Linear Classification

Volker Tresp

Classification

- Classification is the central task of pattern recognition
- Sensors supply information about an object: to which class belongs the object (dog, cat, ...)

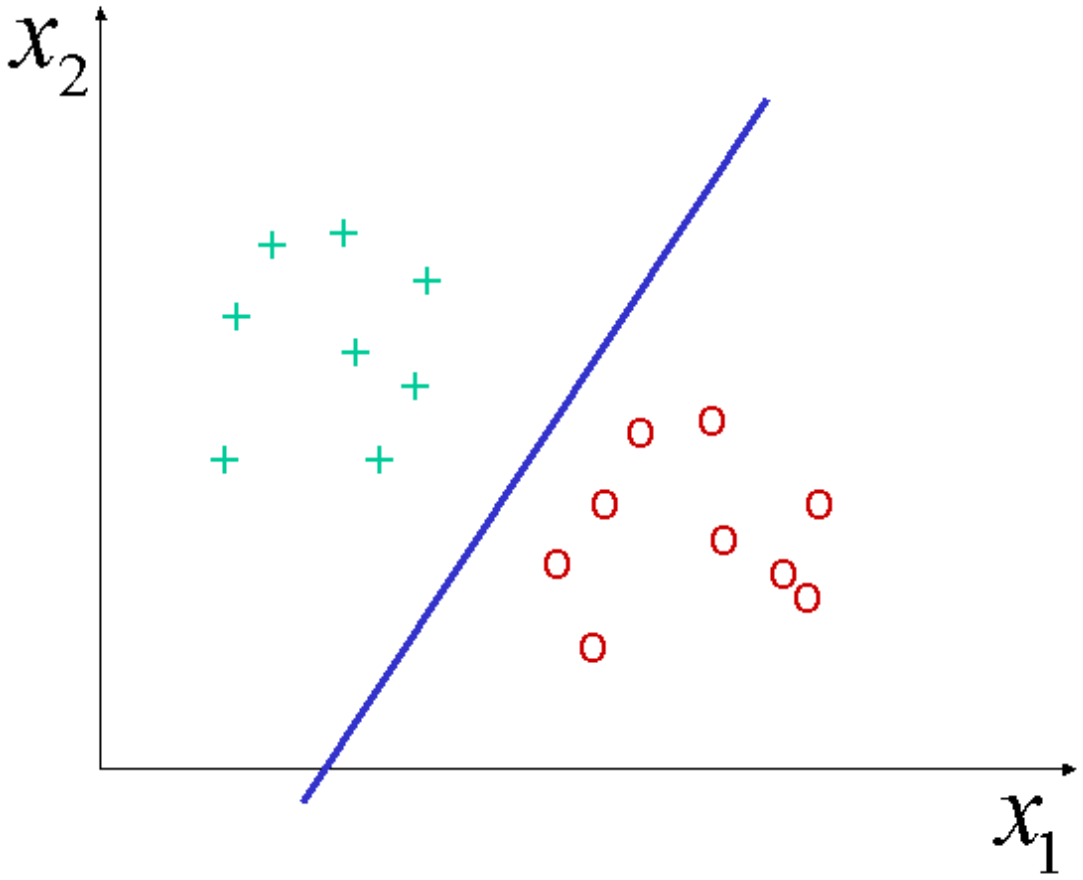
Linear Classifiers

- Linear classifiers separate classes by a linear hyperplane
- In high dimensions a linear classifier often can separate the classes
- Linear classifiers cannot solve the *exclusive-or* problem
- In combination with basis functions, kernels or a neural network, linear classifiers can form nonlinear class boundaries

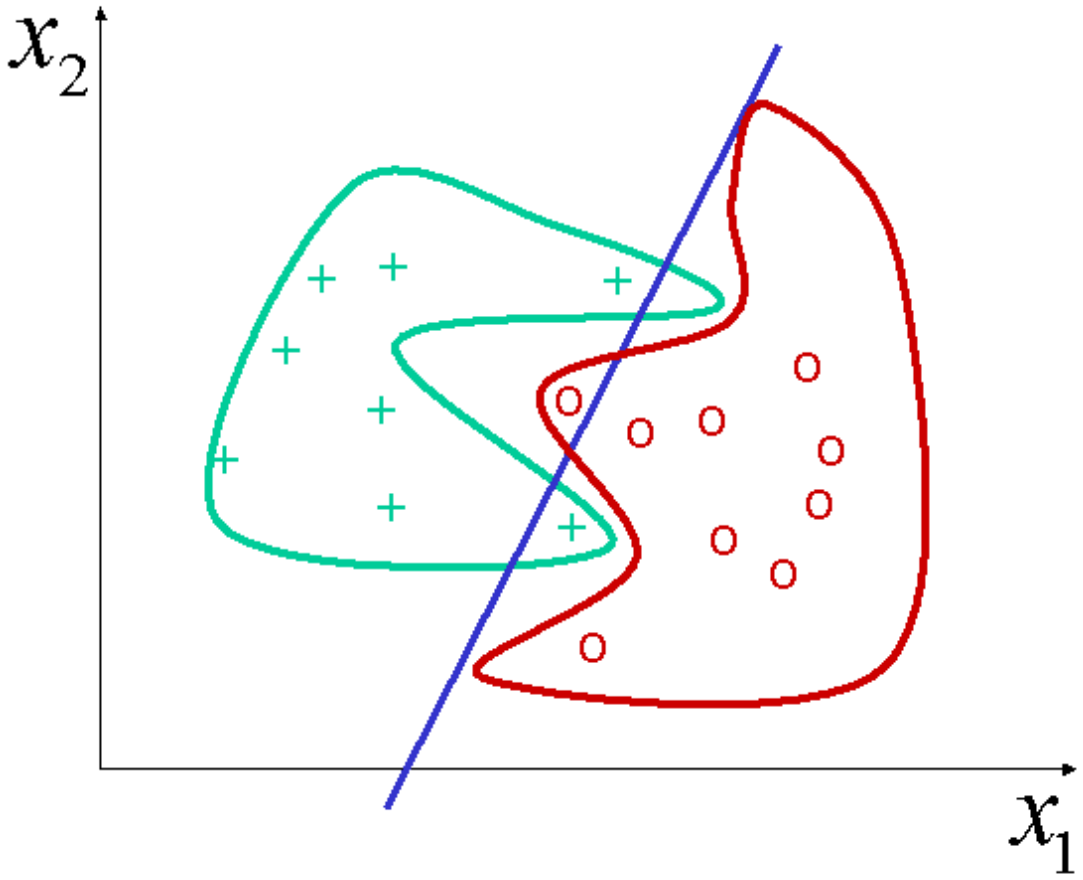
Binary Classification Problems

- We will focus first on binary classification where the task is to assign binary class labels $y_i = 1$ and $y_i = 0$ (or $y_i = 1$ and $y_i = -1$)
- We already know the *Perceptron*. Now we learn about additional approaches
  - I. Generative models for classification
  - II. Logistic regression
  - III. Classification via regression

