# GfKl Data Mining Competition 2005:
# Predicting Liquidity Crises of Companies

**Ilja Bezrukov, Thomas Deselaers, Daniel Keysers, Arne Mauser**
`<surname>@i6.informatik.rwth-aachen.de`

**GfKl 2005 – March 9, 2005**

**Human Language Technology and Pattern Recognition**
**Lehrstuhl für Informatik VI**
**Computer Science Department**
**RWTH Aachen University, Germany**

# Overview

1. Preprocessing
2. Training & Testing
3. Classification 1
4. Classification 2

## Part 1

# Preprocessing

**Ilja Bezrukov**

# Preprocessing

**Real-world data often not suited to achieve good results in classification**

**Problems**

- missing values
- outliers
- "insane distributions"
- noisy values

**Possible countermeasures**

- linear scaling
- creation of binary features
- outlier detection and substitution
- feature selection
- creation of histograms
- using the output of a classifier as new feature

# Data for the task

**Task:** **Prediction of a possible liquidity crisis of a company**

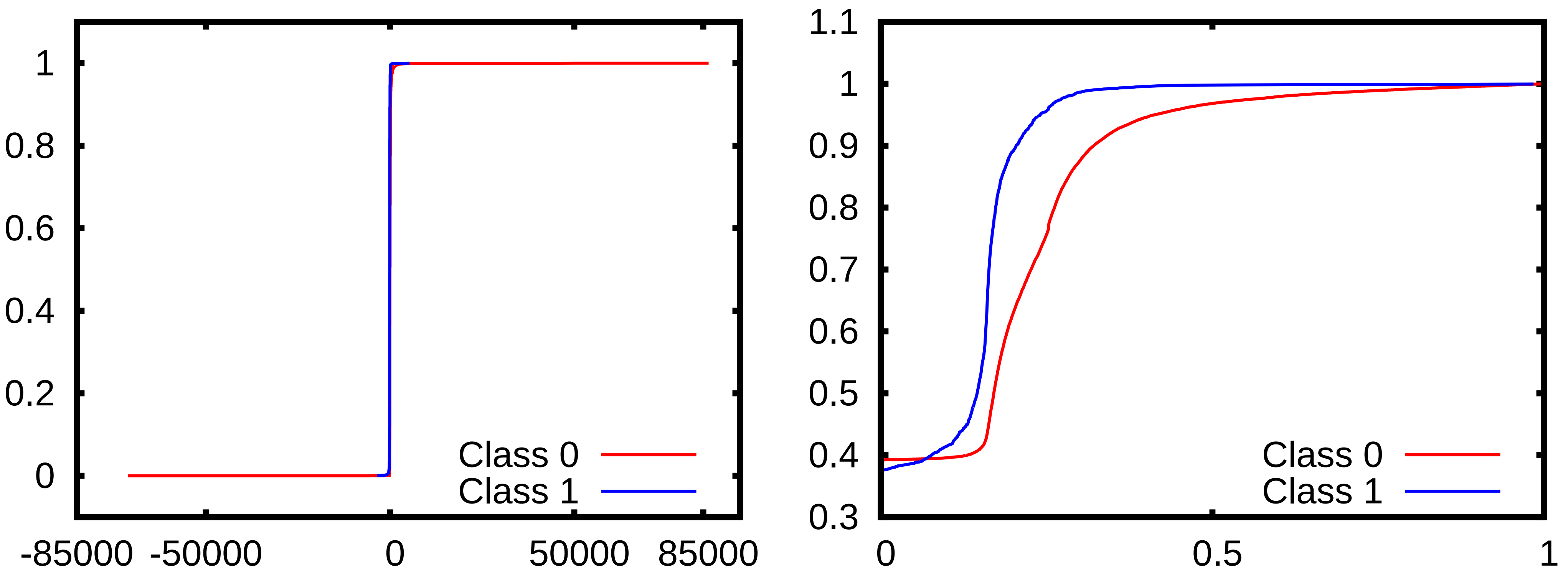
- **20000 training and 10000 test examples**

- **26 numerical features with unknown semantics**
- **24% of missing values per feature on the average**

