STATISTICAL LEARNING SYSTEMS
LECTURES 4 and 5: CLASSIFICATION

Ph. D. Program 2013/2014

J. Koronacki

korona@ipipan.waw.pl

Institute of Computer Science, Polish Academy of Sciences

The project is co-financed by the European Union
within the framework of European Social Fund

Before we turn to (supervised, unsupervised and semisupervised) learning, let us fix some terminology:

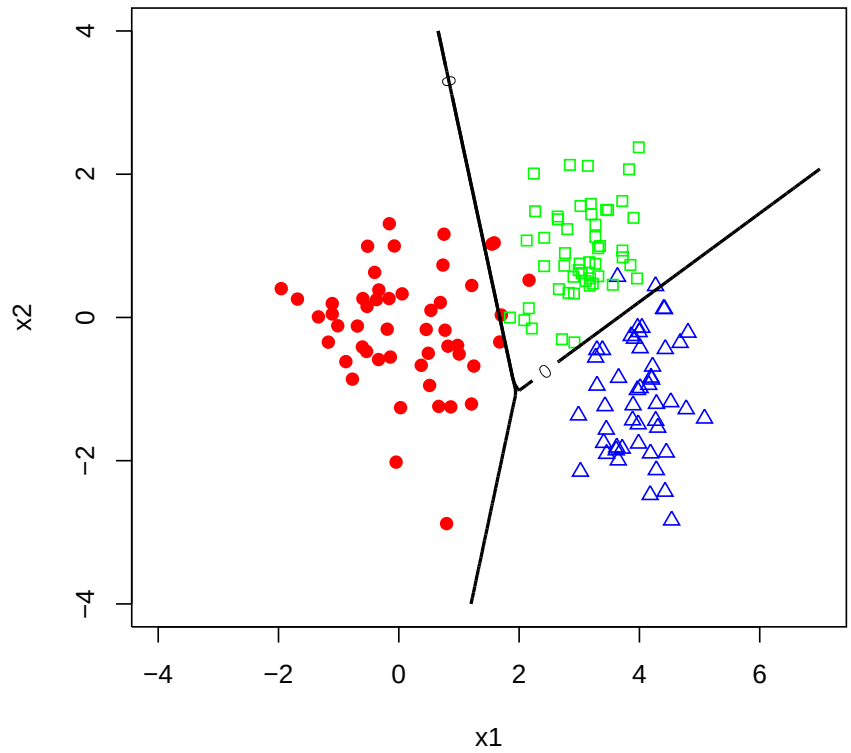
We are given a set (or population) of objects

Each object is described by a vector of features (attributes, descriptor variables) which span a feature (observation) space.

Supervised Learning = Classification (or Discriminant Analysis, DA) + Regression Analysis (including Time Series Analysis)

Within DA we are also given class structure for a sample of objects (training set, learning set)

Rys. 1.2



Supervised classification paradigm

- Given a set of objects with known descriptor variables (attributes, features) and known outcome class membership

- construct a rule which will allow new objects to be assigned to an outcome class on the basis of their attributes

Examples:

- deciding which customers will be good insurance or credit risks (bank scoring);
- identifying customers who are likely to quit or decrease the use of service (churning or attrition in CRM);
- deciding whether a patient is ill on the basis of medical records (blood preasure, sugar level, occurence of heart disease in the family etc.);
- deciding whether somebody is *prone* to succumb to a certain illness within 10 years;
- discriminating between spam and 'genuine' e-mail messages;
- automatic digit recognition, etc. etc.

1. Some Notations and Preliminary Remarks

Measurements $x^{(j)}$, $j = 1, \ldots, p$ are taken on each individual (or object), and the individuals are to be classified into one of g classes (g being finite).

A training sample is available which has data in the form $(\mathbf{x}_i, y_i)$, $i = 1, \ldots, n$, where y_i is the class label of the ith object, $y_i \in \{1, \ldots, g\}$, $\mathbf{x}_i = (x_i^{(1)}, \ldots, x_i^{(p)})^{\mathrm{T}} = (x_{i1}, \ldots, x_{ip})^{\mathrm{T}}$ is the vector of measurements taking values in an p-dimensional space $\mathcal{X}$.

We desire to find a decision (or discriminant, or classification) rule (or, for short, a classifier)

$$d(\mathbf{x}) : \mathcal{X} \to \{1, \ldots, g\}$$

for classifying the objects.

Important remark: Separability of classes is neither assumed nor believed in!

It can happen that there are objects in different classes with the same, or very close, values of all attributes.

That is why probabilistic modeling comes into play.

We regard a feature vector $(x_1, x_2, \ldots, x_p)$ as a value of a random vector with different distribution in each class.

When the distributions admit the same values (their supports are not disjoint) it is possible to get the same, or very close values coming from different classes.

2. Classical Discriminant Analysis

Fisher's Linear Discriminant Analysis

Consider the case when they are are only two classes $g = 2$.

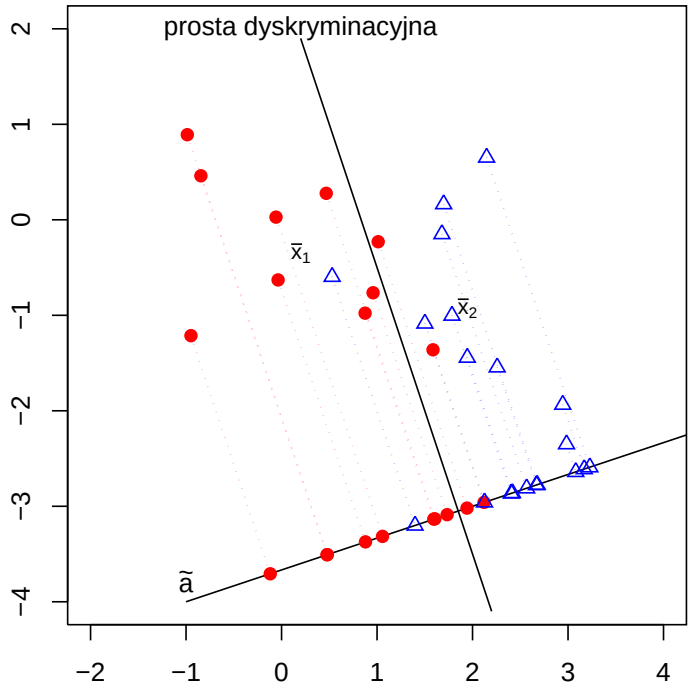
The training sample splits into two subamples corresponding to $y = 1$ and $y = 2$:

n_1 class 1 observations

n_2 class 2 observations

Idea: Find a direction $\mathrm{a} \in R^p$ which best separates observations from two classes when projected on this direction, taking into account within-group variability of projections.

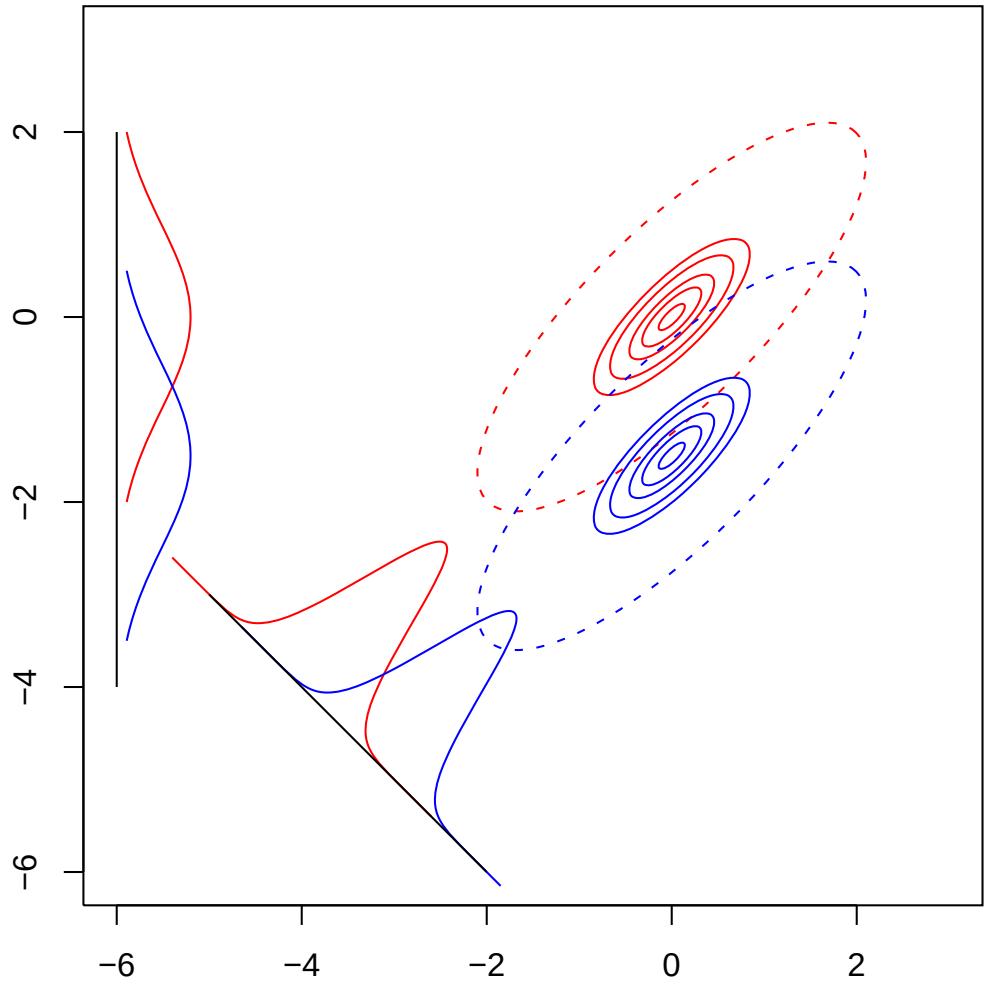
Rys. 1.3.



Summarize the subsamples by their means and adopt the distance between the two means in a given direction as a measure of separation. For the given direction $\mathbf{a}$, the separability between the two subsamples is then defined as the difference of projected means

$$\mathbf{a}^T(\bar{\mathbf{x}}_2 - \bar{\mathbf{x}}_1)$$

Warning: The separating hyperplane in general SHOULD NOT BE the perpendicular bisector of the segment joining the centroids $\bar{x}_1$ and $\bar{x}_2$



Recall that for a given sample $\mathbf{x}_1, \ldots, \mathbf{x}_m$, sometimes to be given in the form of a $(m \times p)$ data matrix

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T \\ . \\ . \\ . \\ \mathbf{x}_m^T \end{bmatrix},$$

the sample mean vector is

$$\bar{\mathbf{x}} = \frac{1}{m} \sum_{r=1}^{m} \mathbf{x}_r = \frac{1}{m} \mathbf{X}^T \mathbf{1},$$

the sample covariance between the ith and jth variable is

$$s_{ij} = \frac{1}{m-1} \sum_{r=1}^{m} (x_{ri} - \bar{x}_i)(x_{rj} - \bar{x}_j)$$

so that the sample covariance matrix is

$$\mathbf{S} = \frac{1}{m-1} \sum_{r=1}^{m} (\mathbf{x}_r - \bar{\mathbf{x}})(\mathbf{x}_r - \bar{\mathbf{x}})^T$$

Useful fact:

Empirical variance of $a^{\mathrm{T}}x_r, r = 1, 2, \ldots, m$ is $\mathbf{a}^{\mathrm{T}}\mathbf{S}\mathbf{a}$.

Assuming common covariance structure of both subsamples the common covariance matrix may be defined as follows: given a sample of size n with n_k items from the kth class, $k = 1, \ldots, g$, each kth class has its "own" sample mean $\bar{\mathbf{x}}_k$ and sample covariance matrix $\mathbf{S}_k$. Accordingly, one can define with $n = n_1 + n_2 + \ldots + n_g$

$$\mathbf{W} = \frac{1}{n-g}\sum_{k=1}^{g}(n_k - 1)\mathbf{S}_k$$

as the within-class (or within-group) covariance matrix.