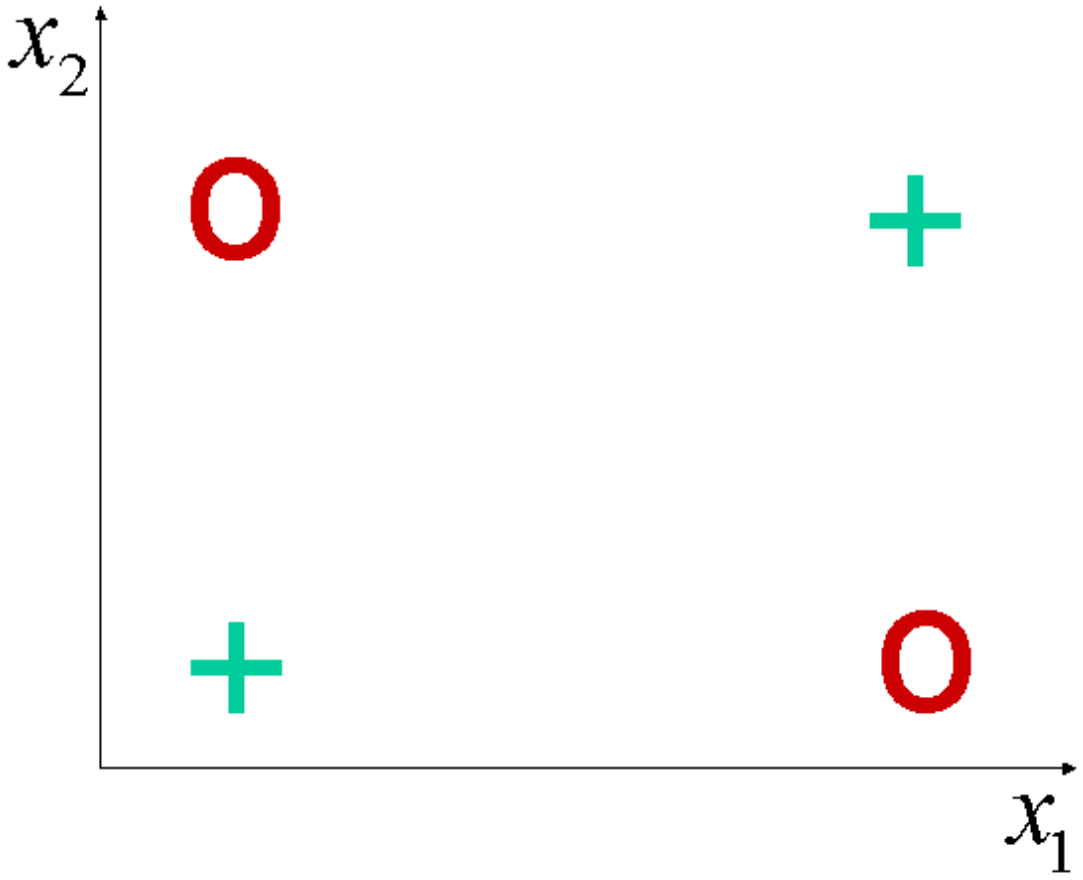
Two Linearly Separable Classes



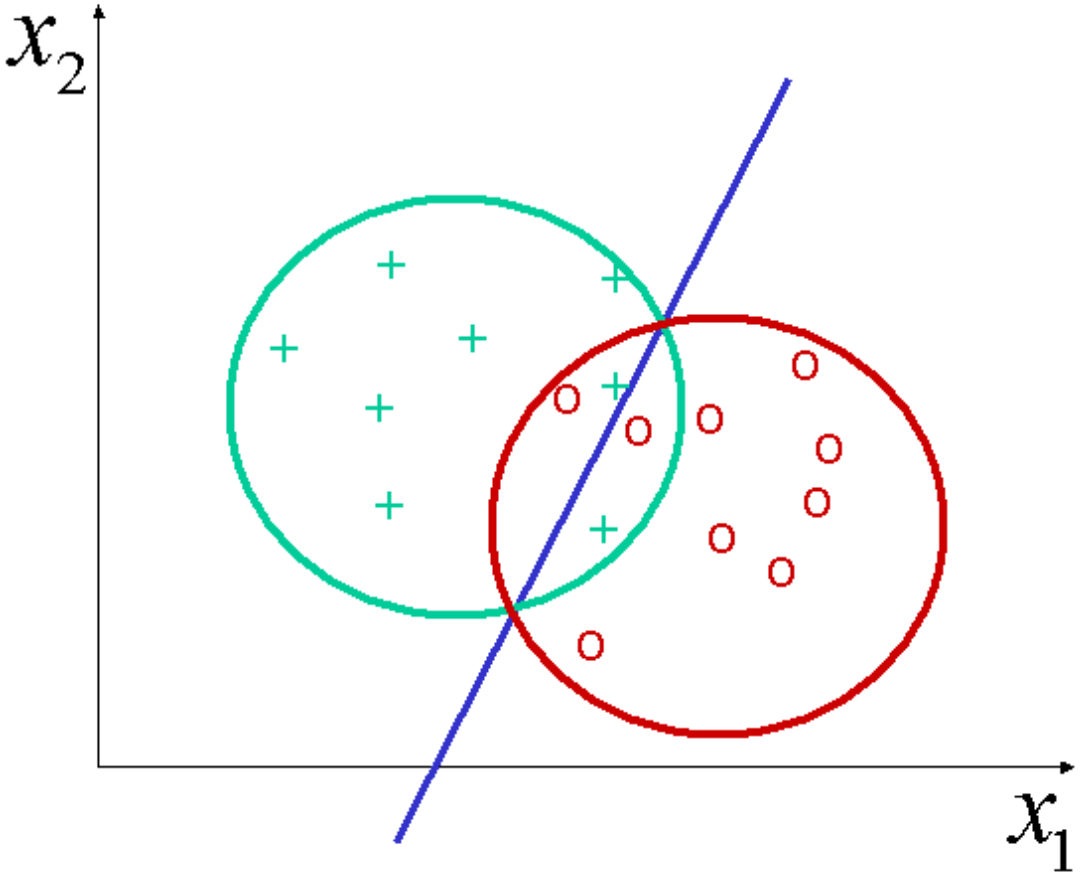
Two Classes that Cannot be Separated Linearly



