Classification in microarray experiments

Sandrine Dudoit and Robert Gentleman

Statistics and Genomics - Lecture 5

Department of Biostatistics
Harvard School of Public Health

January 23-25, 2002

Outline

- Tumor classification using gene expression data.
- Standardization and distance functions.
- Unsupervised learning.
- Supervised learning.
- Biased sampling of classes and differential misclassification costs.
- Performance assessment.
- Aggregating classifiers.

Classification

Task. Assign objects to classes on the basis of measurements made on the objects.

Unsupervised learning. The classes are **unknown** a priori and need to be "discovered" from the data.

a.k.a. cluster analysis; class discovery; unsupervised pattern recognition.

Supervised learning. The classes are **predefined** and the task is to understand the basis for the classification from a set of labeled objects (training or learning set). This information is then used to classify future observations.

a.k.a. discriminant analysis; class prediction; supervised pattern recognition.

Tumor classification using gene expression data

A reliable and precise classification of tumors is essential for successful diagnosis and treatment of cancer.

Current methods for classifying human malignancies rely on a variety of morphological, clinical, and molecular variables.

In spite of recent progress, there are still uncertainties in diagnosis. Also, it is likely that the existing classes are heterogeneous.

DNA microarrays may be used to characterize the molecular variations among tumors by monitoring gene expression profiles on a genomic scale. This may lead to a more reliable classification of tumors.

Tumor classification using gene expression data

There are three main types of statistical problems associated with tumor classification:

1. the identification of new tumor classes using gene expression profiles – **unsupervised learning**;
2. the classification of malignancies into known classes – **supervised learning**;
3. the identification of marker genes that characterize the different tumor classes – **feature selection**.

Gene expression data

Features correspond to expression levels of different genes; classes correspond to different tumor types (e.g. ALL, AML) and are labeled by $\{1, 2, \ldots, K\}$.

Gene expression data on G genes (features) for n mRNA samples (observations)

$$\begin{aligned} \mathbf{x}_i &= \left(x_{i1}, x_{i2}, \ldots, x_{iG}\right) \\ &\quad \text{- gene expression profile / feature vector for sample } i \\ y_i &= \text{tumor class / response for sample } i, \end{aligned}$$

$i = 1, \ldots, n.$

May have covariates such as age, sex.

Gene expression data

Gene expression data on G genes (features) for n mRNA samples (observations)

$$X_{G\times n} = \begin{matrix} \\ \\ \\ \\ \text{Genes} \end{matrix} \overset{\overset{\text{mRNA samples}}{\downarrow}}{\begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{G1} & x_{G2} & \dots & x_{Gn} \end{bmatrix}}$$

x_{gi} = gene expression level of gene g in mRNA sample i

An array of conormalized arrays.

Filtering

It is generally useful to apply the gene–filtering methods introduced in a previous lecture prior to using any of the classification methods presented here – see more specific discussions of feature selection below.

Standardization and distance functions

- Samples are assigned to classes on the basis of their **distance** from (or similarity to) objects known to be in the classes.
- The distance or similarity measure that is used will have a large effect on the performance of the classification procedure.
- The distance measure and its behavior is intimately related to the **scale** on which measurements are made.
- These issues are well-known in the classification literature but there are few methods available for addressing them.

Standardization

- For microarray data, we usually have no a priori reason to favor one gene over another.
- Thus, we would like to put all genes on the same footing.
- This suggests standardizing the expression levels of each gene.
- Two common standardization procedures are

$$x^*_{gi} = \frac{x_{gi} - \bar{x}_g}{s_g}$$

$$\tilde{x}_{gi} = \frac{x_{gi} - x_{g(1)}}{x_{g(n)} - x_{g(1)}}$$

- Could use robust estimates of location and scale, like median and MAD.

Distance functions

- Euclidean distance;
- Mahalanobis distance;
- Manhattan metric;
- Minkowski metric;
- Canberra metric;
- one minus correlation;
- etc.

Unsupervised learning

In classical statistical terminology this is **clustering**.

Associated with each object is a set of G measurements which form the **feature vector** $\mathbf{X} = (X_1, \ldots, X_G)$. The feature vector $\mathbf{X}$ belongs to a feature space $\mathcal{X}$ (e.g. $\Re^G$).

The task is to identify groups of similar objects on the basis of observed measurements $\mathbf{X}_1 = \mathbf{x}_1, \ldots, \mathbf{X}_n = \mathbf{x}_n$.

Clustering gene expression data

- We can **cluster genes (rows)**, e.g. using large numbers of yeast experiments, to identify groups of co-regulated genes. We can cluster genes (rows) to reduce redundancy (cf. feature selection) in predictive models.
- We can **cluster cell samples (cols)**, e.g. tumor cells, for identification (profiles). Also, we might want to estimate the number of different tumor cell types in a set of samples based on gene expression levels.
- We can **cluster both** rows and columns at once.

Clustering gene expression data

- Clustering leads to readily interpretable figures.
- Clustering strengthens the signal when averages are taken within clusters of genes (Eisen).
- Clustering can be helpful for identifying patterns in time or space.
- Clustering is useful, perhaps essential, when seeking new subclasses of cell samples (tumors, etc).

Clustering gene expression data

Clustering can be usefully employed in an exploratory manner. The clusters that obtain from clustering samples should be compared with different experimental conditions such as:

- batch or production order of the arrays;
- batch of reagents;
- technician;
- order.

Any relationships observed here should be considered as potentially serious.

