Statistical Databases and Registers with some statistical learning

a course in
Survey Methodology and Official Statistics
Pages in the book: 9-18, 305-310

Department of Statistics
Stockholm University

December 2011

What is statistical learning?

Statistical learning is the science of making classifications/forecasts from huge amounts of data

Statistical learning is the science where we create/use "algorithms for inferring unknowns from knowns". We extract interesting (non-trivial, implicit, previously unknown and potentially useful) patterns or knowledge from huge amount of data

Alternative names: Data mining, **K**nowledge **D**iscovery in **D**atabases, knowledge extraction, data/pattern analysis, data archeology, information harvesting, business intelligence, etc.

The most known word Data mining is a misnomer.

Statistical learning is better, since we learn from data with aid of statistical methods.

Why statistical learning?

Following are reasons for new data methods:

- explosive growth, from terabytes to petabytes
- manual collection systems are replaced by automatic collection systems.
- data is made public and available over the Web

Major sources of abundant data

Business: e-commerce, transactions, stocks, . . .

Science: Remote sensing, bioinformatics, environmetal, . . .

Society: News, spam, digital cameras, YouTube, . . .

We are drowning in data, but starving for knowledge!

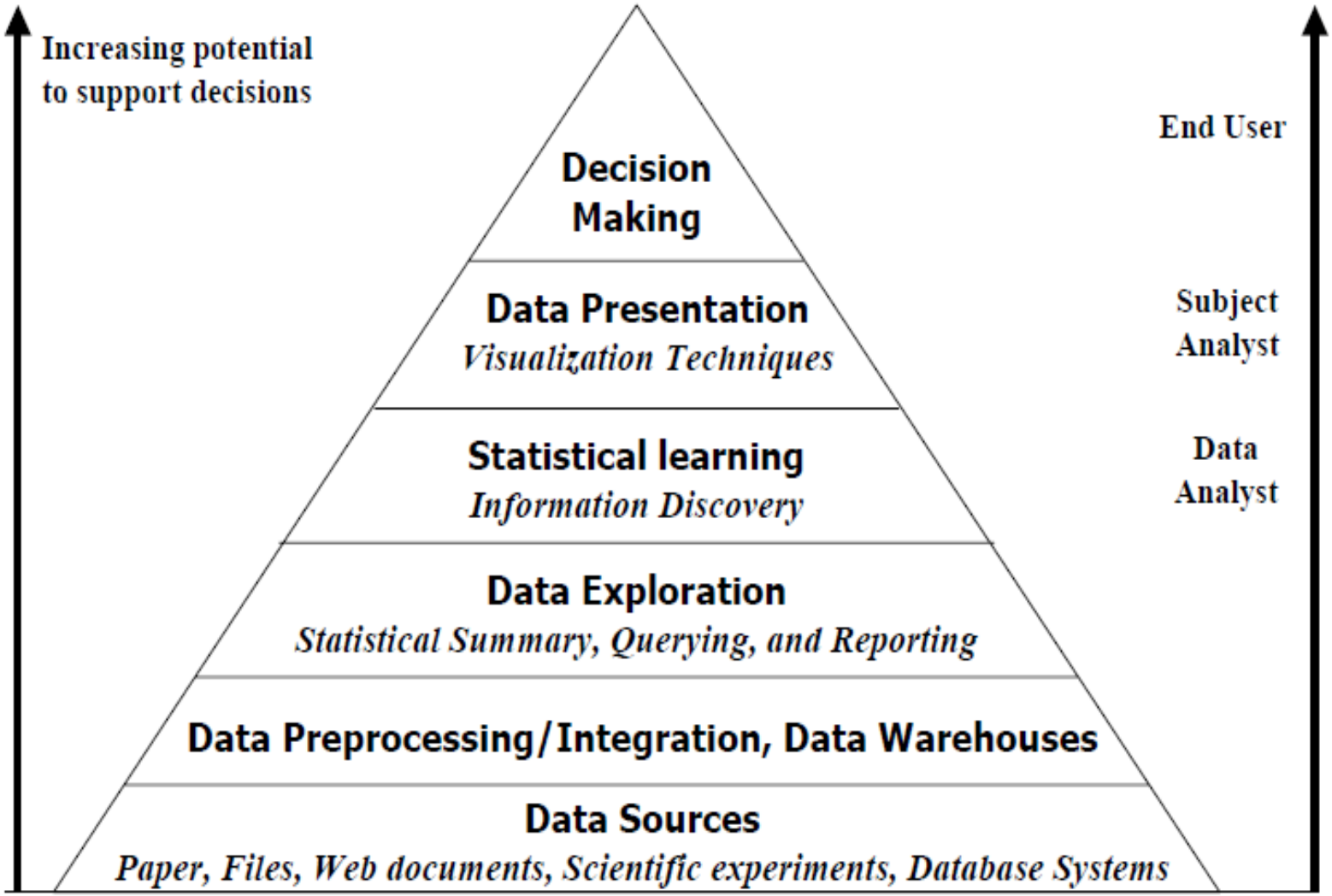
Evolution of database technology

- 1960s: Data collection, database creation, IMS (**I**nformation **M**anagement **S**ystems) and DBMS (**D**ata **B**ase MS)
- 1970s: Relational data model, **R**elational DBMS (RDBMS)
- 1980s:
  - RDBMS, advanced data models (extended-relational, OO (**O**bject **O**riented), deductive, etc.)
  - Application-oriented DBMS (spatial, scientific, . . .)
- 1990s: Data mining, data warehousing, multimedia databases, and Web databases
- 2000s:
  - Stream data management and mining
  - Data mining and its applications
  - Web technology (XML, data integration) and global information systems

Why not traditional statistics

- Tremendous amount of data
  - Algorithms must be highly scalable, to cope with tera-bytes of measurements
  - Vectors of high dimensions (up to tens of thousands)
- High complexity of data
  - Data streams and sensor data
  - Time-series data, temporal data, sequence data
  - Structure data, graphs, social networks and multi-linked data
  - Spatial, spatiotemporal, multimedia, text and Web data
  - Software programs, scientific simulations
- New and sophisticated applications

Statistical learning in decision making



The task for statistical learning

The task for statistical learning is:

