# Machine Learning in Science and Engineering

*Gunnar Rätsch*
*Friedrich Miescher Laboratory*
*Max Planck Society*
*Tübingen, Germany*

*http://www.tuebingen.mpg.de/~raetsch*

# Roadmap

- Motivating Examples
- Some Background
- Boosting & SVMs
- Applications

**Rationale: Let computers learn to automate processes and to understand highly complex data**

# Example 1: Spam Classification

**From:** smartballlottery@hf-uk.org
**Subject:** **Congratulations**
**Date:** 16. December 2004 02:12:54 MEZ

LOTTERY COORDINATOR,
INTERNATIONAL PROMOTIONS/PRIZE AWARD DEPARTMENT.
SMARTBALL LOTTERY, UK.

DEAR WINNER,

WINNER OF HIGH STAKES DRAWS

Congratulations to you as we bring to your notice, the results of the the end of year, HIGH STAKES DRAWS of SMARTBALL LOTTERY UNITED KINGDOM. We are happy to inform you that you have emerged a winner under the HIGH STAKES DRAWS SECOND CATEGORY,which is part of our promotional draws. The draws were held on15th DECEMBER 2004 and results are being officially announced today. Participants were selected through a computer ballot system drawn from 30,000 names/email addresses of individuals and companies from Africa, America, Asia, Australia,Europe, Middle East, and Oceania as part of our International Promotions Program.

…

**From:** manfred@cse.ucsc.edu
**Subject:** **ML Positions in Santa Cruz**
**Date:** 4. December 2004 06:00:37 MEZ

We have a Machine Learning position
at Computer Science Department of
the University of California at Santa Cruz
(at the assistant, associate or full professor level).

Current faculty members in related areas:
Machine Learning: DAVID HELMBOLD and MANFRED WARMUTH
Artificial Intelligence: BOB LEVINSON
DAVID HAUSSLER was one of the main ML researchers in our department. He now has launched the new Biomolecular Engineering department at Santa Cruz

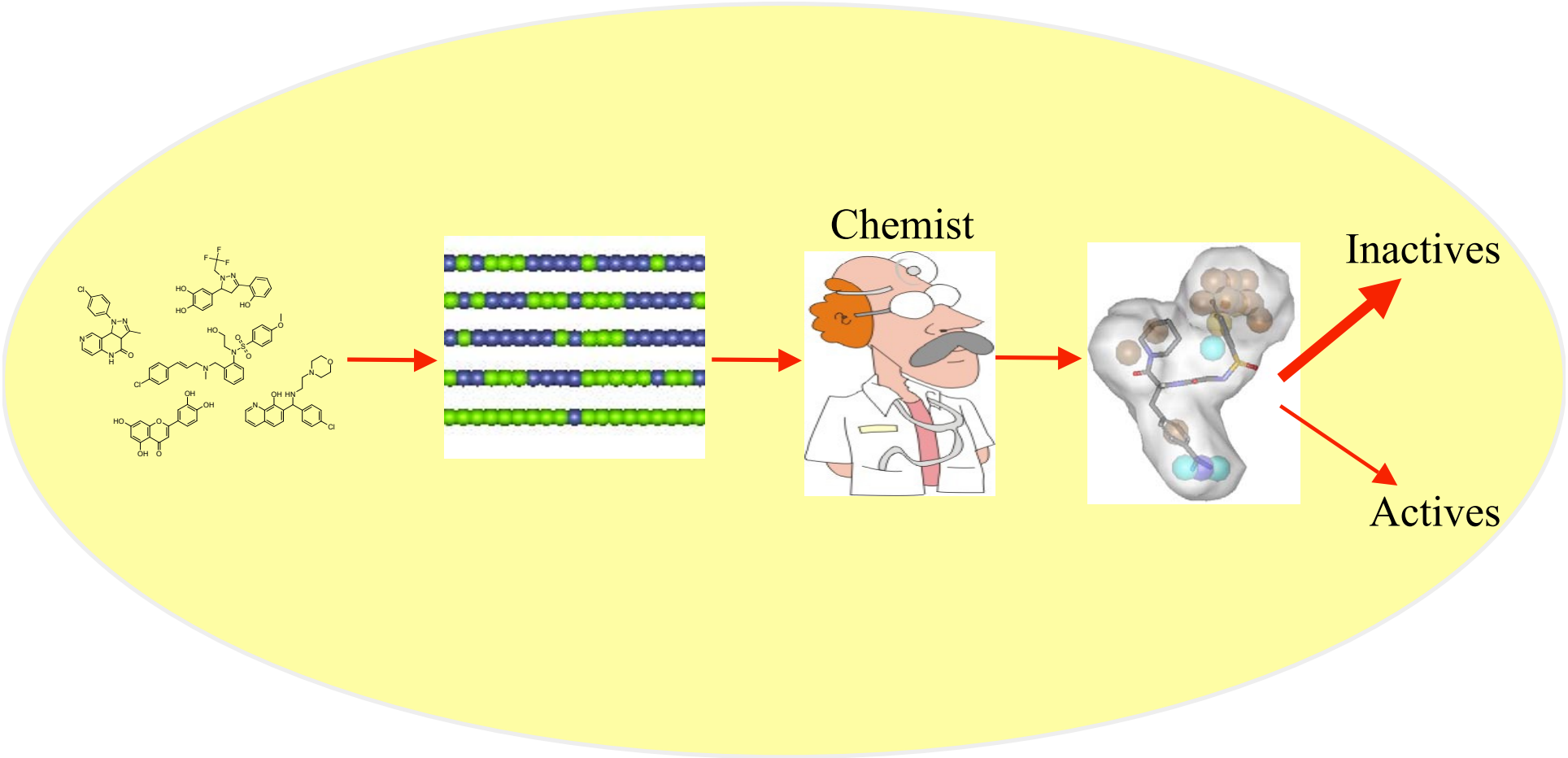
There is considerable synergy for Machine Learning at Santa Cruz:
-New department of Applied Math and Statistics with an emphasis on Bayesian Methods http://www.ams.ucsc.edu/
-- New department of Biomolecular Engineering
http://www.cbse.ucsc.edu/

