Multimedia Data Mining (SoSe 17)

Pattern Recognition Basics - Supervised Learning

Lecture 03

Dr. Damian Borth

German Research Center for Artificial Intelligence (DFKI)

Administration

- News:

S7608 - EXPLORING THE LATENT VISUAL SPACE BETWEEN ADJECTIVES WITH GENERATIVE ADVERSARIAL NETWORKS

Generative adversarial networks (GANs) have been applied for multiple cases, such as generating images and image completion. One interesting feature of GANs is the... View More

50-minute Talk

Federico Raue - Researcher, German Research Center for Artificial Intelligence (DFKI)

Damian Borth - Director Deep Learning Competence Center, German Research Center for Artificial Intelligence (DFKI)

- Lecture Website:
  https://madm.dfki.de/teaching/mdm2017
  - slides for lectures are online
    (current lecture will be posted by today)
  - user: “mdm2017user”, password: ”ai4good!”
  - tutorial tomorrow, 8:15-9:45, room 36-265

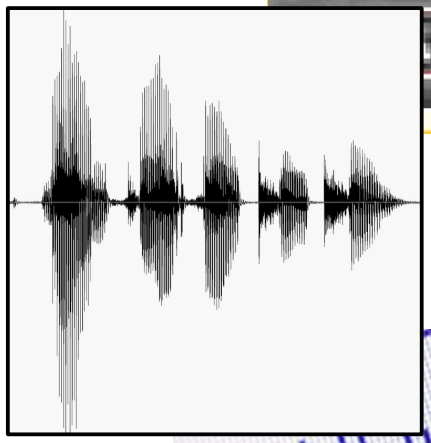
Supervised Learning

- goal: a mapping of real-world objects (for example, images) to pre-defined categories (for example, „face“ vs. „non-face“)
  - as identified by pre-defined "labels"

- Because the relationship between the observed data and the target categories is often too complex to be defined manually („semantic gap“), the mapping should be *learned* automatically!
  - data-driven i.e "samples" representing the category

Applications of Supervised Learning

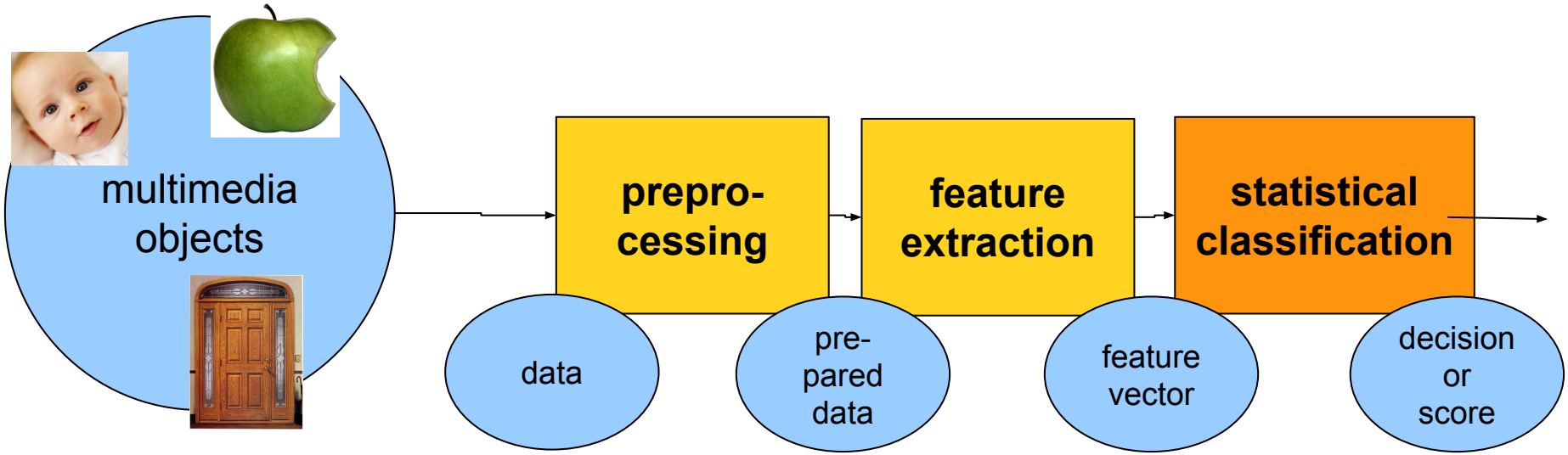
- OCR
- speech recognition
- SPAM filtering
- credit scoring
- biometrics
- …
- …
- ...
- **multimedia retrieval**



Applications in MIR

- image and video categorization
- image annotation / concept detection
- face detection
- face recognition
- speech analysis
- printing technique recognition
- pornography detection
- duplicate detection
- ...

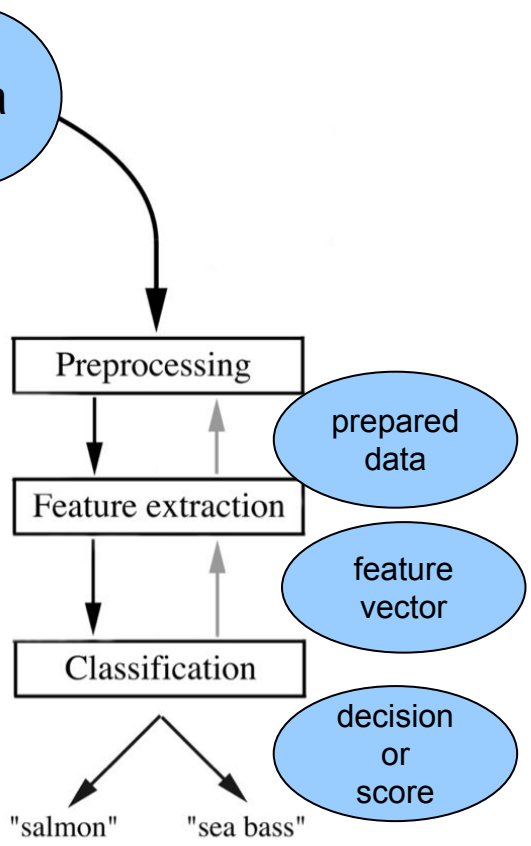
System Setup (traditional)



- **Supervised learning / statistical classification**
  - given: feature vector x (usually, $x \in R^d$)
  - given: class random variable $C \in \{1, ..., K\}$
  - task: make a decision $x \rightarrow C$
  - often, we infer *scores* indicating class membership. For example, probabilistic scores: $P(C=c|x)$, or short $P(c|x)$

Fish Example

X=



C ∈{salmon, sea bass}

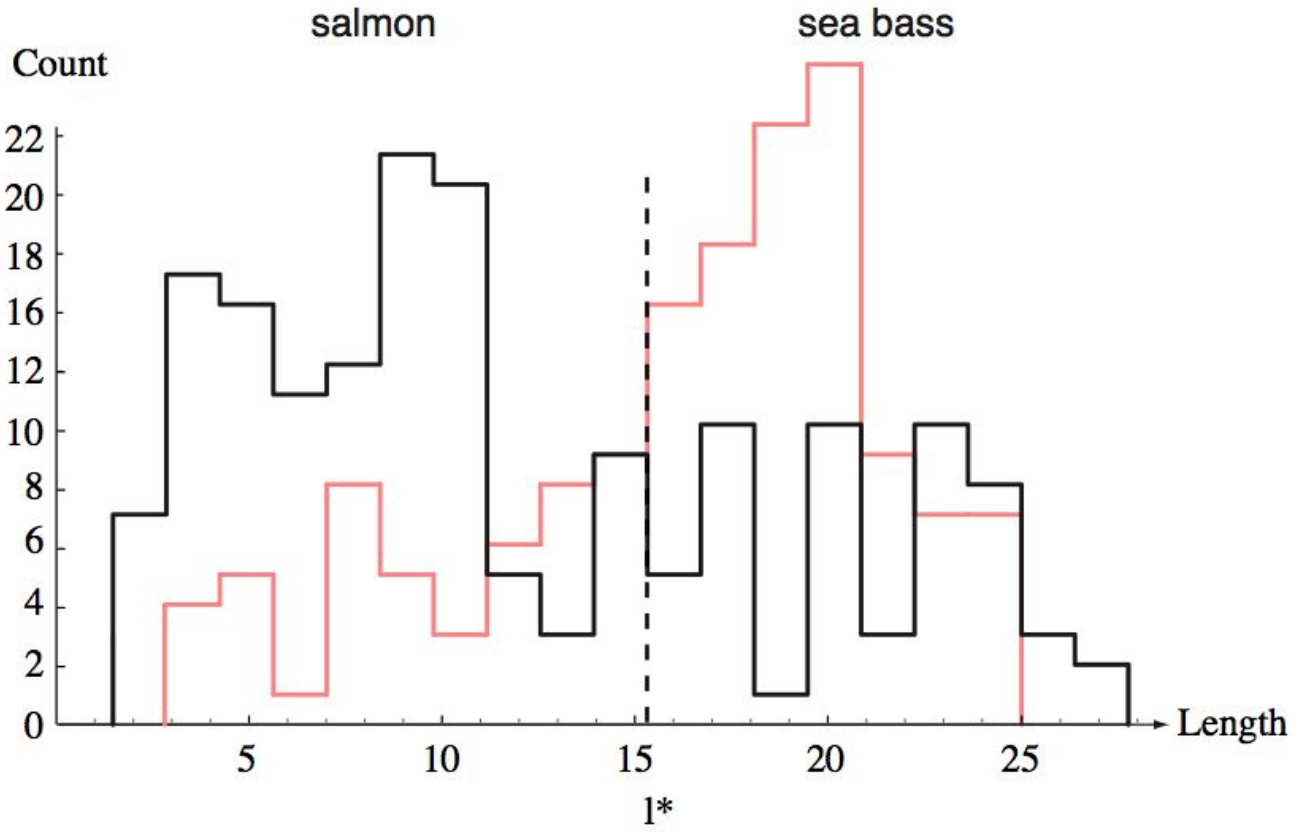
Duda / Hart / Stork: Chapter 1

Fish Example

- assume we have no camera
  - best guess: ?
  - follow distribution of fish classes in ocean
    - apply prior fishermen's knowlege about fish
- assume our factory has a sophisticated camera system extracting features for:
  - length "l"
  - lightness "x"
  - follow evidence as given by observation
    - know distribution of different features per fish classes (condition)
- There will be a dedicated lecture to vis/audio features