- **2 classes, 0: no liqudity crisis, 1: liquidity crisis**
- **11% of the training examples are from class 1**

# Order preserving transformation

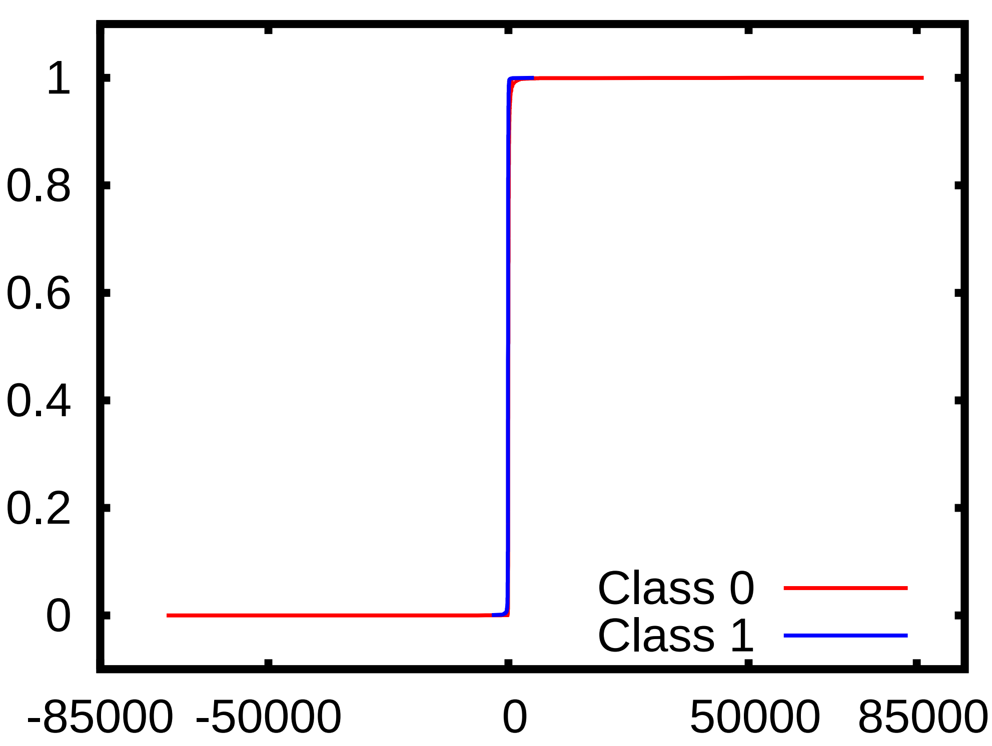
- contains a bin for each unique feature value
- feature values are replaced with the [0..1]-normalized index of their bins