…

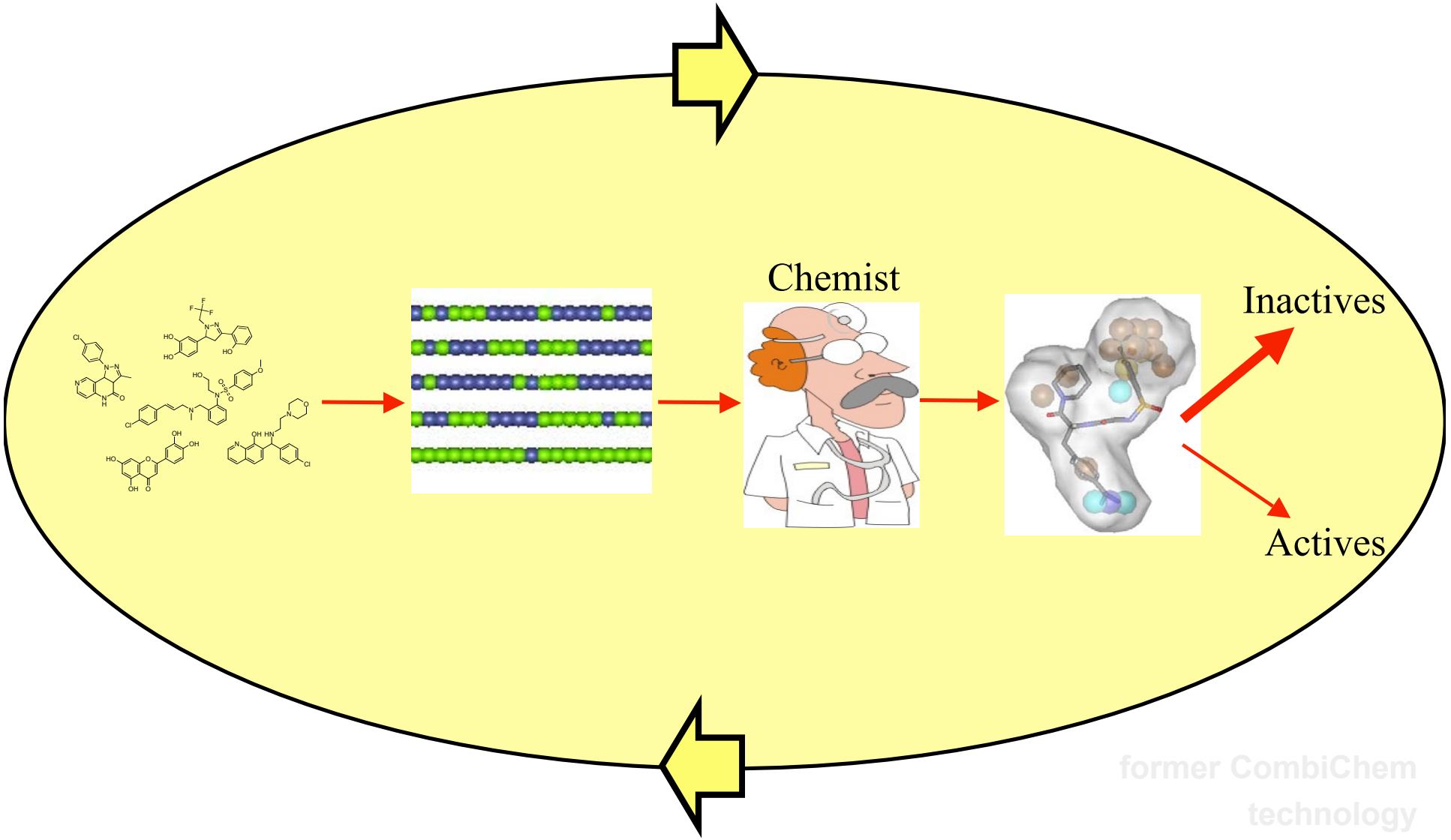
**Goal: Classify emails into spam / no spam**
**How? Learn from previously classified emails!**

**Training:** analyze previous emails
**Application:** classify new emails

# Example 2: Drug Design

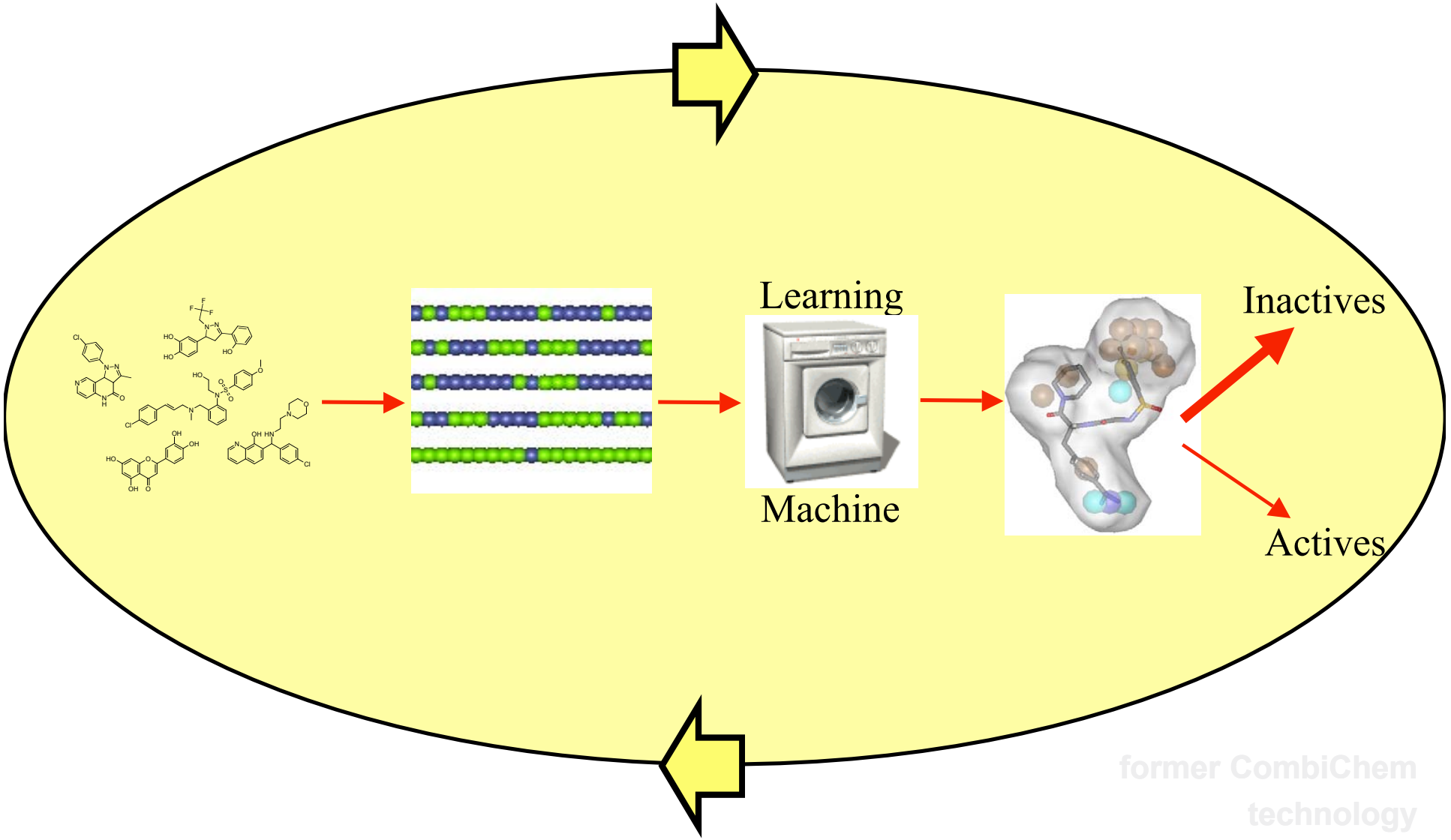


# The Drug Design Cycle




former CombiChem technology

# The Drug Design Cycle




former CombiChem technology

# Example 3: Face Detection

# Premises for Machine Learning

- Supervised Machine Learning
  - Observe N training examples with label
    $$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$$
  - Learn function $f : \mathbf{x} \longrightarrow y$
  - Predict label of unseen example $f(\mathbf{x})$
- Examples generated from statistical process
- Relationship between features and label
- **Assumption:** unseen examples are generated from same or similar process

# Problem Formulation


Natural
+1


Plastic
-1


Natural
+1

. . .


Plastic
-1


?