- Classification - from an overwhelming amount of classify data.
- Regression - predict a new value given information.
- Pattern finding

This is the essence of statistics.

But traditional statistics starts with a model and tries to reject it. If it is not possible to reject the model it is accepted on probable causes.

Statistical learning is the other way around. Learn possible models from data.

Then, due to your objective, different actions will be taken.

Statistical learning examples

A few examples where statistical learning is used

- **Business**: When working with customer data in a local store it was found that many customers on Fridays bought diapers and beers.
- **Physics**: Using readouts of a double-pendulum gives Newton's second law of motion and the law of conservation of momentum

  Eureqa, a program that distills scientific laws from raw data

- **OCR**: To recognize the handwritten post code on letters.
- **PageRank**: How to bring order to the World Wide Web.
- **Medicin**: Identify the risk factors for prostate cancer, based on clinical and demographic variables.

Supervised learning

Supervised learning is when we have an outcome that will guide us in the learning process. So given data $D = \{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n)\}$, where $\mathbf{x}_i$ is a data point and $\mathbf{y}_i$ is class/value, choose a function $\mathbf{f}(\mathbf{x}) = \mathbf{y}$. Here we are concerned with

- Classification - $\mathbf{y}_i \in \{\text{finite set of classes}\}$
- Regression - $\mathbf{y}_i \in \mathbb{R}^d$

(notes here)

Examples of methods are

- Linear Regression and Nearest Neighbour
- Logistic Regression
- Neural Networks

Unsupervised learning

Unsupervised learning is when we try to describe how the data is organized or clustered. So given data $D = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ try to find patterns in data. Here we are concerned with methods like

- Clustering
- Density estimation
- Dimensionality reduction

(notes here)

Examples of methods are

- Cluster Analysis
- Principal Component Analysis
- The Google PageRank Algorithm

Notations

Input variables will be denoted by X and output variables by Y. When these variables are vectors we write $\mathbf{X}$ and $\mathbf{Y}$ and when they are matrices we write $\mathbb{X}$ and $\mathbb{Y}$.

We use small letters to signify observations eg $\mathbf{x}$.

When the output variables are quantitative we talk about *regression* and when they are qualitative *classification*.

In supervised learning the goal is to use the inputs to predict the values of the outputs. For this we need *training* data to construct a prediction rule.