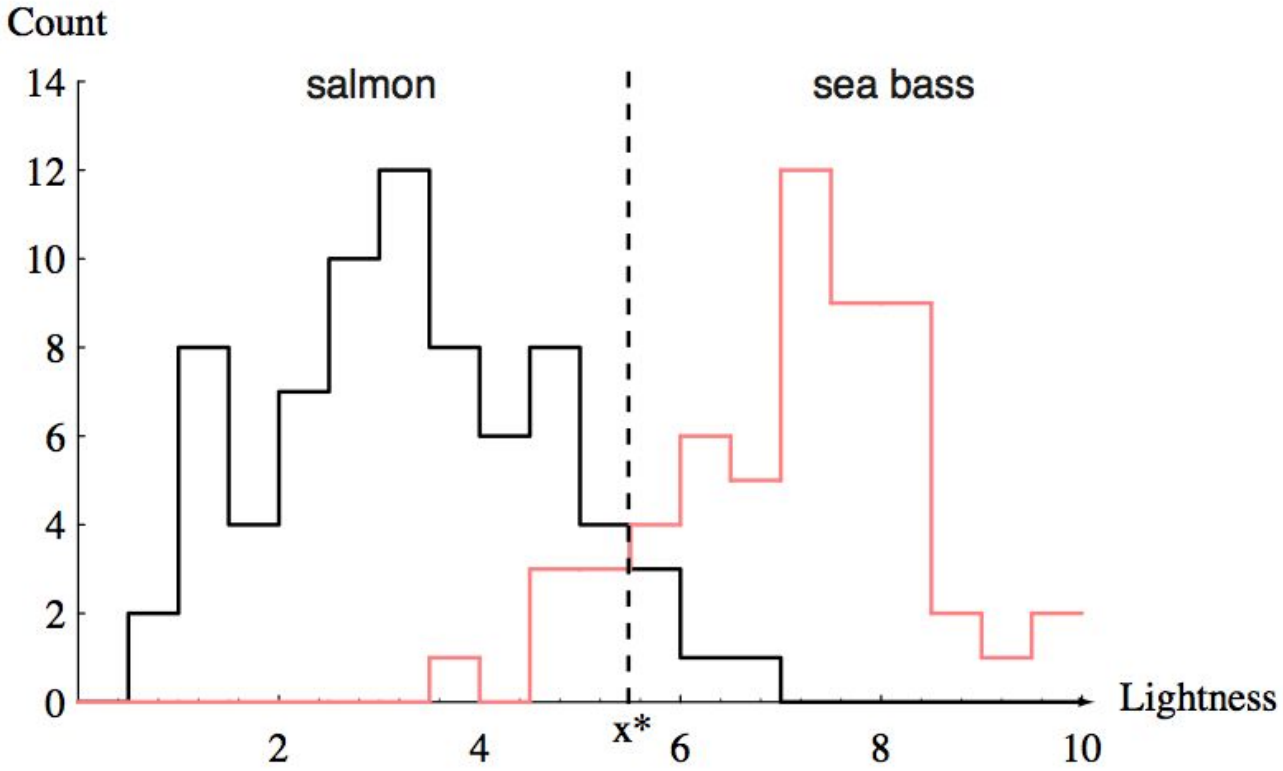
Duda / Hart / Stork: Chapter 1

Fish Example



Duda / Hart / Stork: Chapter 1

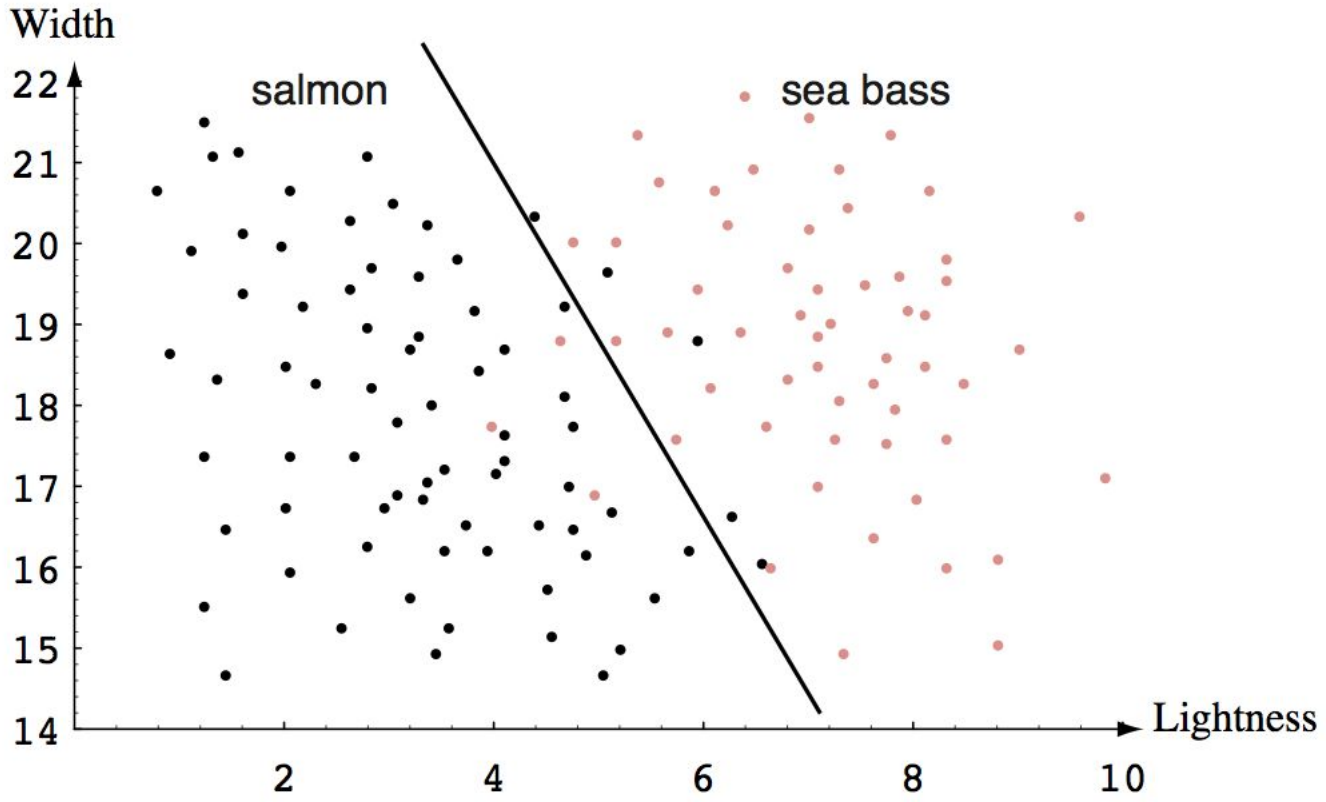
Fish Example



Duda / Hart / Stork: Chapter 1

Fish Example

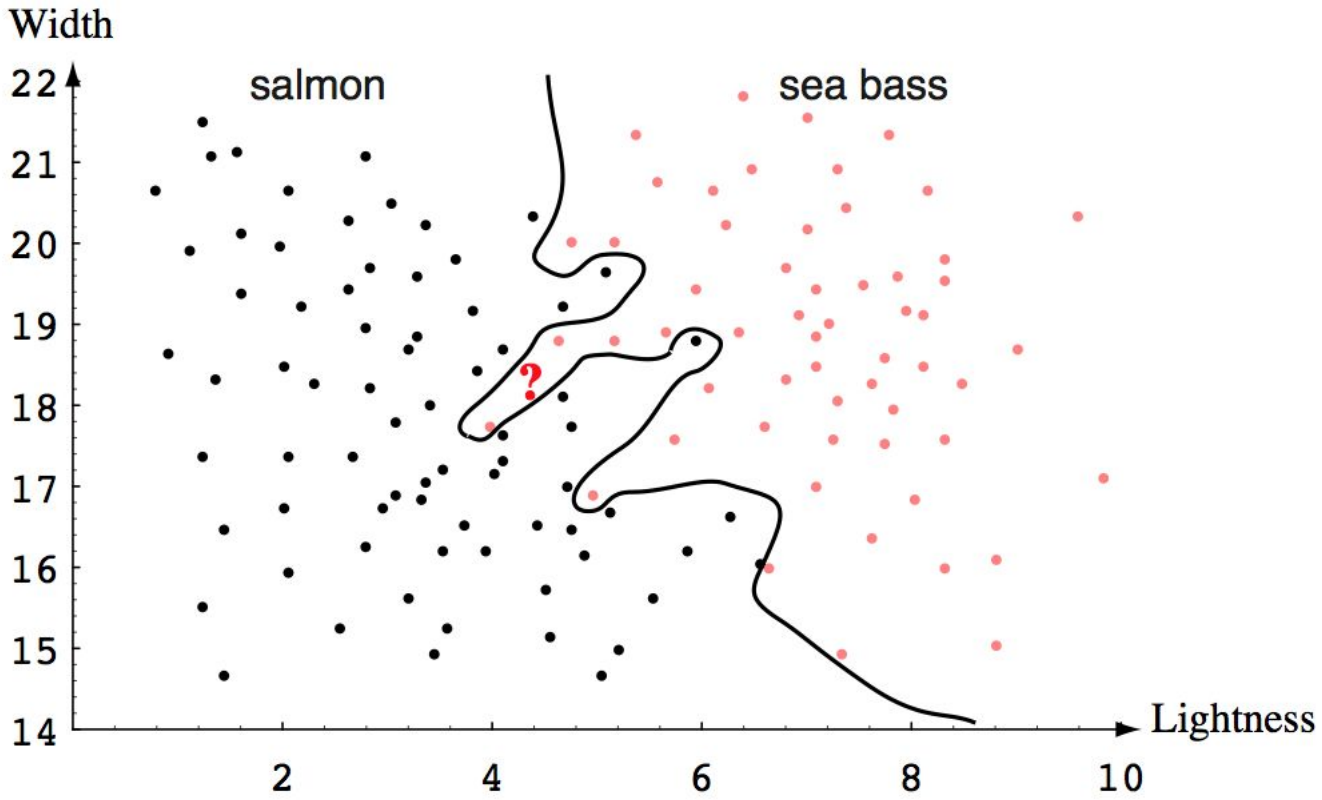
$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}.$$