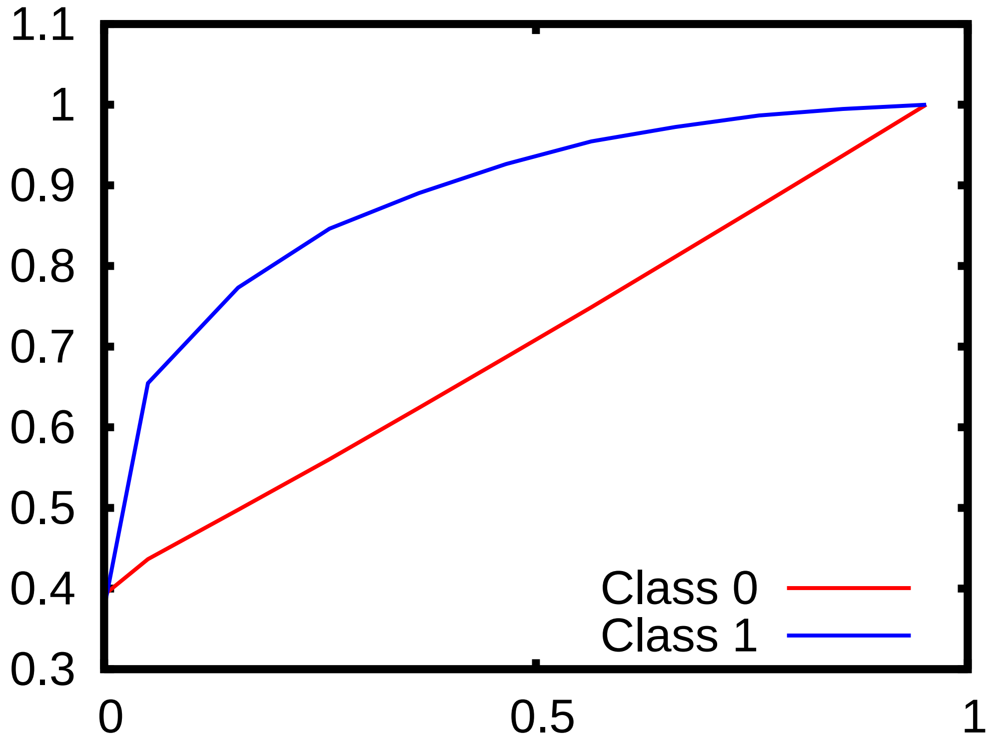



**result:** normalization of distances between neighboring feature values

# Equi-depth histogram

- contains 10 bins, each bin contains approx. the same number of elements
- feature values are replaced with the center of their bins




**result:** **discretization of the feature space and smoothing of feature values**

# Binary features

**Idea: Generalizing or emphasizing particularities**

- **missing values**
- **special values (e.g. zero)**
- **represent categorial values that are expressed numerically**

# Dataset creation


1. merge train and test data for transformation:


2. transform data completely
3. re-separate training and test data


**create two data sets:**

- **“quantile histogram”**
- **“equi-depth histogram”**

## Part 2

# Training and Testing

**Thomas Deselaers**

# Problem

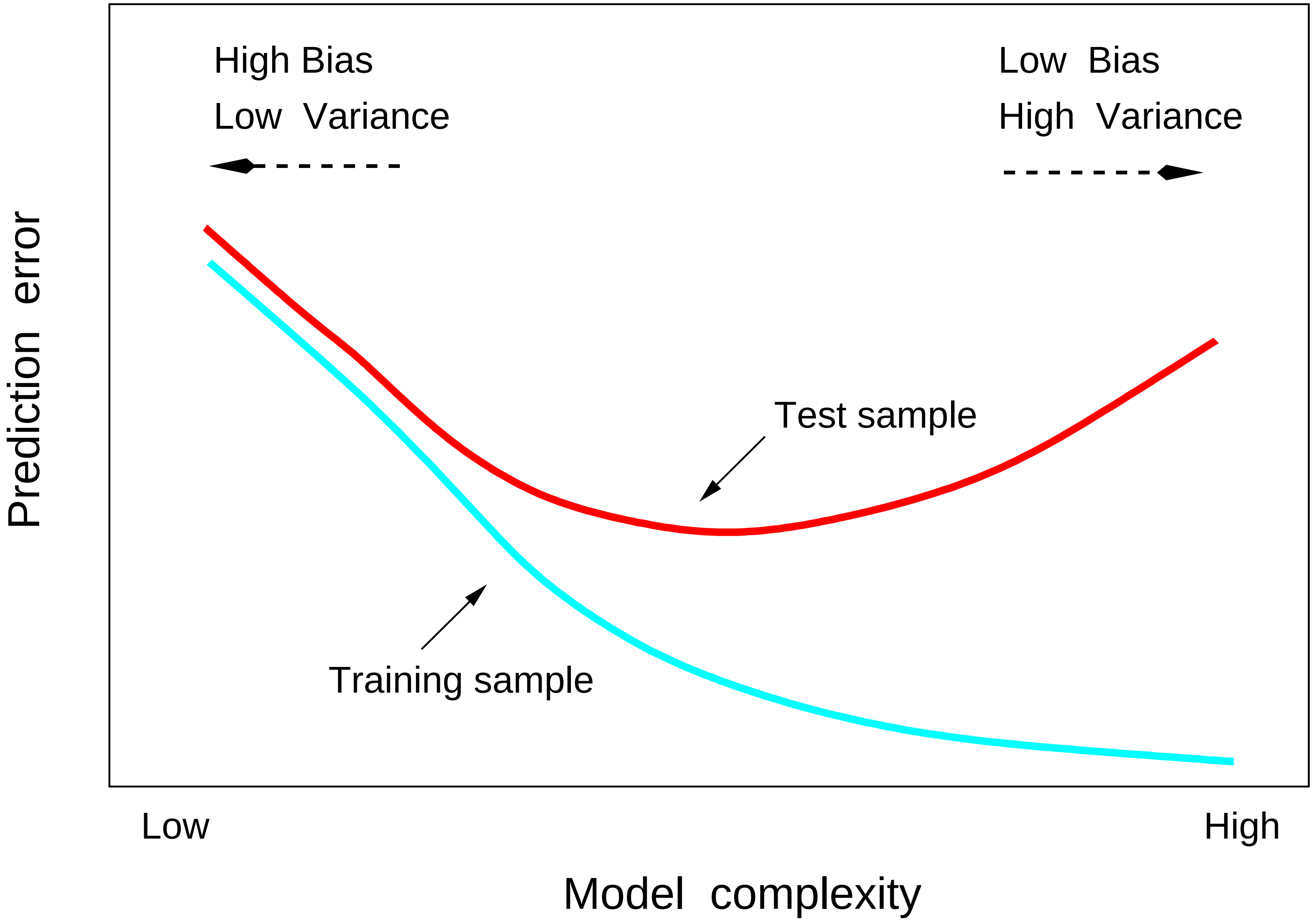
**free parameters:**

- **classifier**
- **preprocessing**
- **feature combination**
- **classifier combination**
- **parameters to classifier**

**problem: model assessment and selection**

- **how to select the best for each of the above parameters?**
- **how good generalizes a considered model to unseen test data?**