Classification with linear regression

Our prerequisite are Y = income (a value) and (X_1, X_2) geographical coordinates.

Our task is to send an advertisement to the areas (ZIP code) where people have more than average income.

A ZIP code in Sweden consists of 5 numbers where each digit signify a more detailed geographical division.

Eg Stockholm starts with a 1 and the ZIP code for Kungsgatan 32-54 is 11135.

We know who lives at 11135[1] and we also know their income

How to choose these areas of prospective customers?

[1] Because of the ever increasing demand of controlling people we soon have a dwelling register and hence it will in the future be possible to attach an income to each dwelling.

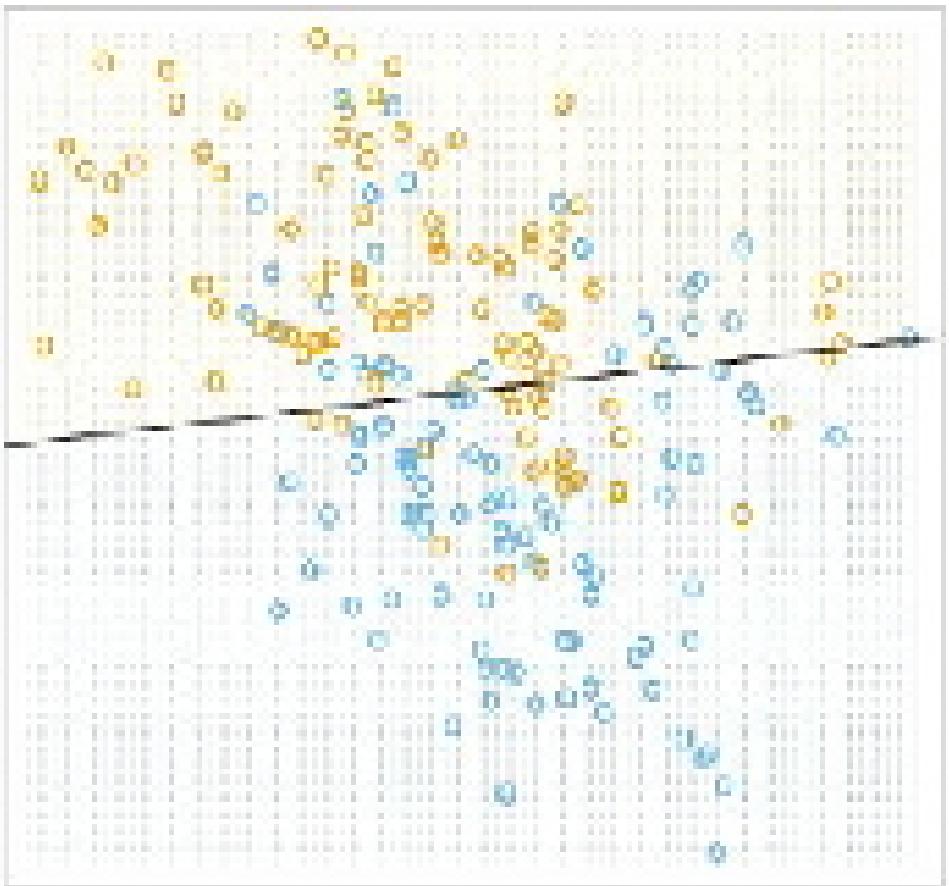
Classification with linear regression (cont)

As a model we choose

$$\mathbf{x} = (x_1, x_2)$$

$$y = \begin{cases} \text{orange} \\ \text{blue} \end{cases}$$

The orange circles denote people with a more than average income and the blue with less than average income.



We will use linear regression to divide the area in two parts.

Classification with linear regression (cont)

From the picture it is easily seen that there is a crude division between orange and blue circles.