**The "World":**

- Data
- Unknown Target Function
- Unknown Distribution
- Objective

$$\{(\mathbf{x}_n, y_n)\}_{n=1}^N, \mathbf{x}_n \in \mathbb{R}^d, y_n \in \{\pm 1\}$$
$$y = f(\mathbf{x})$$
$$\mathbf{x} \sim p(\mathbf{x})$$
Given new $\mathbf{x}$, predict $y$

**Problem: $P(\mathbf{x}, y)$ is unknown**

# Problem Formulation

**The 'Model'**

**Hypothesis class:** $\mathcal{H} = \left\{ h \mid h : \mathbf{R}^d \to \{\pm 1\} \right\}$

**Loss:** $l(y, h(\mathbf{x}))$ (e.g. $\mathbf{I}[y \neq h(\mathbf{x})]$)

**Objective:** Minimize the true (expected) loss – "generalization error"

$$h^* = \text{argmin}_{h \in \mathcal{H}} \; L(h) \text{ with } \boxed{L(h) := \mathbf{E}_{\mathbb{X} \times \mathbb{Y}} \; l(\mathbb{Y}, h(\mathbb{X}))}$$

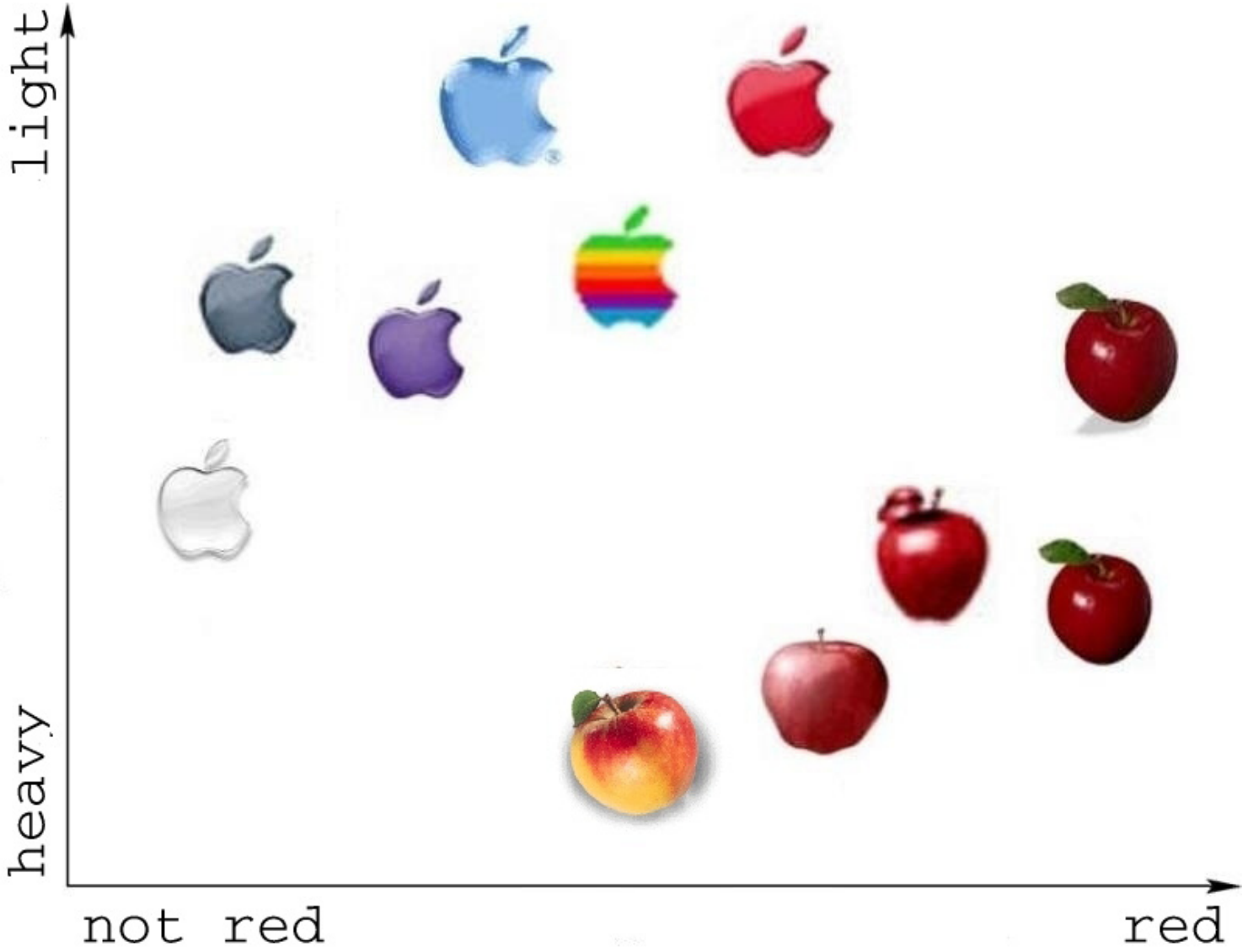
**Problem:** Only have a data sample available, $P(\mathbf{x}, y)$ is unknown!

**Solution:** Find empirical minimizer

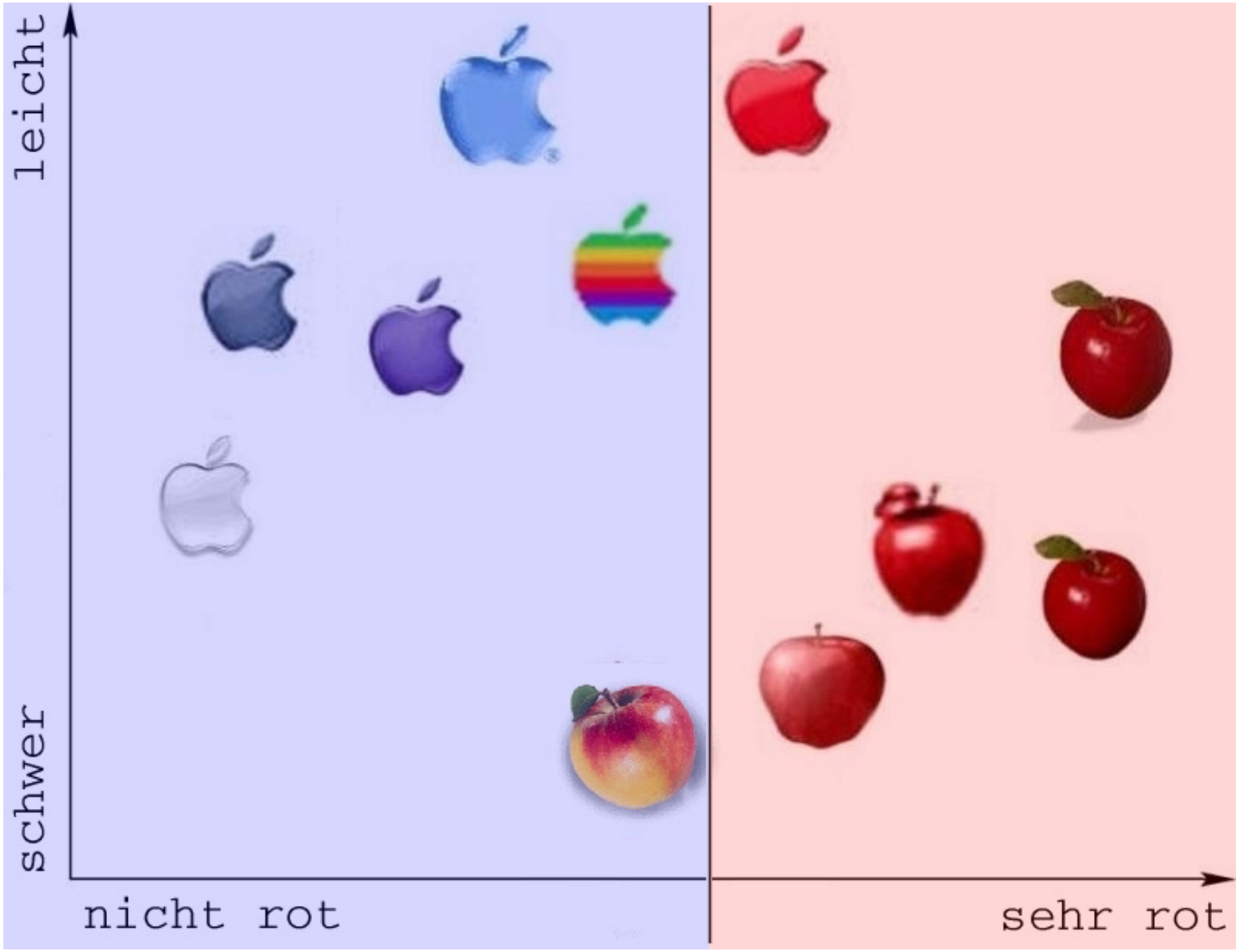
$$\hat{h}_N = \text{argmin}_{h \in \mathcal{H}} \; \frac{1}{N} \sum_{n=1}^{N} l(y_n, h(\mathbf{x}_n))$$