The Classical Example not two Classes that cannot be Separated Linearly: XOR



Separability is not a Goal in Itself: Overlapping Classes



I. Generative Model for Classification

- On assumes a data generating process
- In particular, we assume that the observed classes y_i are generated with probability

$$P(y_i = 1) = \kappa_1 \qquad P(y_i = 0) = \kappa_0 = 1 - \kappa_1$$

  with $0 \leq \kappa_1 \leq 1$. In a next step, a data point $\mathbf{x}_i$ has been generated from $P(\mathbf{x}_i|y_i)$
- (Note, that $\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,M})^T$, which means that $\mathbf{x}_i$ does not contain the bias $x_{i,0}$)

Bayes' Theorem

- To classify a data point $\mathbf{x}_i$, i.e. to determine the y_i, we apply Bayes theorem and get

$$P(y_i|\mathbf{x}_i) = \frac{P(\mathbf{x}_i|y_i)P(y_i)}{P(\mathbf{x}_i)}$$

$$P(\mathbf{x}_i) = P(\mathbf{x}_i|y_i = 1)P(y_i = 1) + P(\mathbf{x}_i|y_i = 0)P(y_i = 0)$$

- Maximum-likelihood estimator for the prior class probabilities are

$$\widehat{P}(y_i = 1) = \widehat{\kappa}_1 = N_1/N$$

and

$$\widehat{P}(y_i = 0) = \widehat{\kappa}_0 = N_0/N = 1 - \widehat{\kappa}_1$$

where N_1 and N_0 is the number of training data points for class 1, respectively class 0

Class-specific Distributions

- To model $P(\mathbf{x}_i|y_i)$ one can chose an application specific distribution
- A popular choice is a Gaussian distribution

$$P(\mathbf{x}_i|y_i = l) = \mathcal{N}(\mathbf{x}_i; \mu^{(l)}, \Sigma)$$

with

$$\mathcal{N}\left(\mathbf{x}_i; \mu^{(l)}, \Sigma\right) = \frac{1}{(2\pi)^{M/2}\sqrt{|\Sigma|}} \exp\left(-\frac{1}{2}\left(\mathbf{x}_i - \mu^{(l)}\right)^T \Sigma^{-1}\left(\mathbf{x}_i - \mu^{(l)}\right)\right)$$

- Note, that both Gaussian distributions have different modes (centers) but the same covariance matrices. This has been shown to often work well.

Maximum-likelihood Estimators for Modes and Covariances

- One obtains a maximum likelihood estimators for the modes

$$\widehat{\mu}^{(l)} = \frac{1}{N_l} \sum_{i:y_i=l} \mathbf{x}_i$$

- One obtains as unbiased estimators for the covariance matrix

$$\widehat{\Sigma} = \frac{1}{N-M} \sum_{l=0}^{1} \sum_{i:y_i=l} (\mathbf{x}_i - \widehat{\mu}^{(l)})(\mathbf{x}_i - \widehat{\mu}^{(l)})^T$$

A Posteriori Distribution

- It follows that

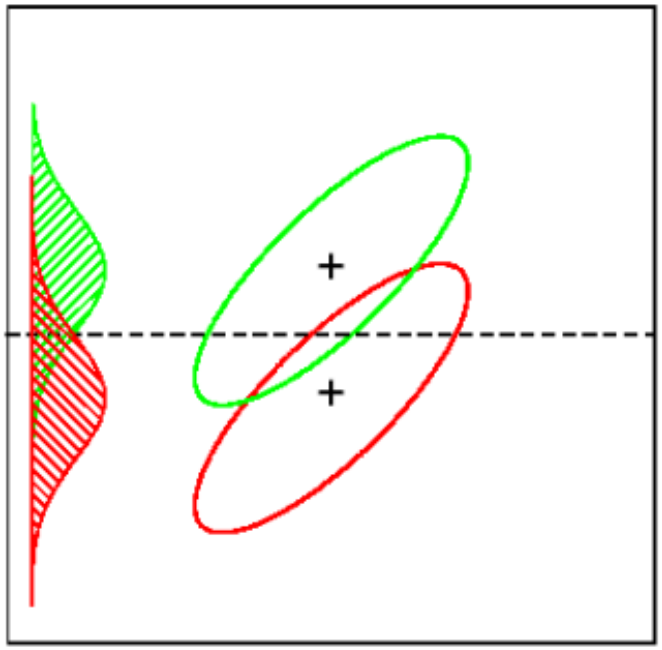
$$P(y_i = 1|\mathbf{x}_i) = \frac{P(\mathbf{x}_i|y_i = 1)P(y_i = 1)}{P(\mathbf{x}_i|y_i = 1)P(y_i = 1) + P(\mathbf{x}_i|y_i = 0)P(y_i = 0)}$$

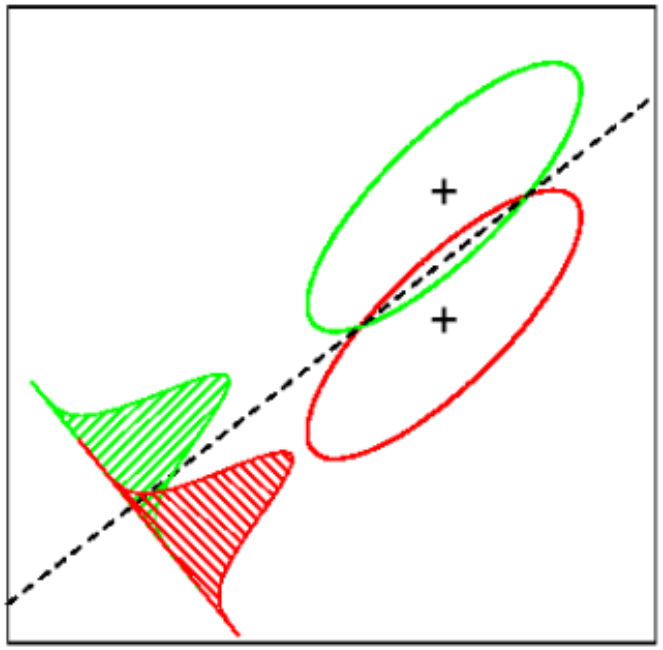
$$= \frac{1}{1 + \frac{\kappa_0}{\kappa_1} \exp\left((\mu^{(0)} - \mu^{(1)})^T \Sigma^{-1} \mathbf{x}_i + \frac{1}{2}\mu^{(0)T}\Sigma^{-1}\mu^{(0)} - \frac{1}{2}\mu^{(1)T}\Sigma^{-1}\mu^{(1)}\right)}$$

$$= \text{sig}\left(w_0 + \mathbf{x}_i^T \mathbf{w}\right) = \text{sig}\left(w_0 + \sum_j^M x_{i,j} w_j\right)$$

$$\mathbf{w} = \Sigma^{-1}\left(\mu^{(1)} - \mu^{(0)}\right)$$

$$w_0 = \log \kappa_1/\kappa_0 - \frac{1}{2}\mu^{(0)T}\Sigma^{-1}\mu^{(0)} + \frac{1}{2}\mu^{(1)T}\Sigma^{-1}\mu^{(1)}$$