To find the decision border we assume the model

$$Y = \mathbf{x}^T \boldsymbol{\beta} + \boldsymbol{\epsilon} = \text{income at } \mathbf{x}$$

and the classification $\{0, 1\}$ where

$$G = \begin{cases} 1 & \text{orange} \quad \text{if } \hat{Y} > 0.5 \\ 0 & \text{blue} \quad \text{if } \hat{Y} \leq 0.5 \end{cases}$$

This classification give us the decision border

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 = 0.5$$

which is seen as the line in the figure.
Our model misclassify a lot of data, on both sides.

k-Nearest Neighbours

The main idea is, given, $\mathbf{x} = (x_1, x_2)$ find the k closest points, in D, to $\mathbf{x}$. Let

$$N_k(\mathbf{x}) = \{\mathbf{z} \in D \mid \text{the } k \text{ points with smallest distances to } \mathbf{x}\}.$$

This in turn will partition the plane $\mathbb{R}^2$ into mutually exclusive regions. Then we compute the average income for these points

$$\hat{Y}(\mathbf{x}) = \frac{1}{k} \sum_{\mathbf{x}_i \in N_k(\mathbf{x})} y(x_i).$$

The point $\mathbf{x}$ is classified, as orange or blue, by the same algorithm as before

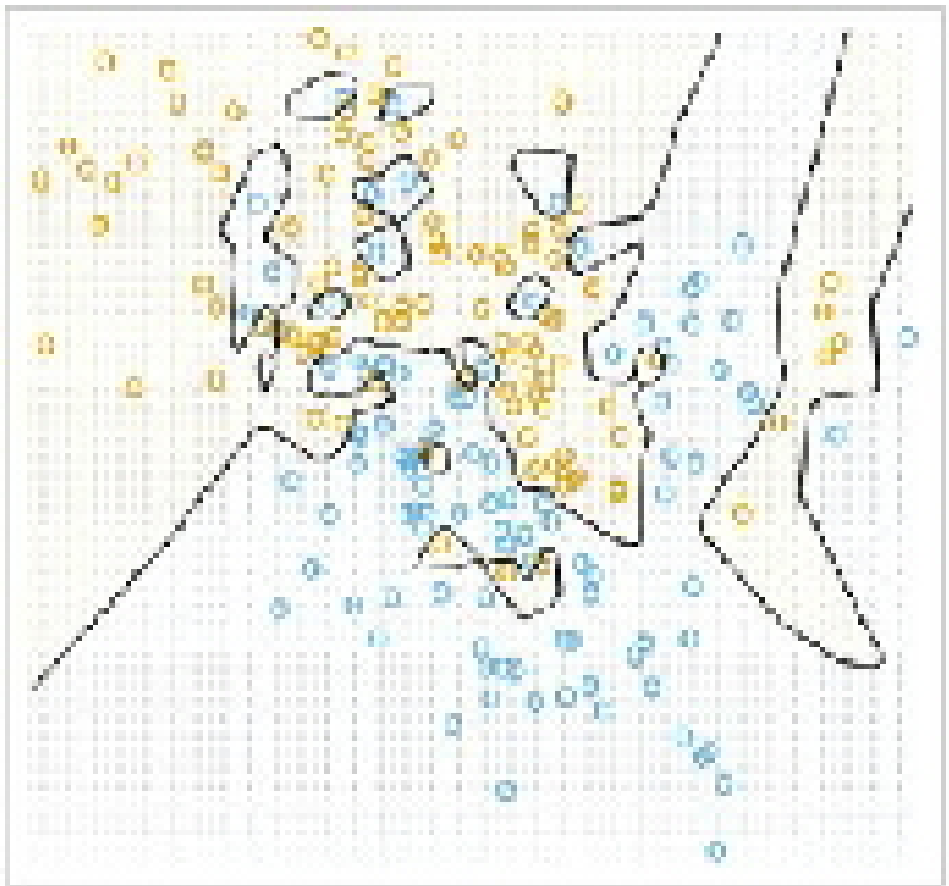
$$G = \begin{cases} \text{orange} & \text{if } \hat{Y} > 0.5 \\ \text{blue} & \text{if } \hat{Y} \leq 0.5 \end{cases}$$

k-Nearest Neighbours k=1

We have

$$\hat{Y}(\mathbf{x}) = y(x_i)$$

Choose a point $\mathbf{x}$ and find the closest point $\mathbf{x}_i$ to it. Draw a line from $\mathbf{x}$ to $\mathbf{x}_i$. Every point on the line will have the same classification as $\mathbf{x}_i$. Do so for all points $\mathbf{x} \in \mathbb{R}^2$ and we will have our partition.



The orange circles are now totally separated from the blue circles and vice versa.