# Model Complexity vs. Classifier Performance

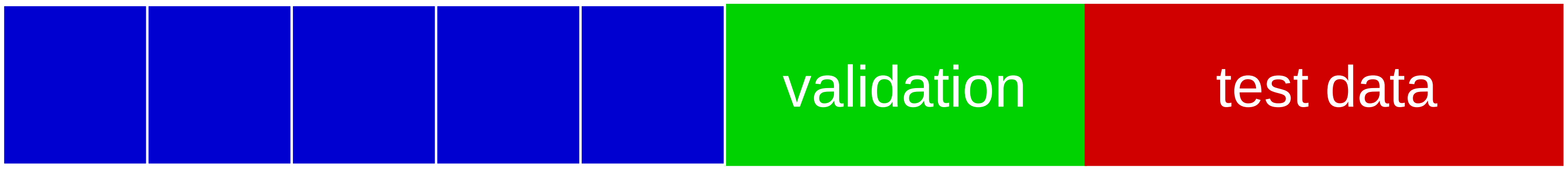


# Splitting the Data

**given situation:**


**take some data from the training data away:**

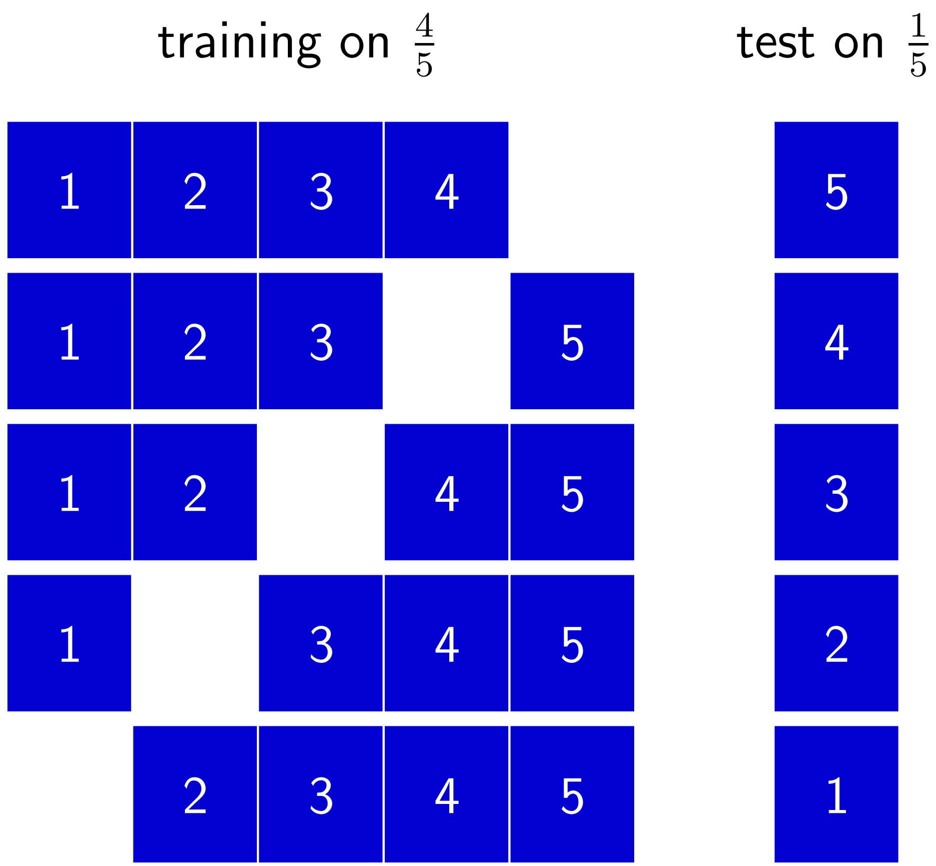

**perform experiments with 5-fold cross validation on remaining training data:**



# Cross-Validation

## assessing classifier performance:

- **cross-validation on reduced training data:**




## aim:

- **determine some “good” setups**

# Evaluation on Validation data

- **given the set of “good” setups**
- **evaluate these on the so-far unseen training data:**


**Result:**

- **performance measure (here: recall) on the validation data**
- **average these performance measures with the according measures from cross-validation experiments to select the “best” method**

# Probabilty of Improvement

**Question: What is the probability that one method is better than another?**
**(or: What is the probability that the one method is only by chance better than the other?)**

**comparing two methods:**


**probability of improvement:**

- **draw $M \cdot N$ independent samples from the data and measure the performance**
- **count how often method $A$ performs better than $B$ on the $M$ sample sets**

**The relative frequence of $A$ performing better $B$ on the $M$ sample sets is the bootstrap estimator for the probability of $A$ outperforming $B$.**

# Classification of the Testdata

**from our current setup**


**go back to the initial setup**


**and classify the test data using all training data.**

Part 3

# Classification 1

**Arne Mauser**

# Classification 1

**utilization of publicly available classifiers:**

- **Weka data mining toolkit for JAVA**
- **Netlab machine learning library for Matlab**

**classifiers were selected according to performance on crossvalidation**

**models employed in the approaches:**

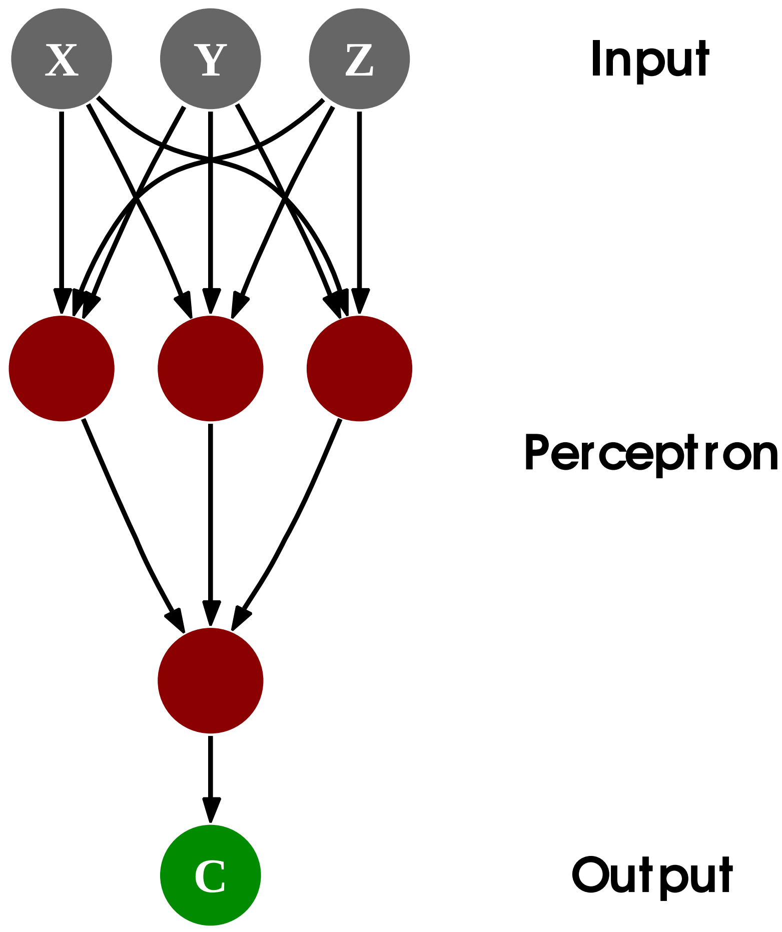
- **logistic model tree, LMT (Weka)**
- **alternating decision tree, ADT (Weka)**
- **multilayer perceptron, MLP (Netlab)**
- **logistic regression (Netlab)**