# Example: Natural vs. Plastic Apples

# Example: Natural vs. Plastic Apples



# Example: Natural vs. Plastic Apples

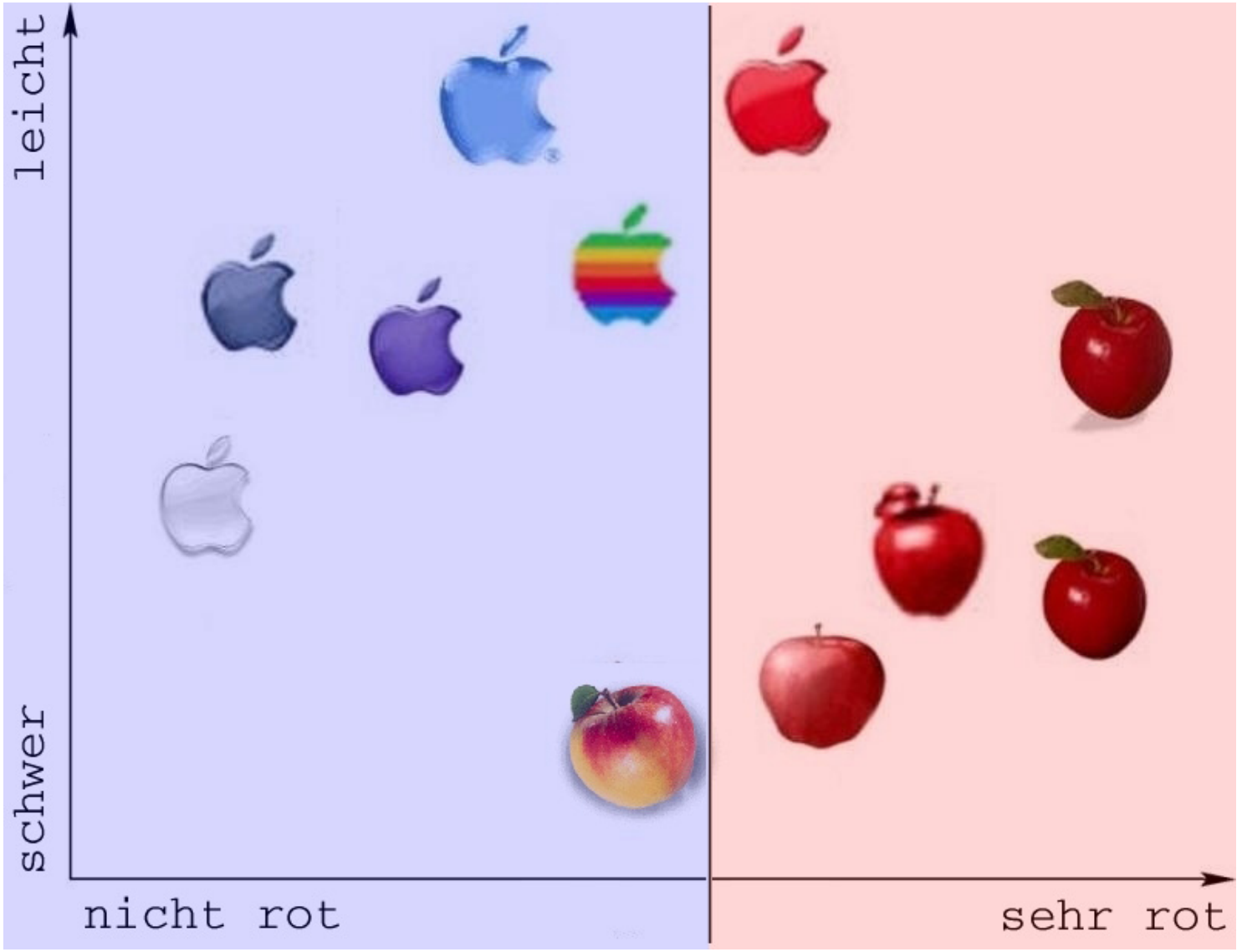


# AdaBoost (Freund & Schapire, 1996)

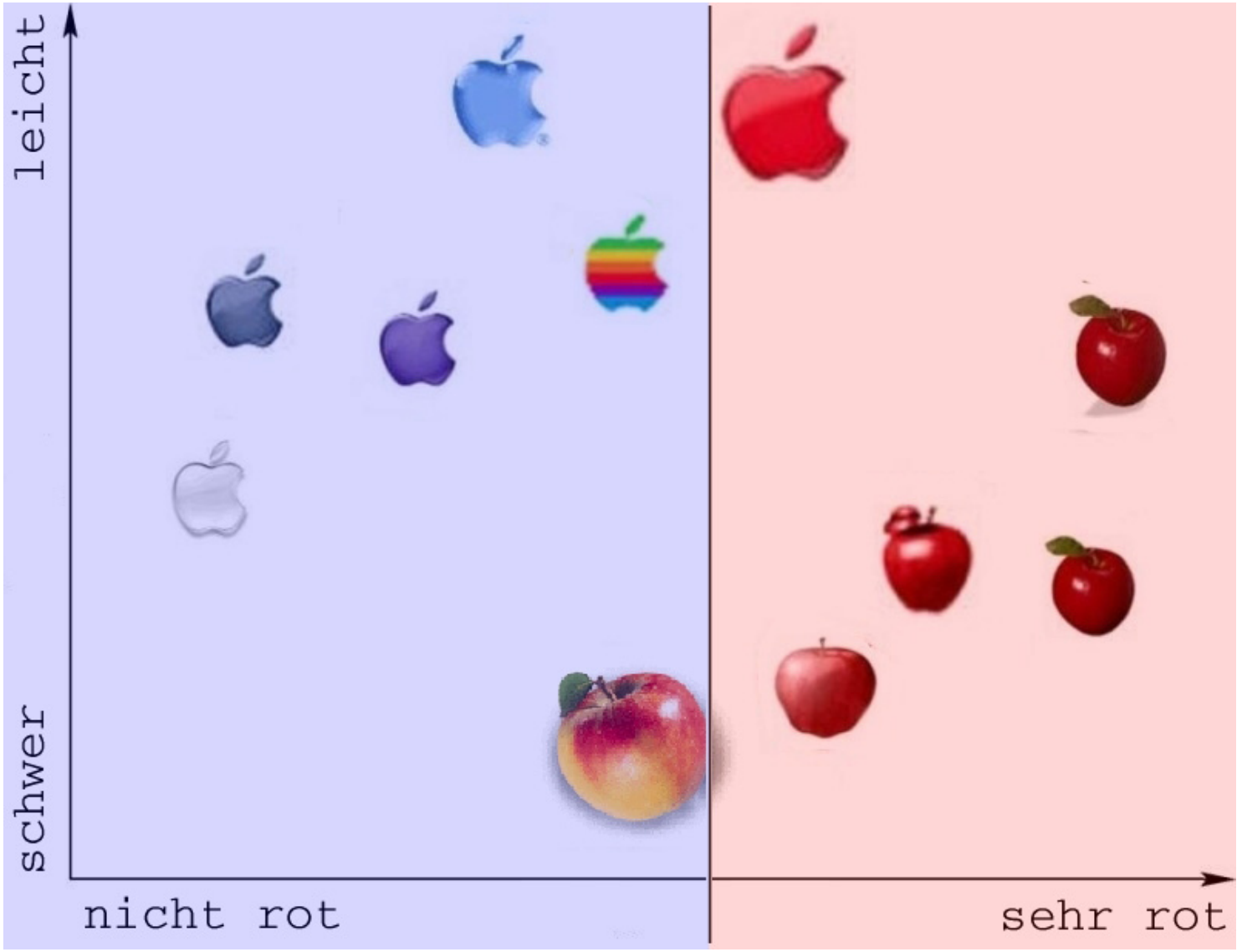
- **Idea:**
  - Use simple many “rules of thumb”
  - Simple hypotheses are not perfect!
  - Hypotheses combination => increased accuracy
- **Problems**