Example: Row and column clustering

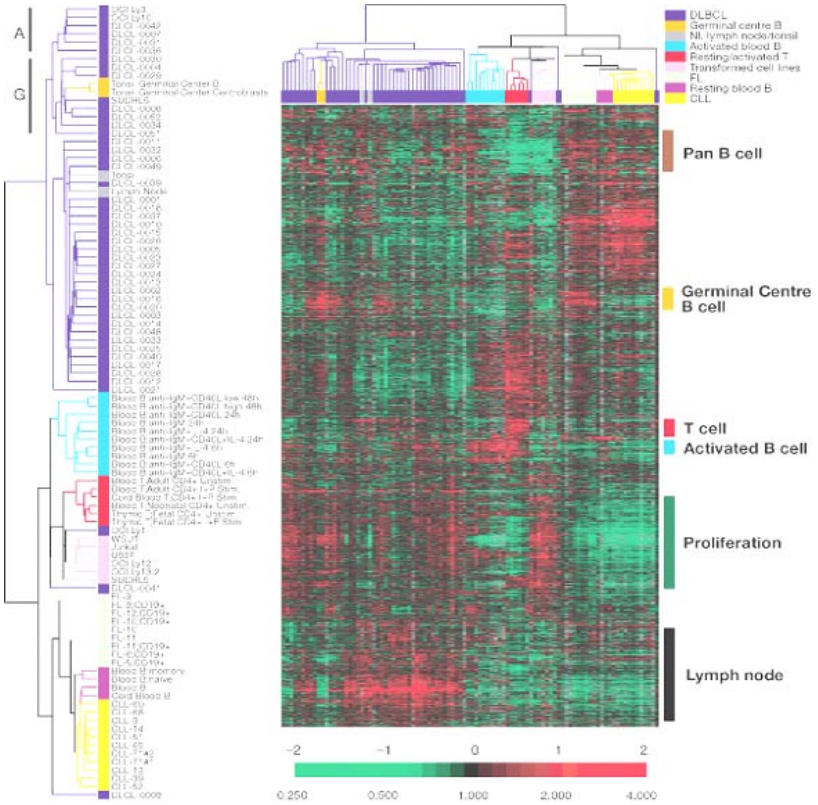


Figure 1: Alizadeh et al. (2000). Distinct types of diffuse large B–cell lymphoma identified by gene expression profiling. *Nature*.

Clustering gene expression data

Questions:

- Which genes / arrays to use?
- Which similarity or dissimilarity measure?
- Which clustering algorithm?

Answers will depend on the biological problem.

Partitioning methods

- Partition the data into a prespecified number K of mutually exclusive and exhaustive groups.
- Iteratively reallocate the observations to clusters until some criterion is met, e.g. minimize within cluster sums of squares.
- Examples: k–means, partitioning around medoids (PAM), self–organizing maps (SOM), model–based clustering (e.g. Gaussian mixtures), fuzzy clustering.
- The `cluster` package in R has many more.

Hierarchical methods

- Hierarchical clustering methods produce a **tree** or **dendogram**.
- They avoid specifying how many clusters are appropriate by providing a partition for each K obtained from cutting the tree at some level.
- The tree can be built in two distinct ways
  - bottom-up: **agglomerative** clustering;
  - top-down: **divisive** clustering.

Agglomerative methods

- Start with n mRNA sample (or G gene) clusters.
- At each step, merge the two closest clusters using a measure of between-cluster dissimilarity which reflects the shape of the clusters.
- Between-cluster dissimilarity measures:
  - *Unweighted Pair Group Method with Arithmetic mean (UPGMA)*: average of pairwise dissimilarities;
  - *Single-link*: minimum of pairwise dissimilarities;
  - *Complete-link*: maximum of pairwise dissimilarities.

Divisive methods

- Start with only one cluster.
- At each step, split clusters into two parts.
- Advantages: Obtain the main structure of the data, i.e., focus on upper levels of dendogram.
- Disadvantages: Computational difficulties when considering all possible divisions into two groups.

Partitioning vs. hierarchical

- **Partitioning**
  - Advantages: Provides clusters that satisfy an optimality criterion (approximately).
  - Disadvantages: Need initial K, long computation time.
- **Hierarchical**
  - Advantages: Fast computation (for agglomerative clustering).
  - Disadvantages: Rigid, cannot correct later for erroneous decisions made earlier.

Clustering

Important tasks (which are generic) are:

1. Estimating the number of clusters;
2. Assigning each observation to a cluster;
3. Assessing the strength/confidence of cluster assignments for individual observations;
4. Assessing cluster homogeneity.

There are a number of methods available for assessing the number of clusters in the data. Gordon (1996) discusses many of them. Recent additions are the *Gap statistic* proposed by Tibshirani et al. (2000) and the *Clest* procedure of Fridlyand & Dudoit (2001).

Supervised learning

Certain objects are to be classified as belonging to one of a number of predefined classes, say, $\{1, 2, \ldots, K\}$ (incl. doubt or outlier classes).

Associated with each object are a class label or **response** $Y \in \{1, 2, \ldots, K\}$ and a set of G measurements which form the **feature vector** or **vector of predictor variables** $\mathbf{X} = (X_1, \ldots, X_G)$. The feature vector $\mathbf{X}$ belongs to a feature space $\mathcal{X}$ (e.g. $\Re^G$).

The task is to classify an object into one of the K classes on the basis of an observed measurement $\mathbf{X} = \mathbf{x}$, i.e., predict Y from $\mathbf{X}$.

Classifiers

A **classifier** or **predictor** for K tumor classes **partitions** the space $\mathcal{X}$ of gene expression profiles into K disjoint and exhaustive subsets, $A_1, \ldots, A_K$, such that for a sample with expression profile $\mathbf{x} = (x_1, \ldots, x_G) \in A_k$ the predicted class is k.

Classifiers are built from past experience, i.e., from observations which are known to belong to certain classes. Such observations comprise the **learning (training) set (LS)**
$\mathcal{L} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$.

Classifier built from a learning set $\mathcal{L}$:

$$C(\cdot, \mathcal{L}) : \mathcal{X} \to \{1, 2, \ldots, K\}.$$

Predicted class for an observation $\mathbf{x}$: $C(\mathbf{x}, \mathcal{L}) = k$ if $\mathbf{x} \in A_k$.

