Business Intelligence
SS 2018

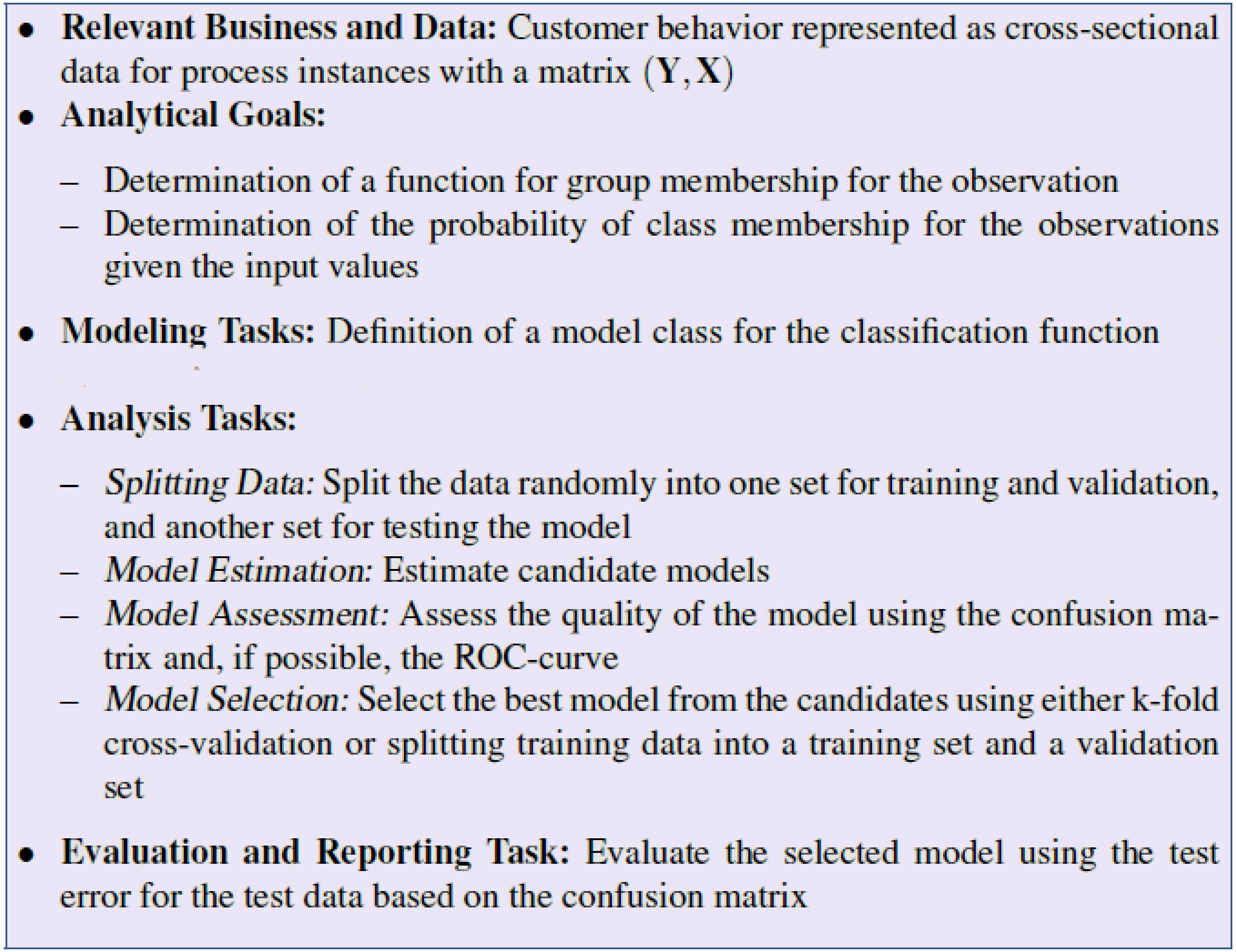
Cross-sectional Analysis 2
Classification Methods

W. Grossmann

Content

- Classification Methodology
- Modell Class: Bayes Classification
- Model Class: Logistic Regression
- Model Class: Classification Trees
- Model Class: Nearest neighbor Classification
- Model Class: Support Vector Machines
- Model Class: Combination Methods

Classification Methodology



- **Relevant Business and Data:** Customer behavior represented as cross-sectional data for process instances with a matrix $(\mathbf{Y}, \mathbf{X})$
- **Analytical Goals:**
  - Determination of a function for group membership for the observation
  - Determination of the probability of class membership for the observations given the input values
- **Modeling Tasks:** Definition of a model class for the classification function
- **Analysis Tasks:**
  - *Splitting Data:* Split the data randomly into one set for training and validation, and another set for testing the model
  - *Model Estimation:* Estimate candidate models
  - *Model Assessment:* Assess the quality of the model using the confusion matrix and, if possible, the ROC-curve
  - *Model Selection:* Select the best model from the candidates using either k-fold cross-validation or splitting training data into a training set and a validation set
- **Evaluation and Reporting Task:** Evaluate the selected model using the test error for the test data based on the confusion matrix

Classification Methodology

- In the first step “Relevant Business and Data” descriptive analysis of the variables should be done (cf. the example for regression of car prices):
  – Considerations about the data generating process (quality)
  – Type of explanatory variables X: (categorical-qualitative or numeric-quantitative)
  – Frequency distribution of the numeric variables by summary measures, histograms, considerations whether normality of the distributions is an issue
  – Identification of outliers
  – Missing values

Classification Methodology

  - Tables for frequencies of categorical variables
  - Correlations for numeric variables, scatterplot matrix
  - Grouped box plots for numeric variables with categorical explanatory variables and with the class identifier Y
  - Tables for frequencies of categorical variables and visualization by bar charts, mosaic plots
  - Tests about the dependency of the numeric variables on the categorical variables
- Short summary of the findings

Model Class: Bayes Classification

- Basic problem formulation for two groups
  - Given $N=n+m$ training data from two classes labeled by $Y = 0$ and $Y = 1$

$$Group1: (\vec{x}_1, 0), (\vec{x}_2, 0) \ldots, (\vec{x}_n, 0)$$

$$Group2: (\vec{x}_{n+1}, 1), (\vec{x}_{n+2}, 1) \ldots, (\vec{x}_{n+m}, 1)$$

  - For the two groups we know prior probabilities

$$P(Y=0) = p_0, \quad P(Y=1) = p_1$$

  - We assume that the attributes x have a certain distribution within the groups with probaiblity densities

$$p(\vec{x} \mid 0), \ \ p(\vec{x} \mid 1)$$

Model Class: Bayes Classification

- If we know the priors and the probabilities in the groups we can define the posterior probabilities of the groups given the data by

$$P(Y=g\,|\,\vec{x}) = \frac{p(\vec{x}\,|\,g)\,p(g)}{p(\vec{x})} \quad g=0,1$$

- Decision rule: Assign a new observation to the class for which $P(Y=g\,|\,\vec{x}_{new})$ is maximal (most probably class)