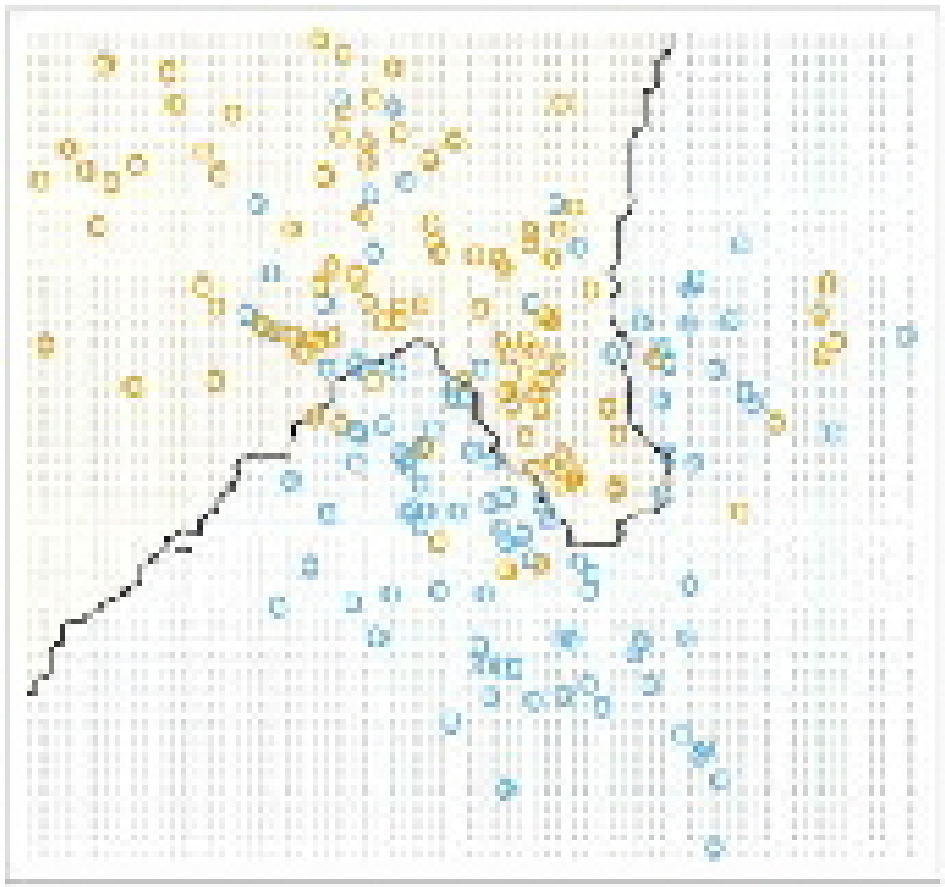
k-Nearest Neighbours k=15

Around each point $\mathbf{x} \in \mathbb{R}^2$ find the 15 closest points.

Classify $\mathbf{x}$ as 0 or 1 according to the algorithm:

$$\hat{Y}(\mathbf{x}) = \frac{1}{k} \sum_{\mathbf{x}_i \in N_k(\mathbf{x})} y(x_i)$$

$$G = \begin{cases} 1 & \hat{Y} > 0.5 \\ 0 & \hat{Y} \leq 0.5 \end{cases}$$



The border is now the black line. This classification is cruder than the previous one but less crude than the first one (the line).

k-Nearest Neighbours

Above we used the simple euclidean distance but one is free to use any distance function.

At first sight this model has only 1 parameter, k, compared to the p $(= 2)$ parameters of the vector $\boldsymbol{\beta}$ in the linear regression case.

But the effective number of parameters are N/k.

To see this note that if we have exactly N/k distinct, non-overlapping, neighbourhoods then we would fit one parameter (the mean) per neighbourhood.

Usually $N/k > p$.

Conclusion: Regression and Neighbour

Linear regression

- gives a border that do not change much with new observations.
- constructs a stable border: Low variance and potential high bias

Nearest Neigbours

- gives, for small values of k, borders that are highly dependent on the training set.
- gives an unstable border: High variance and low bias.

Classification and regression trees – the CART approach

Simple but effective and give good results.

The setup is the data set $D = \{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n)\}$ and the point $\mathbf{x}$ which we want to classify.

Main idea with decision trees is to construct a binary tree and minimize the training error in each leaf of the tree. This makes the tree grow and become big since no errors occur when each data point is its own leave.