Decision theory

Classification can be viewed as **statistical decision theory**. For each object, we need to decide which of the fixed set of classes that object belongs to. We use the observed feature vector $\mathbf{x}$ to aid in that decision.

Assume observations are i.i.d. from an unknown multivariate distribution. The proportion of objects of class k in the population will be denoted as $\pi_k = p(Y = k)$. Objects in class k have feature vectors with density $p_k(\mathbf{x}) = p(\mathbf{x}|Y = k)$.

When (unrealistically) both π_k and $p_k(\mathbf{x})$ are known this problem has a solution. This situation also gives upper bounds on the performance of classifiers in the more realistic setting where these quantities are not known (cf. Bayes rule and Bayes risk).

Decision theory

A reasonable criterion for assessing the quality of a classifier is the **misclassification rate**, $p(C(\mathbf{X}) \neq Y)$.

It will be useful to introduce the notion of a **loss function**. The loss function $L(i, j)$ simply elaborates the loss incurred if a class i case is erroneously classified as belonging to class j.

The **risk function** for a classifier is the expected loss when using it to classify: $R(C) = E[L(Y, C(\mathbf{X}))]$.

Decision theory

Typically $L(i,i) = 0$, and in many cases the loss is symmetric with $L(i,j) = 1$, $i \neq j$ – making an error of one type is equivalent to making an error of a different type.

Then, the risk is simply the misclassification probability. The Bayes rule to be introduced below minimizes this risk.

However, for some important examples such as diagnosis, the loss function is not symmetric.

Bayes rule

For known class conditional densities $p_k(\mathbf{x}) = p(\mathbf{x}|Y = k)$ and class priors π_k, let

$$p(k \mid \mathbf{x}) = \frac{\pi_k p_k(\mathbf{x})}{\sum_l \pi_l p_l(\mathbf{x})}$$

denote the posterior probability of class k given feature vector $\mathbf{x}$.

The **Bayes rule** predicts the class of an observation $\mathbf{x}$ by that with **highest posterior probability**: $C(\mathbf{x}) = \text{argmax}_k \; p(k \mid \mathbf{x})$.

Bayes rule

The Bayes rule minimizes the total risk under a symmetric loss function - **Bayes risk**.

Suitable adjustments can be made for other loss functions and to accommodate the doubt and outlier classes.

Many classifiers can be viewed as versions of this general rule with particular parametric or non-parametric estimates of $p_l(\mathbf{x})$ and π_k used to yield an estimate of $p(k \mid \mathbf{x})$.

Maximum likelihood discriminant rule

For known class conditional densities $p_k(\mathbf{x}) = p(\mathbf{x}|Y = k)$, the **maximum likelihood (ML) discriminant rule** predicts the class of an observation $\mathbf{x}$ by that which gives the largest likelihood to $\mathbf{x}$: $C(\mathbf{x}) = \text{argmax}_k \; p_k(\mathbf{x})$.

In the case of equal class priors π_k, this amounts to maximizing the posterior class probabilities $p(k|\mathbf{x})$, i.e., the Bayes rule.

Fisher linear discriminant analysis

First applied in 1935 by M. Barnard at the suggestion of R. A. Fisher (1936), **Fisher linear discriminant analysis (FLDA)** consists of

1. finding linear combinations $\mathbf{x}\,\mathbf{a}$ of the gene expression profiles $\mathbf{x} = (x_1, \ldots, x_G)$ with large ratios of between-group to within-group sums of squares - **discriminant variables**;
2. predicting the class of an observation $\mathbf{x}$ by the class whose mean vector is closest to $\mathbf{x}$ in terms of the discriminant variables.

Gaussian maximum likelihood discriminant rules

For multivariate Gaussian class densities, i.e., for $\mathbf{X}|Y = k \sim N(\mu_k, \Sigma_k)$ the maximum likelihood classifier is

$$C(\mathbf{x}) = \text{argmin}_k \ \left\{(\mathbf{x} - \mu_k)\Sigma_k^{-1}(\mathbf{x} - \mu_k)' + \log|\Sigma_k|\right\}.$$

In general, this is a **quadratic** rule: **Quadratic discriminant analysis - QDA**.

In practice, population mean vectors and covariance matrices are estimated by corresponding sample quantities, $\bar{\mathbf{x}}_k$ and S_k, resp.

Gaussian maximum likelihood discriminant rules

1. When the class densities have the same covariance matrix, $\Sigma_k = \Sigma$, the discriminant rule is based on the square of the Mahalanobis distance from class means and is linear

$$C(\mathbf{x}) = \text{argmin}_k \ (\mathbf{x} - \mu_k)\Sigma^{-1}(\mathbf{x} - \mu_k)'.$$

Linear discriminant analysis - LDA; FLDA for $K = 2$.

2. In this simplest case, when the class densities have the same diagonal covariance matrix $\Delta = \text{diag}(\sigma_1^2, \ldots, \sigma_G^2)$, the discriminant rule is linear and given by

$$C(\mathbf{x}) = \text{argmin}_k \sum_{g=1}^{G} \frac{(x_g - \mu_{kg})^2}{\sigma_g^2}.$$

Diagonal linear discriminant analysis - DLDA.

Weighted gene voting of Golub et al. (1999)

This is a minor variant on DLDA for two classes, $k = 1,\ 2$.

Sample DLDA classifies an observation $\mathbf{x}$ as 1 iff

$$\sum_{g=1}^{G} \frac{(\bar{x}_{1g} - \bar{x}_{2g})}{s_g^2} \left(x_g - \frac{(\bar{x}_{1g} + \bar{x}_{2g})}{2} \right) \geq 0.$$

The discriminant function can be rewritten as $\sum_g v_g$, where $v_g = a_g(x_g - b_g)$, $a_g = (\bar{x}_{1g} - \bar{x}_{2g})/s_g^2$, and $b_g = (\bar{x}_{1g} + \bar{x}_{2g})/2$.