Duda / Hart / Stork: Chapter 1

Fish Example

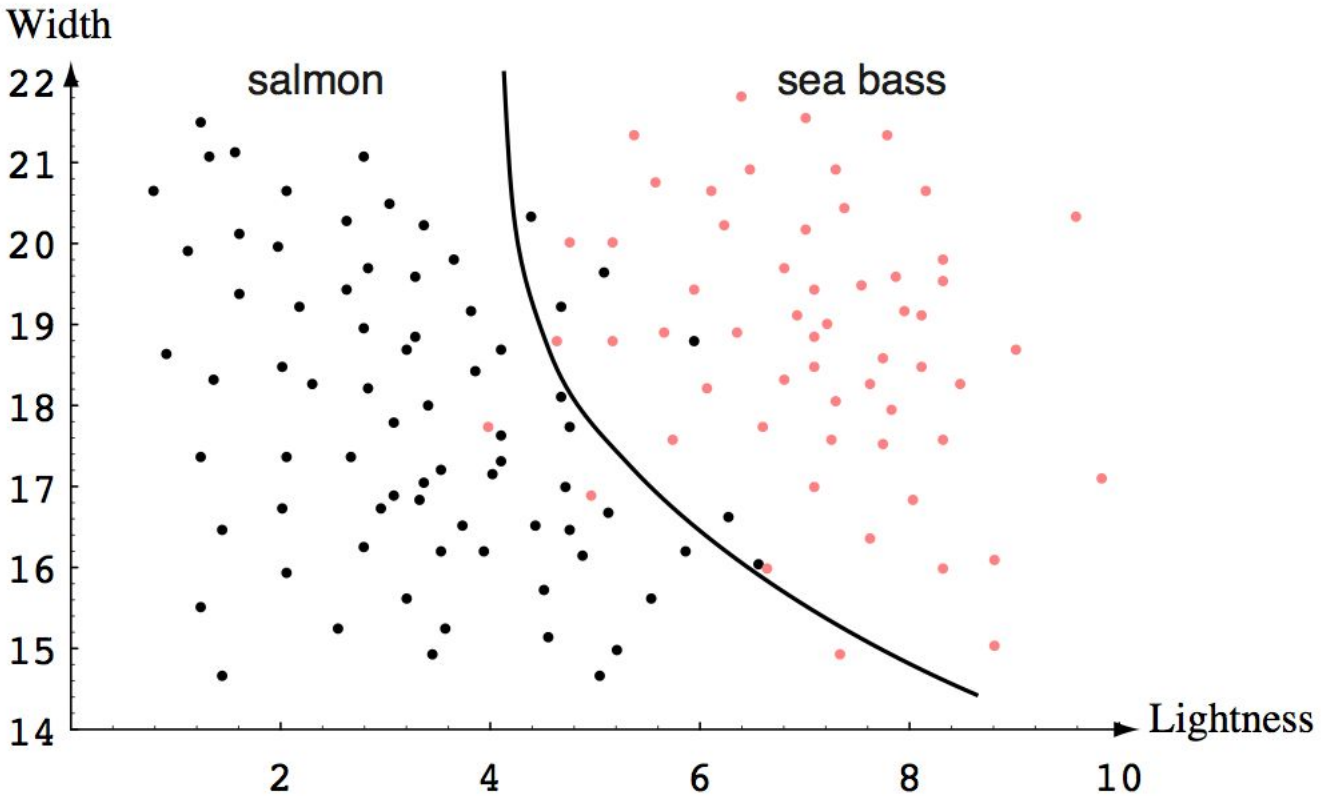
$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}.$$



Duda / Hart / Stork: Chapter 1

Fish Example

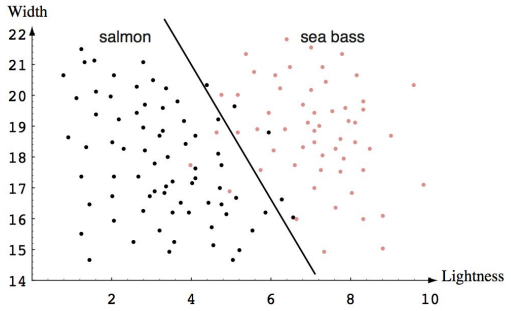
$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}.$$



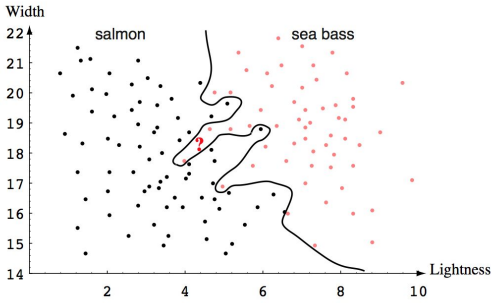
Duda / Hart / Stork: Chapter 1

Fish Example

feature $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$.

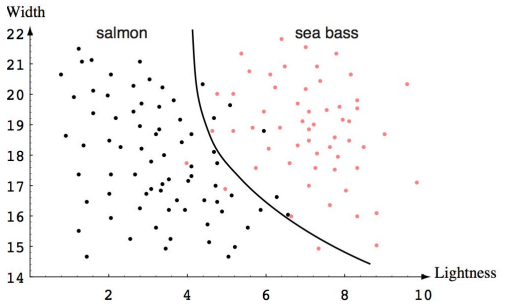


linear
decision boundary



non-linear
decision boundary

!!! “overfitted” !!!



non-linear
decision boundary

“regularized”

Duda / Hart / Stork: Chapter 1

Agenda

1. What is supervised learning ?
2. **How to build an “optimal” classifier ?**
3. What kind of classifiers are there ?
   a. Gaussian CCDs
   b. Nearest Neighbors
   c. Logistic Regression
4. How to combine different classifiers?

Decision Theory

Decision Theory - Classification

- What can we say about the optimal decision of a classifier?
  - assume we could compute P(C=c|x) ∀c
  - P(C=c|x) is called the „class posterior“

- Idea: **minimize the probability of error**
  - the optimal decision becomes:

$$c^* = \arg\max_c P(c|x)$$

- *extension with* cost *(Duda/Hart/Stork):*
  Spam Filtering / Poisoned Fish / Face Detect.

Decision Theory - Ranking

- Now let's assume we build a retrieval system instead of a classification system
  - samples to retrieve: $x_1,..,x_n$ with labels $c_1,..,c_n \in \{0,1\}$
  - label $c_i=1$ (*relevant*), $c_i=0$ (*non-relevant*)
  - the labels are unknown, but estimates $P(c_i=1|x_i)$ are given

- Find the best **ranking** $\pi : \{1,...,n\} \rightarrow \{1,..,n\}$
  - π maps rank r to document $x_{\pi(r)}$
  - user inspects retrieval results up to rank r*

- Idea: maximize the **expected number of relevant documents** retrieved

Decision Theory - Ranking

- Now let's assume we build a retrieval system instead of a classification system
  - samples to retrieve: $x_1,..,x_n$ with labels $c_1,..,c_n \in \{0,1\}$
  - label $c_i=1$ (*relevant*), $c_i=0$ (*non-relevant*)
  - the labels are unknown, but estimates $P(c_i=1|x_i)$ are given

- Find the best **ranking** $\pi : \{1,...,n\} \rightarrow \{1,..,n\}$
  - π maps rank r to document $x_{\pi(r)}$
  - user inspects retrieval results up to rank r*

- Idea: max **vant documen**

$$\pi^* = \arg\max_{\pi} \quad E\left[\sum_{r=1}^{r^*} c_{\pi(r)}\right]$$

$$\pi^* = \arg\max_{\pi} \quad E\left[\sum_{r=1}^{r^*} c_{\pi(r)}\right] = \sum_{r=1}^{r^*} E\left[c_{\pi(r)}\right]$$

$$= \arg\max_{\pi} \sum_{r=1}^{r^*} \quad \left[1 \cdot P(c_{\pi(r)} = 1 | x_{\pi(r)}) + 0 \cdot P(c_{\pi(r)} = 0 | x_{\pi(r)})\right]$$

$$= \arg\max_{\pi} \sum_{r=1}^{r^*} P(c_{\pi(r)} = 1 | x_{\pi(r)})$$

- solution: place the documents x_i with highest $P(c_i=1|x_i)$ at the top r* ranks

- Conclusion:

For both classification and retrieval systems to make optimal decisions, we need to know P(c|x)

- Problem: to know P(c|x) for all x, we would require „infinitely“ dense training samples over the feature space.

- In practice, we need to „extrapolate“ P(c|x) from a finite training set by **supervised learning**.

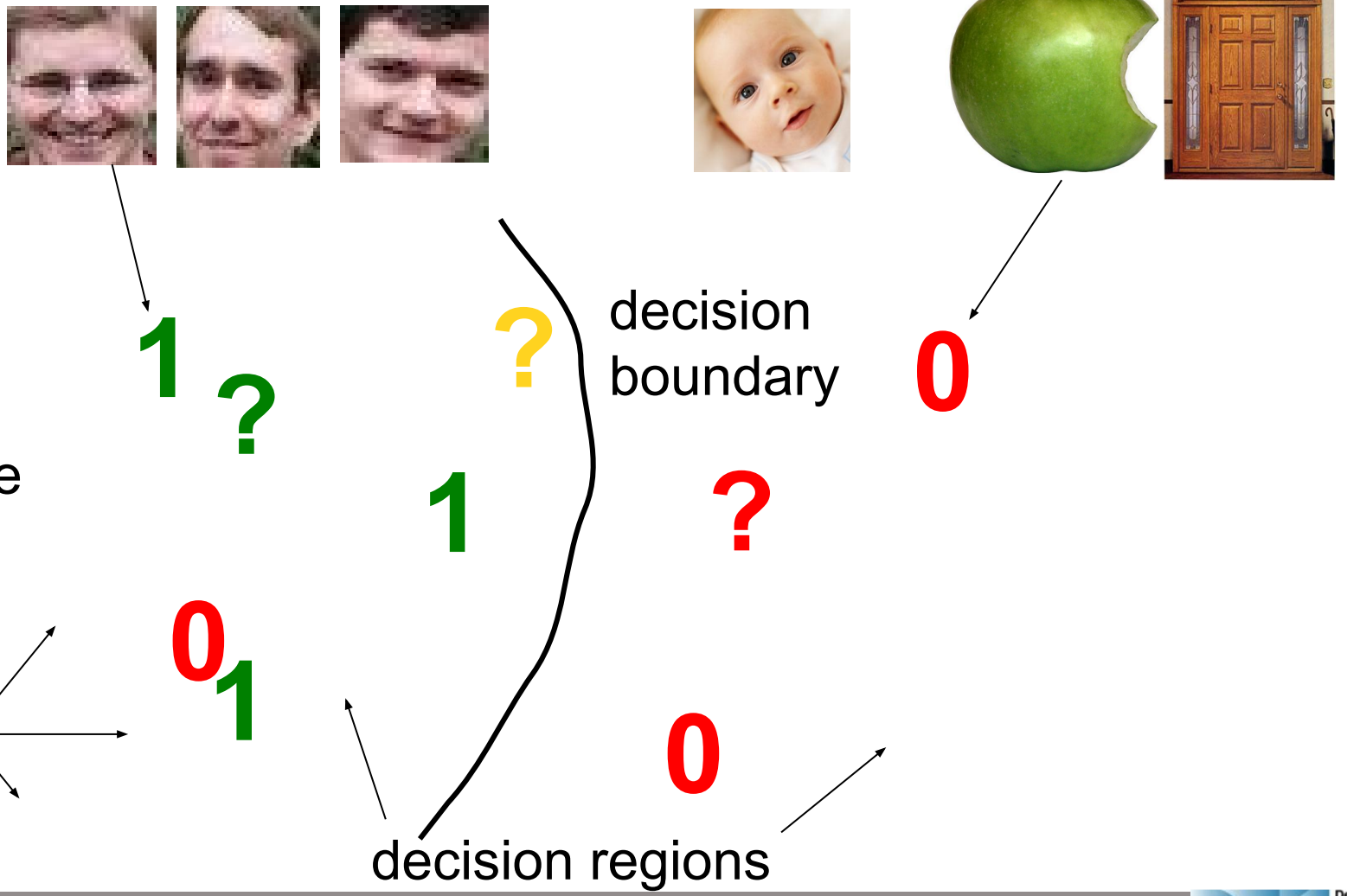
Supervised Learning - Setup

- The supervised learning setup – given is a **training set X**
  - **samples** $x_1,..,x_n \in R^d$
  - **labels** $c_1,..,c_n \in \{1,..,K\}$