Model Class: Bayes Classification

- Operational reformulation:

$$\hat{y} = \begin{cases} 0 & if \ \dfrac{p(\vec{x}_{new} \mid 0)\, p(0)}{p(\vec{x}_{new} \mid 1)\, p(1)} \geq 1 \\ 1 & f \ \dfrac{p(\vec{x}_{new} \mid 0)\, p(0)}{p(\vec{x}_{new} \mid 1)\, p(1)} < 1 \end{cases}$$

- This rule is optimal with respect to the misclassification rate

Model Class: Bayes Classification

- Generalization to more than two classes by deciding for the class with maximal posterior probability
- Generalization for problems with different costs of misclassification by changing the threshold 1 to another value
- Major problem in application:
  - Learning the prior probabilities
  - Learning the distribution of the attributes in the classes

Model Class: Bayes Classification

- Learning prior probabilities:
  - Usually taken from the size of the samples within the groups
- Learning of the probabilities $p(\vec{x} \mid g)$ in the groups is much more difficult
  - Assumption of normal distribution is in most practical cases not justified (quantitative and qualitative attributes)
  - Usually we have a large number of attributes and a common distribution is difficult to estimate

Model Class: Bayes Classification

- In order to overcome these problems the naïve Bayes approach is used
- Basic idea behind naïve Bayes:
  - Assume that the attributes are independent
  - Estimate for each attribute individually the distributions in the classes
  - Calculate the probabilities for all attributes by multiplication

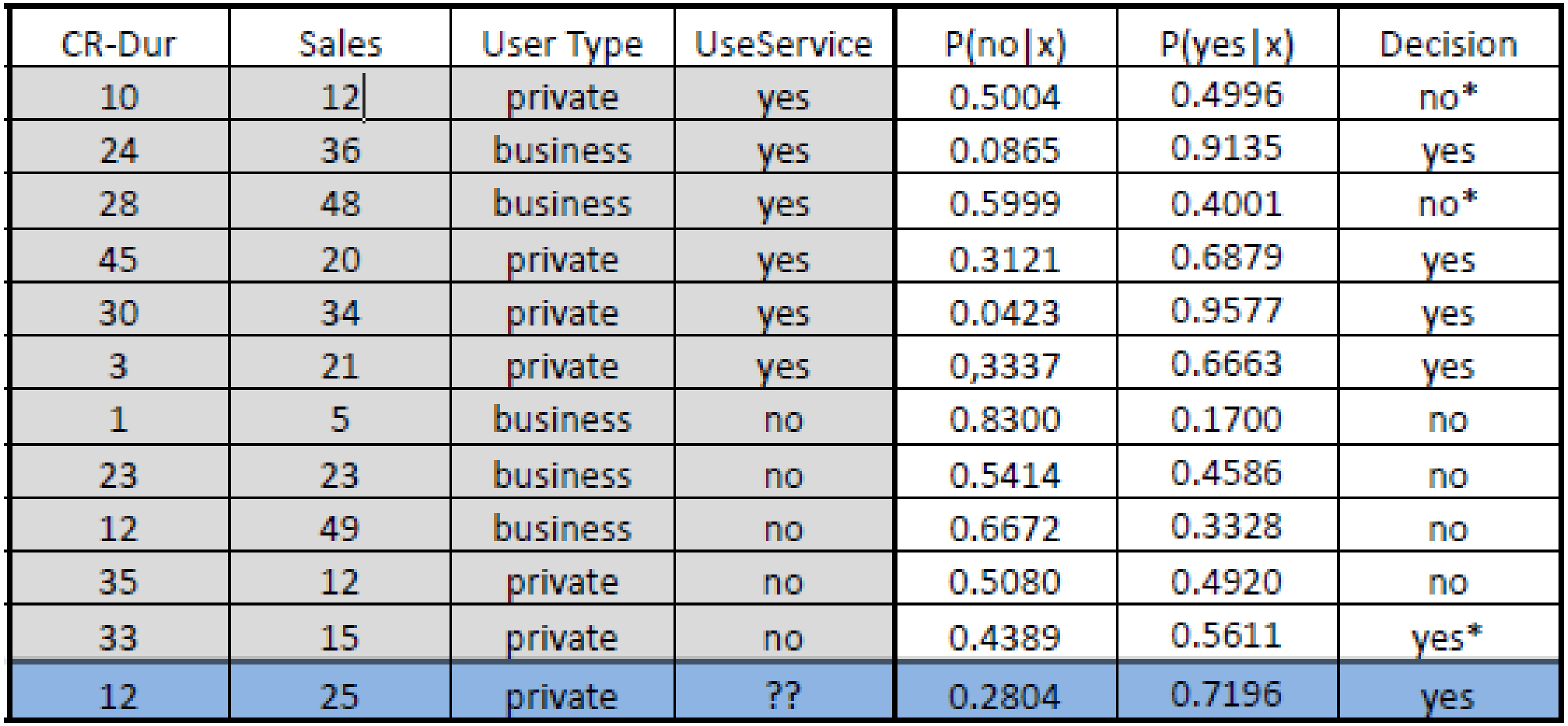
$$\hat{p}(\vec{x} \mid g) = \hat{p}_1(x_1 \mid g) \cdot \hat{p}_2(x_2 \mid g) \cdot \ldots \cdot \hat{p}_k(x_k \mid g)$$

Model Class: Bayes Classification

- For the individual attributes we can use in case of qualitative attributes frequencies in the training data
- In case of quantitative attributes one can use either a defined distribution or density estimation
- Example: Given for 11 customers the duration of the relationship to the company, the sales volume, and the type of usage, we want to learn whether the customer uses a certain service