Comments

- This specific generative model leads to linear class boundaries
- The solution is analogue to Fisher's linear discriminant analysis, where one projects the data into a space in which data from the same class have small variance and where the distance between class modes are maximized
- In other words, one gets the same results from an optimization criterion without assuming Gaussian distributions
- Although we have used Bayes formula, the analysis was frequentist. A Bayesian analysis with a prior distribution on the parameters is also possble

Special Case: Naive Bayes

- If one assumes that the covariance matrices are diagonal, one obtains a *Naive-Bayes* classifier

$$P(\mathbf{x}_i|y_i = l) = \prod_{j=1}^{M} \mathcal{N}(\mathbf{x}_{i,j}; \mu_j^{(l)}, \sigma_j^2)$$

- The naive Bayes classifier has considerable fewer parameters but completely ignores class-specific corelations between features; this is sometimes considered to be naive

- Naive Bayes classifiers are popular in text analysis with often more than 10000 features (key words). For example, the classes might be SPAM and no-SPAM and the features a keywords in the texts. Instead of a Gaussian distribution, a Bernoulli distribution is employed, with $P(word_j|\text{SPAM}) = \gamma_{j,s}$ and $P(word_j|\text{no-SPAM}) = \gamma_{j,n}$

II. Logistic Regression

- The generative model motivates

$$P(y_i = 1|\mathbf{x}_i) = \text{sig}\left(\mathbf{x}_i^T \mathbf{w}\right)$$

(now we include the bias $\mathbf{x}_i^T = (x_{i,0} = i, 1, x_{i,1}, \ldots, x_{i,M-1})^T$). sig() as defined before (logistic funktion).

- One now optimizes the likelihood of the conditional model

$$L(\mathbf{w}) = \prod_{i=1}^{N} \text{sig}\left(\mathbf{x}_i^T \mathbf{w}\right)^{y_i} \left(1 - \text{sig}\left(\mathbf{x}_i^T \mathbf{w}\right)\right)^{1-y_i}$$

Log-Likelihood

- Log-likelihood function

$$l = \sum_{i=1}^{N} y_i \log\left(\text{sig}\left(\mathbf{x}_i^T \mathbf{w}\right)\right) + (1 - y_i) \log\left(1 - \text{sig}\left(\mathbf{x}_i^T \mathbf{w}\right)\right)$$

$$l = \sum_{i=1}^{N} y_i \log\left(\frac{1}{1 + \exp(-\mathbf{x}_i^T \mathbf{w})}\right) + (1 - y_i) \log\left(\frac{1}{1 + \exp(\mathbf{x}_i^T \mathbf{w})}\right)$$

$$= -\sum_{i=1}^{N} y_i \log(1 + \exp(-\mathbf{x}_i^T \mathbf{w})) + (1 - y_i) \log(1 + \exp(\mathbf{x}_i^T \mathbf{w}))$$

Adaption

- The derivatives of the log-likelihood with respect to the parameters

$$\frac{\partial l}{\partial \mathbf{w}} = \sum_{i=1}^{N} y_i \frac{\mathbf{x}_i \exp(-\mathbf{x}_i^T \mathbf{w})}{1 + \exp(-\mathbf{x}_i^T \mathbf{w})} - (1 - y_i) \frac{\mathbf{x}_i \exp(\mathbf{x}_i^T \mathbf{w})}{1 + \exp(\mathbf{x}_i^T \mathbf{w})}$$

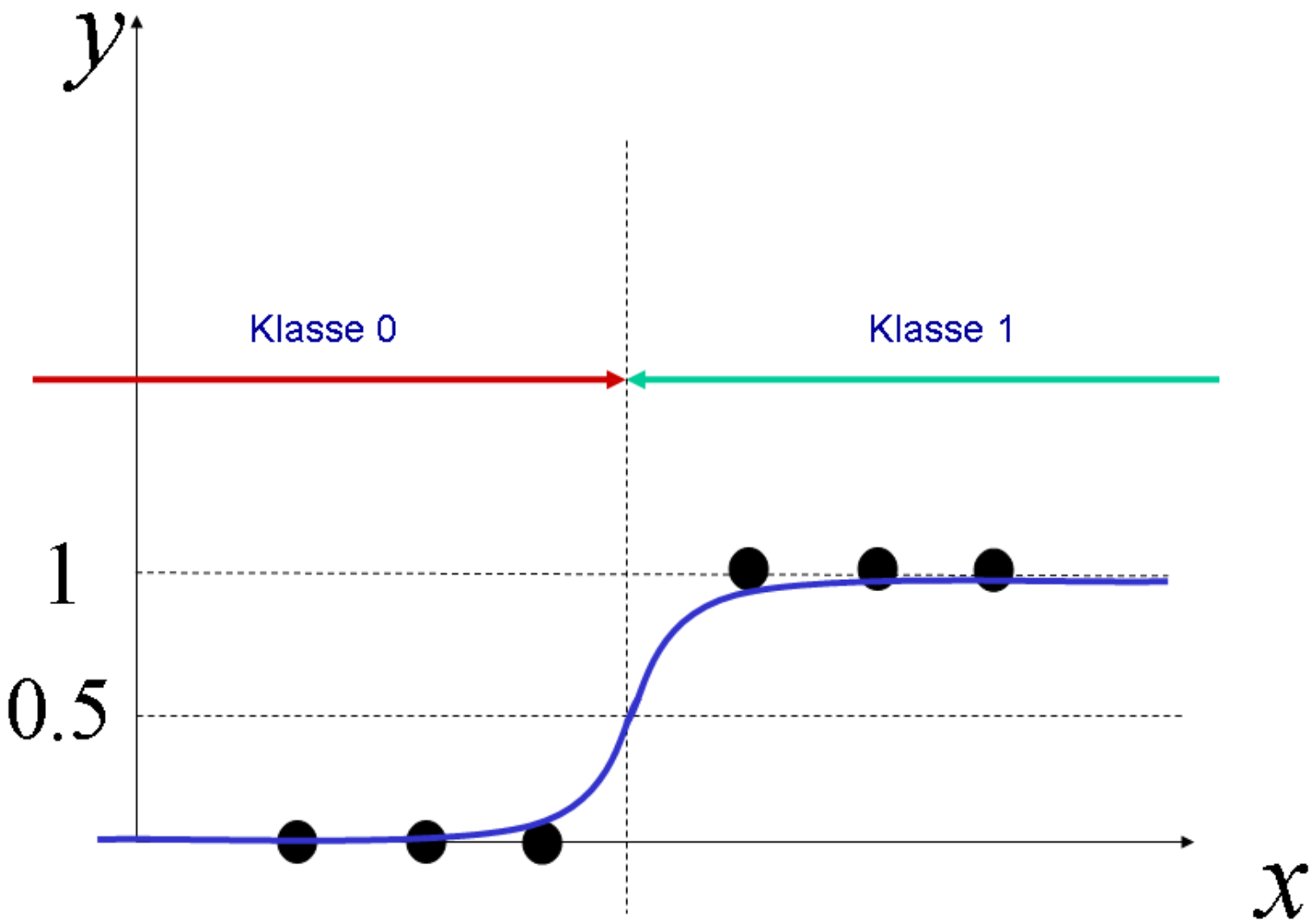
$$= \sum_{i=1}^{N} y_i \mathbf{x}_i (1 - \text{sig}(\mathbf{x}_i^T \mathbf{w})) - (1 - y_i) \mathbf{x}_i \text{sig}(\mathbf{x}_i^T \mathbf{w})$$