Empirical variance of any of the projected subsamples is

$$\mathbf{a}^{\mathrm{T}}\mathbf{W}\mathbf{a}$$

and the squared distance bewteen the projected means $\bar{\mathbf{x}}_1$ and $\bar{\mathbf{x}}_2$ is $(\mathbf{a}^{\mathrm{T}}\bar{\mathbf{x}}_2 - \mathbf{a}^{\mathrm{T}}\bar{\mathbf{x}}_1)^2$.

Standardized measure of separability is

$$\frac{(\mathbf{a}^T\bar{\mathbf{x}}_2 - \mathbf{a}^T\bar{\mathbf{x}}_1)^2}{\mathbf{a}^T\mathbf{W}\mathbf{a}}; \tag{1}$$

We look for direction $\mathbf{a}$ which maximizes this expression and project both subsamples onto this direction. Note: the vector joining $\bar{\mathbf{x}}_1$ and $\bar{\mathbf{x}}_2$ maximizes only the numerator of (??) !!

The optimal direction is

$$\mathbf{a} \propto \mathbf{W}^{-1}(\bar{\mathbf{x}}_2 - \bar{\mathbf{x}}_1).$$

Fisher's LDA Classification rule (Fisher's rule):

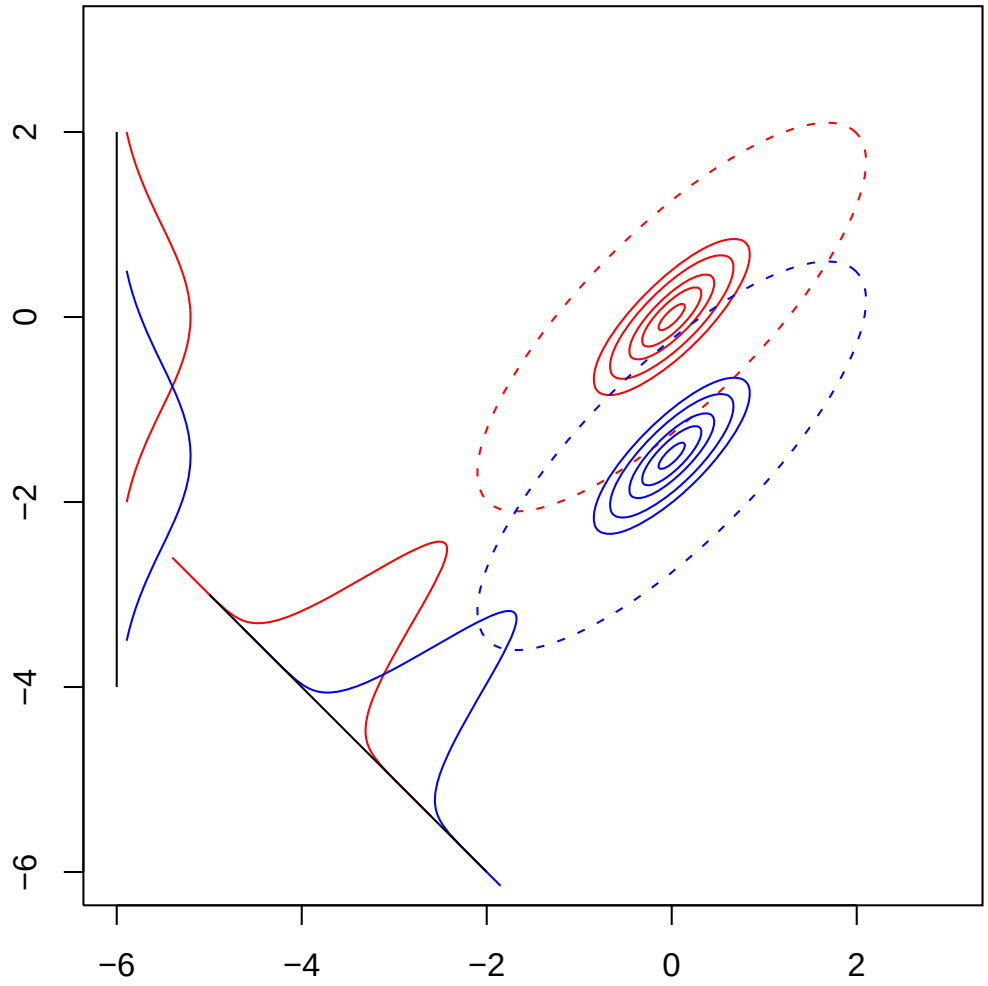
classify new vector $\mathbf{x}$ to class j if

$$|\mathbf{a}^\mathbf{T}\mathbf{x} - \mathbf{a}^\mathbf{T}\bar{\mathbf{x}}_j| < |\mathbf{a}^\mathbf{T}\mathbf{x} - \mathbf{a}^\mathbf{T}\bar{\mathbf{x}}_k|$$

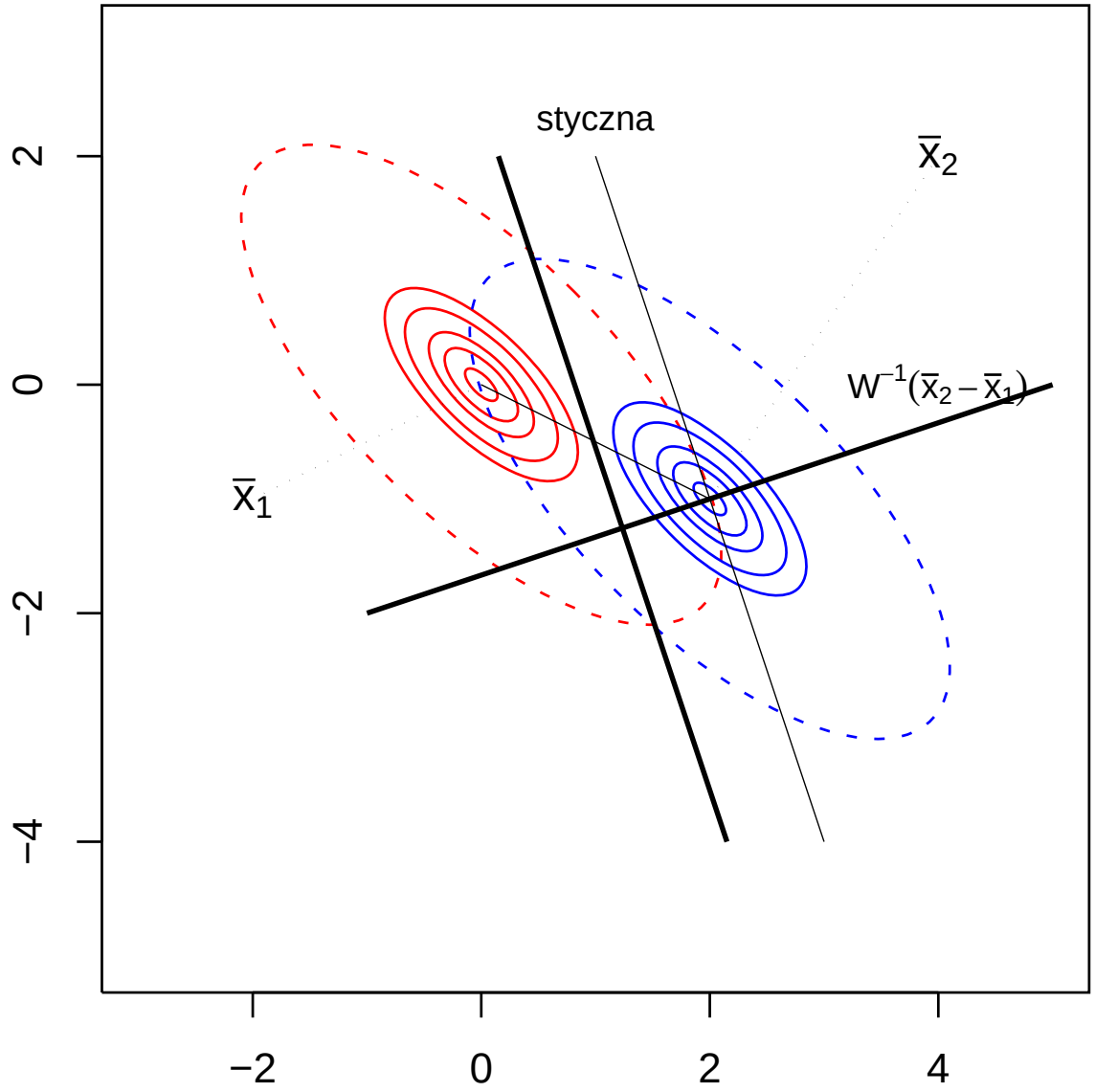
for $k \neq j$, $k, j \in \{1, 2\}$, otherwise classify to the opposite class.

The last equation describes points lying on the same side of a hyperplane perpendicular to direction $\mathbf{a}$ and passing through $\mathbf{a}^T(\bar{\mathbf{x}}_1 + \bar{\mathbf{x}}_2)/2$.

Note: The separating hyperplane in general IS NOT the perpendicular bisector of the segment joining the centroids $\bar{x}_1$ and $\bar{x}_2$



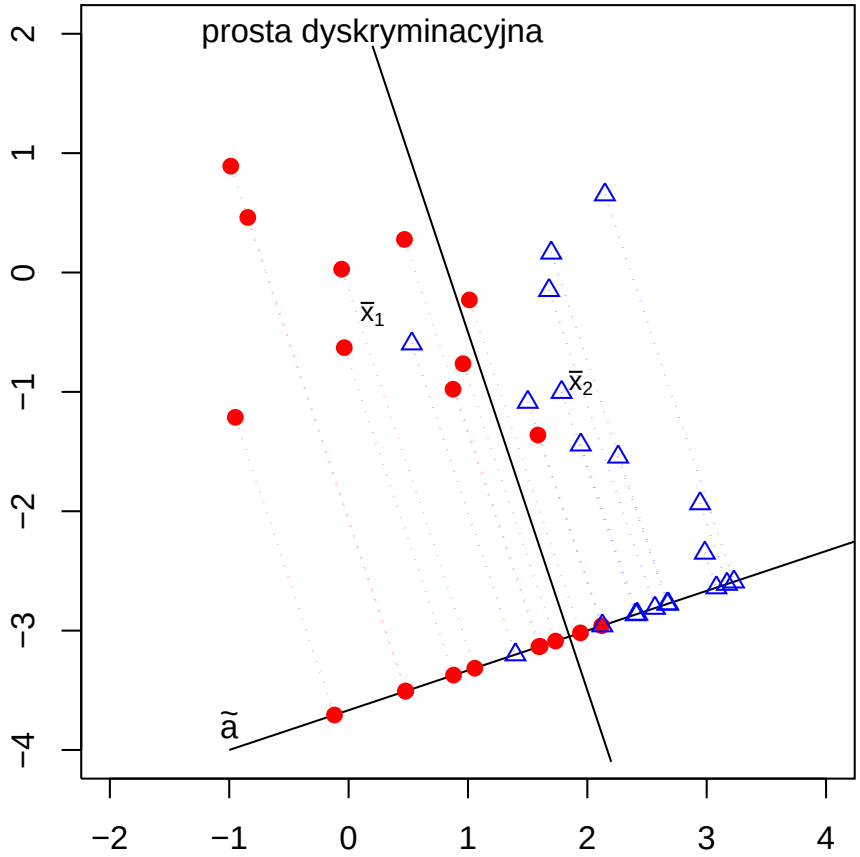
Rys. 1.4.



Equivalently, choose population k for which $g_k(\mathbf{x}) = \max_{i=1,2} g_i(\mathbf{x})$, where

$$g_i(\mathbf{x}) = \mathbf{x}^{\mathbf{T}}\mathbf{W}^{-1}\bar{\mathbf{x}}_i - \frac{1}{2}\bar{\mathbf{x}}_i^T\mathbf{W}^{-1}\bar{\mathbf{x}}_i, \qquad i = 1, 2.$$

Rys. 1.3.