Bayes Classification

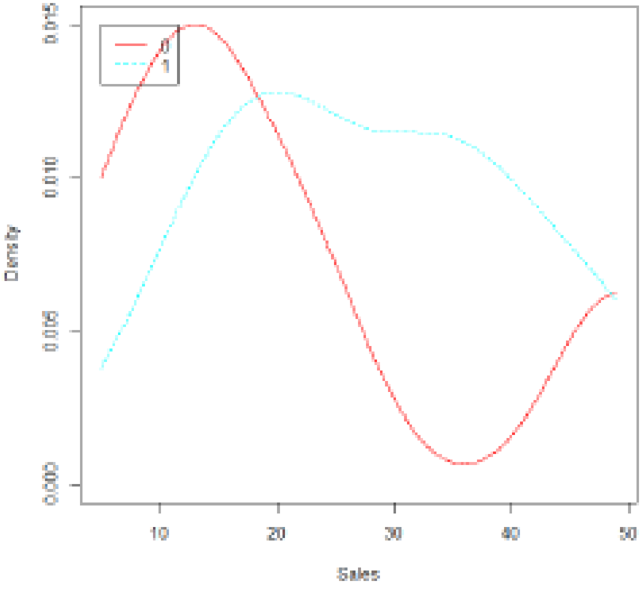
– Data (grey area)

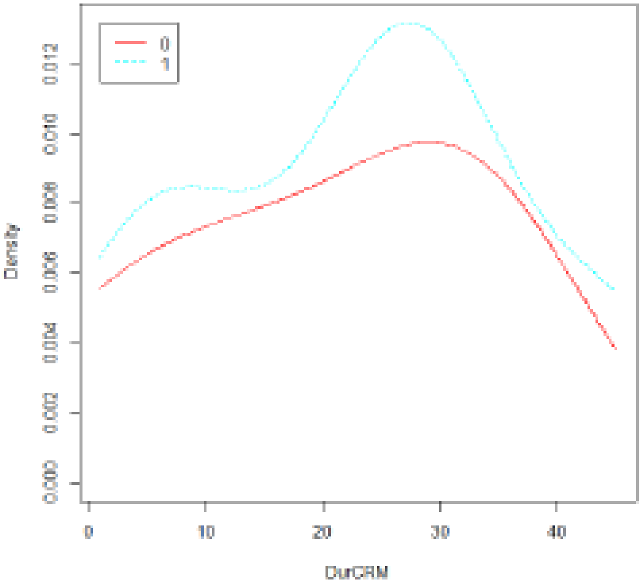


CR-Dur	Sales	User Type	UseService	P(no|x)	P(yes|x)	Decision
10	12	private	yes	0.5004	0.4996	no*
24	36	business	yes	0.0865	0.9135	yes
28	48	business	yes	0.5999	0.4001	no*
45	20	private	yes	0.3121	0.6879	yes
30	34	private	yes	0.0423	0.9577	yes
3	21	private	yes	0,3337	0.6663	yes
1	5	business	no	0.8300	0.1700	no
23	23	business	no	0.5414	0.4586	no
12	49	business	no	0.6672	0.3328	no
35	12	private	no	0.5080	0.4920	no
33	15	private	no	0.4389	0.5611	yes*
12	25	private	??	0.2804	0.7196	yes

Model Class: Bayes Classification

- Our goal is to assign the last customer to one of the two groups (using the service or not)
- Estimation of the probability densities of duration and sales in the two groups:





Model Class: Bayes Classification

  - Results are shown in the right part of the table slide 13
- Advantage of naïve Bayes:
  - Rather simple calculation
  - We obtain rule for classification and probailities
  - Simple adaptation to different misclassification costs
- Disadvantage: no variable selection
- Successful application: spam filter

Model Class: Bayes Classification

- Basic problem formulation for two groups:
  - Given N=n+m training data from two classes labeled by $Y\,=\,0$ and $Y\,=\,\;1$

$$Group1:(\vec{x}_1,0),(\vec{x}_2,0)\ldots,(\vec{x}_n,0)$$
$$Group2:(\vec{x}_1,1),(\vec{x}_2,1)\ldots,(\vec{x}_m,1)$$

  - Our interest is finding a model for the probability that an observation belongs to the group $Y=1$ in dependence of the available k attributes

$$P(Y=1)=P(x_1,x_2,\ldots,x_k)$$

Model Class: Logistic Regression

- For obtaining such a model we use the logarithms of the odds for belonging to the second group:

$$logit = \ln\left(\frac{P(Y=1)}{P(y=0)}\right) = \ln\left(\frac{p}{1-p}\right)$$

- The assumption is that the dependence of the logit from the attributes is a linear function:

$$logit = \beta_0 + \beta_1 x_1 + \ldots + \beta_k x_k$$

Model Class: Logistic Regression

- We interpret now the results for group membership in the training sample as results of a binomial distribution
- This allows application of a statistical model well known under the heading logistic regression
  - From the training data the parameters can be estimated using the method of maximum likelihood, i.e., find the values of $(\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_k)$ which maximize the observed class memberships in the training data

Model Class: Logistic Regression

- After estimation of the coefficients we calculate the probabilities for the second group given attribute values $\vec{x}$ according to the formula

$$p(\vec{x}) = \frac{\exp(\hat{\beta}_0 + \hat{\beta}_1 x_1 + \ldots + \hat{\beta}_k x_k)}{1 + \exp(\hat{\beta}_0 + \hat{\beta}_1 x_1 + \ldots + \hat{\beta}_k x_k)}$$