In Golub et al. $a_g = (\bar{x}_{1g} - \bar{x}_{2g})/(s_{1g} + s_{2g})$ ... (Wrong units).

Logistic discrimination

For Gaussian class conditional densities with common covariance matrix (and other models)

$$\log p(k|\mathbf{x}) - \log p(1|\mathbf{x}) = \alpha_k + \mathbf{x}\beta_k.$$

This suggests modeling $\log p(k|\mathbf{x}) - \log p(1|\mathbf{x})$ more generally by some parametric family of functions, say $g_k(\mathbf{x};\theta)$ (with $g_1(\mathbf{x};\theta) \equiv 0$). Then we estimate

$$\widehat{p(k|\mathbf{x})} = \frac{\exp g_k(\mathbf{x};\hat{\theta})}{\sum_l \exp g_l(\mathbf{x};\hat{\theta})}.$$

Classification is done by using the (estimated) Bayes rule, i.e., $C(\mathbf{x},\mathcal{L}) = \text{argmax}_k \ \widehat{p(k|\mathbf{x})}$.

Logistic discrimination

In the machine learning literature the function $\exp a_k / \sum_l \exp a_l$ is known as the **softmax** function, in earlier statistical literature, it is known as the **multiple logit** function.

In **logistic regression**, take $g_k(\mathbf{x}; \theta) = \alpha_k + \mathbf{x}\beta_k$.

Logistic discrimination provides a more direct way of estimating posterior probabilities (Bayes rule). It is also easier to generalize than classical linear discriminant analysis.

Nearest neighbor classifiers

Nearest neighbor methods are based on a measure of **distance** between observations, such as the Euclidean distance or one minus the correlation between two gene expression profiles.

The k **nearest neighbor** rule, due to Fix and Hodges (1951), classifies an observation $\mathbf{x}$ as follows:

1. find the k observations in the learning set that are closest to $\mathbf{x}$;
2. predict the class of $\mathbf{x}$ by majority vote, i.e., choose the class that is most common among those k observations.

The number of neighbors k can be chosen by **cross-validation**.

Nearest neighbor classifiers

Nearest neighbor classifiers were initially proposed by Fix and Hodges (1951) as consistent non-parametric estimates of maximum likelihood discriminant rules.

Non-parametric estimates of the class conditional densities $p_k(\mathbf{x})$ are obtained by first reducing the dimension of the feature space $\mathcal{X}$ from G to one using a distance function.

The proportions of neighbors in each class are then used in place of the corresponding class conditional densities in the maximum likelihood discriminant rule.

Classification trees

Binary tree structured classifiers are constructed by repeated splits of subsets (nodes) of the measurement space $\mathcal{X}$ into two descendant subsets, starting with $\mathcal{X}$ itself. Each terminal subset is assigned a class label and the resulting partition of $\mathcal{X}$ corresponds to the classifier.

Three main aspects of tree construction: (i) the selection of the splits; (ii) the decision to declare a node terminal or to continue splitting; (iii) the assignment of each terminal node to a class.

Different tree classifiers use different approaches to deal with these three issues. Here, we use **CART - Classification And Regression Trees -** of Breiman et al. (1984).

Classification trees

1. **Splitting rule.** At each node, choose split that maximizes the decrease in impurity.
   E.g. of **impurity functions**: Gini index of diversity
   $i(t) = \sum_{i \neq j} p(i|t)p(j|t)$, also entropy and twoing rule.
2. **Split-stopping rule.** Grow large tree, selectively **prune** the tree upward, getting a decreasing sequence of subtrees, then use **cross-validation** to identify the subtree having the lowest estimated misclassification rate.
3. **Class assignment rule.** For each terminal node, choose the class that minimizes the resubstitution estimate of the misclassification probability, given that a case falls into this node.

Refinements: priors, weighted cost, splits based on linear combinations of variables, missing values (surrogate splits).

Example: Linear discriminant analysis

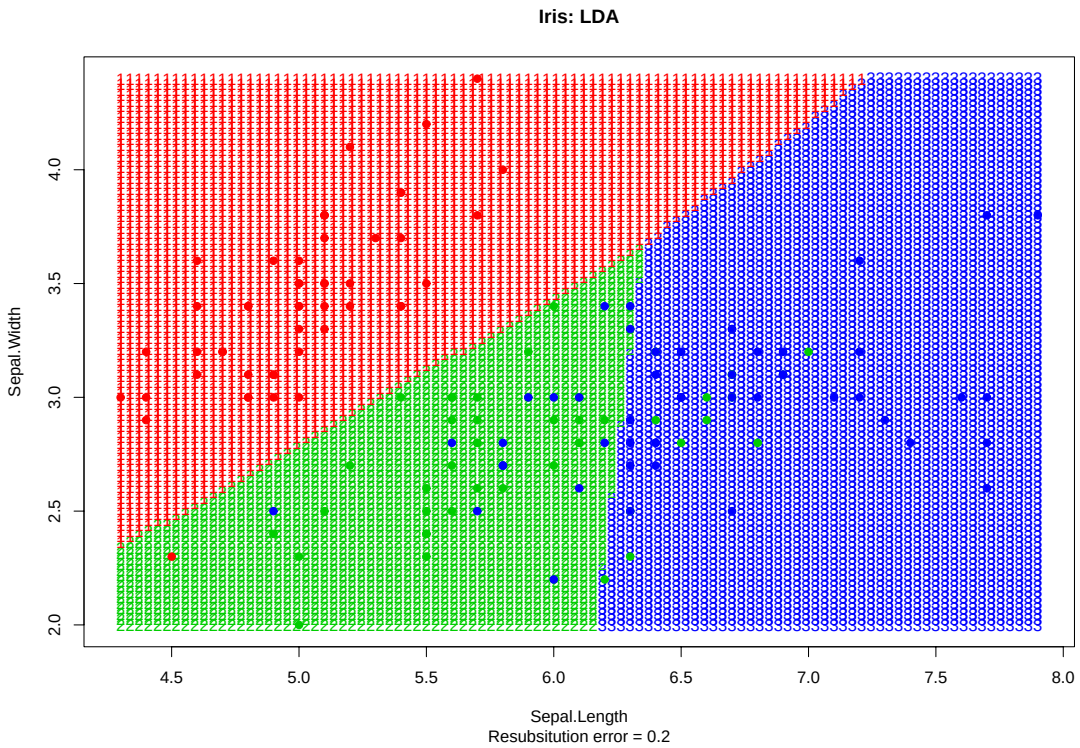


Figure 2: Iris dataset (Fisher 1936): Linear discriminant analysis using sepal length and width.