Solution for general g

Consider, for a fixed class indicator i, the discriminant function

$$g_i(\mathbf{x}) = \mathbf{x}^{\mathrm{T}}\mathbf{W}^{-1}\bar{\mathbf{x}}_i - \frac{1}{2}\bar{\mathbf{x}}_i^T\mathbf{W}^{-1}\bar{\mathbf{x}}_i, \qquad i = 1, \ldots, g,$$

where $\mathbf{W}$ is within-group covariance matrix for the *whole* data set. Choose class k such that $g_k(\mathbf{x}) = \max_{i=1,\ldots,g} g_i(\mathbf{x})$.

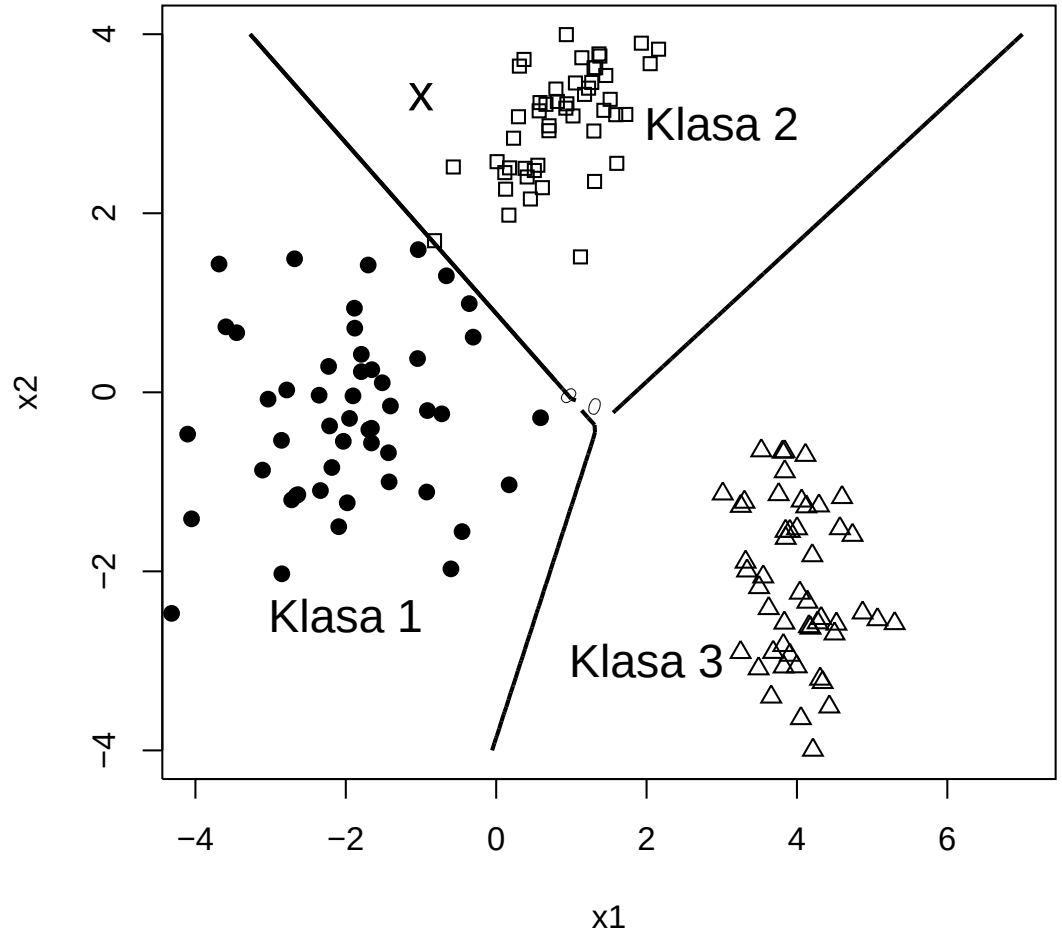
This, in fact, is equivalent, to the following procedure:

Consider, for fixed class indicators $i \neq j$ observations belonging to these classes only and solve classification problem for classes i and j with $\mathbf{W}$ as an estimator of within-group covariance. Call the corresponding rule $d_{ij}(\cdot)$. Thus we solve $\binom{g}{2}$ separate two-class classification problems.

Fisher's LDA classification rule for general g: Classify $\mathbf{x}$ to a class i_0 such that

$$d_{i_0 j}(x) = i_0 \qquad \text{for} \qquad j \neq i_0$$

Rys. 1.9a



Note that in this example a projection on a single direction yields poor separation of the three classes!

Canonical variables

For $g = 2$ direction $\mathrm{a}_1 = W^{-1}(\bar{\mathrm{x}}_2 - \bar{\mathrm{x}}_1)$ is characterised by the property that the data projected onto this direction maximizes

$$\frac{\text{between} - \text{class variance of projections}}{\text{within} - \text{class variance of projections}} = \frac{\mathbf{a}^{\mathrm{T}}\mathbf{B}\mathbf{a}}{\mathbf{a}^{\mathrm{T}}\mathbf{W}\mathbf{a}} \qquad (2)$$

where between-class variance of projections is a variance of the projected means and between-class covariance matrix B is

$$\mathrm{B} = \frac{1}{(g-1)} \sum_{i=1}^{g} n_k (\bar{\mathrm{x}}_k - \bar{\mathrm{x}})(\bar{\mathrm{x}}_k - \bar{\mathrm{x}})^{\mathrm{T}},$$

$g = 2$ (note that g can in fact be any positive integer, thus giving way to multiclass generalization). Indeed,

$$(\mathbf{a}^{\mathrm{T}}\bar{\mathrm{x}}_2 - \mathbf{a}^{\mathrm{T}}\bar{\mathrm{x}}_1)^2 = \mathbf{a}^{\mathrm{T}}(\bar{\mathrm{x}}_2 - \bar{\mathrm{x}}_1)(\bar{\mathrm{x}}_2 - \bar{\mathrm{x}}_1)^{\mathrm{T}}\mathbf{a}$$

and it is easy to show that the last quantity is equal to

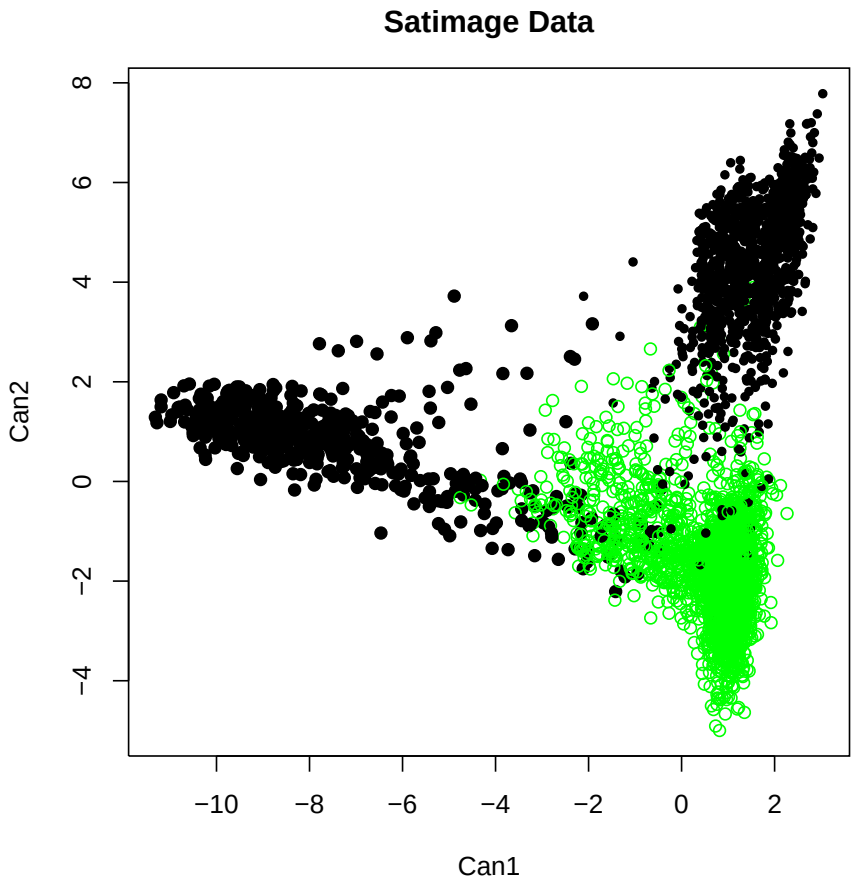
$$\frac{n_1 + n_2}{n_1 n_2}\, \mathbf{a}^{\mathrm{T}} [\sum_{k=1}^{2} n_k (\bar{\mathrm{x}}_k - \bar{\mathrm{x}})(\bar{\mathrm{x}}_k - \bar{\mathrm{x}})^{\mathrm{T}}] \mathbf{a}.$$

For any number of classes g vector a_1 maximizing (??) is called the first canonical direction and $\mathrm{a}_1^{\mathrm{T}} x$ the first canonical variable.

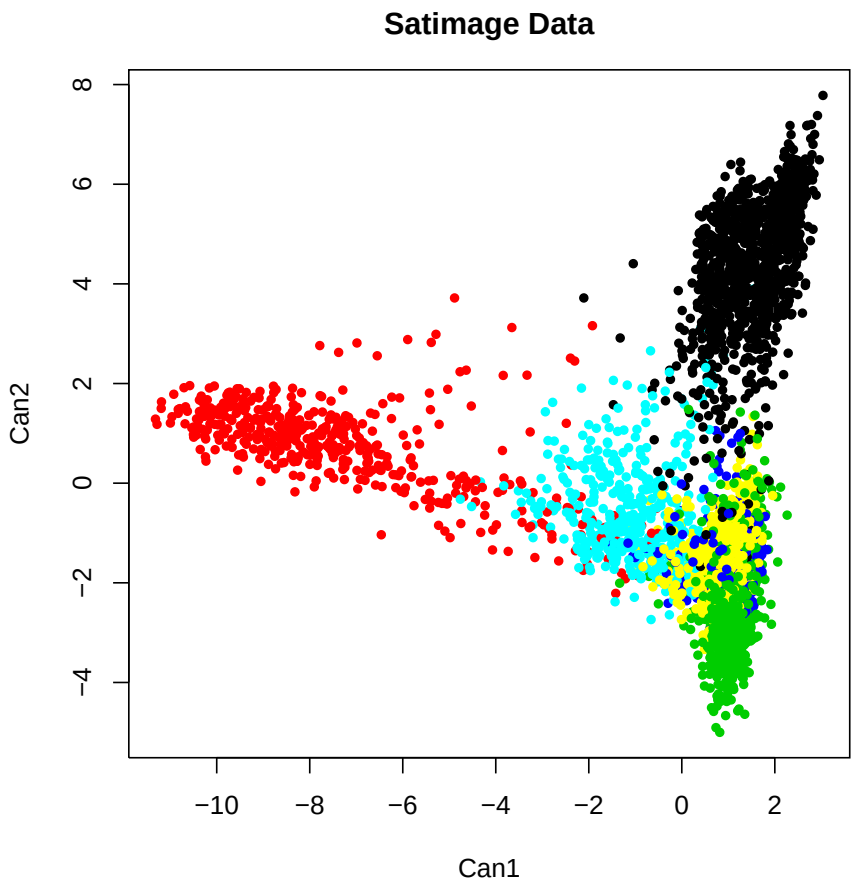
Similarly, the second canonical direction is the direction a_2 such that it maximizes (??) in directions perpendicular to a_1 and so on.