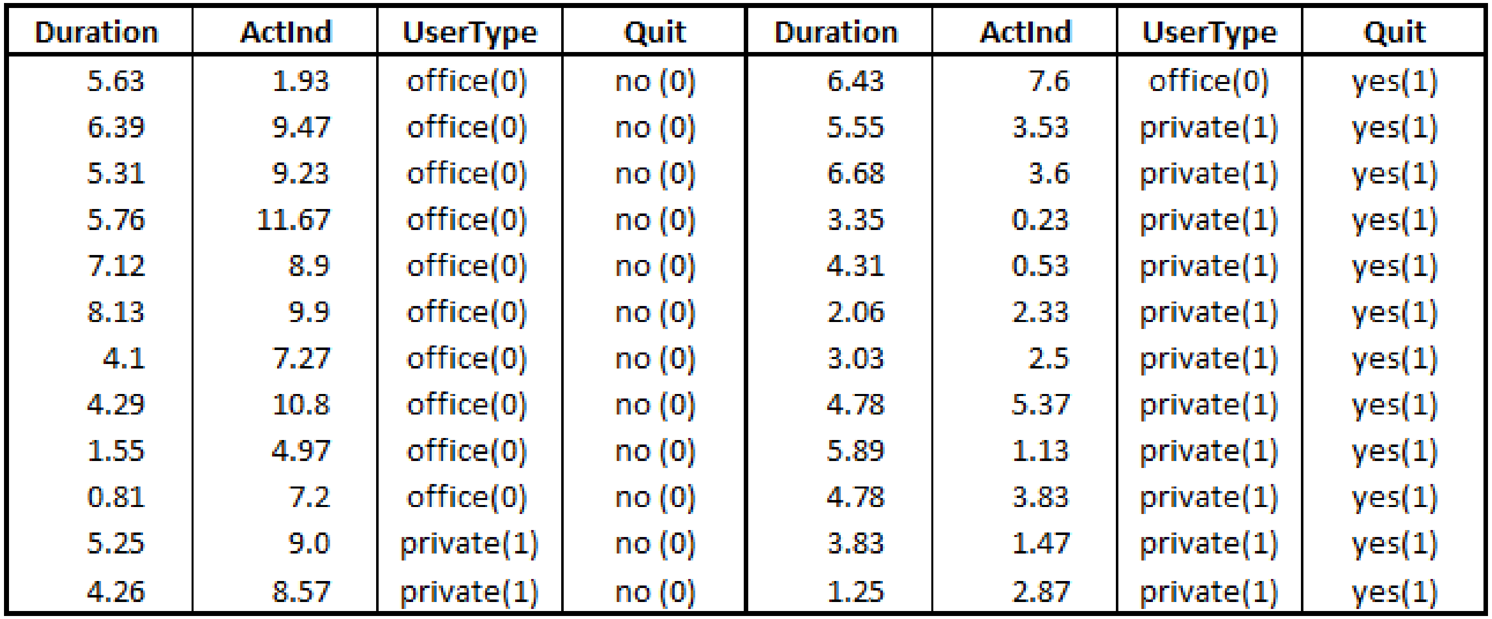
- The decision is defined by

$$\hat{y} = \begin{cases} 1 & if \;\; p(\hat{x}_{new}) \geq tr \\ 0 & f \;\; p(\hat{x}_{new}) < tr \end{cases}$$

tr defines a threshold (standard case $tr = 0.5$)

Model Class: Logistic Regression

- Example: data about churn risk of 24 customers



Duration	ActInd	UserType	Quit	Duration	ActInd	UserType	Quit
5.63	1.93	office(0)	no (0)	6.43	7.6	office(0)	yes(1)
6.39	9.47	office(0)	no (0)	5.55	3.53	private(1)	yes(1)
5.31	9.23	office(0)	no (0)	6.68	3.6	private(1)	yes(1)
5.76	11.67	office(0)	no (0)	3.35	0.23	private(1)	yes(1)
7.12	8.9	office(0)	no (0)	4.31	0.53	private(1)	yes(1)
8.13	9.9	office(0)	no (0)	2.06	2.33	private(1)	yes(1)
4.1	7.27	office(0)	no (0)	3.03	2.5	private(1)	yes(1)
4.29	10.8	office(0)	no (0)	4.78	5.37	private(1)	yes(1)
1.55	4.97	office(0)	no (0)	5.89	1.13	private(1)	yes(1)
0.81	7.2	office(0)	no (0)	4.78	3.83	private(1)	yes(1)
5.25	9.0	private(1)	no (0)	3.83	1.47	private(1)	yes(1)
4.26	8.57	private(1)	no (0)	1.25	2.87	private(1)	yes(1)

Model Class: Logistic Regression

– Parameter estimates for the logistic model

$$logit(quit) = 1.385 + 3.058 \cdot UserType - 0.577\, ActInd$$

– Interpretation:

– Risk for churning for private users is $exp(3.058)=21.3$ times the risk of office users

– Increase of activity index by 1 unit decreases the risk of churning by a factor $exp(- 0.577)=0.56$

Model Class: Logistic Regression

- Advantages of the logistic model
  - Procedures for automatic selection of important attributes can be used similar to regression
  - Interpretation of the results as risk
  - We obtain probabilities as well as class memberships
  - Adaptation to different costs using different thresholds

Model Class: Logistic Regression

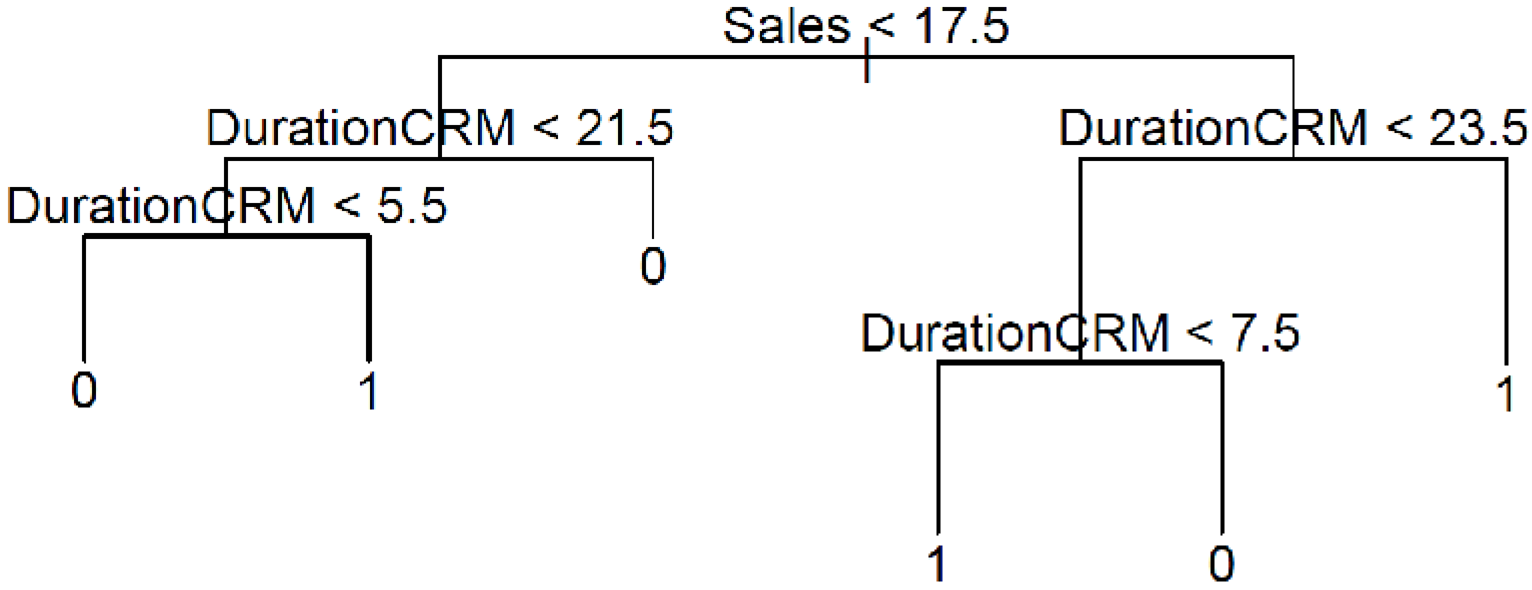
- Generalization to classification with more than two groups
  - Usually done by comparing all pairs of classes and use a majority vote, i.e., the class which is selected most frequently for a case is used
  - Alternative: classification one versus the rest and assign the case to the class with highest probability
- Typical applications: churn management, credit risk