- Because we have labels for all training samples, we speak of **supervised** learning
  - labels for some (but not all) samples: *semi-supervised* learning
  - no labels: *unsupervised* learning

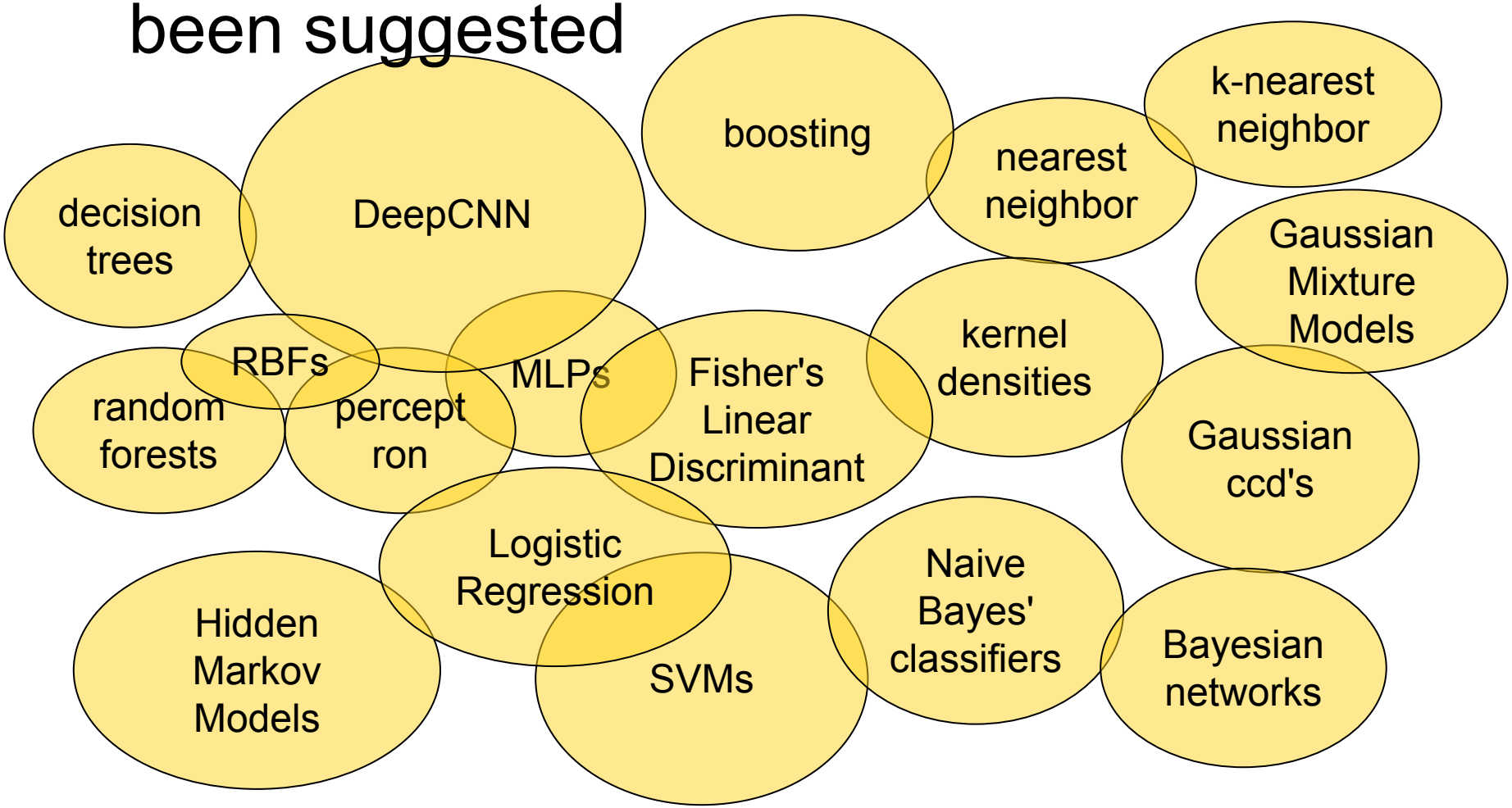
Approach – Supervised Learning

- Example illustration



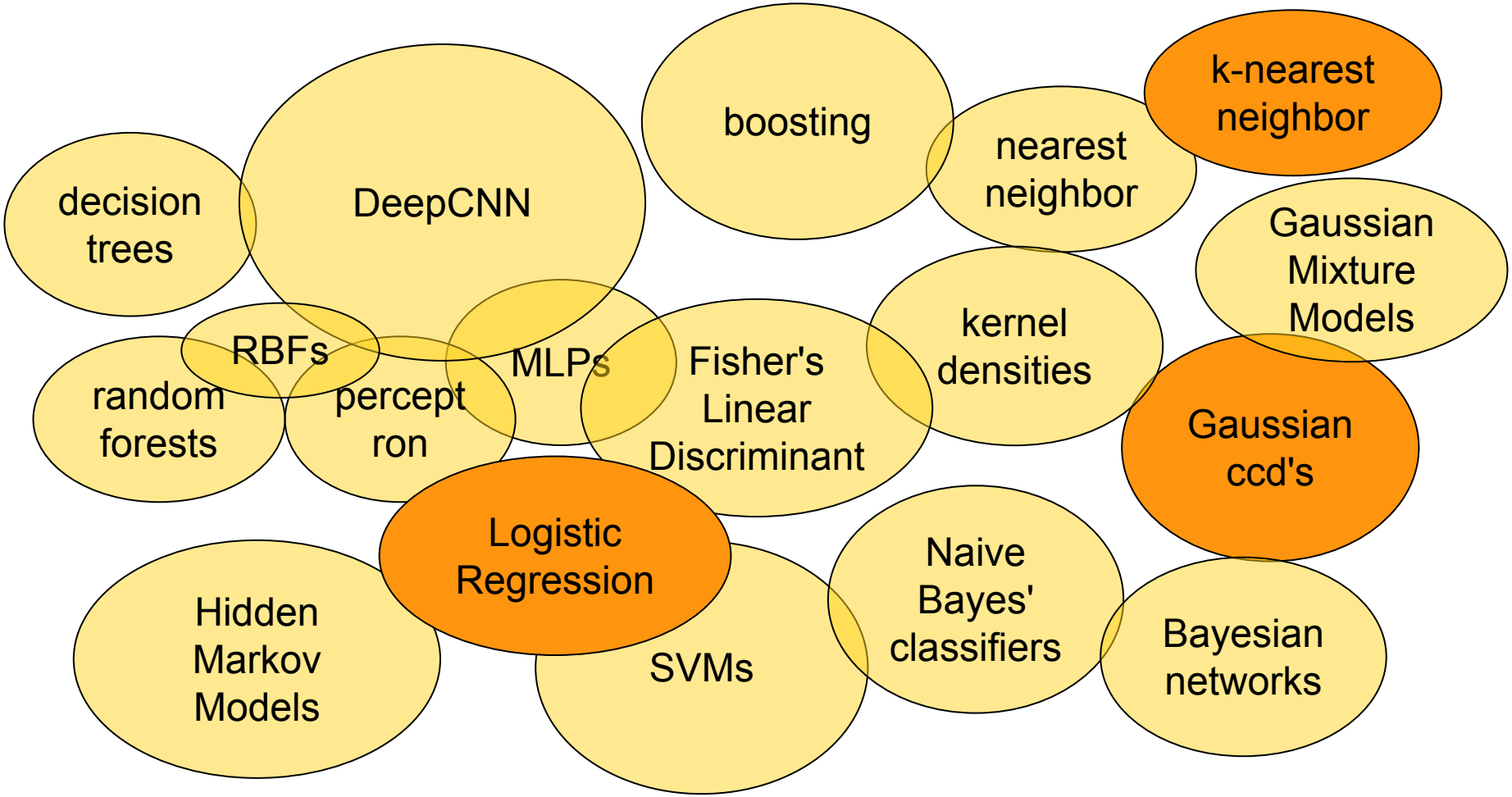
Classification – Approaches

- A lot of different statistical models have been suggested



Classification – Approaches

- Let's have a look at



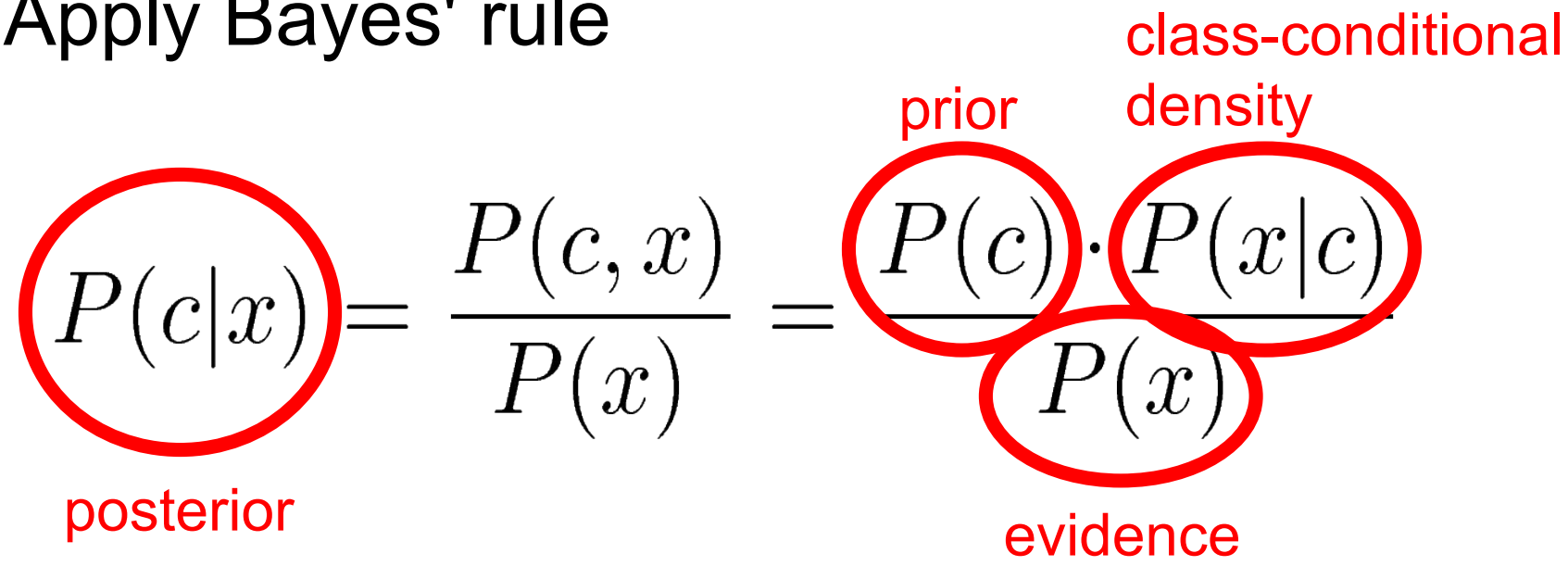
Agenda

1. What is supervised learning ?
2. How to build an “optimal” classifier ?
3. **What kind of classifiers are there ?**
   a. Gaussian CCDs
   b. Nearest Neighbors
   c. Logistic Regression
4. How to combine different classifiers?

