high-dimensional data (e.g., DNA microarray data, spectral data with a high frequency resolution and complex image and video data). High-dimensional data sets are more likely than others to provide clusterings which are not significant and meaningful. Especially in those cases, but also when clustering any other sample data, the use of resampling methods is recommend as a valuable aid for a statistical model quality assessment.

The above description and review of resampling schemes and their performance as well as the presentation of a new approach (NNR) may help users to select an appropriate method in future studies.

8. Acknowledgement

This presentation was supported by the HKI Jena and the International Leibniz Research School for Microbial and Biomolecular Interactions (ILRS) Jena (http://www.ilrs. hki-jena.de).

9. References

Asuncion, A. & Newman, D.J. (2007). UCI Machine Learning Repository [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, School of Information and Computer Science.

Ayad, H.G. & Kamel, M.S. (2008). Cumulative voting consensus method for partitions with variable number of clusters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 30, No. 1, 160-173

Bertoni, A. & Valentini, G. (2006). Randomized maps for assessing the reliability of patients clusters in DNA microarray data analyses. *Artificial Intelligence in Medicine*, Vol. 37, 85-109

Bezdek, J.C. & Pal, N.R. (1998). Some new indexes of cluster validity, *IEEE Transactions on Systems, Man and Cybernetics – Part B*, Vol. 28, 301-315

Bittner, M. & 27 co-authors (2000). Molecular classification of cutaneous malignant melanoma by gene expression profiling. *Nature*, Vol. 406, 536-540

Borgelt, C. & Kruse, R. Finding the number of fuzzy clusters by resampling. *Proceedings of the IEEE Int. Conf. on Fuzzy Systems*, pp. 48-54, ISBN: 0-7803-9488-7, Vancouver, Canada, Sept. 2006, IEEE Press, Piscataway, NJ, USA

Dudoit, S. & Fridlyand J. (2003). A prediction-based resampling method for estimating the number of clusters in a dataset. *Genome Biology* 3:7, Online ISSN 1465-6914

Efron, B. & Tibshirani, R.J. (1993). *An introduction to the bootstrap.* Chapman & Hall/CRC, ISBN 0-412-04231-2

Fred, A. & Jain, A.K. (2006). Learning pairwise similarity. *Proceedings of the Int. Conf. on Pattern Recognition (ICPR)*, pp. 892-895, ISBN 0-7695-2521-0, Hong-Kong, August 2006, IEEE Computer Society Press

Gana Dresen, I.M.; Boes, T.; Huesing, J.; Neuhaeuser, M. & Joeckel, K.-H. (2008). New resampling method for evaluating stability of clusters. *BMC Bioinformatics*, 9:42, doi:10.1186/1471-2105-9-42

Guthke, R.; Kniemeyer, O.; Albrecht, D.; Brakhage, A.A. & Möller, U. (2007). Discovery of Gene Regulatory Networks in Aspergillus fumigatus, In: *Knowledge discovery and emergent complexity in bioinformatics,* Tuyls, K. et al. (Ed.), *Lecture Notes in Bioinformatics 4366*, 22-41, Springer, ISBN 978-3-540-71036-3, Berlin/Heidelberg

Hastie, T.; Tibshirani, R. & Friedman, J. (2001). *The Elements of Statistical Learning*. Springer, ISBN 978-0-387-95284-0

Hennig, C. (2007). Cluster-wise assessment of cluster stability. *Computational Statistics and Data Analysis* Vol. 52, 258-271

Iffert, W. (2007). Investigations for the prognosis of diseases by simultaneous analysis of gene expression data and survival time data. (in German), Diploma Thesis in Bioinformatics, August 2007, Friedrich Schiller University, Jena, Germany

Kell, D.B. & Oliver, S.G. (2004). Here is the evidence, now what is the hypothesis? The complementary roles of inductive and hypothesis-driven science in the post-genomic era. *BioEssays,* Vol. 26, 99-105

Lunneborg, C.E. (2000). *Data Analysis by Resampling. Concepts and Applications*, Duxbury Press, ISBN 0-534-22110-6, Pacific Grove, CA, USA

McLachlan, G.J. & Khan, N. (2004). On a resampling approach for tests on the number of clusters with mixture model-based clustering of tissue samples. *Journal of Multivariate Analysis*, Vol. 90, 90-105

Minaei-Bidgoli, B.; Topchy, A. & Punch, W.F. (2004). A comparison of resampling methods for clustering ensembles. *Proceedings of the International Conference on Machine Learning; Models, Technologies and Applications (MLMTA)*, pp. 939-945, Las Vegas, Nevada, June 2004