Model Class: Classification Trees

- Basic problem formulation for tree classifiers
  - Given a training sample composed from k different classes we want to apply a sequence of questions about properties of the attributes which are answered either by “yes” or “no”
  - The sequence of questions defines a binary tree
  - The leaf nodes of the tree correspond to a decision about class membership

Model Class: Classification Trees

Example: Given the information from 11 customers about usage (1 = yes, 0 = no) of a certain service, find a rule which allows predict of usage for customer 12 (cf. naïve Bayes)



Model Class: Classification Trees

- Main tasks for finding the classification tree:
  - Find splitting rules: For each node formulate a rule which attribute has to be selected for the next split and which threshold has to be used
  - Find pruning rules: In order to avoid complex trees which overfit the training data define rules which allow pruning the tree
- The most frequently used method for tree classification is CART (Classification and Regression Trees)

Model Class: Classification Trees

- Splitting Rules (CART)
  - Basic for splitting is a definition of the node impurity $Q(t)$, i.e., a numerical value which measures the mixture of different classes in the node t defined by the relative frequencies $\hat{p}(j\,|\,t)$ of the classes in the node
  - Two specifications: Gini index and Entropy

$$Q(t) = \begin{cases} \sum\sum \hat{p}(i\,|\,t)\cdot\hat{p}(j\,|\,t) = 1 - \sum \hat{p}(j\,|\,t)^2 & \textit{Gini index} \\ -\sum \hat{p}(j\,|\,t)\ln\left(\hat{p}(j\,|\,t)\right) & \textit{Entropy} \end{cases}$$

Model Class: Classification Trees

- For a variable the split is defined by the conditions

$$X < tr \vee X > tr \quad (\mathit{quantitative\ variables})$$
$$X = a_k \vee X \neq a_k \quad (\mathit{qualitative\ variables})$$

- The variable used for split is that one with minimal impurity measure (greedy search)
- Splitting stops if the node has impurity 0 or a small number of cases

Model Class: Classification Trees

- Pruning the tree
  - Pruning of the tree is based on a criterion which takes into account the complexity of the tree by penalization of the empirical risk with the number of nodes $|T|$

$$R_{emp}(\alpha) = R_{emp}(\alpha) + \alpha \cdot |T|$$

  - It can be shown that only a final number of penalization parameters α have to be checked and that there exists a unique tree which minimizes the empirical risk
  - The final tree is afterward defined by crossvalidation from the candidate trees

Model Class: Classification Trees

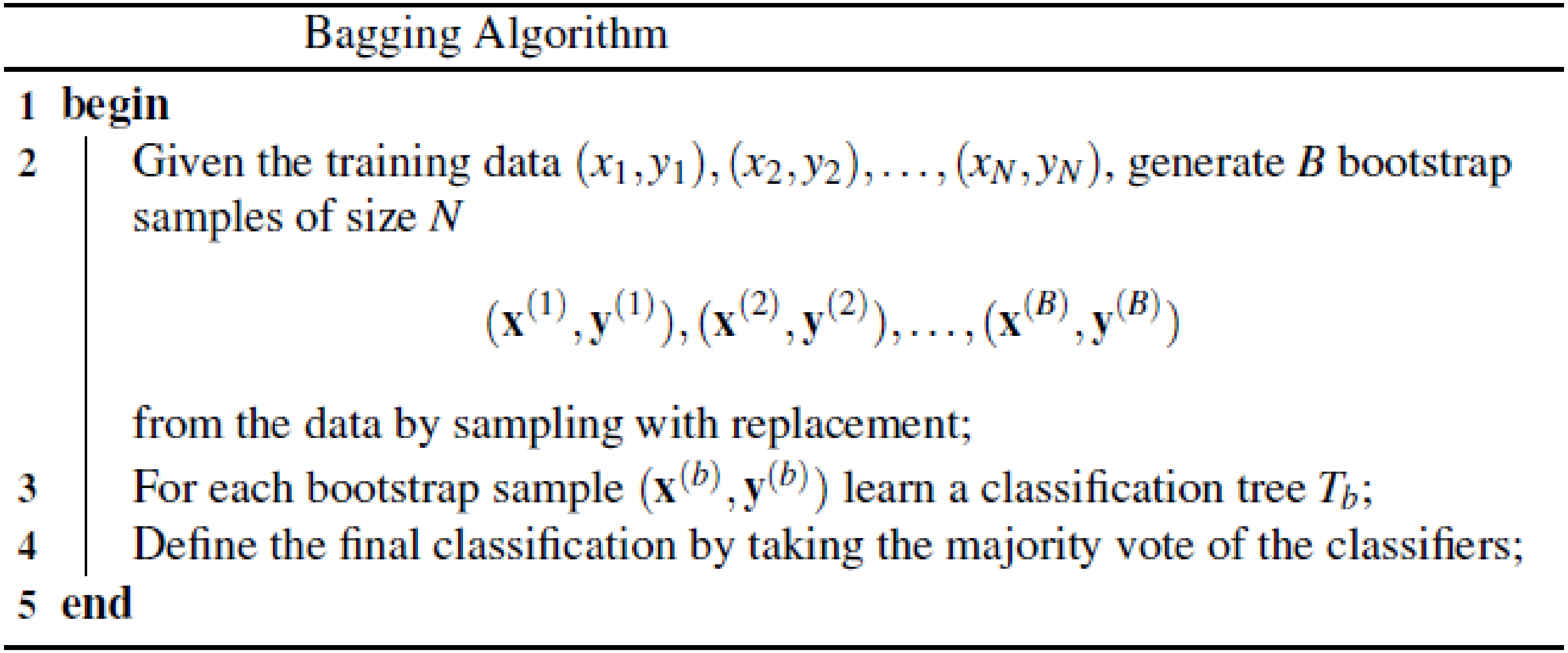
- Advantages of CART
  - The procedure can be generalized to problems with different weights for misclassification costs
  - Missing values can be handle within the procedure by so called surrogate splits (alternative variable if the information for split is missing)
  - Applicable to an arbitrary number of classes
  - Estimation of class probabilities is rather simple