An Example of a Generative Model - Gaussian Class-conditional Densities

Decision Theory and Bayes' Rule

- We want to estimate P(c|x)
- Apply Bayes' rule



- two kinds of methods
  - *discriminative*: only model P(c|x)
  - *generative*: model P(c,x)

Class-conditional Densities

- Let us focus on generative models for now
- Most important: class-conditional densities
  - two classes (1/0, relevant/non relevant)
  - evidence: marginalizes out / same for all classes
  - prior: has no influence on ranking
  - *(assume P(c)=1/2 ∀c)*
- simplify

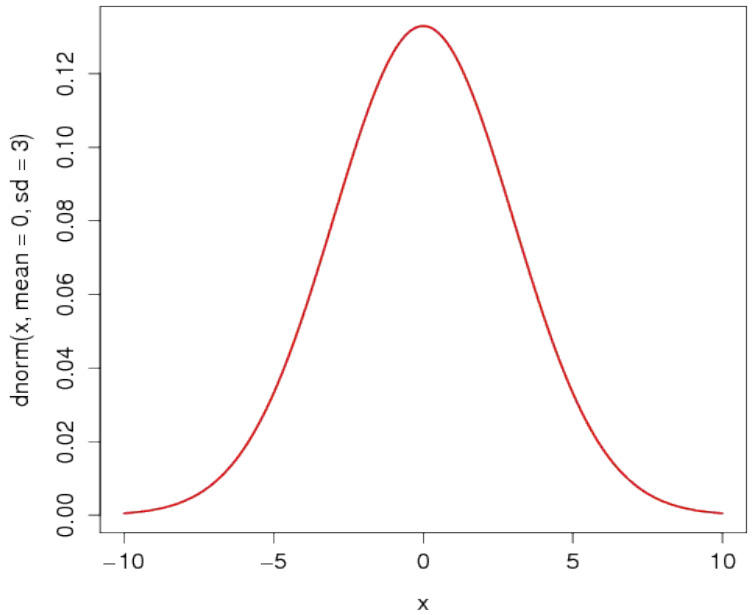
$$P(c=1|x) = \frac{\frac{1}{2} \cdot P(x|c=1)}{\frac{1}{2} \cdot P(x|c=1) + \frac{1}{2} \cdot P(x|c=0)}$$

$$= \frac{P(x|c=1)}{P(x|c=1) + P(x|c=0)}$$

Example: Gaussians

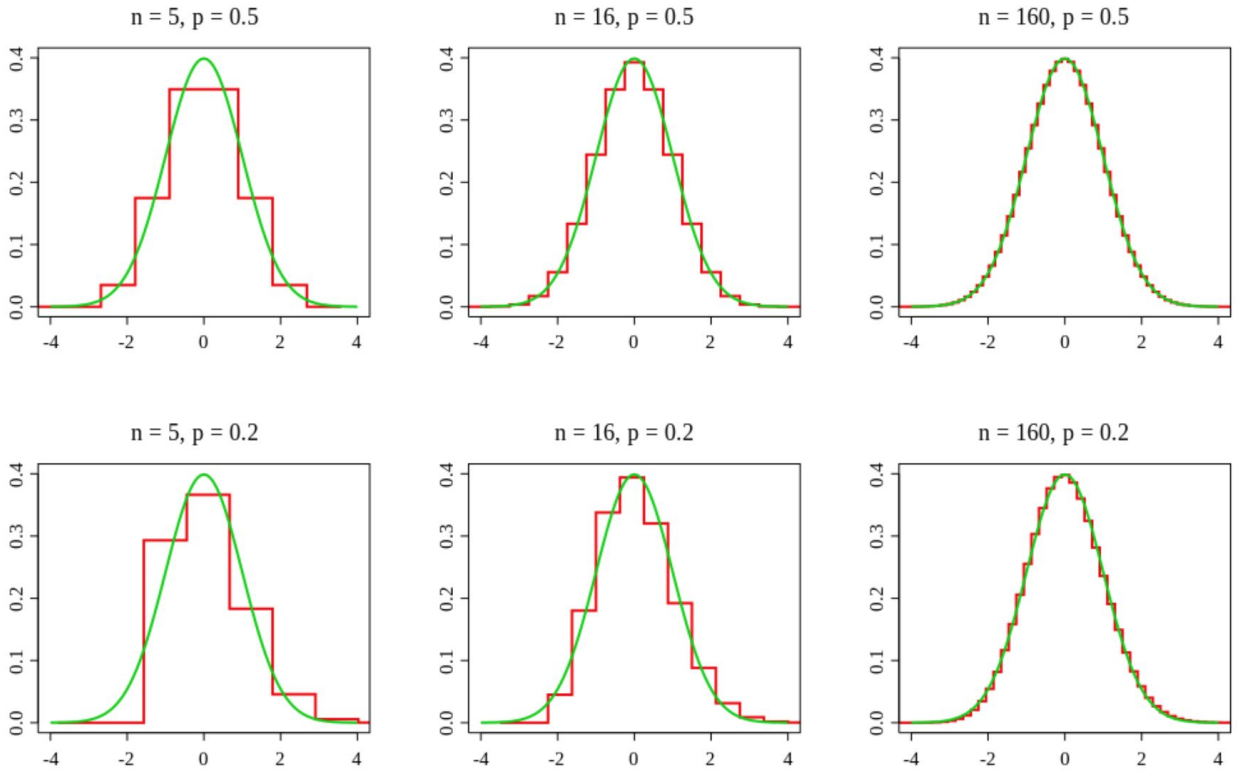
- frequently used class-conditional densities (CCDs): **normal (= Gaussian) distribution**

$$\mathcal{N}(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\{-\frac{(x-\mu)^2}{2\sigma^2}\}$$



“Central Limit Theorem”

- *“when **independent random variables** are added, their sum tends toward a **normal distribution**”*



Source: https://de.wikipedia.org/wiki/Zentraler_Grenzwertsatz

Gaussian CCDs: Example

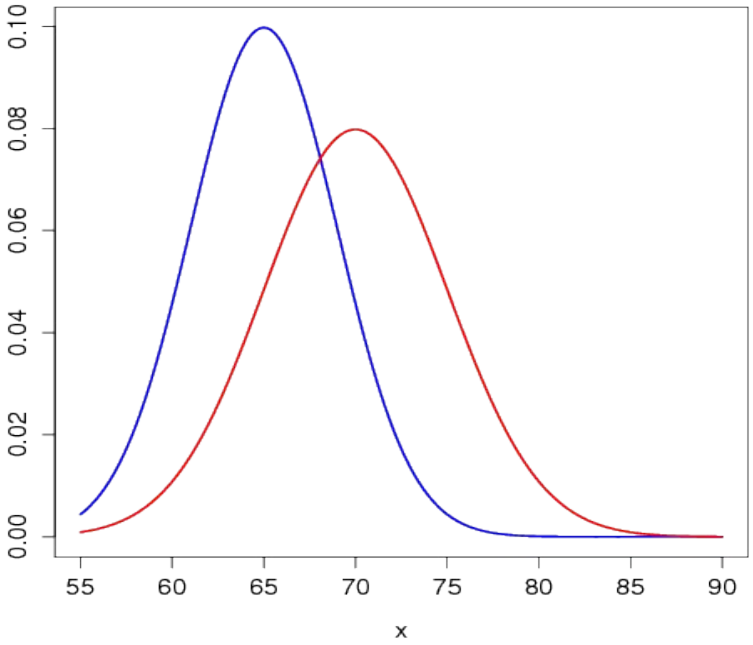
- example: determine the gender c of a person (c=0/1 for women/men) by his/her height x

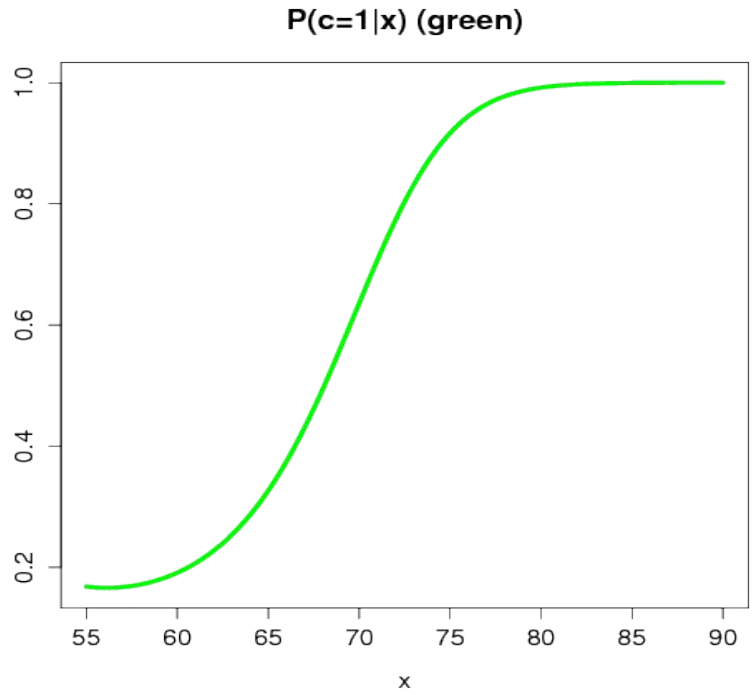
$$P(x|c=0) = \mathcal{N}(x;\mu_0,\sigma_0)$$
$$P(x|c=1) = \mathcal{N}(x;\mu_1,\sigma_1)$$

„The Living Histogram“

Gaussian CCDs: Example

$$P(c=1|x) = \frac{\mathcal{N}(x;\mu_1,\sigma_1)}{\mathcal{N}(x;\mu_1,\sigma_1)+\mathcal{N}(x;\mu_0,\sigma_0)}$$