$$= \sum_{i=1}^{N} (y_i - \text{sig}(\mathbf{x}_i^T \mathbf{w})) \mathbf{x}_i$$

- A gradient-based optimization of the parameters to maximize the log-likelihood

$$\mathbf{w} \longleftarrow \mathbf{w} + \eta \frac{\partial l}{\partial \mathbf{w}}$$

- Typically one uses a Newton-Raphson optimization procedure


y
Klasse 0
Klasse 1
1
0.5
x

Cross-entropy Cost Function

- The likelihood cost function is also called cross entropy cost function and is written for $y_i \in \{0, 1\}$

$$cost = \sum_{i=1}^{N} y_i \log(1 + \exp(-\mathbf{x}_i^T \mathbf{w})) + (1 - y_i) \log(1 + \exp(\mathbf{x}_i^T \mathbf{w}))$$

$$= \sum_{i=1}^{N} \log\left(1 + \exp\left((1 - 2y_i)\mathbf{x}_i^T \mathbf{w}\right)\right)$$

- ... and for $y_i \in \{-1, 1\}$

$$cost = \sum_{i=1}^{N} \log\left(1 + \exp\left(-y_i \mathbf{x}_i^T \mathbf{w}\right)\right)$$

Logistic Regression in Medical Statistics

- Logistic regression has become one of the the most important tools in medicine to analyse the outcome of treatments
- $y_i = 1$ means that the patient has the disease. $x_1 = 1$ might represent the fact that the patient received the treatment (medication) and $x_1 = 0$ might mean that the patient did not receive the treatment. The other inputs are often typical confounders (age, sex, ...)
- Logistic regression then permits the prediction of the outcome for any patient
- Of course, of great interest is if w_1 is significantly negative (i.e., the treatment is effective)

Log-Odds and Log-Odds-Ratio

- The logarithm of the Odds is defined as

$$\log(Odds(\mathbf{x}_i)) = \log \frac{P(y_i = 1|\mathbf{x}_i)}{P(y_i = 0|\mathbf{x}_i)}$$

- For logistic regression,

$$\log(Odds(\mathbf{x}_i)) = \log \frac{P(y_i = 1|\mathbf{x}_i)}{P(y_i = 0|\mathbf{x}_i)} = \log \frac{1}{1 + \exp(-\mathbf{x}_i^T \mathbf{w})} \frac{1 + \exp(-\mathbf{x}_i^T \mathbf{w})}{\exp(-\mathbf{x}_i^T \mathbf{w})}$$

$$= \log \frac{1}{\exp(-\mathbf{x}_i^T \mathbf{w})} = \mathbf{x}_i^T \mathbf{w}$$

- One is often only interested in the effect of the treatment

$$w_1 = \log(Odds(\mathbf{x}_1 = 1)) - \log(Odds(\mathbf{x}_1 = 0)) = \log \frac{Odds(\mathbf{x}_1 = 1)}{Odds(\mathbf{x}_1 = 0)}$$

- This is the logarithm of the so called odds ratio (OR). If $w_1 \approx 0$, then the treatment is ineffective: the odds ratio is commonly used in case-control studies!

Exponential Family

- Many common distributions (Bernoulli, Gauss, Binomial, multinomial, Poisson, ...) belong to the *exponential family of distribution* and can be written as

$$P(y|\eta) = \frac{1}{Z(\eta)} \exp\left(\eta^T F(y)\right) = \frac{1}{Z(\eta)} \exp\left(\sum_{j=0}^{M} \eta_j t_j(y)\right)$$

$$F(y) = (t_0(y), t_1(y), t_2(y), ...)^T$$

  where $t_1(y), t_2(y), ...$ is a sufficient statistics of the respective distribution

- Thus

$$\log P(y|\eta) = -\log Z(\eta) + \sum_j \eta_j t_j(y)$$

- Idea: model $\log P(y|\eta)$ as a weighted sum of basis functions $t_j(y)$. $Z(y)$ is needed for proper normalization, $Z(y) = \int \exp \sum_{j=0}^{M} \eta_j t_j(y) dy$ normalizes the distribution. Special parameter: $\eta_0 = 1$

Exponential Family: Bernoulli Distribution

- Consider a Bernoulli distribution with $P(y|\theta) = \theta^y(1-\theta)^{1-y}$. Then:

$$\log P(y|\theta) = y\log\theta + (1-y)\log(1-\theta) = \log(1-\theta) + y(\log\theta - \log(1-\theta)) =$$

$$\log(1-\theta) + y\log\frac{\theta}{1-\theta} = \log(1-\theta) + y\eta$$

- Where $\eta_1 = \log\frac{\theta}{1-\theta}$. We then also obtain the inverse $\theta = \frac{1}{1+\exp -\eta_1}$ and can write

$$\log(1-\theta) + y\eta_1 = -\log(1+\exp\eta_1) + y\eta_1$$

- We can identify: $t_1(y) = y$, $Z(\eta) = 1+\exp\eta$, $t_0(y) = 0$ and

$$P(y|\eta) = \frac{1}{1+\exp\eta}\exp y\eta$$

- Note that the use of the sigmoid function is motivated from the definition of the natural parameter