Model Class: Classification Trees

- Disadvantages of CART
  - The procedure is sensitive to the ordering of the variables
  - Dependency on training data
  - Splits do not take the dependency of the variables into account
- In general CART is one of the most popular methods for classification

Model Class: Classification Trees

- A generalization of CART to overcome the disadvantages is Bagging (Bootstrap Aggregation)



```
Bagging Algorithm
begin
    Given the training data $(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$, generate $B$ bootstrap
    samples of size $N$

        $$(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), (\mathbf{x}^{(2)}, \mathbf{y}^{(2)}), \ldots, (\mathbf{x}^{(B)}, \mathbf{y}^{(B)})$$

    from the data by sampling with replacement;
    For each bootstrap sample $(\mathbf{x}^{(b)}, \mathbf{y}^{(b)})$ learn a classification tree $T_b$;
    Define the final classification by taking the majority vote of the classifiers;
end
```

Model Class: Classification Trees

- Another generalization are Random Forests
  - Similar to Bagging but in each Bootstrap sample only a random subset of the attributes is used
- Other methods for tree classification
  - C4.5, C5: allows trees with more than two child nodes
  - CHAID (Chi-Square Automatic Interaction Detection): partitioning tables from qualitative variables

Model Class: Nearest Neighbor Classification

- Basic problem formulation for k-nearest neighbor classifiers
  - Given the training data from k classes
  - Define a distance between the observations based on the attributes
  - Find for a new observation $\vec{x}_{new}$ the k observations in the training sample which are closest to the new observation
  - Assign to the new observation the class which has the majority in the k nearest neighbors

Model Class: Nearest Neighbor Classification

- Main computational problems
  - Definition of the distance between the observations
    - In case of quantitative variables the distance is usually the Euclidean distance
    - In case of binary variables the Hamming distance is frequently used
    - Combination of qualitative and quantitative variables needs special consideration. In R the function `daisy` allows such a calculation
  - Choice of k: The simplest case is $k = 1$, many times $k = 5$ is recommended

Model Class: Nearest Neighbor Classification

- Advantages of nearest neighbor classifiers.
  - Easy to calculate
  - From theoretical point of view close to Bayes classifiers
  - No explicit learning step (instance based learning, lazy learning)
- Disadvantages
  - Sensitive to the local structure of the data (outliers)
  - No variable selection

Model class: Support Vector Machines

- Basic problem formulation for support vector machines in case of two classes labeled by –1 and +1
  - Training data from two classes:

$$(\vec{x}_i, y_i),\ \ y_i \in \{-1, 1\},\ \ i = 1,2,\ldots,n$$

  - We want to separate the two classes by a linear function (hyperplane)

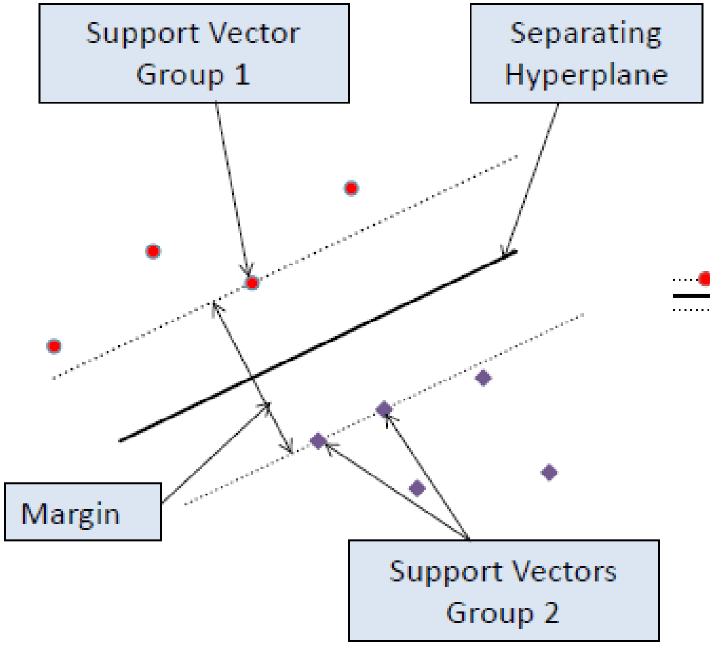
$$\vec{w}^t \vec{x} = b$$

  - The quality of the separation is measured by the so called margin , i.e., the minimum distance of points in the classes from the hyperplane

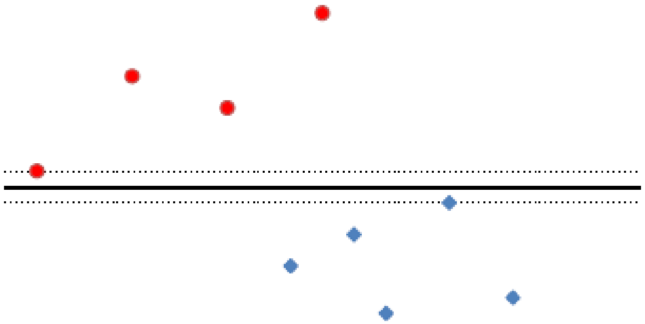
Model class: Support Vector Machines

- Points with smallest distance are called support vectors
- Visualization of the margin in case of two attributes

good separation



bad separation



Model class: Support Vector Machines

  - We want to find a linear function which makes the idea of good separation in the previous example precise

- Main problems in solving this task
  - Is it always possible to separate the classes?
  - How can we calculate the best separation?