  - How to generate different hypotheses?
  - How to combine them?
- **Method**
  - Compute distribution $d_1, \ldots, d_N$ on examples
  - Find hypothesis on the weighted sample
  - Combine hypotheses $h_1, \ldots, h_J$ linearly:

$$f(\mathbf{x}) = \sum_{j=1}^{J} \alpha_j h_j(\mathbf{x})$$

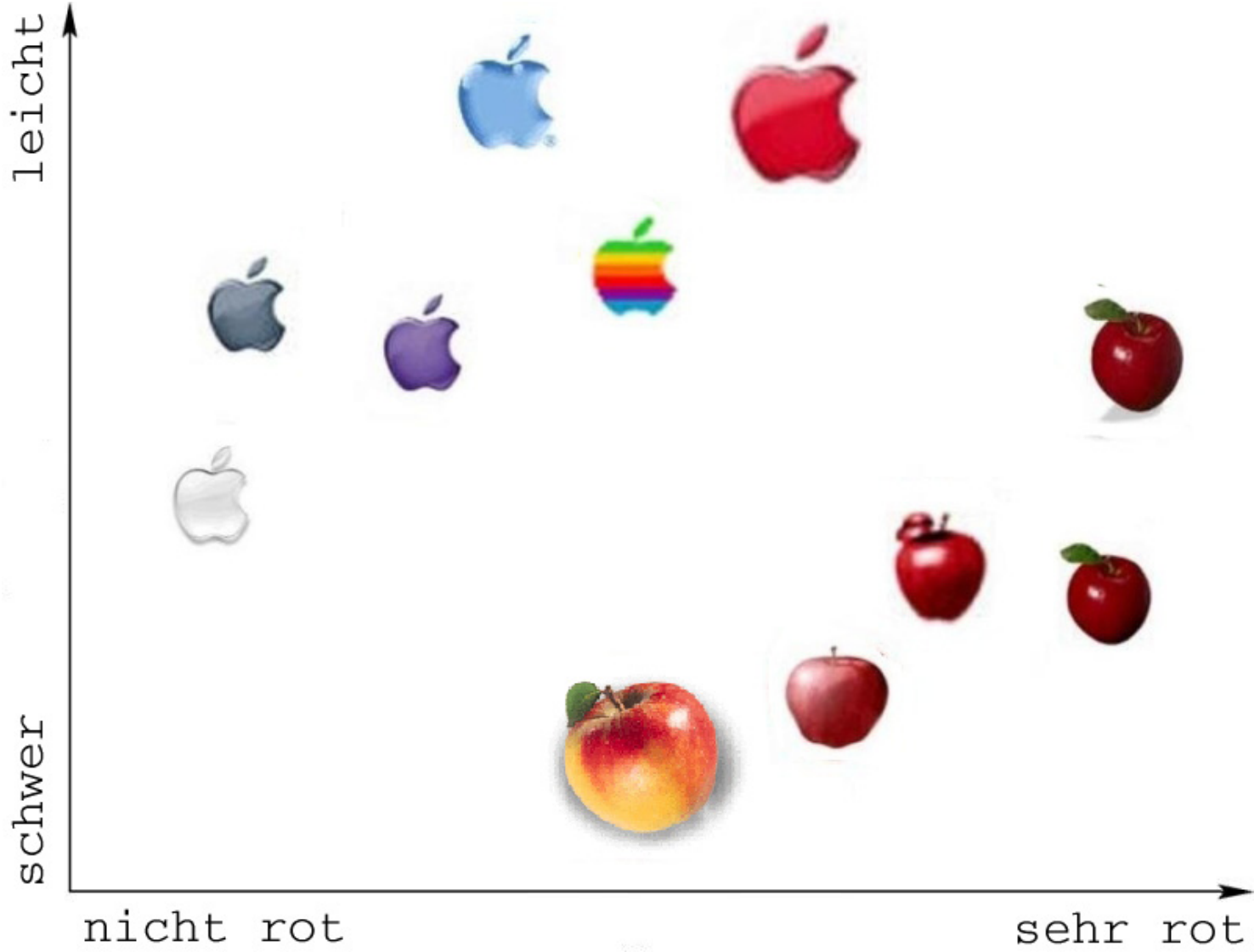
# Boosting: 1st iteration (simple hypothesis)