Example: Quadratic discriminant analysis

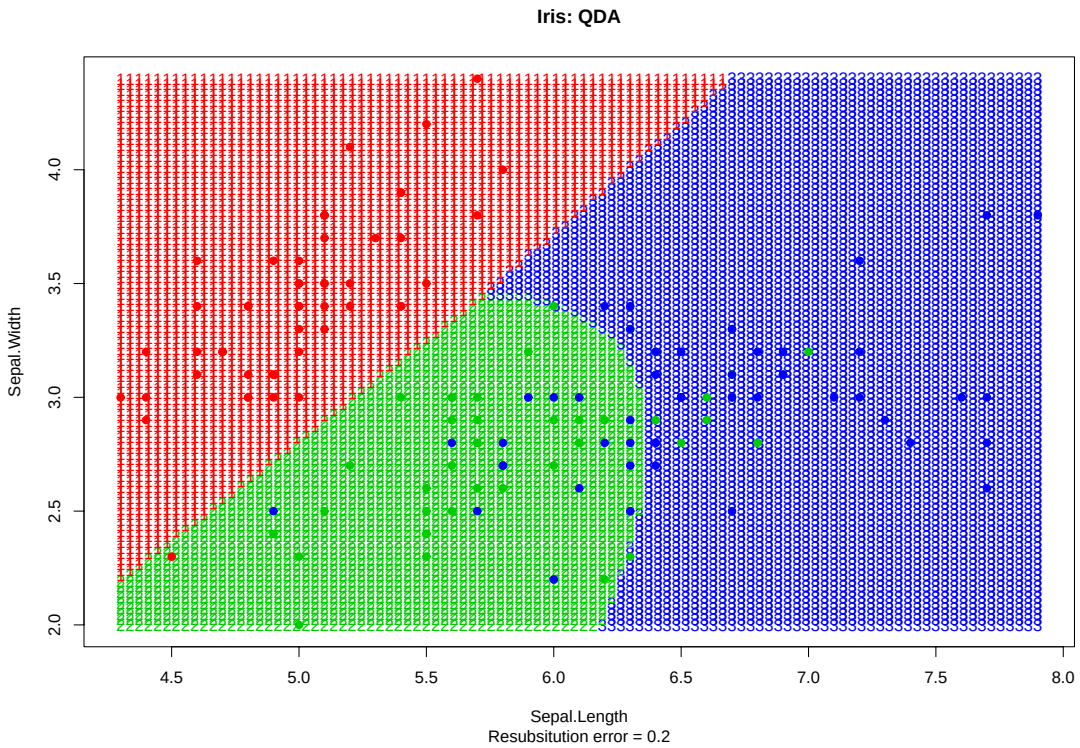


Figure 3: Iris dataset (Fisher 1936): Quadratic discriminant analysis using sepal length and width.

Example: Nearest neighbor classifier

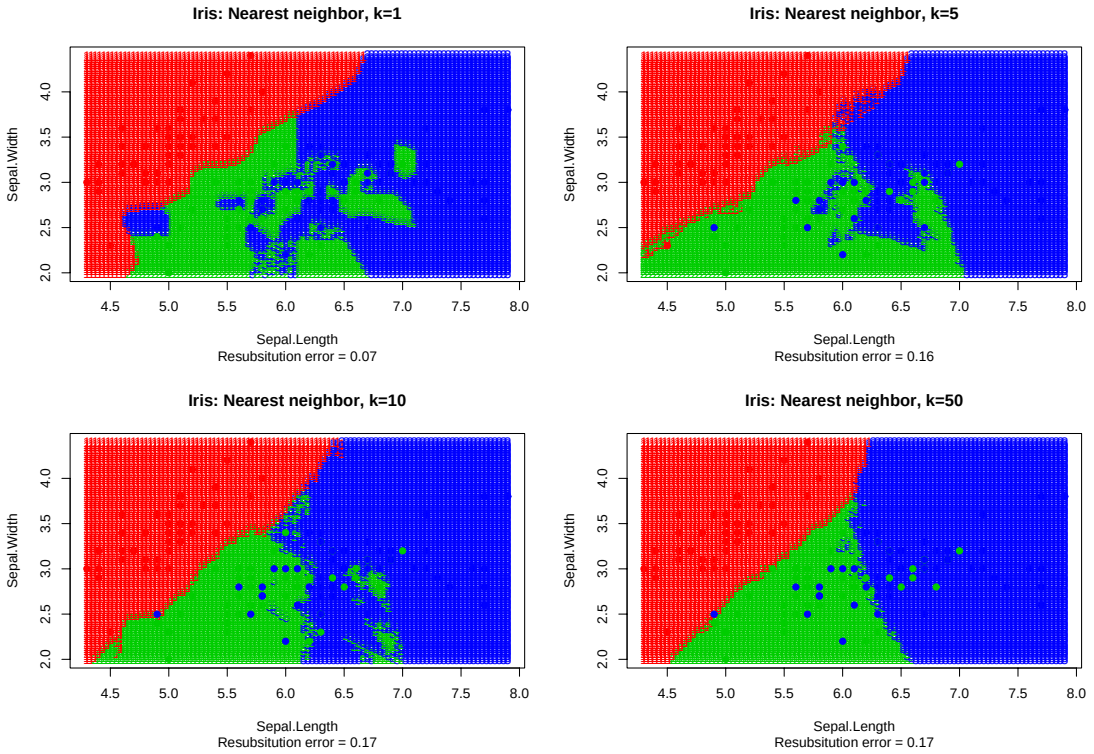
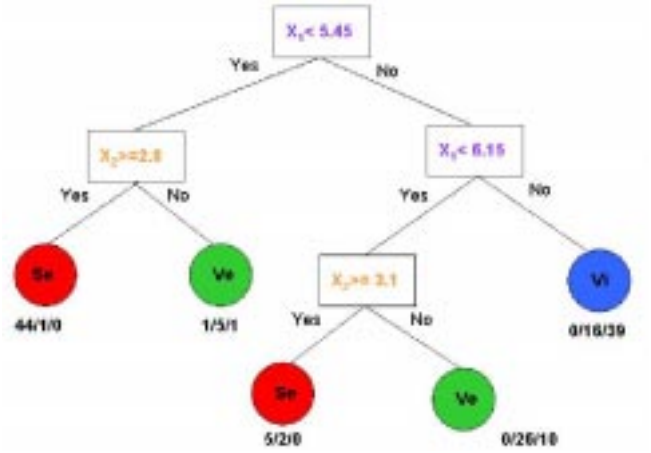