Model class: Support Vector Machines

- Vapnik-Chernovenkis dimension answers the first question
  - How many points can be separated in k dimensions by a hyperplane, independent from the position?
  - In two dimensions obviously only three points
  - Vapnik Chernovenkis dimension of a class of functions (VC-dimension): The maximum number of points in any configuration, which can be separated by a function in the class
  - In case of linear functions in k dimensions the VC-dimension is $k + 1$

Model class: Support Vector Machines

- Development of the algorithm for finding the best separating hyperplane is based on three ideas
  - First idea: solution in case of possible separation of the points
    - Solution obtained by a quadratic minimization problem: Find a vector $\vec{w}$ and a constant b solving the optimization problem

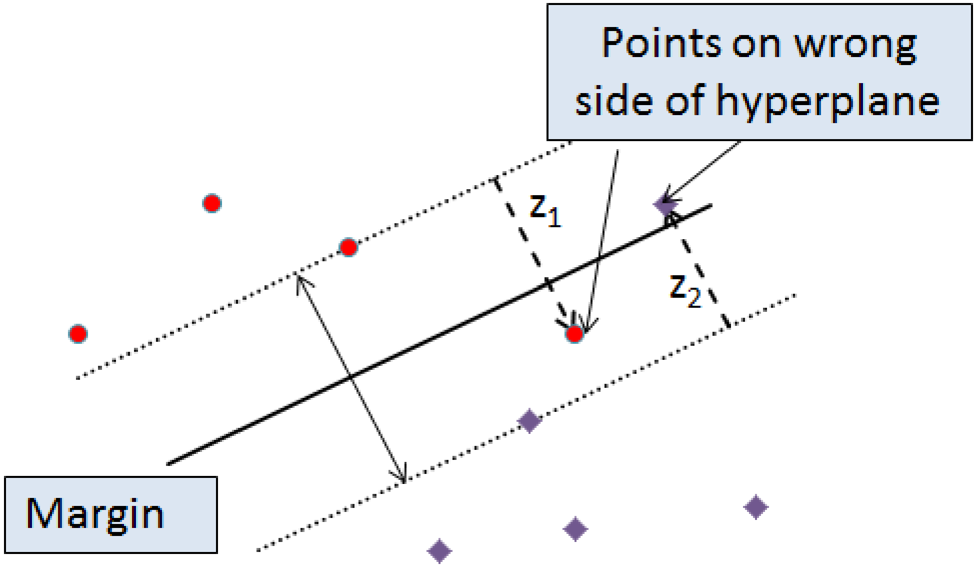
$$\min \frac{1}{2} \| \vec{w} \|$$

$$y_i \vec{w}^t \vec{x} + b \geq 1, \; i = 1, 2, \ldots, n$$

    - This is a high dimensional quadratic optimization problem

Model class: Support Vector Machines

- Second idea: allow not perfect separation by penalization of points which are misclassified by a so called soft margin



Model class: Support Vector Machines

- Solution obtained by a quadratic minimization problem with penalization terms: Find a vector $\vec{w}$ a constant b and penalty terms ξ_i solving the optimozation problem

$$\min \frac{1}{2} \| \vec{w} \| + C \sum \xi_i$$

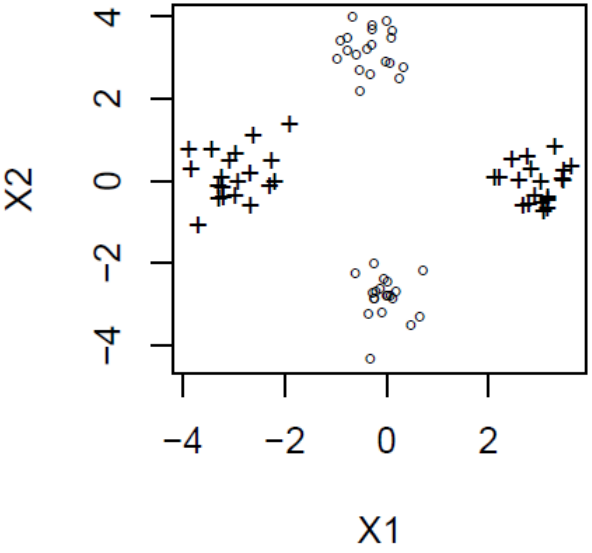
$$y_i \vec{w}^t \vec{x} + b \geq 1 - \xi_i,\ \xi_i \geq 0\ ,\ i = 1,2,\ldots,n$$

- Here C is a parameter measuring the trade off between separation and classification

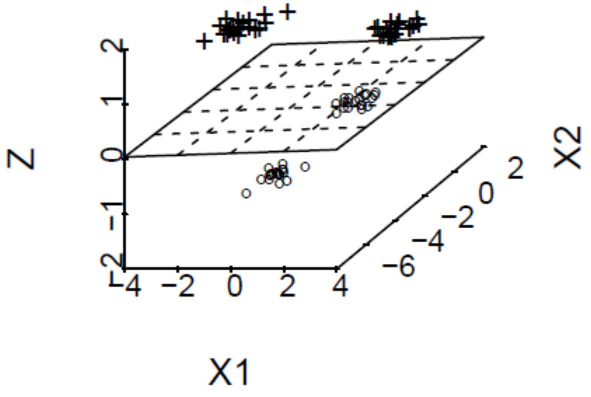
Model class: Support Vector Machines

- Third idea: Transform the problem in a new more complex space with higher VC-dimension with respect to linear separation functions
- Example: XOR-problem