Generalized Linear Models (GLMs)

- Express the natural parameter as a linear function of the inputs to get the generalized linear model (GLM). For Bernoulli we get,

$$\eta_i = \mathbf{x}_i^T \mathbf{w}$$

$$P(y_i|\mathbf{x}_i, w) = \frac{1}{1 + \exp \mathbf{x}_i^T \mathbf{w}} \exp y_i \mathbf{x}_i^T \mathbf{w}$$

and in particular

$$P(y_i = 1|\mathbf{x}_i, \mathbf{w}) = \frac{1}{1 + \exp \mathbf{x}_i^T \mathbf{w}} \exp \mathbf{x}_i^T \mathbf{w} = \mathrm{sig}(\mathbf{x}_i^T \mathbf{w})$$

- *Thus logistic regression is the GLM for the Bernoulli likelihood model*

III. Classification via Regression

- Linear Regression:

$$f(\mathbf{x}_i, \mathbf{w}) = w_0 + \sum_{j=1}^{M-1} w_j x_{i,j}$$

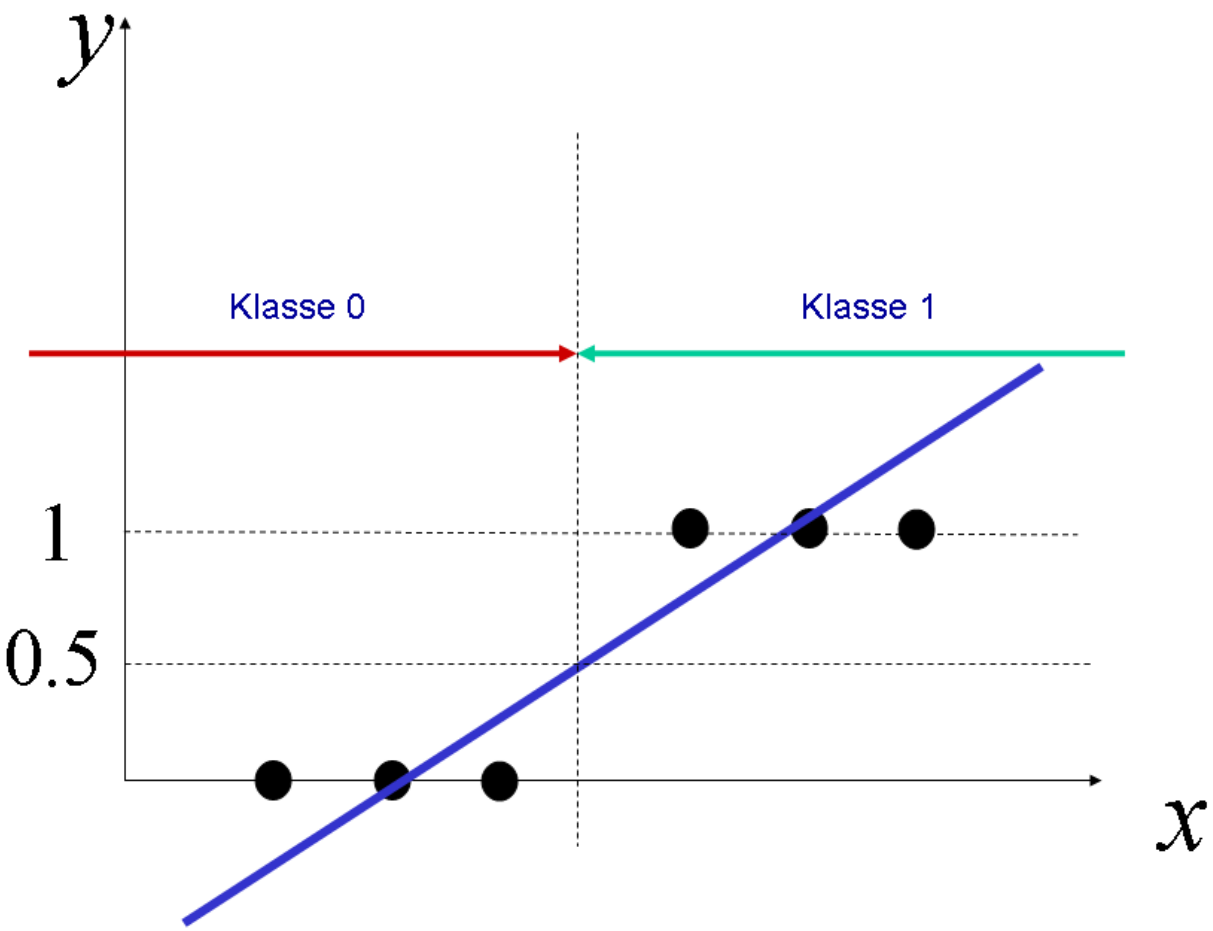
$$= \mathbf{x}_i^T \mathbf{w}$$

- We define as target $y_i = 1$ if the pattern $\mathbf{x}_i$ belongs to class 1 and $y_i = 0$ if pattern $\mathbf{x}_i$ belongs to class 0
- We calculate weights $\mathbf{w}_{LS} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$ as LS solution, exactly as in linear regression
- For a new pattern $\mathbf{z}$ we calculate $f(\mathbf{z}) = \mathbf{z}^T\mathbf{w}_{LS}$ and assign the pattern to class 1 if $f(\mathbf{z}) > 1/2$; otherwise we assign the pattern to class 0