Figure 4: Iris dataset (Fisher 1936): $k = 1, 5, 10, 50$ nearest neighbor classifier, using sepal length and width.

Example: Classification tree



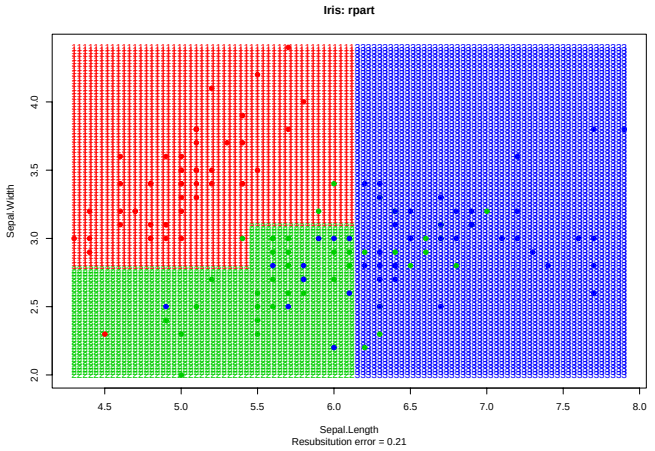


Figure 5: Iris dataset (Fisher 1936): classification tree (10-fold CV), using sepal length and width.

Other classifiers

- Perceptron;
- Neural networks;
- Support vector machines (SVMs);
- Learning vector quantization (LVQ).

Standardization and distance functions

We noted previously that both distance and the scale that observations are measured on are important. Some classifiers further modify the distance function.

- **Linear or quadratic discriminant analysis.** Based on the Mahalanobis distance from class means, thus features are already standardized in some sense. Is this an appropriate standardization?
- **Nearest neighbor classifiers.** One must decide on appropriate standardization and distance function for the problem under consideration.
- **Classification trees.** Invariant under monotone transformations of individual features, e.g. standardizations introduced above.

Asymmetric losses

- In many diagnosis settings, the loss incurred from misclassifying a diseased ($\mathbf{d}$) person as healthy ($\mathbf{h}$) far outweighs the loss incurred by making the error of classifying a healthy person as diseased.
- In this case, we must modify our loss function. The simple expedient of stating that the first error is e times higher than the second yields a plug-in decision rule. Classify a patient as diseased if $p(\mathbf{d} \mid \mathbf{x}) > c = 1/(1+e)$.
- Again, we can use any estimate of the class posterior probabilities.

Assymetric class sample sizes

- In many situations, such as medical diagnosis, the representation of the classes in the learning set does not reflect their importance in the problem.
- For example, in a binary classification problem with a rare disease class ($\mathbf{d}$) and a common healthy class ($\mathbf{h}$), a learning set obtained by random sampling from the population would contain a vast majority of healthy cases.
- Unequal class sample sizes could possibly lead to serious biases in the estimation of posterior probabilities $p(\mathbf{d} \mid \mathbf{x})$ and $p(\mathbf{h} \mid \mathbf{x})$.

Assymetric class sample sizes

Consider the case of linear discriminant analysis which assumes a common covariance matrix estimated using both samples. This estimate will be dominated by the more abundant sample.

This is fine if the covariance matrix is really the same for both classes. However, in the case of unequal covariance matrices, the bias in the class posterior probabilities $p(k \mid \mathbf{x})$ is more severe when the classes are unequally represented in the learning set.

Biased sampling of classes

Here are some suggestions that might help to alleviate biases from unequal class representation in the learning set.

- Subsample the abundant population so that both classes are on an equal footing in the parameter estimation, thus reducing estimation biases.
  Problem: this would be wasteful of training data when the biased learning set is obtained from a larger learning set.
- Downweight cases from the abundant class so that the sum of the weights is equal to the number of cases in the less abundant class.
- Obtain less biased estimates of the class posterior probabilities, either by theory (as can be done for LDA) or via bias reduction techniques such as the jackknife.

Biased sampling of classes

N.B. The above solutions effectively make the sample proportions differ from the population proportions and can in turn lead to biased estimates of the class posterior probabilities.

To see this, let n_k denote the number of class k cases in the learning set. The plug-in estimators of class posterior probabilities are in fact estimating quantities proportional to $p(k \mid \mathbf{x})n_k/\pi_k$.

Adjustment is thus required to ensure that our preferred estimator of the class posterior probabilities $p(k \mid \mathbf{x})$ is approximately unbiased.