# Multi-Layer Perceptron

**single perceptron:**
**compute single output (class) value as function of the weighted sum of input values**

**multi-layer perceptron:**
**connect the outputs of perceptrons with inputs of other perceptrons**

**approach used 2 layers of perceptrons**



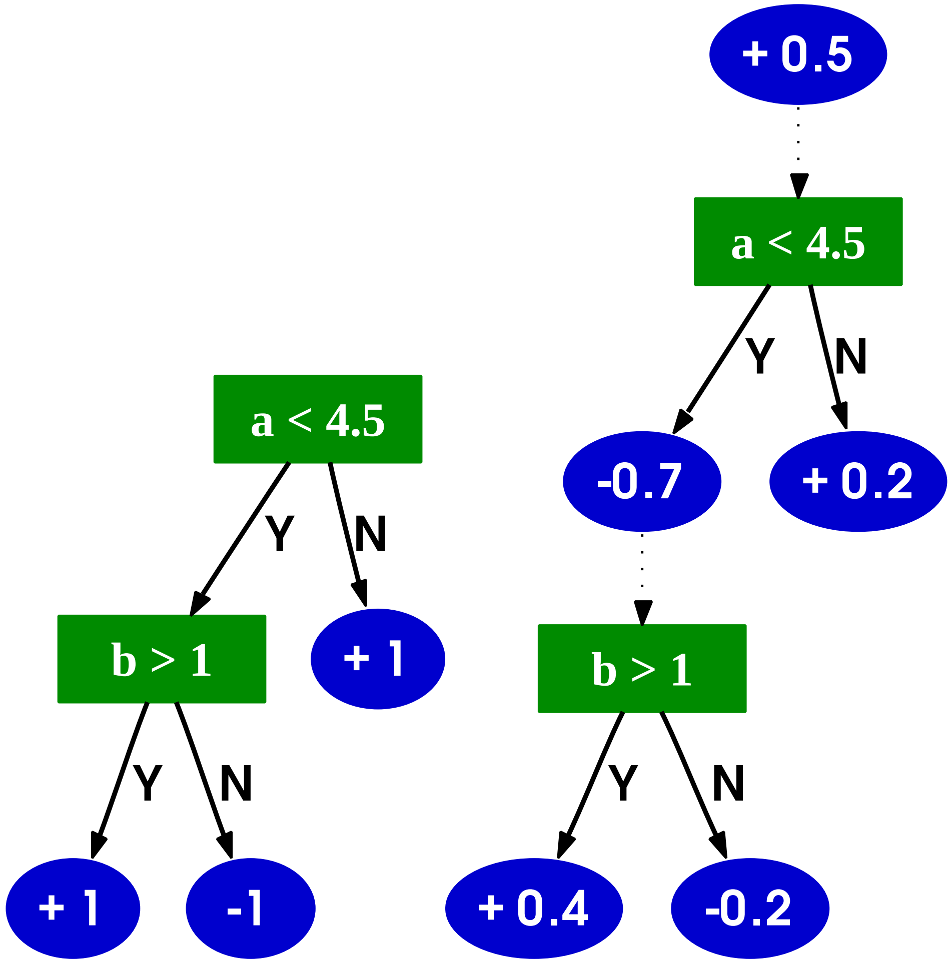
# Alternating Decision Tree

**decision tree with alternating levels of decision nodes and prediction nodes**

**training performed using modified AdaBoost algorithm**

**provides confidence measure for predictions**

**classification by combining all predictions along the path from the root to the leaf**



# Logistic Model Tree

**decision tree with logistic regression functions at the leaves**

**combine the advantages of logistic regression**

- **restricted model space**
- **models posterior probibilities $p(k|x)$**
- **stability of fitting process**