(notes here)

CART approach – Regression

It is quite easy to make binary splits – visually – in two dimensions. But what to do with many dimensions?

Suppose $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$ with $\mathbf{x}_i = (x_{i1}, x_{i2,\ldots,}x_{ip})$ and that we partition into M mutually exclusive regions $R_1, R_2, \ldots, R_M$ ie $\mathbb{R}^p = R_1 \cup R_2 \cup \cdots \cup R_M$. For each region we model the response as a constant value c_m. In short we have

$$f(\mathbf{x}) = \sum_{m=1}^{M} c_m I(\mathbf{x} \in R_m) = \begin{cases} c_1 & \mathbf{x} \in R_1 \\ c_2 & \mathbf{x} \in R_2 \\ \vdots & \\ c_M & \mathbf{x} \in R_M \end{cases}$$

How do we choose the regions $R_1, R_2, \ldots, R_M$?

CART approach – Regression (cont)

The first idea is to minimize a sum like

$$Q = \sum_{i=1}^{n} \left(y_i - f\left(\mathbf{x}_i\right)\right)^2$$

from which we get $\hat{c}_m = ave\left(y_i \mid \mathbf{x}_i \in R_m\right)$. But this method turns out to be too computational complicated.

To find our binary splits we need to simplify and we do that by regarding one variable x_{ij} at a time.

For variable x_{ij} we try a split at some number s and calculate

$$Q_{js} = \min_a \sum_{\mathbf{x}_i \in R_{\leq}(j,s)} \left(y_i - a\right)^2 + \min_b \sum_{\mathbf{x}_i \in R_{>}(j,s)} \left(y_i - b\right)^2$$

where $R_{\leq}\left(j,s\right) = \left\{\mathbf{x} \mid x_{ij} \leq s\right\}$ and $R_{>}\left(j,s\right) = \left\{\mathbf{x} \mid x_{ij} > s\right\}$. We then choose the j and s that gives the minimum of Q_{js}.

CART approach – Regression (cont)

We now need some kind of stopping rule since otherwise we will end up with a tree where each $\mathbf{x}_i$ has its own region.

One such stopping rule is

$$C_\alpha(T) = \sum_{m=1}^{|T|} \sum_{\mathbf{x}_i \in R_m} (y_i - \hat{c}_m)^2 + \alpha |T|$$

where

$$\begin{aligned} |T| &= \text{number of leaves} \\ &= \text{number of regions} \end{aligned}$$

We note that the function $C_\alpha(T)$ is a penalty function (compare eg AIC) which we want to minimize.

CART approach – Classification

The classification procedure of CART assumes the number of classes to be K. Here we only indicate the stopping rule.

The difference between the Regression and Classification methods are in the penalty function. In the Classification case we start with the proportion/frequence of class k observations in node m.

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{\mathbf{x}_i \in R_m} I\left(y_i = k\right)$$

An observation $\mathbf{x}$ in node m is classified as class $k\left(m\right)$ where $k\left(m\right) = \arg\max_k \hat{p}_{mk}$, the majority class in node m.

As measure of dispersion (compare $Q_m\left(T\right)$) we take e.g. the Gini index

$$\sum_{k=1}^{K} \hat{p}_{mk}\left(1 - \hat{p}_{mk}\right) = 1 - \sum_{k=1}^{K} \hat{p}_{mk}^2$$

Advantages of CART

- CART is nonparametric. Therefore the method does not require specification of any functional form
- CART does not require variables to be selected in advance. The algorithm will itself identify the most significant variables and eliminate non-significant ones.
- CART-results are invariant to monotone transformations. Changing one or several variables to its logarithm or square root will not change the structure of the tree. Only the splitting values (but not variables) will be different.
- CART can easily handle outliers.
  Outliers can negatively affect the results of some statistical models. The splitting algorithm of CART handles noisy data easily. Outliers are isolated in a separate node.

Disadvantages of CART

- CART may have unstable decision trees.
  Insignificant modification of learning sample, such as eliminating several observations, could lead to radical changes in the decision tree: increase or decrease of tree complexity, changes in splitting variables and values.
- CART splits only by one variable and not combinations of variables. That is; all splits are perpendicular to the axis under consideration.
- CART-splits are locally optimal at each split – they may not be globally optimal.