This can be done by specifying appropriate priors for DA and CART, and by using weighted voting for nearest neighbors.

Features

- **Feature selection.** Automatic with trees, no need to preselect features (no over-fitting).
  For DA and nearest neighbor classifiers one should perform preliminary feature selection, otherwise performance degrades with a large number of uninformative features.

  Must take this into account when estimating error rates; feature selection is an aspect of building the predictor (cf. performance assessment below).

- **Missing data.** Automatic imputation with trees; for other classifiers either ignore missing data or use imputed data (e.g. nearest neighbor imputation).

Performance assessment

1. **Resubstitution estimation.** Error rate on the learning set.
   Problem: can be severely biased downward.

2. **Test set estimation.** Cases in the learning set $\mathcal{L}$ are divided into two sets, $\mathcal{L}_1$ and $\mathcal{L}_2$; classifier is built using $\mathcal{L}_1$ and error rate is computed for $\mathcal{L}_2$.
   Must ensure that $\mathcal{L}_1$ and $\mathcal{L}_2$ are i.i.d.: e.g.
   two-thirds/one-third repeated random sampling.
   Problem: reduces effective sample size.

Performance assessment

3. **V-fold cross-validation (CV) estimation.** Cases in $\mathcal{L}$ are randomly divided into V subsets $\mathcal{L}_v$, $v = 1, \ldots, V$, of as nearly equal size as possible. Classifiers are built on $\mathcal{L} - \mathcal{L}_v$, test set error rates are computed for $\mathcal{L}_v$, and averaged over v.

   Special case $V = n$, leave-one-out cross-validation (LOOCV). More computation with $V = n$.

   Bias-variance trade-off: taking a smaller V can give a larger bias, but a smaller variance and mean-square error.

4. **Out-of-bag estimation.** See discussion of random forests below.

Performance assessment

The use of cross–validation (or any other process) is intended to provide accurate estimates of the classification error rate.

N.B. These estimates relate only to the experiment that was (cross-) validated.

Performance assessment

There is a common practice in this area of doing feature selection using all of the data and then using cross–validation only on the model building and classification portion.

In that case, inference can only be applied to the latter portion of the process.

However, in most cases, the features are unknown and the intended inference includes feature selection. Then, CV estimates as above tend to suffer from a downward bias and inference is not warranted.

Features should be selected only on the basis of the learning set $\mathcal{L}_1$ for test set estimation or the samples in $\mathcal{L} - \mathcal{L}_v$ for CV estimation.

Performance assessment

Building a classifier can be viewed as model building. In that sense, we are attempting to make inference about the parameters in a model. In many cases these parameters have standard errors that are proportional to $1/\sqrt{n}$.

If we use leave–one–out cross–validation, we are perturbing the data by amount $1/n$. This means that we are trying to make inference about the parameters from perturbations that are much smaller than the variability in the parameters.

It would probably be beneficial to consider leaving out more observations (e.g. $V = 10$). This can give larger bias, but smaller variance and mean-square error.

Some form of stratified sampling may need to be carried out to ensure balance across important classes in all samples.

Aggregating classifiers

Breiman (1996, 1998) found that gains in accuracy could be obtained by **aggregating predictors** built from perturbed versions of the learning set. In classification, the multiple versions of the predictor are aggregated by **voting**.

Let $C(\cdot, \mathcal{L}_b)$ denote the classifier built from the bth perturbed learning set $\mathcal{L}_b$ and let w_b denote the weight given to predictions made by this classifier. The predicted class for an observation $\mathbf{x}$ is given by

$$\text{argmax}_k \sum_b w_b \, I(C(\mathbf{x}, \mathcal{L}_b) = k).$$

Key to improved accuracy: instability of classifier.

Bagging

Standard bagging. In the simplest form of **bagging - b**ootstrap **agg**regat**ing -** perturbed learning sets of the same size as the original learning set are non-parametric bootstrap replicates of the learning set, i.e., drawn at random with replacement from the learning set.

Predictors are built for each perturbed dataset and aggregated by plurality voting ($w_b = 1$).

Parametric bootstrap. Perturbed learning sets are generated according to a mixture of multivariate Gaussian distributions.

Convex pseudo-data. Breiman (1996).

Random forests

Random forest. A combination of tree classifiers (or other), where each tree depends on the value of a random vector, i.i.d. for all trees in the forest.

- **Random learning set - bagging.** Each tree is formed from a bootstrap sample of the learning set (same size) – random vector consists of the outcomes of n draws at random with replacement from $\{1, \ldots, n\}$
- **Random features.** For a fixed parameter $G_0 << G$ (e.g. $\sqrt{G}$), G_0 features are randomly selected at each node and only these are searched though for the best split – random vector consists of the outcomes of G_0 draws at random without replacement from $\{1, \ldots, G\}$.

Random forests

- **Random forest.** Combine above two sources of randomness.

A maximal exploratory tree is grown (pure terminal nodes) for each bootstrap learning set and the forest obtains a classification by plurality voting.

By-products of aggregation

I. Out-of-bag estimate of error rate. No need for CV or test set estimates of error rate; an unbiased estimate is obtained as a by-product of the bootstrap.

For each bootstrap sample, about 1/3 of the cases are left out and not used in the construction of the tree $\Rightarrow$ test set.

For the bth bootstrap sample, put the "out-of-bag" cases down the bth tree to get a test set classification. Let the final test set classification of the forest be the class having the most votes. Compare this classification to the class labels of the learning set to get the out-of-bag estimate of the error rate.

By-products of aggregation

II. Case-wise information.

- *Class probability estimates (Votes):* the proportion of votes for the "winning" class ($\in [0, 1]$).
  This gives a measure of confidence for the prediction.
- *Vote margins:* the proportion of votes for the true class minus the maximum of the proportion of votes for each of the other classes ($\in [-1, 1]$).
  Vote margins can be used to detect mislabeled learning set cases.