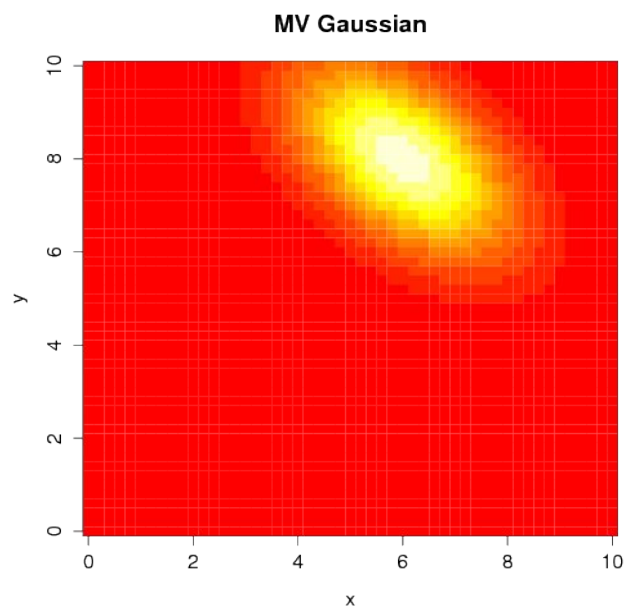


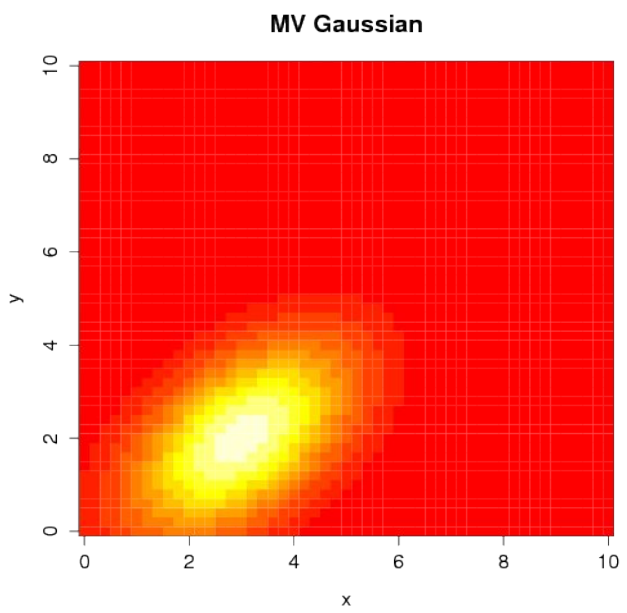


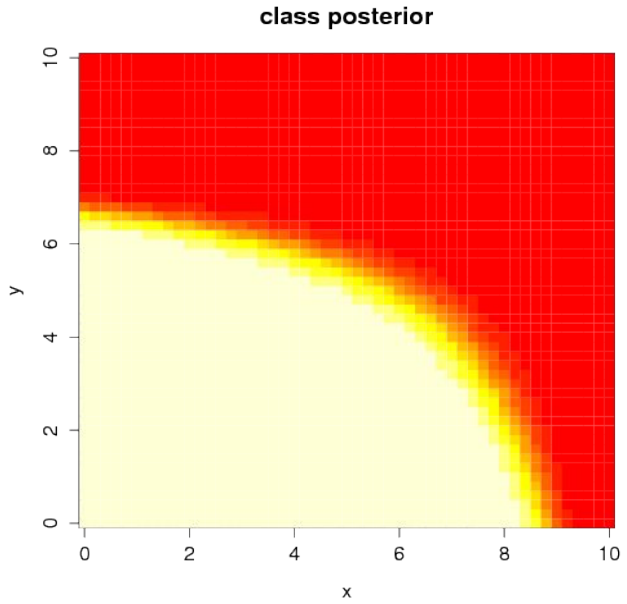
Multi-variate Gaussians

- This can be extended to multiple dimensions using **multi-variate** Gaussians (with dimension d)

$$p(x;\mu,\Sigma) = \frac{1}{(2\pi)^{d/2} \cdot |\Sigma|^{1/2}} \cdot exp\{-\frac{1}{2}(x-\mu)\Sigma^{-1}(x-\mu)\}$$

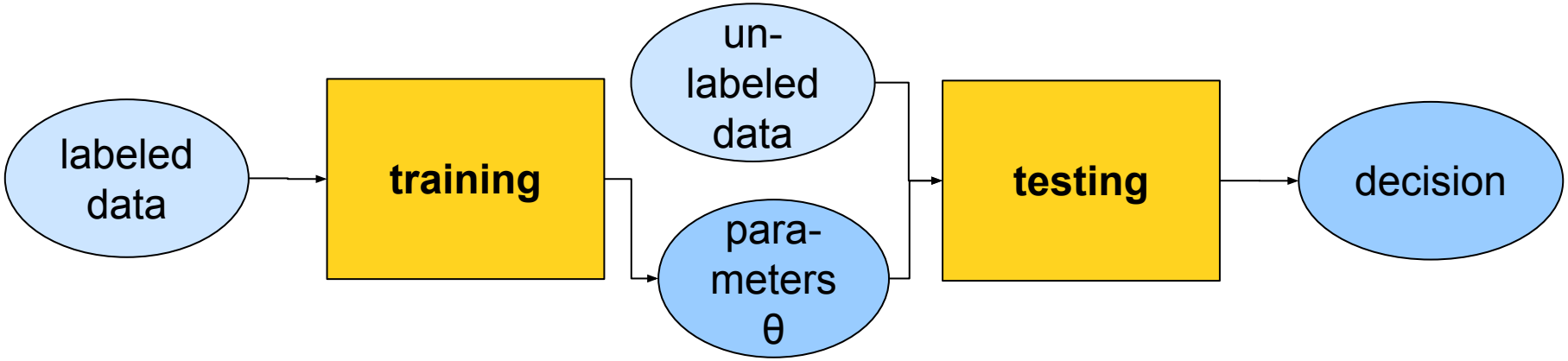






Training

- Question: How do we know the model parameters $\theta = (\mu_1, \Sigma_1, \mu_0, \Sigma_0)$?
- Answer: training
- Input: a labeled training set $(x_1,c_1),..,(x_n,c_n)$



- Example: train μ_1 on $X_1 = \{x_i \mid c_i=1\}$
- Popular approach:

  maximum likelihood (ML)

  = *choose the parameters that make the observed training data most likely*

$$\mu_1^* = \arg\max_{\mu} P(X_1|\mu)$$

$$...derive...$$

$$= \frac{1}{|X_1|} \sum_{x \in X_1} x$$

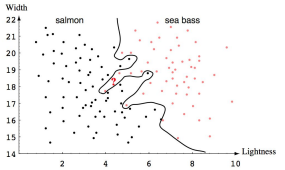
Gaussian CCDs – Training

- Conclusion:
- **ML estimate for mean = empirical mean**
- Works similar for variances (ML estimate = empirical variance)
- Not as obvious for other distributions

more information on parameter estimation:
see Duda/Hart/Stork

Overfitting

- We have previously observed a fundamental challenge to pattern recognition systems: **overfitting**

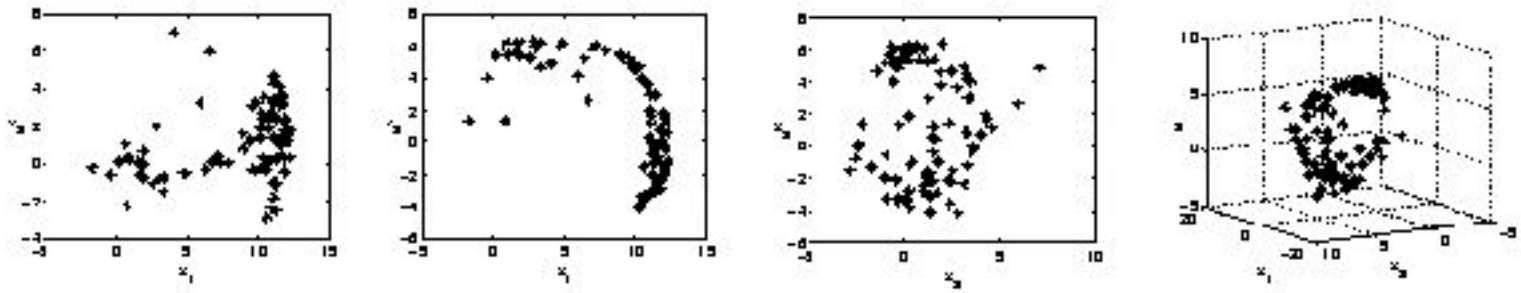


- „symptoms“: error on the training set is significantly lower than on the test set
- causes:
  - few training samples
  - high dimensions („curse of dimensionality“)
  - many system parameters
  - model does not fit the data

A Non-parametric Model - Nearest Neighbor

Parametric vs. Non-parametric Approaches

- So far, we assumed P(x|c) to be Gaussian.
- What about these distributions?



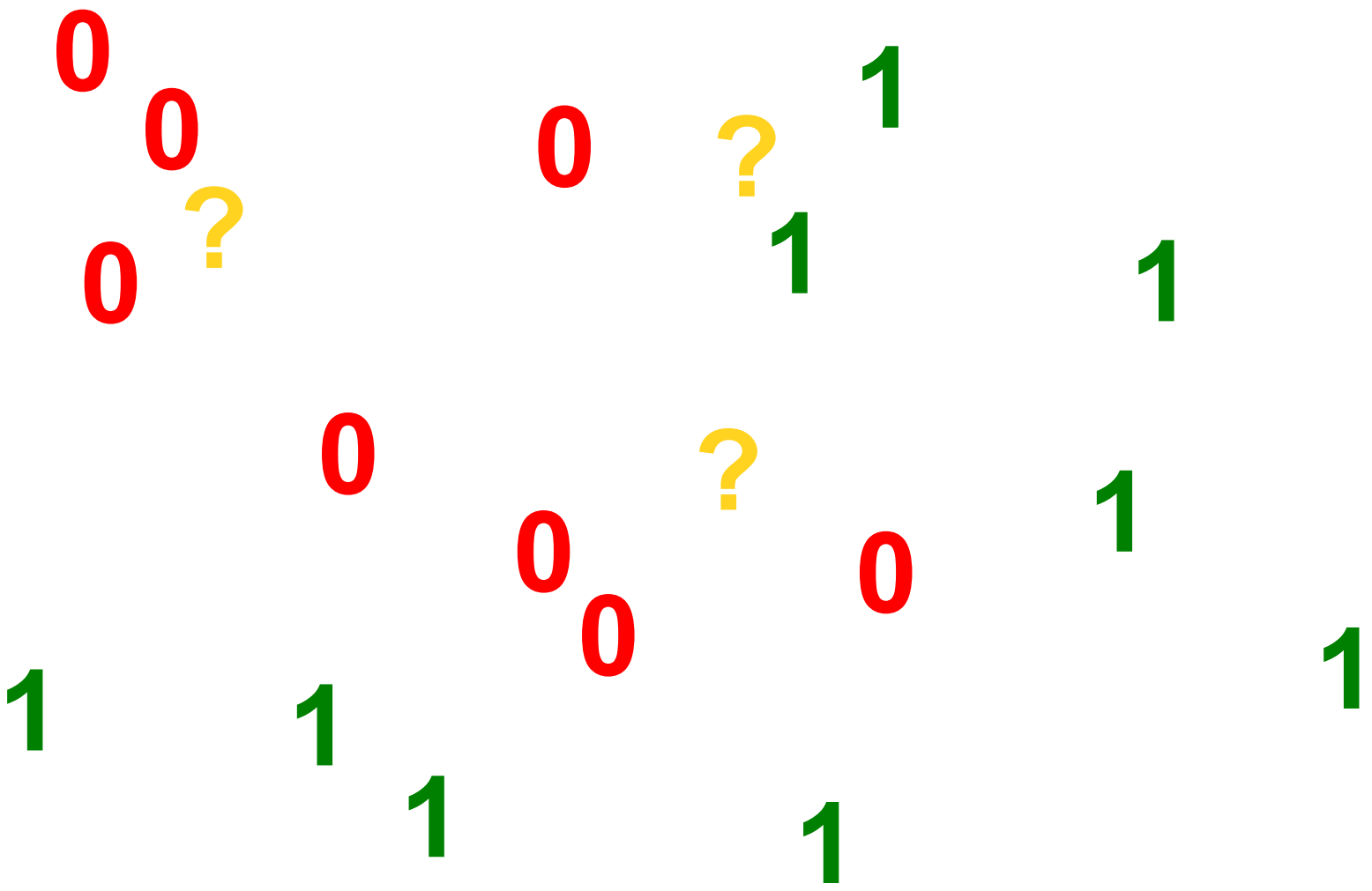
- Often, we don't know the parametric form of P(x|c)
- Possible approaches:
  - *mixtures* of Gaussians
  - **non-parametric methods** (no parameters, no training)