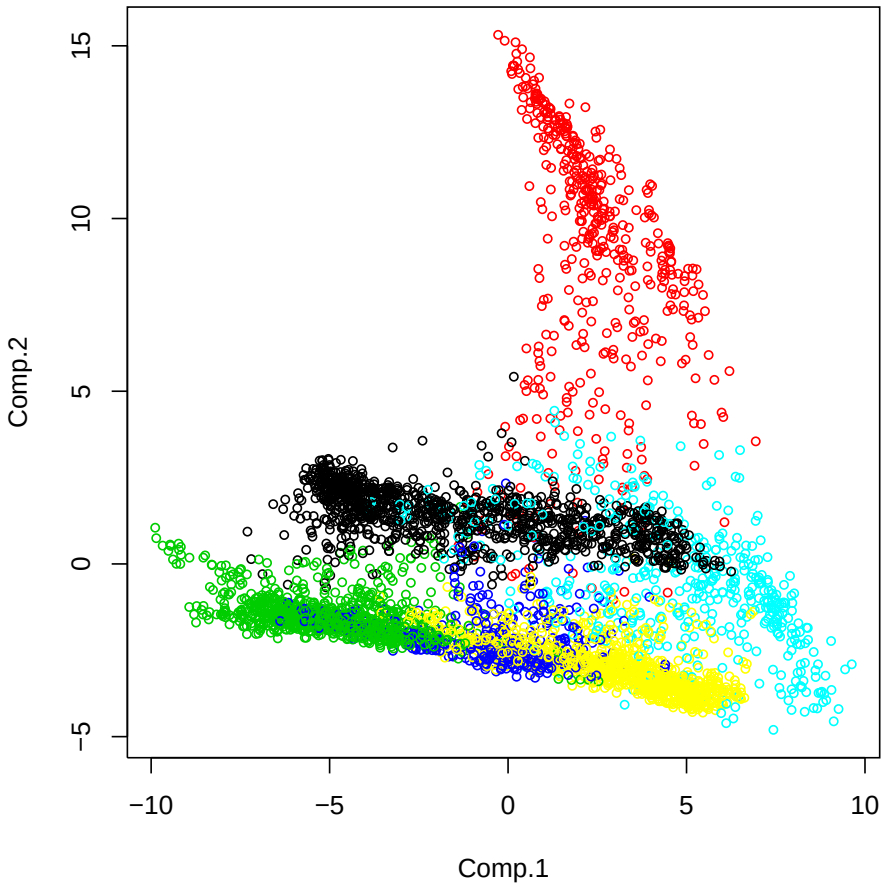
$\mathrm{a}_i^{\mathrm{T}} \mathrm{x}_j$ - score of i^{th} canonical variable on j^{th} observation.

It is useful for visual purposes to view a plot of the first two canonical variables. Satimage data: classification of soils based on satellite images (6 classes, green color corresponds to 4 classes)



Canonical directions should not be confused with principal component directions (for which class indicators play no role).





Scatterplot of scores of the first two canonical variables and the first two principal components.

3. Supervised Classification and Linear Regression

Supervised classification is in fact a problem of estimating a function, albeit with nominal response variable. Moreover, assuming $g = 2$ and coding one class as 0 and the other by 1, $\mathcal{G} = \{0, 1\}$, one obtains

$$E(y|\mathrm{x}) = P(y = 1|\mathrm{x}). \tag{3}$$

Thus, the discriminant analysis problem can be described as that of regression analysis with regression function given by $P(y = 1|\mathrm{x})$.

Let us apply the linear regression model and, once solved, assign new observation x to class 1 if the estimate for $P(y = 1|\mathrm{x})$ is larger than 1/2, and to class 0 otherwise.

This approach can readily be extended to g classes with $g \geq 2$.

Let us denote class labels by row vectors of the form

$$\mathbf{y} = (y^{(1)}, y^{(2)}, \ldots, y^{(g)})$$

with label for class k

$$\mathbf{y} = (0, \ldots, 0, 1, 0, \ldots, 0),$$

where the k**-th coordinate of the indicator vector** $\mathbf{y}$ **is 1. In the matrix form, the sample** $(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n)$ **is described by two matrices,**

$$\mathbf{X}_{(n,p+1)} = \begin{bmatrix} 1 & x_1^{(1)} & x_1^{(2)} & \ldots & x_1^{(p)} \\ 1 & x_2^{(1)} & x_2^{(2)} & \ldots & x_2^{(p)} \\ & \ldots & & \ldots & \\ & \ldots & & \ldots & \\ 1 & x_n^{(1)} & x_n^{(2)} & \ldots & x_n^{(p)} \end{bmatrix},$$

and

$$\mathbf{Y}_{(n,g)} = \begin{bmatrix} y_1^{(1)} & y_1^{(2)} & \ldots & y_1^{(g)} \\ y_2^{(1)} & y_2^{(2)} & \ldots & y_2^{(g)} \\ & \ldots & \ldots & \\ & \ldots & \ldots & \\ y_n^{(1)} & y_n^{(2)} & \ldots & y_n^{(g)} \end{bmatrix}.$$

Our problem consists in building a linear model relating the vector of explanatory variables

$$\mathbf{x} = (x^{(1)}, x^{(2)}, \ldots, x^{(p)})$$

with the vector of response variables

$$\mathbf{y} = (y^{(1)}, y^{(2)}, \ldots, y^{(g)}).$$

This is the multivariate (linear) regression analysis problem (not to be confused with that of multiple regression).

The matrix of unknown parameters, $\hat{\mathbf{B}}_{(p+1,g)}$, is given as the solution of the following problem

$$\min_{\mathbf{B}} \sum_{i=1}^{n} \|\mathbf{y}_i - [1, \mathbf{x}_i]\mathbf{B}\|^2, \tag{4}$$

where $\|\alpha\|^2$ is the squared Euclidean norm.

In this way, the following model is obtained

$$\hat{\mathbf{Y}} = \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y},$$

or, equivalently,

$$\hat{\mathbf{Y}} = \mathbf{X}\hat{\mathbf{B}}, \tag{5}$$

where

$$\hat{\mathbf{B}} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y}. \tag{6}$$

For a new observation $\mathbf{x}$ one gets the response g-vector of the form

$$\hat{\mathbf{y}}(\mathbf{x}) = [1, \mathbf{x}]\hat{\mathbf{B}}. \tag{7}$$

Analogously, as when $g = 2$, we have

$$p(k|\mathbf{x}) = E(y^{(k)}|\mathbf{x}),$$

and thus $\hat{\mathbf{y}}(\mathbf{x})$ is a linear estimator of a posteriori probabilities $p(k|\mathbf{x})$, $k = 1, 2, \ldots, g$.

One can show that for each $\mathbf{x}$,

$$\sum_{k=1}^{g} \hat{y}^{(k)}(\mathbf{x}) = 1. \tag{8}$$

The classification rule for observation x becomes:

- **Choose the class which corresponds to the greatest coordinate of the response vector $\widehat{y}(\mathrm{x})$,**

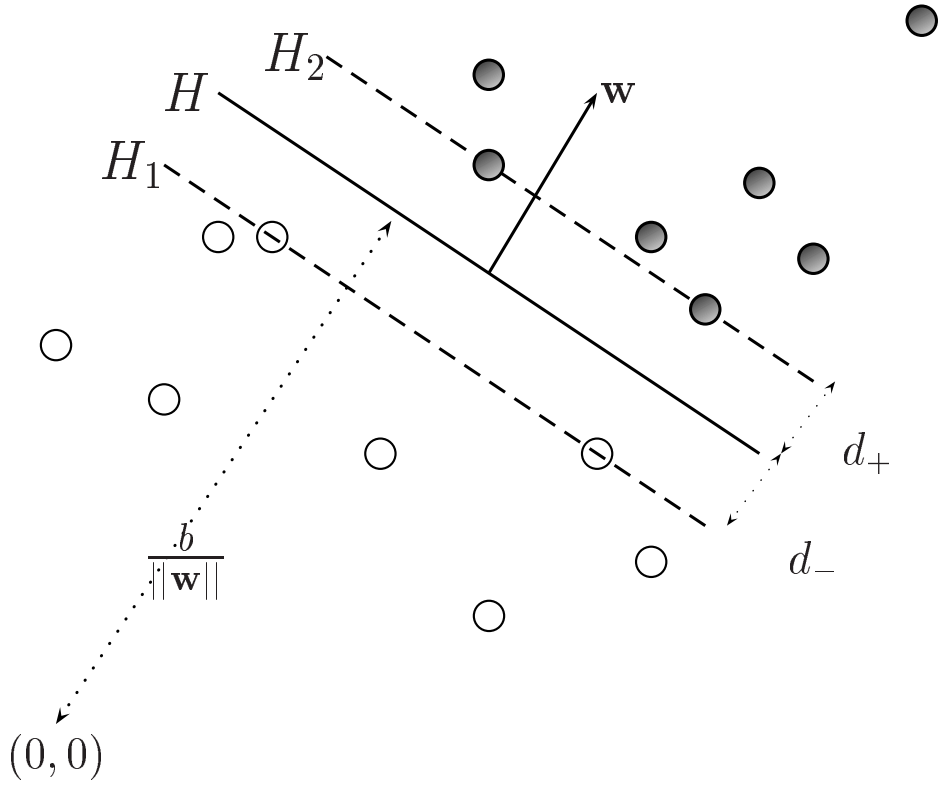
$$\widehat{c}(\mathrm{x}) = \underset{k \in \{1,\ldots,g\}}{\operatorname{argmax}} \ \widehat{y}^{(k)}(\mathrm{x}). \tag{9}$$

For $g = 2$, Fisher's LDA and linear regression DA are strictly related.

Still, the given solution has serious disadvantages. The obvious one is that posterior probabilities are modeled as linear functions. The (much) less obvious one is that a class (or some classes) can be "masked" by others.

4. Linear Support Vector Machine
(a brief remark)

For two linearly separable classes the method relies on construction of two parallel hyperplanes separating the classes with the widest possible margin between them (to be discussed in the sequel in some detail).



Rys. 9.15. Przypadek liniowo separowalny

5. Bayes rule for known distributions

For a moment assume that probability densities $p(\mathrm{x}|k)$ are known for all $k = 1, 2, \ldots, g$ i.e. we know the distribution of attributes in all classes.

For a given vector x one can calculate $p(k|\mathrm{x})$ - conditional probability of class k given the value x of a vector of attributes. $p(k|\mathrm{x})$ is the so called posterior probability of class k given x.

Intuitively, $p(k|\mathrm{x})$ should be large if x comes from class k.

Bayes paradigm:

Classify x to population i such that the value of posterior probability $p(i|\mathrm{x})$ is maximal among $p(1|\mathrm{x}), \ldots, p(g|\mathrm{x})$.

Denote by $\pi_k = P(Y = k)$ i.e. probability that a randomly chosen element belongs to class k (a priori probability of k^{th} class).

Observe that by Bayes theorem

$$p(k|\mathbf{x}) = \frac{\pi_k p(\mathbf{X} = \mathbf{x}|Y = k)}{p(\mathbf{X} = \mathbf{x})} = \frac{\pi_k p(\mathbf{x}|k)}{\sum_{i=1}^{g} \pi_i p(\mathbf{x}|i)}$$

and the denominator does not depend on k. Thus the Bayes rule is equivalent to:

Allocate observation $\mathbf{x}$ to the population for which

$$\pi_k p(\mathbf{x}|k)$$

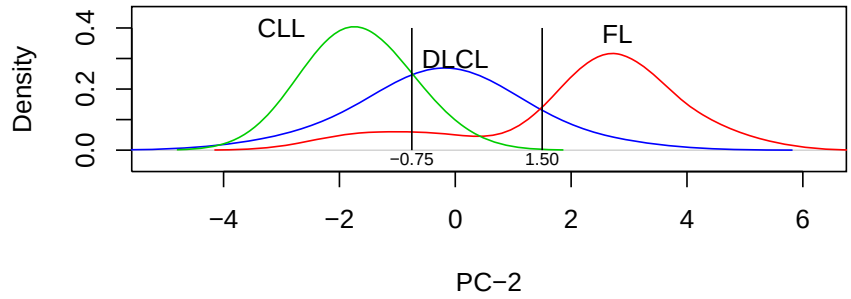
is maximized.