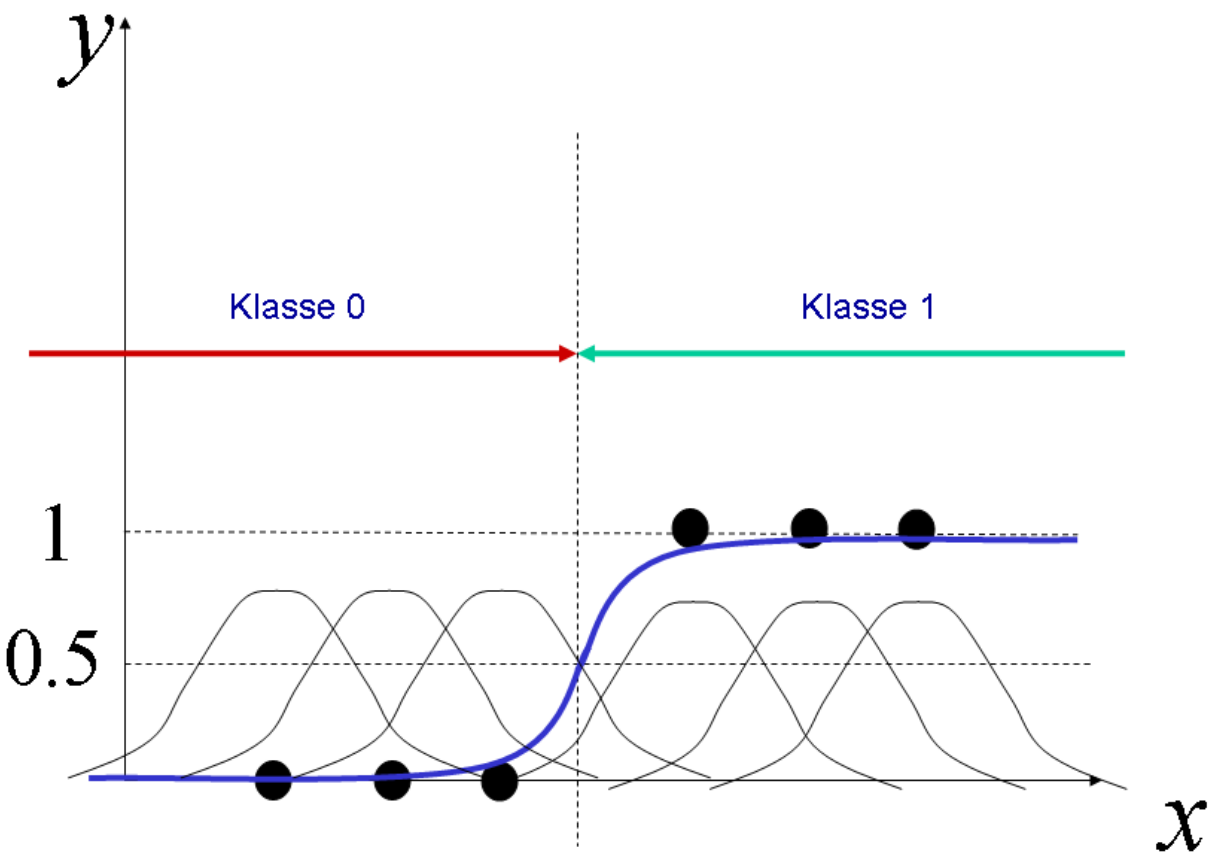
Bias

- Asymptotically, a LS-solution converges to the posterior class probabilities, althogh a linear functions is typically not able to represent $P(c = 1|\mathbf{x})$. The resulting class boundary can still be sensible
- One can expect good class boundaries in high dimensions and/or in combination with basis functions, kernels and neural networks; in high dimensions sometimes consistency can be achieved

Classification via Regression with Linear Functions



Classification via Regression with Radial Basis Functions



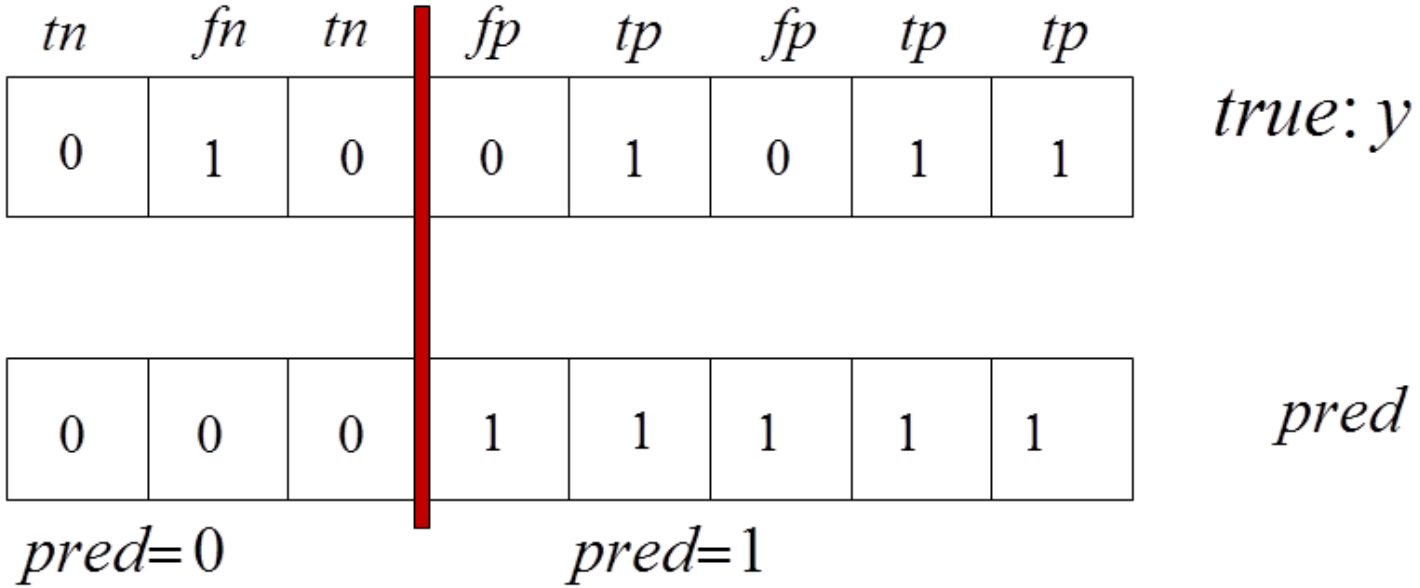
Performance

- Although the approach might seem simplistic, the performance can be excellent (in particular in high dimensions and/or in combination with basis functions, kernels and neural networks). The calculation of the optimal parameters can be very fast!