K-Nearest Neighbor Classification

- given: labeled training samples $(x_1,c_1),..,(x_n,c_n)$
- training: *None*
- classification:

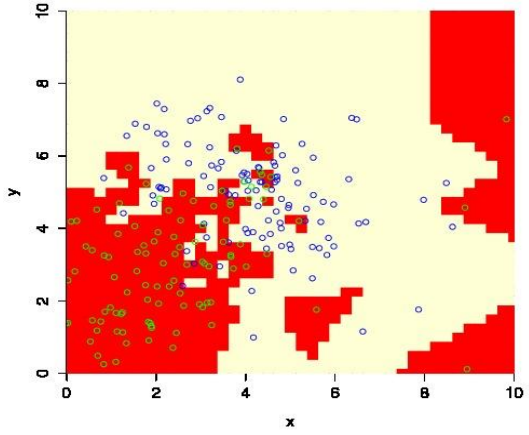
- given: unlabeled sample x
- rank training samples by their distance from x:
  - $x_{\pi(1)}, x_{\pi(2)}, \ldots x_{\pi(K)}, x_{\pi(K+1)}, \ldots, x_{\pi(n)}$
- set $$P(c|x) = \frac{\sum_{k=1}^{K} \delta(c_{\pi(k)}, c)}{K}$$

3-Nearest Neighbor: Example

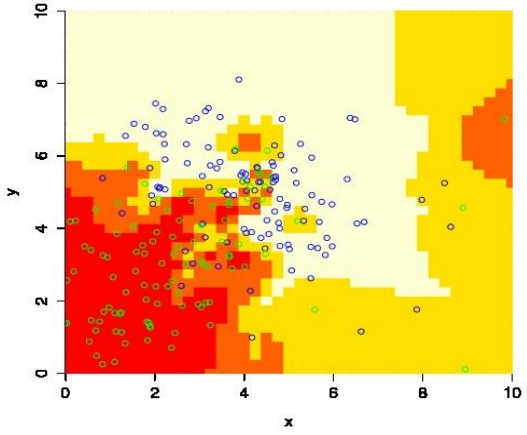


Nearest Neighbor - Example

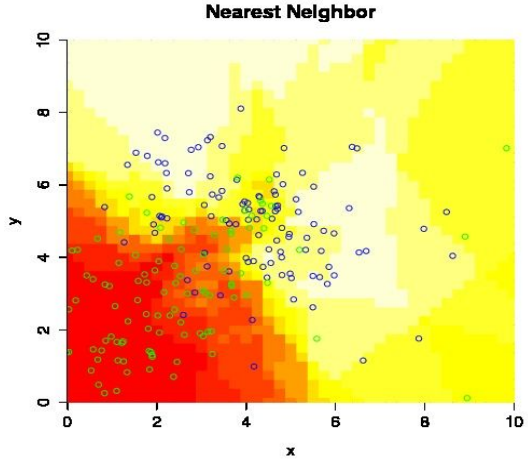
K=1



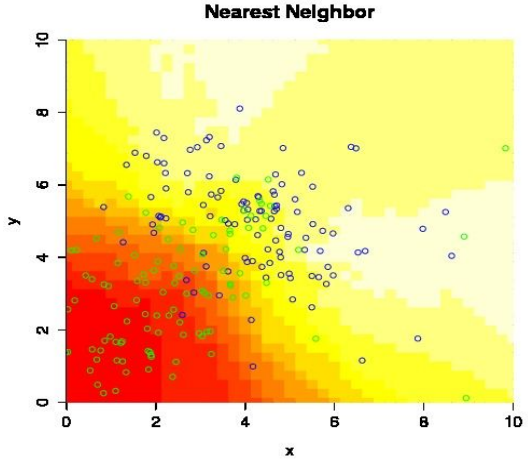
K=3



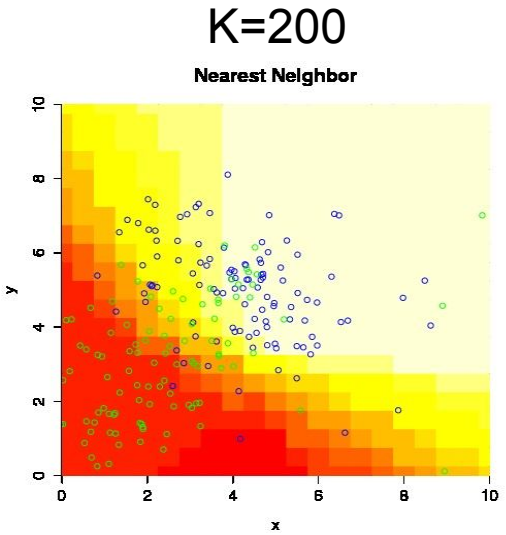
K=10



K=50

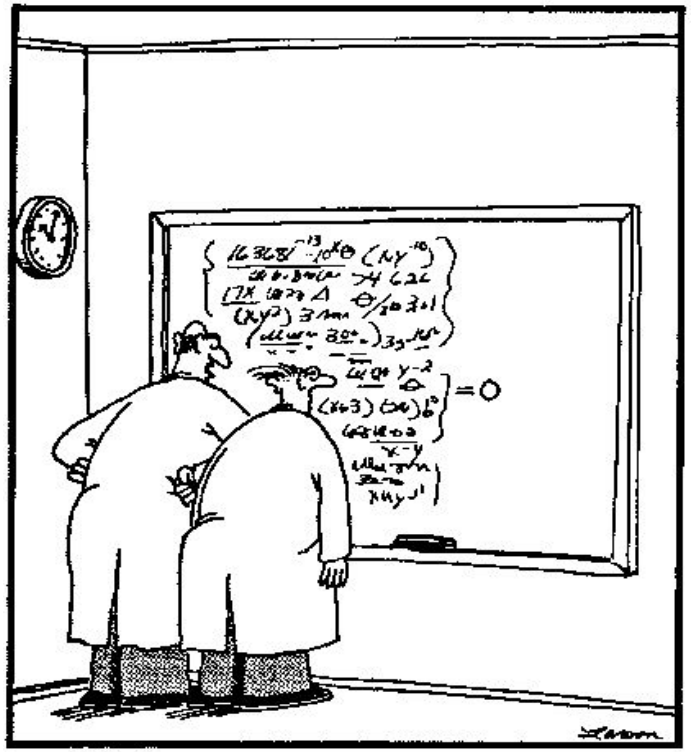


K=200



Nearest Neighbor – Statistical Motivation

- derivation why this works:
- Duda/Hart/Stork (Chapter 4)



"No doubt about it, Ellington - we've mathematically expressed the purpuse of the universe. God, how I love the thrill of scientific discovery!"

Nearest Neighbor - Discussion

- When is nearest neighbor (NN) successful?
  - we need many samples in small regions!
- Is nearest neighbor better than Gaussians?
  - not necessarily – **if** the underlying class-conditional densities are truly Gaussian and we can determine parameters reliably, Gaussians are the optimal model!
- Are there really no parameters?
  - there's K!
    - low K = high variance
    - high K = oversmoothing
    - good compromise in practice: $K=\sqrt{n}$

A Discriminative Model - Logistic Regression

Discriminative Models

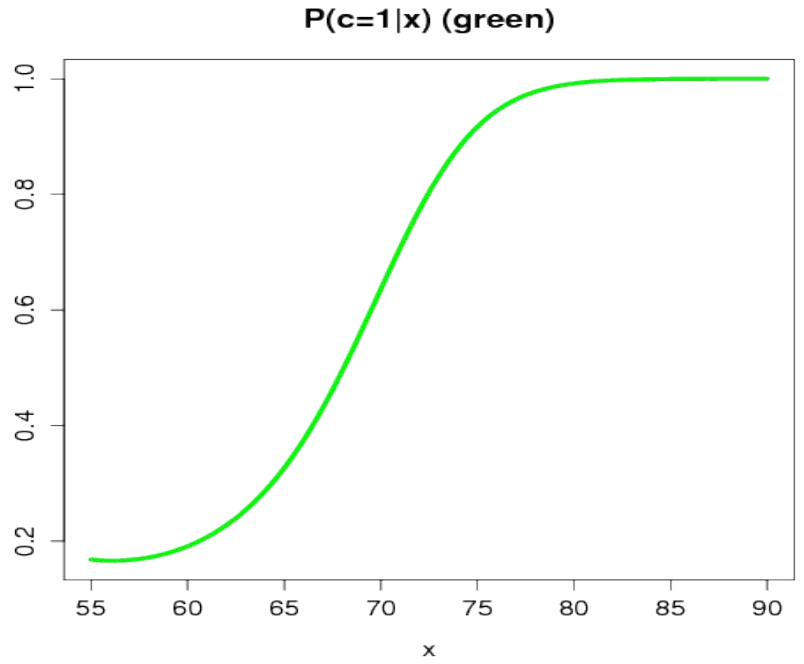
- We saw a *generative* model - Gaussians
  - we know P(x|c) and P(c), i.e. we know P(c,x)
  - we can „generate“ samples from P(c,x)
    - draw c' from P(c)
    - draw x from P(x|c')
- Alternative: omit P(x|c) and P(c), and directly estimate P(c|x)! → ***discriminative models***