Three-class classification problem: classses CLL, DLCL, FL denote types of disease. Two cases: 1. $\pi_1 = \pi_2 = \pi_3 = 1/3$

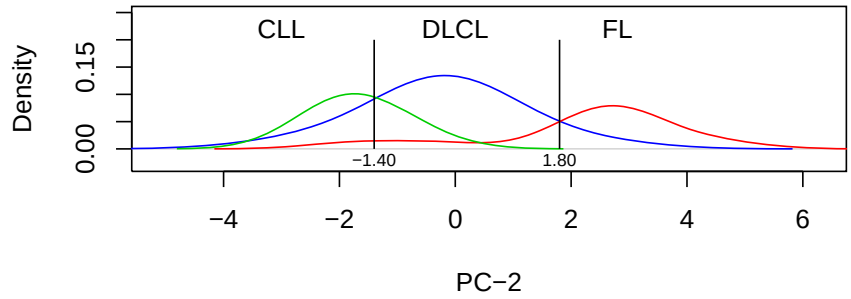
$PC - 2 < -0.75 \Longrightarrow CLL$

2. $\pi_1 = \pi_3 = 0.25, \pi_2 = 0.5$. $PC - 2 < -1.40 \Longrightarrow CLL$

Rys. 2.3a



Rys. 2.3b



Besides its heuristic appeal, Bayes rule yields decision rule with the smallest probability of error !

Let $\widehat{d} : \mathcal{X} \to \{1, \ldots, g\}$ be a classification rule (a classifier). Let

$$\text{Loss}(i, j) = \begin{cases} 0, & i = j \\ 1, & i \neq j \end{cases}$$

be the loss function representing the cost of making decision j when the true class is i.

The risk function for classifier $\widehat{d}$ is the expected loss when using it, as a function of the unknown class k:

$$R(\widehat{d}, k) = E[\text{Loss}(k, \widehat{d}(\mathbf{x}))|\text{class} = k] = P(\widehat{d}(\mathbf{x}) \neq k|\text{class} = k)$$

probability of misclassification (misallocation) of a random vector from class k. The total (or the Bayes) risk is the total expected loss,

$$R(\widehat{d}) = ER(\widehat{d}, k) = \sum_{k=1}^{g} \pi_k R(\widehat{d}, k).$$

Fact. The Bayes rule minimizes the total risk.

What is the relation between the Bayes rule and Fisher's LDA

Fact. If class k is the $\mathcal{N}_p(\mathbf{m}_k, \Sigma)$ population, $k = 1, \ldots, g$, and $\Sigma > 0$, then the Bayes discriminant rule allocates $\mathbf{x}$ to class j, where $j \in \{1, \ldots, g\}$ is that value of k

$$\text{maximizing } g_k(\mathbf{x}) = \mathbf{m}_k^T \Sigma^{-1} \mathbf{x} - \frac{1}{2} \mathbf{m}_k^T \Sigma^{-1} \mathbf{m}_k + \log \pi_k$$

This rule together with its empirical version is called Linear Discriminanat Analysis (LDA) rule.

For $g = 2$ allocate $\mathbf{x}$ to class 1 if

$$(\mathbf{m_1} - \mathbf{m_2})^T \Sigma^{-1} \mathbf{x} - \frac{1}{2} (\mathbf{m_1} - \mathbf{m_2})^{\mathbf{T}} \Sigma^{-1} (\mathbf{m_1} + \mathbf{m_2}) > \log \frac{\pi_2}{\pi_1}$$

and replacing means $\mathbf{m}_i$ and Σ by $\bar{\mathbf{x}}_i$ and $\mathbf{W}$, respectively, we get for $\pi_1 = \pi_2$ exactly Fisher's LDA rule:

Allocate $\mathbf{x}$ to class 1 if

$$(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)^{\mathbf{T}} \Sigma^{-1} \mathbf{x} - \frac{1}{2} (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_2)^{\mathbf{T}} \mathbf{W}^{-1} (\bar{\mathbf{x}}_1 + \bar{\mathbf{x}}_2) > 0$$

If $\pi_1 = \pi_2$ - optimal cut-point midway between projected means;

If $\pi_1 \neq \pi_2$ - optimal cut-point moves toward less likely class.

Fisher's LDA, by its very definition, aims at obtaining a linear discriminant rule. It is, however, constrained by assumption of the same covariance structure in clases. Moreover, it does not take into account that apriori probabilities π_i may be different.

Bays rule allows us to get rid of this assumption and to construct a more flexible discriminant rule in the case of multinormal populations with different covariance matrices and different $\pi_i's$.

Let the model for class k be $\mathcal{N}_p(\mathbf{m}_k, \Sigma_k)$, $k = 1, \ldots, g$. It is easily seen that the Bayes discriminant rule is quadratic:

$$\text{maximize} \quad -\frac{1}{2}\log|\Sigma_k| - \frac{1}{2}(\mathbf{x} - \mathbf{m}_k)^T \Sigma_k^{-1}(\mathbf{x} - \mathbf{m}_k) + \log \pi_k$$

over $k = 1, \ldots, g$.

For $g = 2$ this yields the following rule: allocate to class 2 if

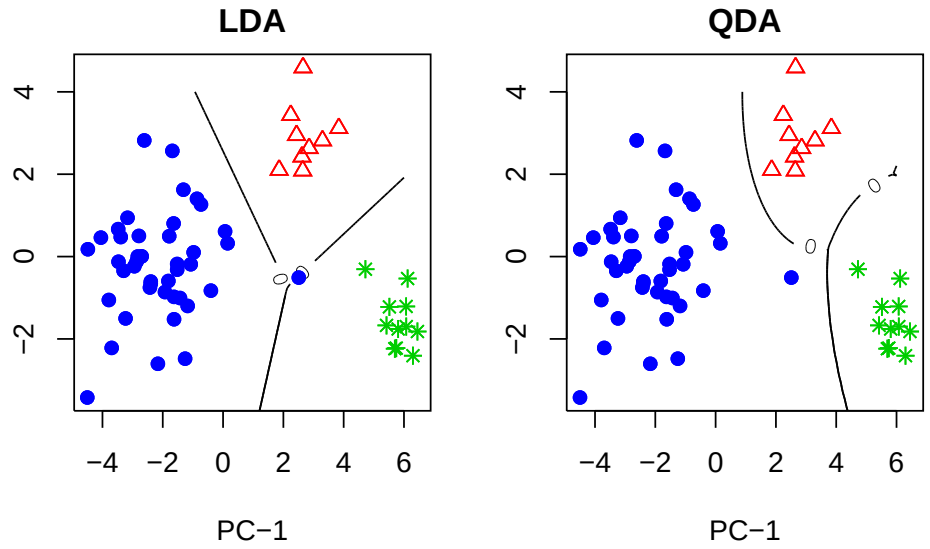
$$\begin{aligned} \log(\pi_2/\pi_1) \;&+\; \frac{1}{2}\log(|\Sigma_1|/|\Sigma_2|) \\ &+\; \mathbf{x}^T(\Sigma_2^{-1}\mathbf{m}_2 - \Sigma_1^{-1}\mathbf{m}_1) - \frac{1}{2}\mathbf{x}^T(\Sigma_2^{-1} - \Sigma_1^{-1})\mathbf{x} \\ &-\; \frac{1}{2}\mathbf{m}_2^T\Sigma_2^{-1}\mathbf{m}_2 + \frac{1}{2}\mathbf{m}_1^T\Sigma_1^{-1}\mathbf{m}_1 > 0 \end{aligned}$$

- Quadratic term $2^{-1}\mathbf{x}^T(\Sigma_2^{-1} - \Sigma_1^{-1})\mathbf{x}$ in discriminant function !

- If the class covariances are the same, the above rule reduces to the linear rule from the previous section.

- If $\pi_1 = \pi_2$ discriminant function does not depend on π_1 or π_2.

QDA rule is obtained by plugging in estimates of means and covariance matrices for all classes.



One should note that, in practice, QDA may prove inferior to LDA even if the class covariances are different.

The plug-in approach is appealing and usually safe provided that n_k, the size of the sample from class k, is large enough (for each k) when compared with the number of measurements (predictor variables) p and the number of classes g. Otherwise, however, it can give poor results.

In particular, QDA requires estimation of $g(p + p(p + 1)/2)$ parameters, while the best linear rule requires estimation of only $gp+p(p+1)/2$ parameters. Thus, unless the sample sizes are very large, estimates needed to perform QDA will have larger variances. Under such circumstances, QDA can very well be outperformed by LDA – although LDA runs the risk of underfitting the data, QDA risks overfitting them.

There is vast empirical evidence that LDA and QDA perform well on an amazingly large set of classification tasks.

Rules provided by LDA and QDA behave stably- they have smaller variance than most competitors. Because of the bias-variance trade-off it is frequently better to put up with some bias of the decision boundary if it can be estimated with much lower variance than more exotic alternatives.

Note that Fisher's rule was derived without any distributional assumptions. LDA should be robust against mild departures from normality. It is indeed the case.

LDA performed on augmented set of features (e.g. with their squares included) yields nonlinear boundaries and performs comparably to QDA.

Concluding this section, let us briefly discuss Bayes rules for problems with different misclassification costs, e.g., such as:

$$\text{Loss}(i,j) = \begin{cases} 0, & i = j \\ l_{ij}, & i \neq j \end{cases}$$

The Bayes rule, which minimizes the total risk, has then the form: allocate $\mathbf{x}$ to class k if

$$\sum_{i=1}^{g} L(i,k)p(i|\mathbf{x}) = \min_{l \in \mathcal{G}} \sum_{i=1}^{g} L(i,l)p(i|\mathbf{x}).$$

This rule minimizes the total risk.

6. Bayesian and the like classification in practice

Logistic Classification

Idea: **Abandon the linear regression model and try a more flexible model of posterior probabilities.**

For the so-called logit function we get a logistic model in which log-odds are linear:

Assuming again $g = 2$ (and classes coded as 1 and 2):

$$\log \frac{p(2|\mathbf{x})}{1 - p(2|\mathbf{x})} = \alpha + \beta^T \mathbf{x},$$

with the inverse

$$p(2|\mathbf{x}) = \frac{\exp(\alpha + \beta^T \mathbf{x})}{1 + \exp(\alpha + \beta^T \mathbf{x})}$$

(As we know well, $\log(v/(1-v))$ is called the logit function and is denoted by logit(v)).

The parameters in the logistic model can be estimated (iteratively!) by maximizing the likelihood function

$$\prod_{i=1}^{n} p(2|\mathbf{x}_i)^{y_i} p(1|\mathbf{x}_i)^{1-y_i},$$

where y_i is the value of the indicator function for class 2 for the ith object.

Classification is done by using the empirical Bayes rule, i.e.:

allocate x to a class 1 if $\widehat{p}(1|\mathrm{x}) \geq \widehat{p}(2|\mathrm{x})$

The approach easily generalizes to the case with more than two classes (it is another matter that the greater is g, the more involved the estimation process becomes). We assume

$$\Theta(k|\mathbf{x}) \equiv \log \frac{p(k|\mathbf{x})}{p(1|\mathbf{x})} = \alpha_k + \beta_k^T \mathbf{x},$$

for $k = 2, \ldots, g$, with the inverses

$$p(k|\mathbf{x}) = \frac{\exp \Theta(k|\mathbf{x})}{\exp \Theta(1|\mathbf{x}) + \ldots + \exp \Theta(g|\mathbf{x})}$$

with $\Theta(1|\mathrm{x}) = 0$.

Classification is done by using the empirical Bayes rule, i.e.:

allocate x to class k for which the value of $\widehat{p}(k|\mathrm{x})$ is maximal.

Remark. Both LDA (but not Fisher's LDA) and logistic classification model imply that

$$\log p(k|\mathbf{x})/p(1|\mathbf{x})$$

is a linear function of the predictors. Thus both methods yield a linear classification boundary. The difference between the methods:

- Logistic classification uses conditional likelihood to estimate β and α - no information on distribution of X is used.