Data



Transformed Data



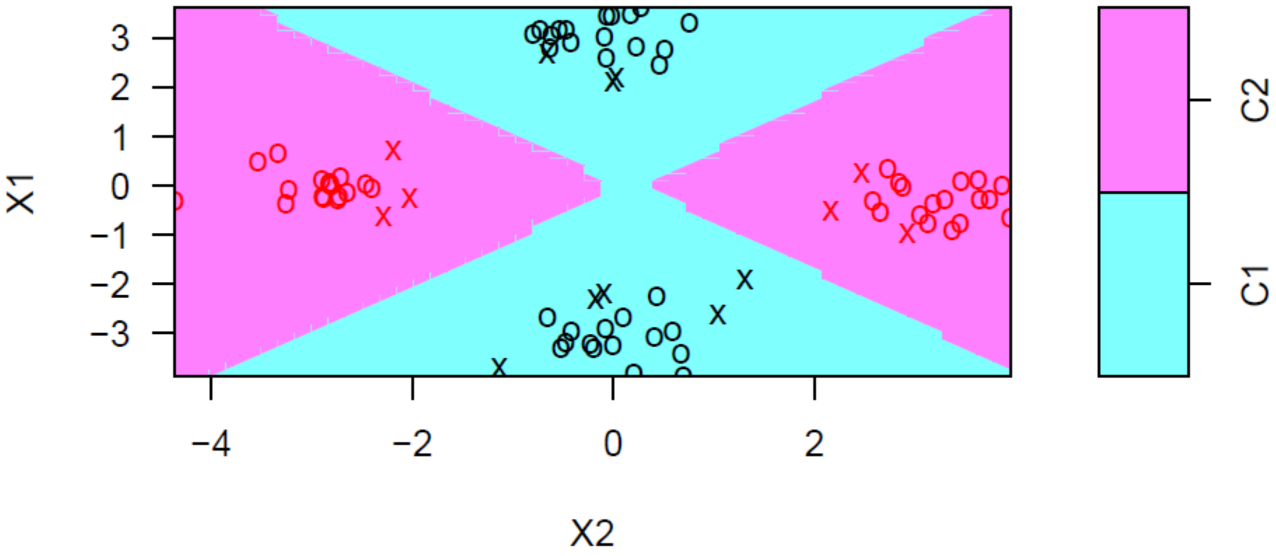
Model class: Support Vector Machines

- This idea is realized by the so called kernel trick: we do not transform the problem explicit but use a kernel function which allows the calculation of inner products and distances in the new space
- The most frequently used kernel function for transformation of standard problems is the radial basis kernel

$$H(\vec{x}, \vec{x}_1) = \exp\left\{ \frac{-\| \vec{x} - \vec{x}_1 \|^2}{\sigma^2} \right\}$$

Model class: Support Vector Machines

- Example: Solving an XOR problem
  - Visualization of the solution with radial basis kernel in the two dimensional space (crosses mark support vectors)



Model class: Support Vector Machines

- Properties of support vector machines
  - The solution is from theoretical point of view of great interest because it minimizes directly the empirical risk
  - The VC dimension allows theoretical estimation of the generalization error
  - The kernel trick allows application to not numeric data, e.g., classification of graphs (graph kernels), or classification of text data (string kernels)
  - Contrary to logistic regression or trees the solution is usually different to interpret

Model class: Support Vector Machines

- The probability for class assignment has to be calculated separately, e.g., in R logistic regression is used
- Classification of more than two classes is mostly done by comparing all pairs of classes and taking the majority vote

Model class: Combination Methods

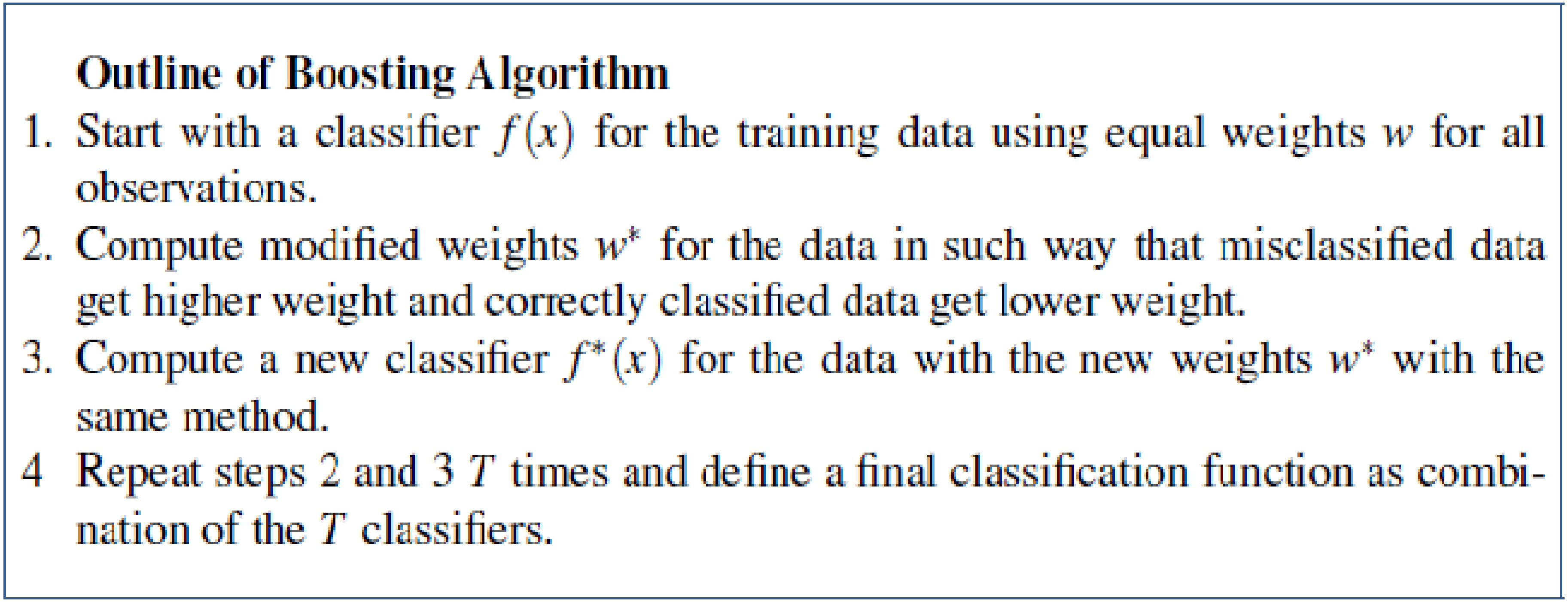
- Basic problem formulation for support vector machines in case of two classes labeled by –1 and +1
  - Training data from two classes:

    $$(\vec{x}_i, y_i),\;\; y_i \in \{-1, 1\},\;\; i = 1, 2, \ldots, n$$

  - We want to learn a classification rule by repeated application of a so called weak classifier, i.e., a classification function which is only slightly better than a naïve classification without knowledge of the attributes leading to a misclassification rate slightly smaller than 0.5

Model class: Combination Methods

- This idea is realized the Adaboost algorithm



Outline of Boosting Algorithm

1. Start with a classifier $f(x)$ for the training data using equal weights w for all observations.
2. Compute modified weights w^* for the data in such way that misclassified data get higher weight and correctly classified data get lower weight.
3. Compute a new classifier $f^*(x)$ for the data with the new weights w^* with the same method.

4 Repeat steps 2 and 3 T times and define a final classification function as combination of the T classifiers.

Model class: Combination Methods

- As weak classifier in most cases classification trees with only few nodes are used, sometimes called stumps
- Calculation of the weights for the data and weighting of the different classifiers uses the concept of exponential loss of a classifier $f(x)$ defined by

$$L(y, f) = \exp(-\beta \cdot y \cdot f(x))$$

- The method allows theoretical estimation of the generalization error and has similarities with logistic regression