- remember the Gaussian case P(c|x) was a *sigmoid* function

$$P(c = 1|x) = \sigma(w \cdot x + b)$$

- where

$$\sigma(t) = \frac{1}{1 + e^{-t}}$$



Sigmoids in more Dimensions

- In more dimensions, we have a weight **vector** w

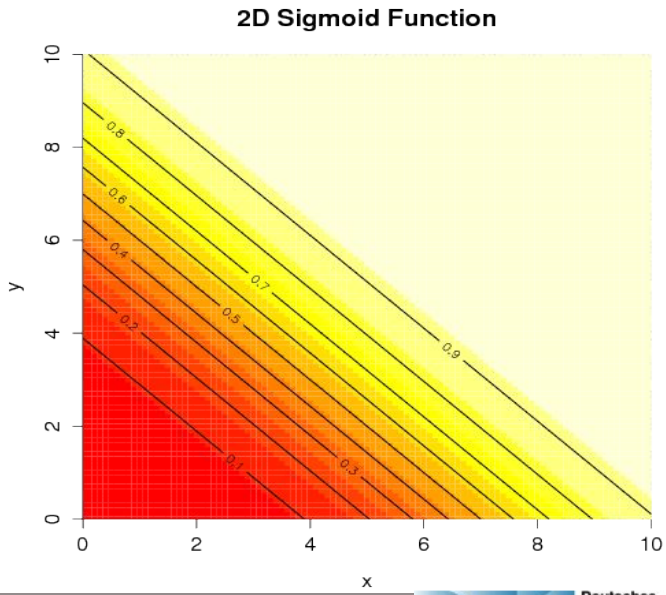
$$P(c = 1|x) = \sigma(\mathbf{w} \cdot x + b)$$

- The decision boundary becomes a (linear) hyperplane

- We can omit b using *augmented vectors*

$$x := (x_1, .., x_d, \mathbf{1})$$

$$w := (w_1, .., w_d, b)$$



Logistic Regression: Approach

- The parameter vector **w** determines the decision function
- Training: again, estimate **w** by Maximum Likelihood (ML)

$$
\begin{aligned}
w^* &= \arg\max_w P(c_1, .., c_n | x_1, ..., x_n, w) \\
&= \arg\max_w \prod_{i:c_i=1} P(c_i = 1 | x_i, w) \cdot \prod_{i:c_i=0} [1 - P(c_i = 1 | x_i, w)] \\
&= \arg\max_w \prod_{i:c_i=1} \sigma(w \cdot x_i) \cdot \prod_{i:c_i=0} [1 - \sigma(w \cdot x_i)]
\end{aligned}
$$

Logistic Regression: Approach

- Optimization: use gradient descent
  - see [Bishop, 205f]

- initialize w^0
- until convergence:
  - set

$$w^{k+1} = w^k + \lambda \sum_i [(c_i - \sigma(w \cdot x_i)) \cdot x_i]$$

  - set k=k+1

Logistic Regression - Discussion

- Logistic Regression – What's good about it?
  - learn only what we need → less overfitting
    - Gaussians: 2D + 2*D*D parameters (means+variances)
    - Logistic Regression: D+1 parameters
- What's bad about it?
  - *linear decision boundaries only*
  - gradient descent optimization may end up in local optima
  - gradient descent optimization is difficult to tune (step size)
- There are other (linear and non-linear) classifiers with different optimization criteria
  - error minimization (perceptron, multi-layer perceptrons)
  - margin maximization (SVMs)

Agenda

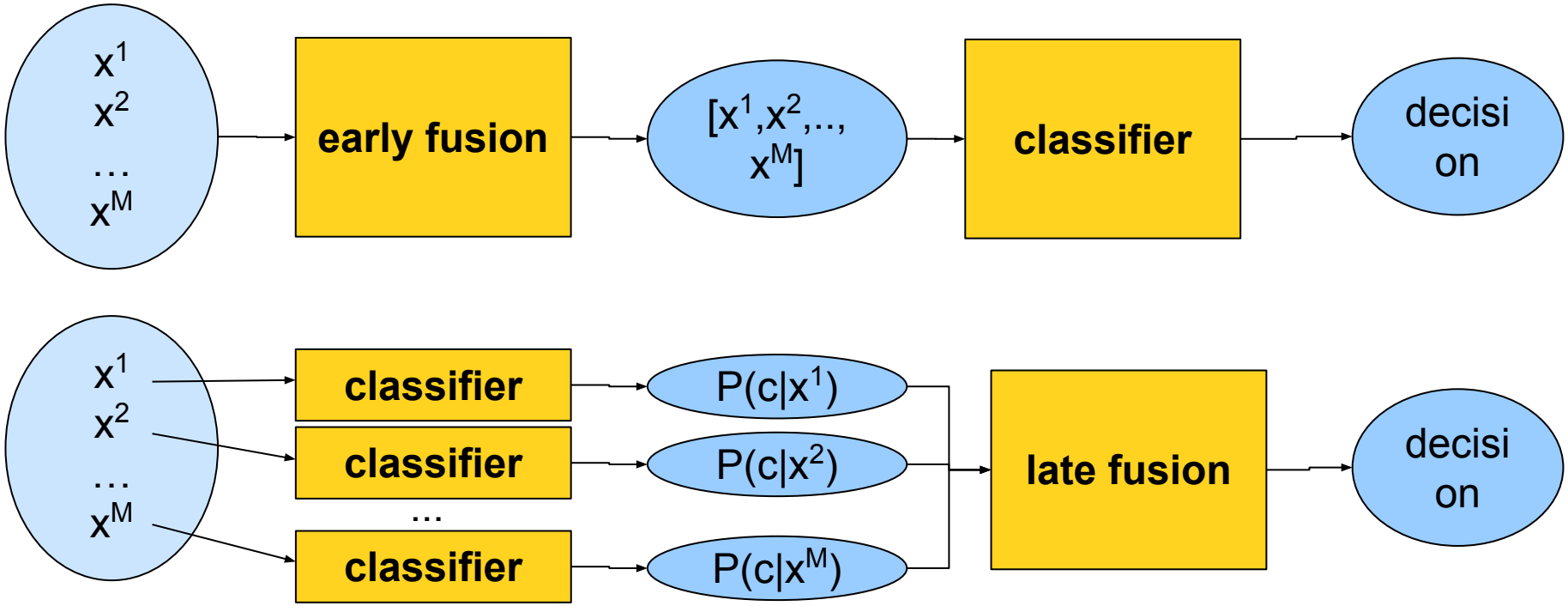
1. What is supervised learning ?
2. How to build an “optimal” classifier ?
3. What kind of classifiers are there ?
   a. Gaussian CCDs
   b. Nearest Neighbors
   c. Logistic Regression
4. **How to combine different classifiers?**

Classifier Combination - Heuristic Rules

- In Multimedia Information Retrieval, classifiers frequently combine different pieces of evidence
  - multiple features
  - multiple modalities
  - multiple classifiers
  - multiple training sets

Early Fusion vs. Late Fusion

- Different combination strategies
  - early fusion = concatenate features
  - **late fusion** = combine classification results



Late Fusion: Problem Formulation

- Assumption: two classes, M classifiers
- Test set $x_1,..,x_n$
- Each sample consists of different modalities: $x_i = (x_i^1,..,x_i^M)$
- Each classifier gives scores $P^m(c=1|x_i^m)$

$$P(c=1|x_i) = F\left[P^1(c=1|x_i^1), P^2(c=1|x_i^2), ..., P^{M-1}(c=1|x_i^{M-1}), P^M(c=1|x_i^M)\right]$$

Classifier Combination – Heuristic Rules

- Other heuristic rules
  - product rule

$$P(c|x) = \prod_m P^m(c|x)$$

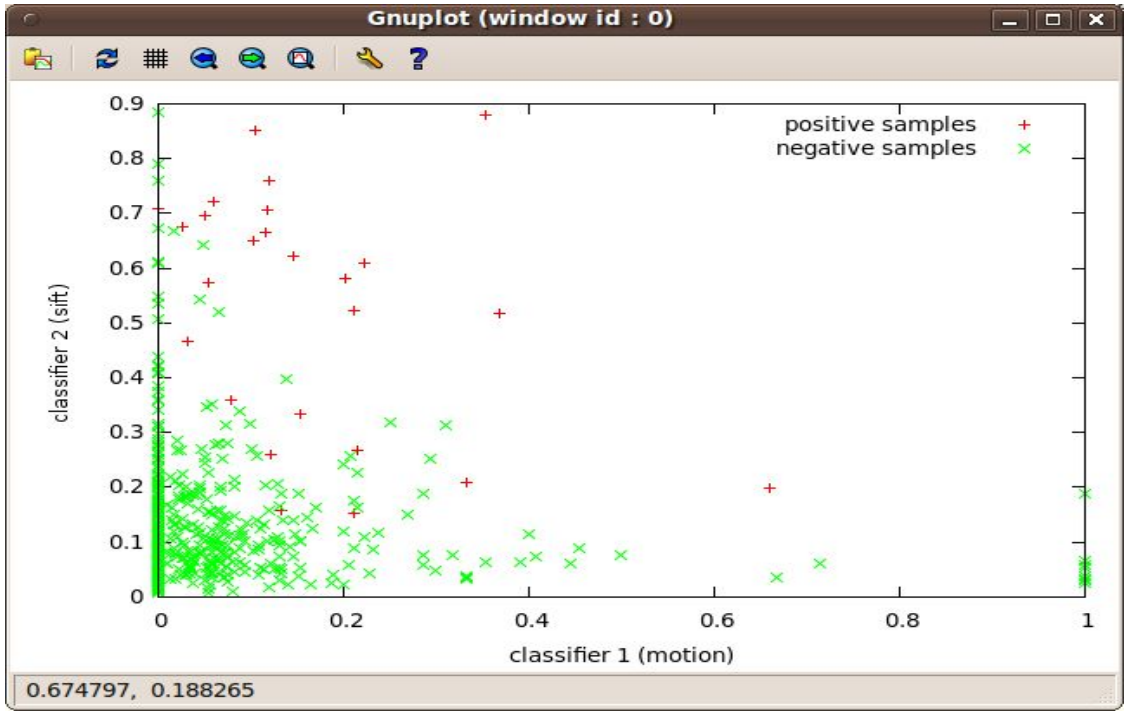
    - *statistical motivation by applying Bayes' rule*
  - sum rule

$$P(c|x) = \frac{1}{M} \sum_m P^m(c|x)$$

    - *statistical motivation: Bishop, p.656f*
    - *if P^m are estimated from subsamples of the training data, this approach is called **bagging***
  - min/max rule

Late Fusion - Example

- Example: Detecting „basketball“ in YouTube clips



- *What's the problem here?*

Weighted Sum Fusion

- Weighted sum fusion

$$P(c|x_i) = \sum_m w_m \cdot P^m(c|x_i) \qquad \left[\sum_m w_m = 1\right]$$

- We can give different weights to classifiers of different accuracy
- *How to learn weights?*
  → example: grid search

Weighted Sum Fusion

- On what data should we learn the weights?
  - determine on the test set?
  - determine on the training set?
  - determine on a separate validation set!

Conclusion

Discussion

- This lecture – 3 sample classifiers
  - Generative models (with Gaussian CCDs)
  - K-nearest neighbor
  - Logistic regression
- The Big Answer – Which one is the best?
  - the right classifier depends on the distribution of the target data...
  - … on the preprocessing ...
  - … on the features...
  - … on the amount of training data.
- → ***no-free-lunch theorem***