By-products of aggregation

III. Variable importance statistics. Here, variable importance is defined in terms of the contribution to predictive accuracy, i.e., predictive power.

For each tree, randomly permute the values of the jth variable for the out-of-bag cases, put these new covariates down the tree, and get new classifications for the forest.

By-products of aggregation

The importance of the jth variable can be defined as

- *Importance measure 1:* the difference between the out-of-bag error rate for randomly permuted jth variable and the original out-of-bag error rate.
- *Importance measure 2:* the average across all cases of the differences between the margins for randomly permuted jth variable and for the original data.
- *Importance measure 3:* the number of lowered margins minus the number of raised margins.
- *Importance measure 4:* sum of all decreases in impurity in the forest due to a given variable, normalized by number of trees.

By-products of aggregation

- **IV. Intrinsic proximities between cases.** Proportion of trees for which cases i and j are in the same terminal node.
  - Clustering;
  - Multidimensional scaling;
  - Outlier detection: 1/(sum of squared proximities of cases in same class).

Boosting

Freund and Schapire (1997), Breiman (1998).

The data are **resampled adaptively** so that the weights in the resampling are increased for those cases most often misclassified.

The aggregation of predictors is done by **weighted voting**.

Boosting

For a learning set $\mathcal{L} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$, let $\{p_1, \ldots, p_n\}$ denote the resampling probabilities, initialized to be equal.

For bth step of the boosting algorithm (adaptation of `AdaBoost`):

1. generate a perturbed learning set $\mathcal{L}_b$ of size n by sampling with replacement from $\mathcal{L}$ using $\{p_1, \ldots, p_n\}$;
2. build a classifier $C(\cdot, \mathcal{L}_b)$ based on $\mathcal{L}_b$;
3. run the learning set $\mathcal{L}$ through the classifier $C(\cdot, \mathcal{L}_b)$ and let $d_i = 1$ if the ith case is classified incorrectly and $d_i = 0$ o.w.;
4. define

   $$\epsilon_b = \sum_i p_i d_i, \ \beta_b = (1 - \epsilon_b)/\epsilon_b \text{ and } w_b = \log(\beta_b)$$

   and update the resampling probabilities for the $(b+1)$st step by

   $$p_i = \frac{p_i \beta_b^{d_i}}{\sum_i p_i \beta_b^{d_i}}.$$

Comparison of classifiers

Ref. S. Dudoit, J. Fridlyand, and T. P. Speed. (2002). Comparison of discrimination methods for the classification of tumors using gene expression data. *JASA*. Vol. 97, No. 457.

- Linear and quadratic discriminant analysis
  - Fisher linear discriminant analysis (FLDA);
  - Diagonal linear discriminant analysis (DLDA)
    $\longrightarrow$ gene voting scheme of Golub et al. is a variant of DLDA;
  - Diagonal quadratic discriminant analysis (DQDA);
- Nearest neighbor classifiers;
- Classification trees (CART);
- Bagging and boosting.

Other studies: Support vector machines (SVMs), penalized logistic regression, Bayesian regression analysis.

Comparison study - datasets

- **Lymphoma.** Alizadeh et al. (2000).
  $n = 81$ samples, $p = 4,682$ genes,
  3 classes (B-CLL, FL, DLBCL).

- **Leukemia.** Golub et al. (1999).
  $n = 72$ samples, $p = 3,571$ genes,
  3 classes (B-cell ALL, T-cell ALL, AML).

- **NCI 60.** Ross et al. (2000).
  $n = 64$ samples, $p = 5,244$ genes,
  8 classes.

Comparison study - results

- In the main comparison, the nearest neighbor classifier and DLDA had the smallest error rates, while FLDA had the highest error rates.
- Aggregation improved the performance of CART, the largest gains being with boosting and bagging with convex pseudo-data.
- For the lymphoma and leukemia datasets, increasing the number of predictor variables (genes) didn't affect much the performance of the various classifiers. There was an improvement for the NCI 60 dataset.
- A more careful selection of a small number of genes improved the performance of FLDA dramatically.

Comparison study - discussion

- "Diagonal'x' LDA vs. "correlated" LDA: ignoring correlation between genes helped here.
- Unlike classification trees and nearest neighbors, LDA is unable to take into account gene interactions.
- Although nearest neighbors are simple and intuitive classifiers, their main limitation is that they give very little insight into mechanisms underlying the class distinctions.
- Classification trees are capable of handling and revealing interactions between variables.
- Useful by-product of aggregated classifiers: error rates, prediction votes, variable importance statistics.
- With larger training sets, we expect an improvement in the performance of aggregated classifiers and to gain more insight into the relationship between tumor class and gene expression levels.

Discussion

There are two main goals in analyzing the data

- **Prediction.** Predict response of future observations (tumor samples) using a collection of predictor variables (genes).
- **Information.** Extract information about the underlying data generating mechanism, i.e., on the relationship between response and predictor variables.

Discussion

Data models: High-dimensional data, unknown distribution, many models will provide similar predictive accuracy.

Accuracy vs. simplicity (interpretability): a predictor does not have to be simple to provide accurate prediction or reliable information about the relationship between response and predictor variables (e.g. random forests).

Dimensionality: newer classification procedures thrive on the number of variables, the more the better (trees, SVMs vs. DA, nearest neighbors).

Ref. L. Breiman (2001). Statistical modeling: the two cultures. *Statistical Science.*, Vol. 16, No. 3, p. 199–231.

Lab 5

In Lab 5, we will apply the methods introduced in the lecture to gene expression data from the leukemia study of Golub et al. (1999).

Acknowledgments

- **Brown Lab**, Biochemistry, Stanford.
- **Leo Breiman**, Statistics, UC Berkeley.
- **Jane Fridlyand**, UCSF Cancer Center.
- **Yee Hwa (Jean) Yang**, Statistics, UC Berkeley.
- **Sabina Chiaretti**, DFCI.