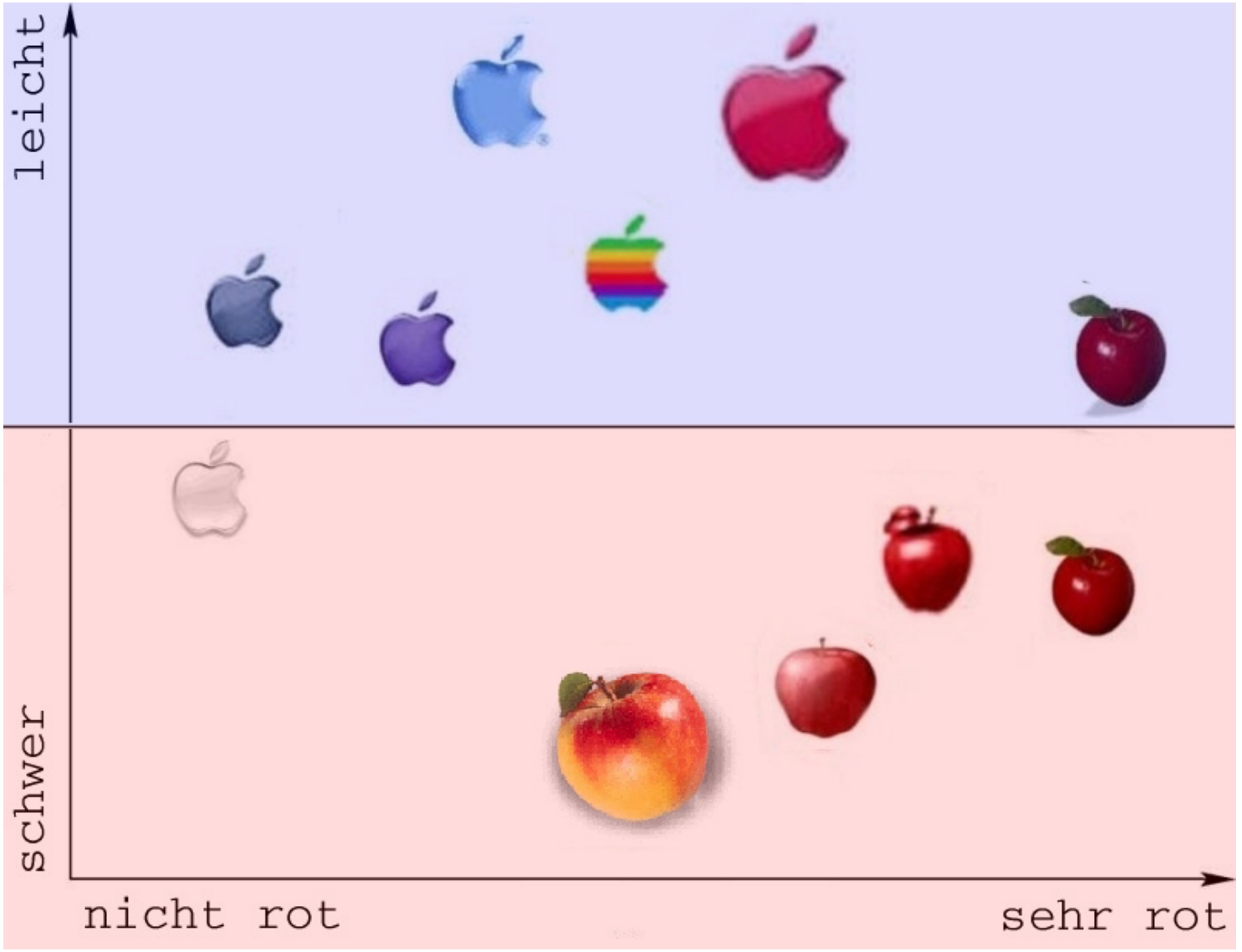
# Boosting: recompute weighting



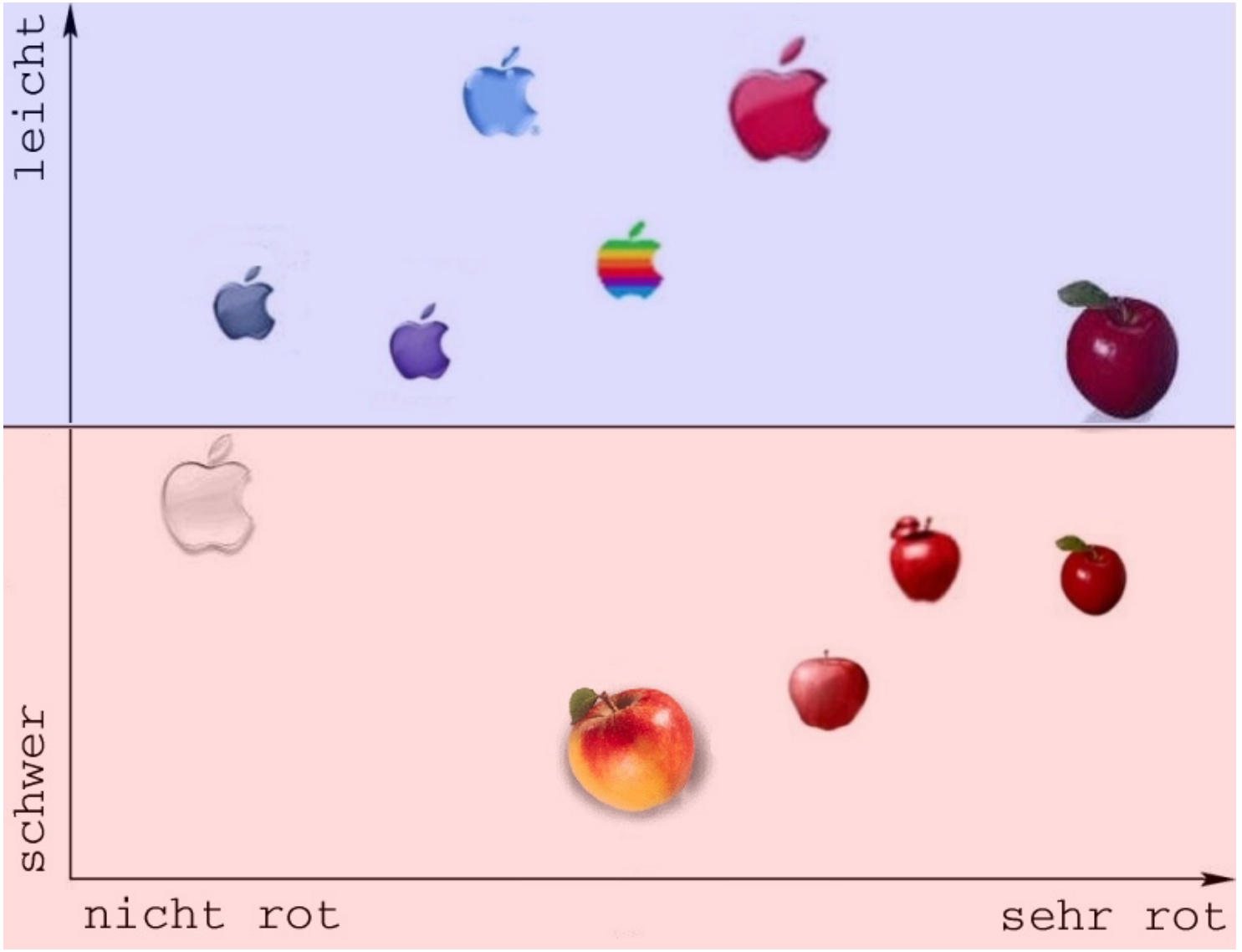
# Boosting: 2nd iteration



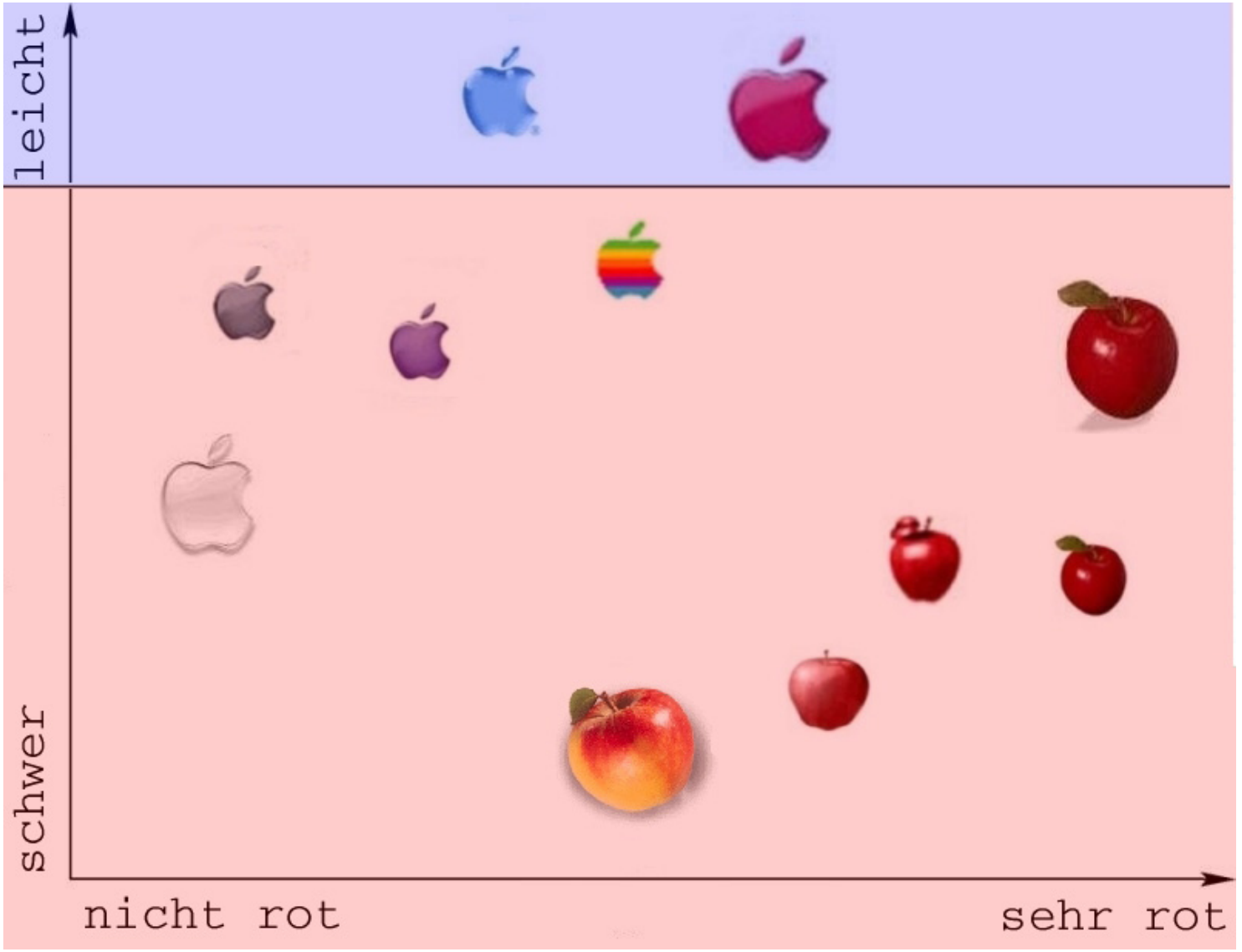