Möller, U. & Radke, D. (2006a). Performance of data resampling methods for robust class discovery based on clustering. *Intelligent Data Analysis* Vol. 10, No. 2, 139-162

Möller, U. & Radke, D. (2006b). A cluster validity approach based on nearest neighbor resampling, *Proceedings of the Int. Conf. on Pattern Recognition (ICPR)*, pp. 892-895, ISBN 0-7695-2521-0, Hong-Kong, August 2006, IEEE Computer Society Press

Möller, U. (2007). Missing clusters indicate poor estimates or guesses of a proper fuzzy exponent. In: *Applications of Fuzzy Sets Theory*, Masulli, F.; Mitra, S.; Pasi, G. (Ed.), *Lecture Notes in Artificial Intelligence 4578,* 161-169, Springer, ISBN 978-3-540-73399-7, Berlin-Heidelberg

Möller, U. (2008). Methods for robust class discovery in gene expression profiles of tissue samples. Poster presentation at the conference *Bioinformatics Research and Development (BIRD)*, July 2008, Vienna, Austria

Monti, S.; Tamayo, P.; Mesirov, J. & Golub, T. (2003). Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. *Machine Learning*, Vol. 52, 91–118

Smolkin, M. & Ghosh, D. (2003). Cluster stability scores for microarray data in cancer studies. *BMC Bioinformatics*, 4:36, www.biomedcentral.com/1471-2105/4/36

Strehl, A. & Gosh, J. (2002). Cluster ensembles: A knowledge reuse framework for combining multiple partitions, *J. of Machine Learning Research,* Vol. 3, 583–617

Suzuki, R. & Shimodaira, H. (2004). An application of multiscale bootstrap resampling to hierarchical clustering of microarray data: How accurate are these clusters? *Proceedings of the Int. Conf. on Genome Informatics (GIW),* p. P034

Suzuki, R. & Shimodaira, H. (2006). Pvclust: an R package for assessing the uncertainty in hierarchical clustering. *Bioinformatics*, Vol. 22, No. 12, 1540-1542

Theodoridis S. & Koutroumbas, K. (2006). *Pattern recognition.* 3rd ed., Academic Press, ISBN 0-12-369531-7, San Diego

Topchy, A.; Jain, A.K. & Punch, W. (2005). Clustering ensembles: models of consensus and weak partitions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 12, 1866-1881

Tseng, G.C. & Wong, W.H. (2005). Tight clustering: a resampling-based approach for identifying stable and tight patterns in data. *Biometrics*, Vol. 61, 10-16

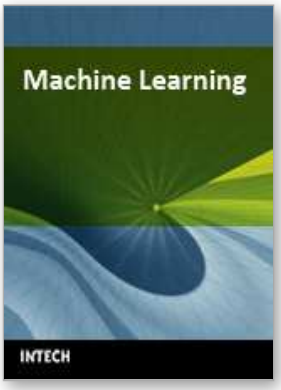
Ulbrich, B. (2006). Improvements of tumor classification based on molecular-biological patterns by using new methods of unsupervised learning. (in German), Diploma Thesis in Bioinformatics, August 2007, Friedrich Schiller University, Jena, Germany

Valentini, G. (2006). Clusterv: a tool for assessing the reliability of clusters discovered in DNA microarray data. *Bioinformatics*, Vol. 22 No. 3, 369-370

Yeung, K.Y.; Haynor, D.R. & Ruzzo, W.L. (2001). Validating clustering for gene expression data. *Bioinformatics*, Vol. 17, No. 4, 309-318

Yu, Z.; Wong, H.-S. & Wang, H. (2007). Graph-based consensus clustering for class discovery from gene expression data. *Bioinformatics*, Vol. 23, No. 21, 2888-2896

Yu, C.H. (2003). Resampling methods: concepts, applications, and justification. Practical Assessment, Research & Evaluation, 8(19). Retrieved January 6, 2005 from http://PAREonline.net/getvn.asp?v=8&n=19



Machine Learning

Edited by Abdelhamid Mellouk and Abdennacer Chebira

ISBN 978-953-7619-56-1
Hard cover, 450 pages
Publisher InTech
Published online 01, January, 2009
Published in print edition January, 2009

Machine Learning can be defined in various ways related to a scientific domain concerned with the design and development of theoretical and implementation tools that allow building systems with some Human Like intelligent behavior. Machine learning addresses more specifically the ability to improve automatically through experience.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Ulrich Möller (2009). Resampling Methods for Unsupervised Learning from Sample Data, Machine Learning, Abdelhamid Mellouk and Abdennacer Chebira (Ed.), ISBN: 978-953-7619-56-1, InTech, Available from: http://www.intechopen.com/books/machine_learning/resampling_methods_for_unsupervised_learning_from_sample_data

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821