**and decision trees**

- **less restricted model space**
- **capture non-linear patterns**

**decision tree learning algorithm based on C4.5**

**regression estimated using LogitBoost algorithm**

## Part 4

# Classification 2

**Daniel Keysers**

# Naive Bayes

**naive Bayes classifier:**
**assumption of (conditional) feature independence**

$$Pr(x|k) \; = \; \prod_i Pr(x_i|k)$$

**often good results despite the obvious incorrectness of assumption**

**Can we use a similar naive combination for the posteriors $Pr(k|x_i)$ ?**

# Classifier Combination of Feature Posteriors

view $Pr(k|x_i)$ as (weak) classifier (cp. boosting)
use classifier combination on these

classifier combination:
usually sum rule gives better results than product rule

leads to “naive posterior” rule:

$$Pr(k|x) \;\propto\; \sum_i Pr(k|x_i)$$

estimate importance of features in log-linear model:

$$Pr(k|x) \;\propto\; \exp\left(\sum_i \lambda_i Pr(k|x_i)\right)$$

here:
estimate $Pr(k|x_i)$ as relative frequencies
after histogramization of features $x_i$

# Maximum Entropy

**resulting distribution has the log-linear or exponential functional form:**

$$p_\Lambda(k|x) = \frac{\exp\left[\sum_i \lambda_i f_i(x,k)\right]}{\sum_{k'} \exp\left[\sum_i \lambda_i f_i(x,k')\right]}$$

**optimization problem is convex and has a unique global maximum**

**algorithm to compute the global maximum given a training set:**

$\longrightarrow$ **generalized iterative scaling**

**crucial problem in maximum entropy modeling:**

**– choice of the appropriate feature functions $\{f_i\}$**

**– here: use $Pr(k|x_i)$**

# Classifier Combination

**widely used method to smooth disadvantages of different classifiers**

- **parallel combination method**
- **different classifiers (and maybe parameters)**
- **using same features & training data**
- **combination method: sum of posteriors**

**one of the best methods used in the competition combines:**

- **naive posterior & maximum entropy**
- **alternating decision tree**
- **logistic regression**

# Results

| Method | CV-Score | V-Score | testdata-score | rank |
| --- | --- | --- | --- | --- |
| combination | 1445 | 360 | 894 | 2 |
| LMT | 1408 | 358 | 894 | 2 |
| MLP | 1395 | 358 | 884 | 6 |
| ADT | 1426 | 357 | 883 | 7 |
| NB-ME | 1412 | 362 | 881 | 9 |
| maximum | 1796 | 448 | 1111 | - |
| winner (D.Vogel) | | | 896 | 1 |

# Conclusion

– use appropriate data preprocessing

– avoid overfitting (cross-validation, hold-out, classifier combination)

– have a set of suitable classifiers ready for evaluation

– it is not necessary to use support vector machines

**Thank you for your attention!**

# Maximum Entropy

**idea: we have information about a probability distribution from training data**
**$\rightarrow$ choose consistent distribution and with highest possible entropy**

**feature functions:** $$(x, k) \longmapsto f_i(x, k)$$

**maximum entropy principle:**

$$Pr'(k|x) = \arg \max_{Pr(k|x)} \Big\{ -\sum_n \sum_k Pr(k|x_n) \log Pr(k|x_n) \Big\}$$

**with the requirements:**
**– normalization constraint for each observation $x$:**

$$\sum_k Pr(k|x) = 1$$

**– feature constraint for each feature $i$:**

$$\sum_n \sum_k Pr(k|x_n) f_i(x_n, k) = \sum_n f_i(x_n, k_n)$$