- LDA uses the normality assumption: $(\mathrm{X}|Y = k) \sim N(\mu_k, \Sigma)$.

Logistic discrimination is more appropriate when distribution of X significantly differs from normal or class covariances are significantly different.

Empirical Bayes rules (kernel and nearest neighbor DA)

Bayes rule :

Allocate an observation to the population for which

$$\pi_k p(\mathbf{x}|k)$$

is maximized.

This is a theoretical rule as we neither know π_k nor $p(\mathbf{x}|k)$.

We have to estimate these quantities:

Estimation of π_k: $\widehat{\pi}_k = \frac{n_k}{n}$ where $n_k = \#\{i : Y_i = k\}$. Warning: Appropriate for a random sample drawn from distribution of $(Y, \mathbf{X})$, not for stratified sampling when separate samples are drawn from $(\mathbf{X}|Y = i)$.

Estimation of $p(\mathbf{x}|k)$ is much more complex: we want to estimate density using i.i.d. sample drawn from this density.

Consider $X_1, X_2, \ldots, X_n$ iid sample in R^p pertaining to (continuous) density f.

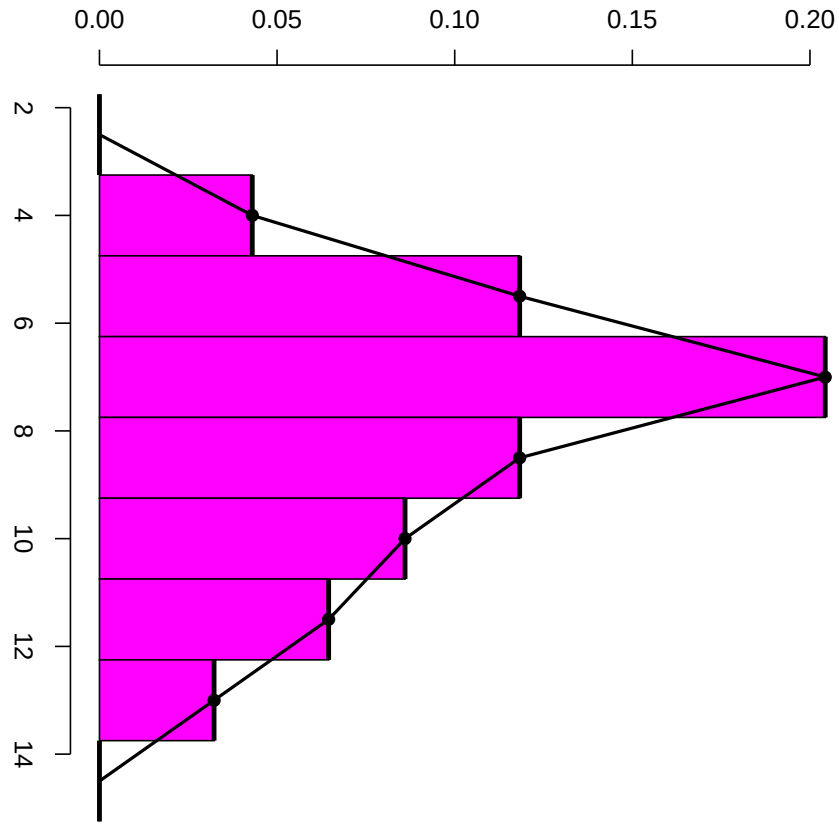
Consider $p = 1$ first. A simple estimate of f is given by a histogram. We consider a sequence of equidistant points (adjacent points differ by h_n)

$$\ldots x_{-1}(n) < x_0(n) < x_1(n) < \ldots$$

such that $x_{i+1}(n) - x_i(n)(n) = h_n$ and for $x \in (x_i(n), x_{i+1}(n)]$

$$\widehat{f}_{hist}(x) = \frac{1}{h_n}(F_n(x_{i+1}(n)) - F_n(x_i(n)),$$

where $F_n(x) = \#\{X_i \leq x\}/n$. The figure shows a typical histogram with its piecewise linear approximation.



It is known that that one gets the best approximation (in probabilistic sense) for midpoints of partition intervals.

Idea: move the histogram together with a point at which we want to estimate a density

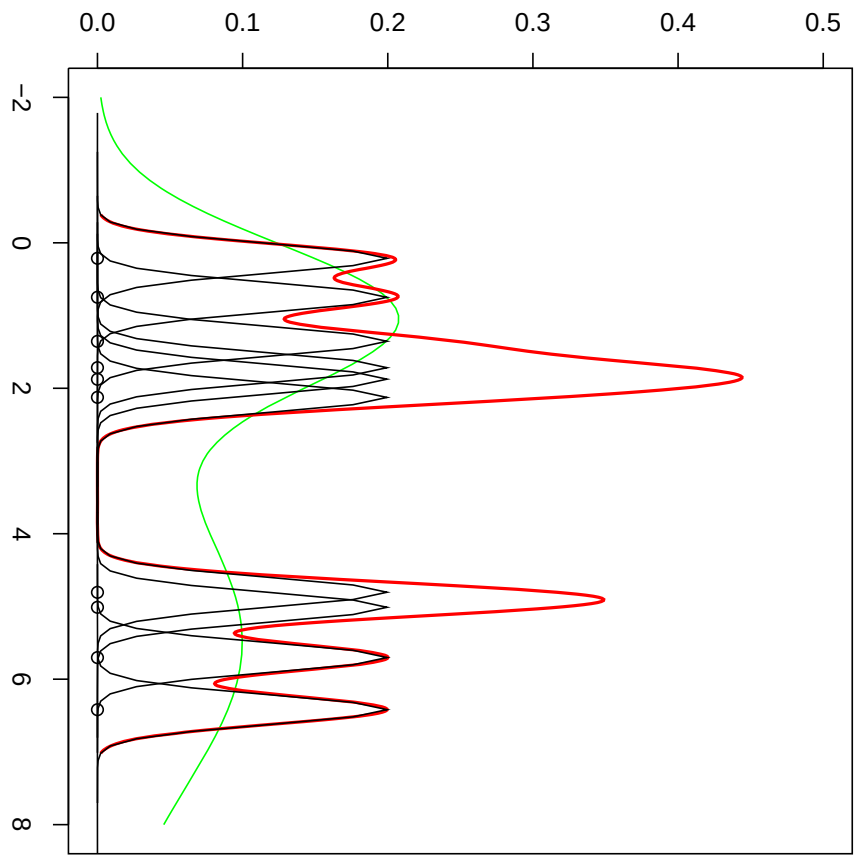
Moving-window histogram

$$\widehat{f}_n(x) = \frac{1}{h_n}(F_n(x + h_n/2) - F_n(x - h_n/2)) = \frac{1}{nh_n}\sum_{i=1}^{n} e\Big(\frac{x - X_i}{h_n}\Big).$$

where $e(t) = 1$ **for** $-0.5 \leq t < 0.5$**, otherwise 0.**

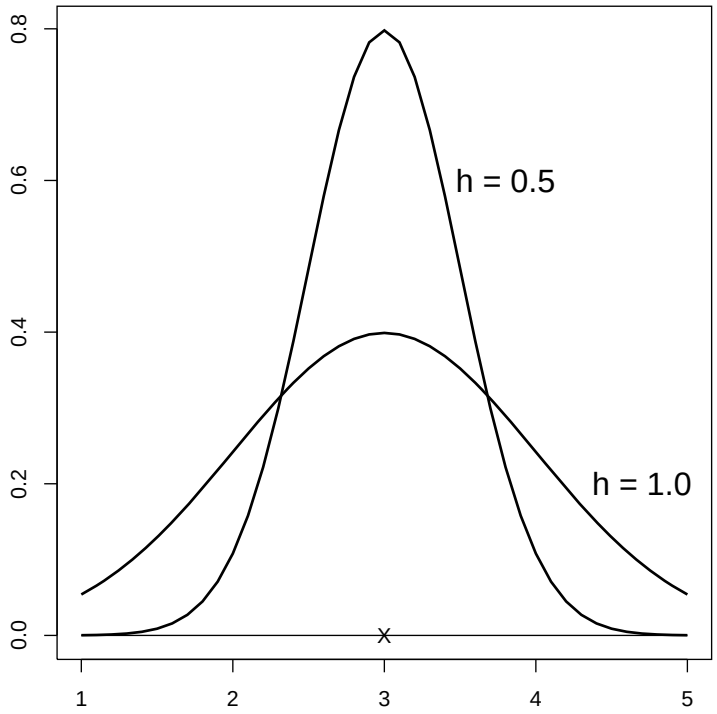
We can choose arbitrary density function K **(kernel) instead of** e**. Rosenblatt-Parzen estimator of** f

$$\widehat{f}_n(x) = \frac{1}{nh_n}\sum_{i=1}^{n} K\Big(\frac{x - X_i}{h_n}\Big).$$



Usually K is taken as the standard normal density or some other smooth symmetric density.

Choice of h_n greatly influences the shape of the estimate obtained.



Usually h_n is chosen as a minimizer of some measure of accuracy of $\widehat{f}$, in particular of $MISE = \int E(\widehat{f}_n(x) - f(x))^2\,dx$.

For normal f we get in this way

$$h_n = (4/3)^{1/5} \min(s_1, s_2) n^{-1/5},$$

where s_1=empirical standard deviation of X_i and

$$s_2 = \text{interquantile range}/1.34$$

is another range-based estimate of σ. This works pretty well for unimodal densities.

Another solution: define h_n in such a way that it depends on the data and point x in such a way that $h_n(x, X_1, X_2, \ldots, X_n)$ is large for x in regions when the data is sparse and is small in the opposite case. The most obvious candidate: Nearest neighbour (NN) distance. We take $k(n) \uparrow$ sequence of integers and define

$$h_n = R_n(x, X_1, X_2, \ldots, X_n)$$

as the distance from x to its $k(n)^{th}$ nearest neighbour.

Kernel estimation of f is readily generalized to p dimensions. The 'only' problem is the curse of dimensionality: for large p we need enormous sample sizes to estimate density in this way. The ways to circumvent it are based on finding interesting low-dimensional projections of f - projection pursuit density estimation.

How does this translate to classification rules?

Empirical Bayes rules are obtained:

Allocate an observation to the population for which $\widehat{\pi}_k \widehat{p}(\mathbf{x}|k)$ is maximized.