Evaluating Classifiers

- So far we focussed on accuracy: we maximized the number of correctly classified patterns
- Accuracy is not always a good measure: Let's assume that I want to generate a classifier that predicts if web pages are relevant for my query.
- If the classifier always predicts 0 (uninteresting), then the classifier has an accuracy very close to 100%

Definitions



- TP (True Positive) = #tp (here: 3)
- FP (False Positive) = #fp (here: 2)
- TN (True Negative) = #tn (here: 2)
- FN (True Negative) = #fn (here: 1)

Probabilistic Interpretation

- with $N = TP + FP + TN + FN$ test patterns,

$$\hat{P}(pred = 1, y = 1) = \frac{TP}{N}$$

$$\hat{P}(pred = 1, y = 0) = \frac{FP}{N}$$

$$\hat{P}(pred = 0, y = 0) = \frac{TN}{N}$$

$$\hat{P}(pred = 0, y = 1) = \frac{FN}{N}$$

Accuracy

- **Accuracy** :

$$Accuracy \rightarrow \frac{TP + TN}{N}$$

- If we assign the label *correct* to the events $(pred = 1, y = 1)$ and $(pred = 0, y = 0)$, then

$$Accuracy = P(correct)$$

- The error rate is (1-Accuracy).
- Accuracy is not a useful measure for highly imbalanced classes (see search engine example)

Precision

- **Precision** (Relevance, positive predicted value)

$$Precision = \frac{TP}{TP + FP}$$

- This approximates

$$P(y = 1 | pred = 1)$$

- If a search engine classifies a web page to be relevant, it should be relevant!

Negative Predictive Value

- **Negative Predictive Value** (NPR)

$$NPR = \frac{TN}{TN + FN}$$

- This approximates

$$P(y = 0 | pred = 0)$$

- If a search engine classifies a web page to be irrelevant, it should be irrelevant!

Recall

- **Recall** (sensitivity, true positive rate):

$$Recall = \frac{TP}{TP + FN}$$

- This approximates

$$P(pred = 1 | y = 1)$$

- A measure for classifier performance, independent of class-mix
- A search engine that classifies all pages as being interesting has a $Recall = 1$. A fire detector should not miss any fires and should have a Recall close to 1.

Specifity

- **Specificity** (Specifity, true negative rate, 1 - false-positive-rate)

$$Specifity = \frac{TN}{TN + FP}$$

- This approximates

$$P(pred = 0 | y = 0)$$

- A measure for classifier performance, independent of class-mix
- Specificity should be high for a fire detector: if there is no fire, it should not alarm

F-Measure

- **F-measure**

$$F = 2\frac{Precision \times Recall}{Precision + Recall}$$

The F-measure combines precision and recall. Trivial search engines, that either predict all pages to be relevant or irrelevant, would have an F-measure of 0.

Connection to Odds and Odds Ratio

- We can interpret the treatment as $pred$ and outcome as y
- Then

$$(Odds|treatment = 1) = \frac{Precision}{1 - Precision} = \frac{TP}{FP}$$

$$(Odds|treatment = 0) = \frac{1 - NPR}{NPR} = \frac{FN}{TN}$$

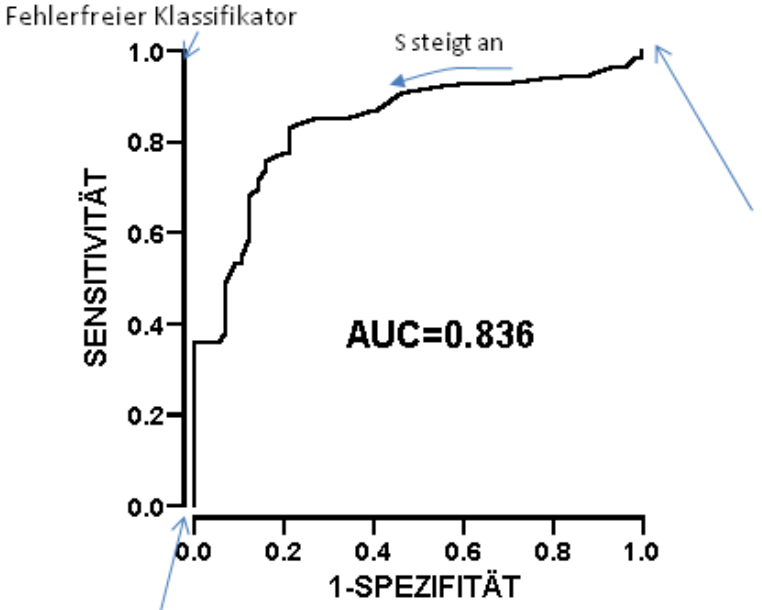
$$OR = \frac{Precision \times NPR}{(1 - Precision) \times (1 - NPR)} = \frac{TP \times TN}{FP \times FN}$$

ROC, AUC-ROC, AUC-PR

- Consider the classifier $\mathbf{sign}(f(x) - \alpha)$ for $-\infty < \alpha < \infty$.
- With $\alpha \to \infty$, $TP, FP \to 0$, $Recall \to 0$, $Specificty \to 1$, and Precision should be $>> 0$.
- With $\alpha \to -\infty$, $TN, FN \to 0$, $Recall \to 1$, $Specificty \to 0$, and Precision becomes the the percentage of class one.
  - ROC (Receiver operating characteristic). For the ROC-curve one varies α and plots Recall (y-axis) against (1-Specific $=$ FPR) (x-achis)
  - PR (Precision-Recall). Same, but we plot Precision as a function of Recall
- The AUC-ROC is the integral under the ROC curve. A random classifier has an AUC-ROC of 0.5, a perfect classifier of 1
- The AUC-PR is the integral under the PR curve. A random classifier has an AUC-ROC of around 0. The AUC-PR can be more relevant in highly unbalanced classes (search engines

Die Receiver Operating Characteristic (ROC) – Kurve

•Gibt mein Klassifikator eine Klassenwahrscheinlichkeit aus, dann entscheide ich mich für Klasse 0, wenn dieser Wert unter einem Schwellwert S ist und ansonsten entscheide ich mich für Klasse 1
•(0,0): S=1 ($\alpha=-\infty$) (1,1): S= ist 0 ($\alpha=\infty$) (0.3, 0.85): S=0.5 (Beispiel)



- Ich sage nur Klasse 1 voraus
- Mein Klassifikator sagt, dass jede Webseite ein Treffer ist
- Alle Patienten sind krank (Klasse 1)
- Spezifität: P(pred = 0 | y = 0) = 0 Sensitivität: P(pred = 1 | y = 1) = 1

- Ich sage nur Klasse 0 voraus
- Mein Klassifikator sagt, dass keine Webseite ein Treffer ist
- Alle Patienten sind gesund (Klasse 0)
- Spezifität: P(pred = 0 | y = 0) = 1 Sensitivität: P(pred = 1 | y = 1) = 0

•Das Integral unter der Kurve (area under curve, AUC-ROC) ist bei perfekter Klassifikation gleich 1 und bei Zufallsklassifikation gleich 0.5

Optimizing Specific Measures

- It is possible to derive algorithms which directly optimize certain measures, e.g., F-Measure, AUC