For $g = 2$ allocate to population 1 if

$$\frac{n_1}{n}\frac{1}{n_1 h_n}\sum_{i=1}^{n_1} K\Big(\frac{x - X_{i1}}{h_n}\Big) \geq \frac{n_2}{n}\frac{1}{n_2 h_n}\sum_{i=1}^{n_2} K\Big(\frac{x - X_{i2}}{h_n}\Big).$$

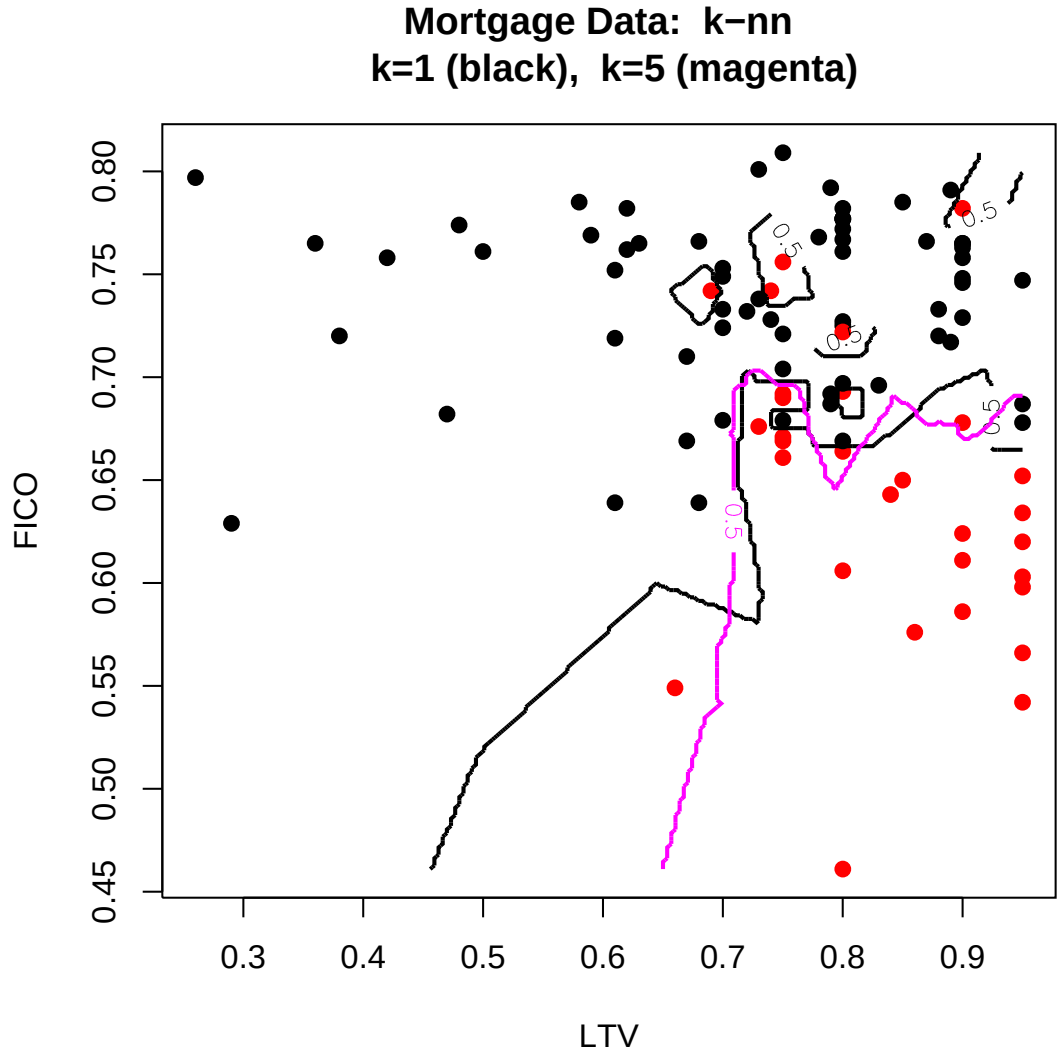
h_n is chosen usually the same for both samples. For $h_n = R_n$ and uniform kernel we get the celebrated kNN rule:

Consider the sphere around x containing exactly $k(n)$ nearest neighbors in the merged sample. Classify x to the class having the most representatives in the neighbourhood. In the case of ties classify arbitrarily to one of the classes with most representatives.

This works usually well even for $k(n) \equiv 1$ when all coordinates of x are quantitative and have comparable variability. Otherwise we have to standardize.

Observe that for 1-NN rule resubstitution estimator of err is always equal to 0, indicating how inadequate this estimator can be.

Example *Mortgage data* **concerns default on house credit payments and its two predictors FICO (Fair Isaac scoring) and LTV (ratio of credit to the value of the house).**



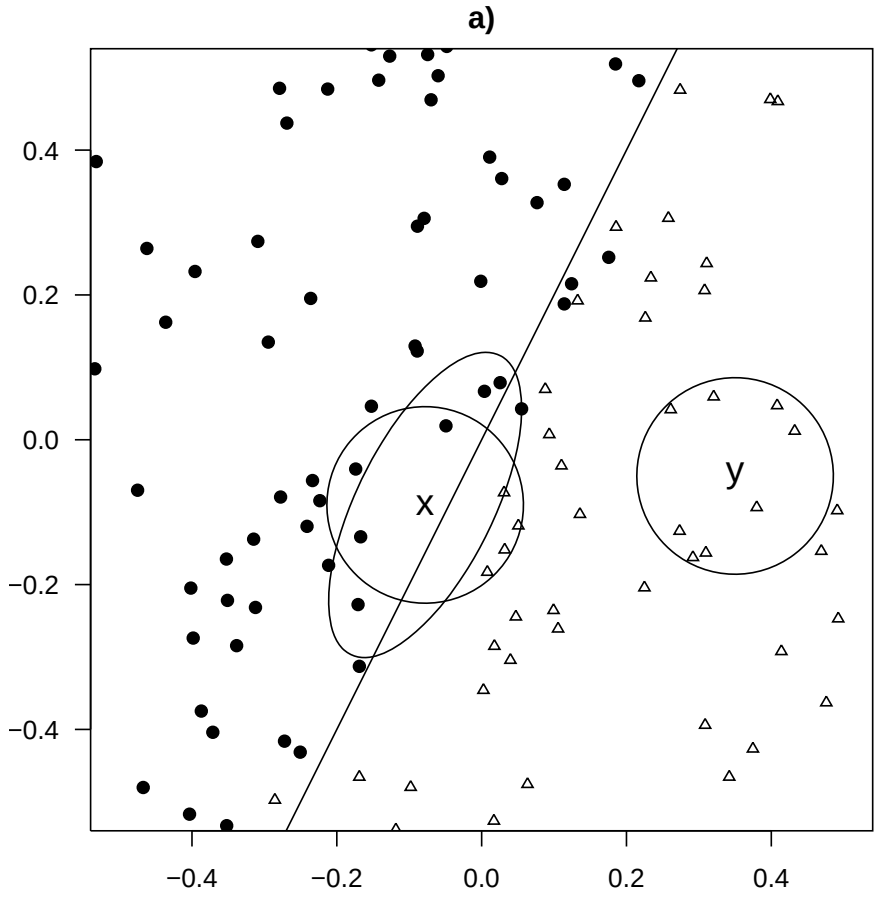
Small 'islands' containing few points indicate overfitting. kNN with $k = 5$ seems to be much less prone to overfitting in this case.

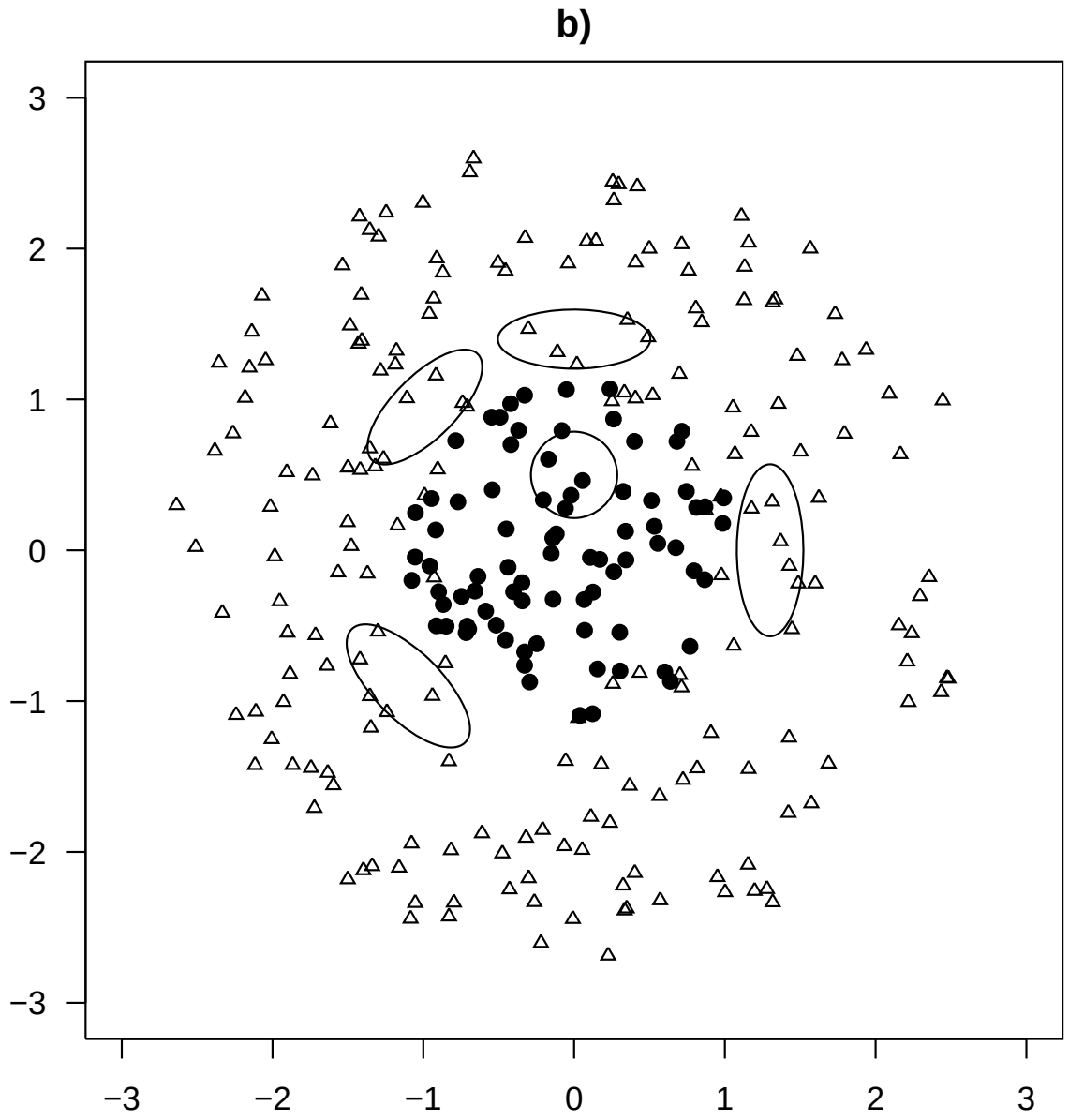
Mixture Discriminant Adaptive Nearest Neighbors (DANN)

Idea: Adapt

$$d(\mathbf{x}, \mathbf{y}) = ((\mathbf{x} - \mathbf{y})' \mathbf{A} (\mathbf{x} - \mathbf{y}))^{1/2}$$

arrordingly.




b)
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3

Mixture Discriminant Analysis

Assume that each of the k, $k = 1, 2, \ldots, g$, classes is decribed by a mixture normal distribution with s_k compnents,

$$p(\mathbf{x}|k) = \sum_{r=1}^{s_k} p_{kr}\phi(\mathbf{x}; \mu_{kr}, \Sigma_{kr}),$$

where

$$0 < p_{kr} < 1 \quad \text{i} \quad \sum_{r=1}^{s_k} p_{kr} = 1,$$

and $\phi(\mathbf{x}; \mu_{kr}, \Sigma_{kr})$ is a p-dimensional normal density with mean μ_{kr} and covariance matrix Σ_{kr}.

Usually, it is also assumed that $\Sigma_{kr} = \Sigma$, $r = 1, 2, \ldots, s_k$, $k = 1, 2, \ldots, g$.

Parameters of each mixture distribution with known s_k are usually estimated via EM algorithm; cf [HTF]. If s_k is not known in advance, it can be found experimentally, or other approaches can be used.

Prototype methods

Training data is represented by a set of points (prototypes) in feature space. Classification of a query point x is made to the class of the closest prototype. 'Closest' is usually defined by Euclidean distance after standardization of each predictor.

Crucial: prototypes should be well positioned to capture the distribution of each class.

Drawback: LVQ is not based on optimization of some criterion what makes it difficult to understand its properties.

Example: Learning Vector Quantization (LVQ1)

1. Choose R initial prototypes for each class: $m_1(j), \ldots, m_R(j)$, $j = 1, \ldots, g$, e.g. by random sampling.

2. Sample a training point x_i (with replacement) and let (k, j) be the index of the closest prototype $m_k(j)$ to x_i . Now,

(a) if $y_i = j$ (i.e., they are in the same class) move the prototype towards the training point.

$$m_k(j) \longleftarrow m_k(j) + \varepsilon(\mathbf{x}_i - m_k(j)),$$

where ε is positive learning rate;

(b) if $y_i \neq j$ (i.e., they are in different classes) move the prototype away the training point.

$$m_k(j) \longleftarrow m_k(j) - \varepsilon(\mathbf{x}_i - m_k(j)).$$

3. Repeat step 2 decreasing the learning rate ε with each iteration to 0.

Log-linear and location models

The former is used for multinomially distributed discrete data when a parsimonious model is required. We shall present the log-linear model for data from only one class, assuming additionally that the data are two-dimensional ($p = 2$; generalization to $p > 2$ is straightforward). Say, $x^{(1)}$ assumes one of m_1 values, and $x^{(2)}$ one of m_2 values. Writing probabilities $P_{ij} = P(x^{(1)} = i, x^{(2)} = j)$ as $P_1, \ldots, P_{m_1 m_2}$ we get a multinomial distribution

$$P(n_1, n_2, \ldots, n_m) = \frac{n!}{n_1! n_2! \ldots n_m!} P_1^{n_1} P_2^{n_2} \ldots P_m^{n_m},$$

where $m_1 m_2 = m$. Assuming independence of $x^{(1)}$ and $x^{(2)}$, and using obvious notations, we easily obtain

$$\ln \mu_{ij} = \ln \mu_{i.} + \ln \mu_{.j} - \ln n.$$

Denoting

$$\theta = \frac{1}{m_1 m_2} \sum_{i=1}^{m_1} \sum_{j=1}^{m_2} \ln \mu_{ij},$$

$$\alpha_i = \frac{1}{m_2} \sum_{j=1}^{m_2} \ln \mu_{ij} - \theta,$$

$i = 1, 2, \ldots, m_1$**, and**

$$\beta_j = \frac{1}{m_1} \sum_{i=1}^{m_1} \ln \mu_{ij} - \theta,$$

$j = 1, 2, \ldots, m_2$**, yields (after some manipulations)**

$$\ln \mu_{ij} = \theta + \alpha_i + \beta_j,$$

with

$$\sum_{i=1}^{m_1} \alpha_i = 0 \quad \textbf{oraz} \quad \sum_{j=1}^{m_2} \beta_j = 0.$$

The model obtained has only $1 + (m_1 - 1) + (m_2 - 1)$ **independent parametrs,** θ**,** α_i **and** β_j**.**

If we include interactions, we get

$$\ln \mu_{ij} = \theta + \alpha_i + \beta_j + \gamma_{ij},$$

with

$$\sum_{i=1}^{m_1} \gamma_{ij} = 0,$$

$j = 1, 2, \ldots, m_2$**, and**

$$\sum_{j=1}^{m_2} \gamma_{ij} = 0,$$

$i = 1, 2, \ldots, m_1$**. We have** $(m_1 - 1)(m_2 - 1)$ **independent parameters** γ_{ij}**. We also know that**

$$\sum_i \mu_{i\cdot} = \sum_j \mu_{\cdot j} = n.$$

Hence, for two-dimensional data, the full log-linear model is equivalent to the multinomial model. Of course, we have greater flexibility when choosing a log-linear model for data of greater dimension - the greater dimension, the greater the flexibility.

Location model is often used when we deal with discrete-continuous data. Say, our observations come from g classes, each observation described by a c-dimensional multinomial distribution and a d-dimensional normal distribution.

Then, for class k, $k = 1, 2, \ldots, g$, the joint distribution of getting l-th value from the multinomial distribution,

$$l = 1, 2, \ldots, m_1 m_2 \cdots m_c$$

and $\mathbf{v}$ from d-dimensional normal distribution is given by the density

$$\frac{P_{kl}}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{v} - \mu_k^{(l)})'\Sigma_k^{-1}(\mathbf{v} - \mu_k^{(l)})\right).$$

7. Performance assessment of a classifier

The main measure of performance of a classification rule $\hat{d}$ is actual error rate (called also conditional error rate)

$$err = P(\hat{d}(X) \neq Y|\mathcal{X}),$$

where $\mathcal{X} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ is a training sample on which classification rule $\hat{d}$ is based and (X, Y) is a new element to be classified independent of training sample. This is the total risk for 0-1 loss function introduced earlier.

Error rate err is conditional on the training sample on which our rule is based - and it should be ! In practice we are interested in performance of a classification rule constructed on a given sample.

Another measure of performance is unconditional error rate (expected error rate) $E(err) = R(\hat{d})$, where expectation is taken over all possible training samples.

Note that err is not directly observed and we have to estimate it.

Estimates of err

(i) Resubstitution which provides resubstitution or apparent error rate, obtained by reusing the training set:

$$\frac{1}{n}\sum_{i=1}^{n} I(\widehat{d}(\mathbf{x}_i) \neq y_i)$$

Estimates obtained are clearly overly optimistic (or biased in statistical terminology): they yield too small estimate of err as the classifier is tested on the *same* sample from which it was constructed. Thus we can expect relatively more misclassifications on an independent test set.

(ii) Using a test set to obtain an estimate of the error rate.

We split available sample consisting of n elements into e.g. two halves (training and test sample). Classifier $\widehat{d}_{[n/2]}$ is constructed based on the training sample , and its error is estimated using the test sample:

$$\frac{1}{[n/2]} \sum_{i:(x_i,y_i)\in \text{test sample}} I(\widehat{d}(\mathbf{x}_i) \neq y_i)$$

drawbacks:

– depends on the split (usually random);

– we asses the performance of $\widehat{d}_{[n/2]}$, when we would like to use $\widehat{d}_n$....

(iii) Main method: cross-validation. It has the following versions:

- Rotation: say $K = 10$ mutually exclusive subsets are defined, each one being used in turn as a test set for the classifier built on the remainder. K obtained estimators are averaged. This is called $CV-K$.

- Leave-one-out: a single observation is used as a test set for the classifier built on the other $n-1$; this is repeated n times. (jacknife is a similar method)

- Bootstrap: a random subset of size equal to the complete data set is taken, with replacement, for use as the training set, and the remaning data set is used as the test set. Error rate is averaged over bootstrap samples (usually 25-100). Call it $\widehat{err}_{boot}$.

- **Bootstrap 0.632 estimate. Based on the observation that $\widehat{err}_{boot}$ is a pessimistic estimate of err since a bootstrap sample contains on average approx. $1 - e^{-1}$ different elements only. We combine it with analogous (but optimistic!) estimate $\widehat{err}_{boot-opt}$ when the whole sample is taken as a test sample:**

$$\widehat{err} = 0.632\widehat{err}_{boot} + 0.368\widehat{err}_{boot-opt}.$$

This estimator is known to usually have a smaller bias than $\widehat{err}_{boot}$ but its modification, proposed by Efron and Tibshirani and called bootstrap 0.632+ estimator, has been found to be more reliable.

8. Choice of a classifier based on performance assessment

Till now only one clssifier was considered. Let $\widehat{d}_1, \ldots, \widehat{d}_k$- be k classifiers. We would like to choose the one having the smallest error rate.

Original sample is usually split into three parts:

- training sample (50-60% of all observations);

- validation sample used to estimate error rate when selecting a classifier (20-25%);

- test sample used to asess performance of the chosen classifier (20-25%).

Another possibility is to use crossvalidation to choose the 'best' classifier and to asess its performance.

Error rates provide just one type of performance assessment. They describe classifier's performance by only one summary statistic, whose use is justified only when all types of misclassifications are equally serious.

Confusion (classification) matrix

The confusion matrix provides more information about a classifier. It is a matrix with elements

$$e_{ij} = P\{\text{decision } j \mid \text{class } i\},$$

$i, j = 1, \ldots, g$**, or a matrix with elements** $\tilde{e}_{ij} = e_{ij}\pi_i$**. It is easily seen that, in the second case, the sum of the matrix elements falling off the leading diagonal gives the error rate.**

For $g = 2$ e_{ij} are often interpreted in terms of errors comitted when testing $H_0 : y = 1$ (non-disease) against $H_1 : y = 2$ (disease).

$$e_{22} = P\{\text{decision 2} \mid \text{class 2}\} = \mathbf{sensitivity} = \frac{TP}{TP + FN}$$

(probability of predicting disease given true state is disease)

$$e_{11} = P\{\text{decision 1} \mid \text{class 1}\} = \mathbf{specificity} = \frac{TN}{TN + FP}$$

(probability of predicting non-disease given true state is non-disease)

Clearly,

$$\frac{TN}{TN + FP} = 1 - \frac{FP}{TN + FP}$$

Thus

$$\text{sensitivity} = \text{power of the test} = \beta$$
$$\text{specficity} = \text{1-type I error of the test} = 1 - \alpha$$

Example

Classification of healthy individuals ($y = 1$) and ill individuals ($y = 2$), $n_1 = 200, n_2 = 100$.

	person diagnosed as healthy	person diagnosed as ill
healthy	TN	FP
ill	FN	TP

	person diagnosed as healthy	person diagnosed as ill
healthy	176	24
ill	3	97

In the example, sensitivity = 97/100=0.97 and specificity= 176/200=0.88

$$\widehat{err} = \frac{FP + FN}{TN + FP + FN + TP} = 0.09$$

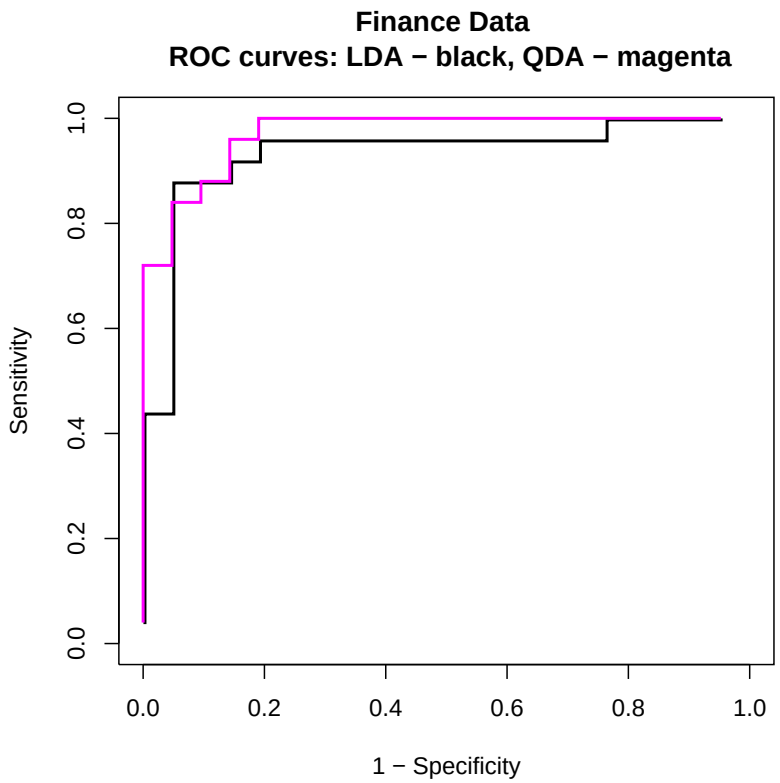
What happens to sensitivity and specificity when we change the threshold in a classification rule ?

We get ROC curve

ROC= Receiver Operating Characteristic

ROC curve is a plot of sensitivity against 1-specificity (type 1 error).

One can compare two classification rules also by comparing their ROC curves. Preferable situation: ROC curve as close to the sides emanating from north-western corner of the square as possible. In what follows we show ROC curves for LDA and QDA applied to the Pima Indians Diabetes Data.



QDA and LDA ROC curves for example data. QDA ROC curve dominates that of LDA for most of the thresholds.

Since we have only 2 classes, the Bayes rule (for different misclassification costs) reads: allocate x to class 2 if

$$l_{21}p(2|\mathbf{x}) > l_{12}(1-p(2|\mathbf{x})), \text{ i.e. } p(2|\mathbf{x}) > \frac{l_{12}}{l_{12}+l_{21}}.$$