# Boosting: 2nd hypothesis



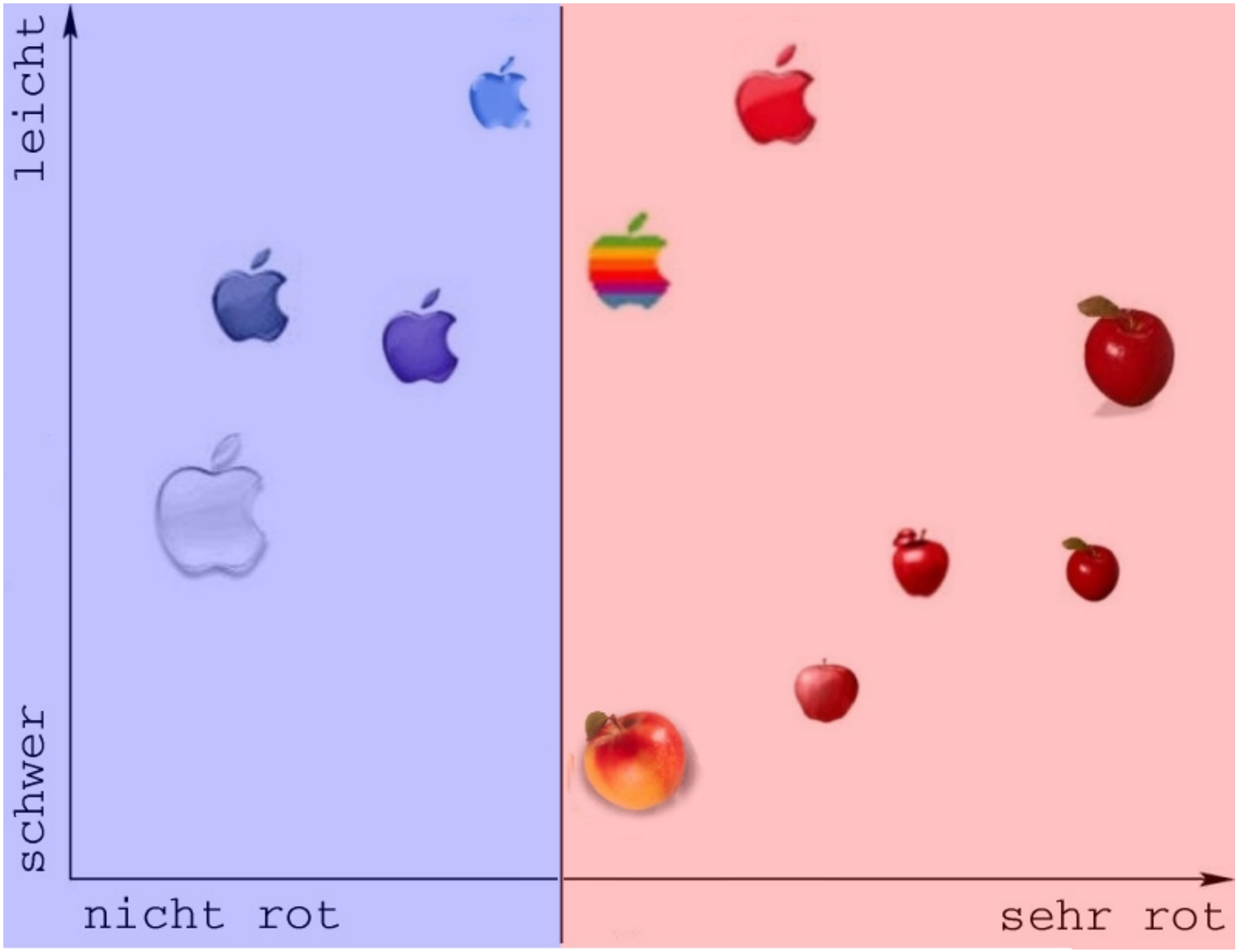
# Boosting: recompute weighting



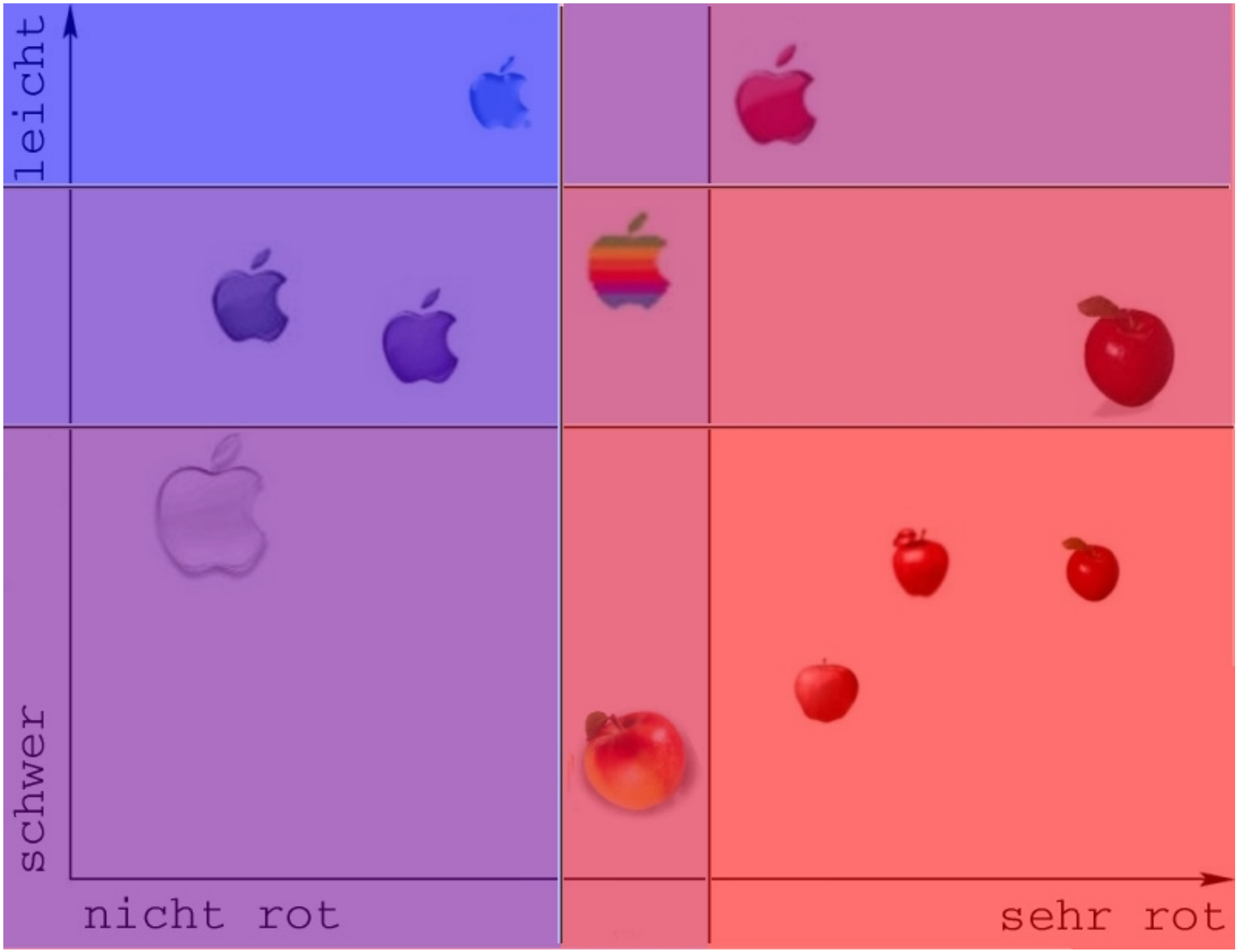
# Boosting: 3rd hypothesis



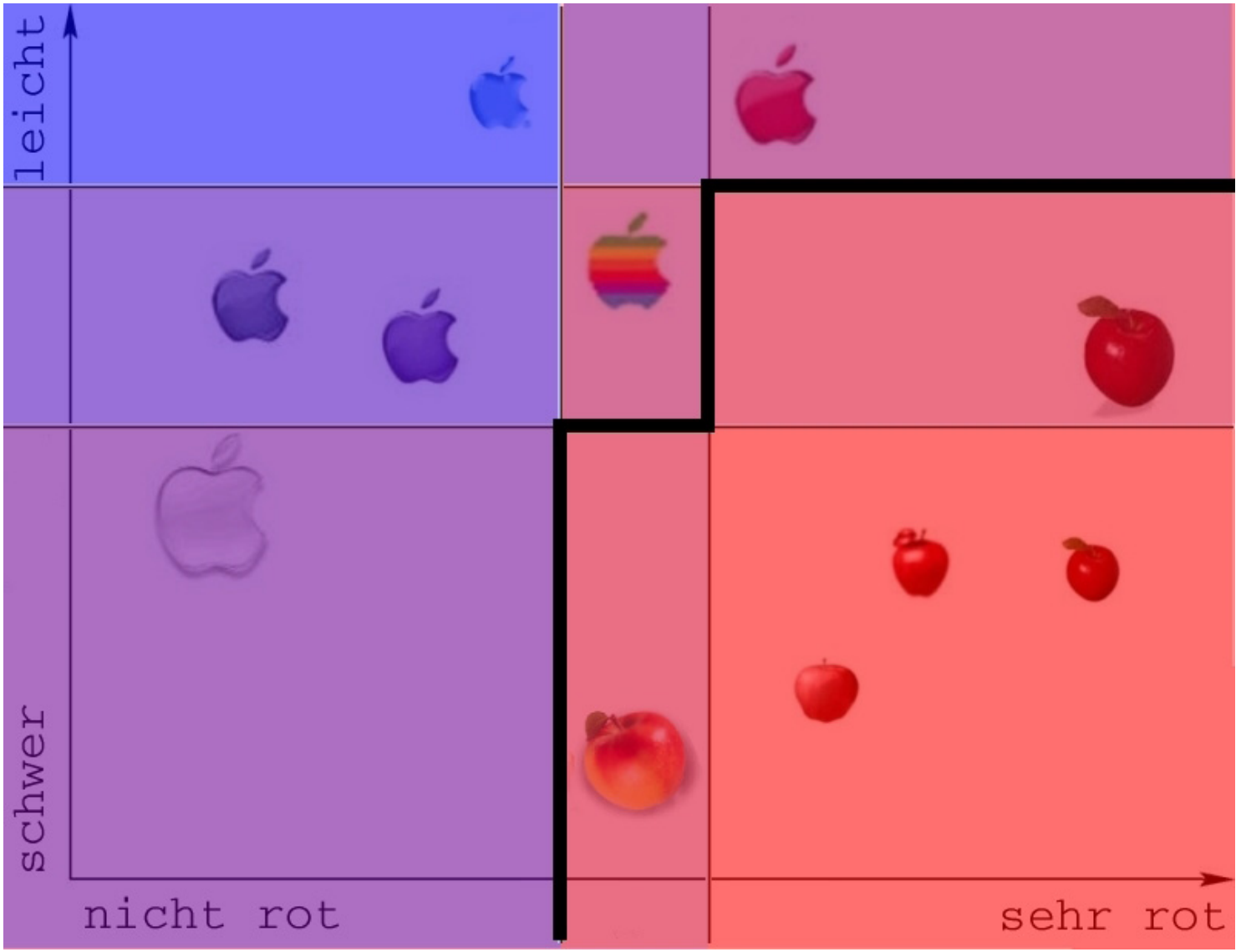
# Boosting: 4rd hypothesis



# Boosting: combination of hypotheses



# Boosting: decision



# AdaBoost Algorithm

**Input:** $N$ examples $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$
**Initialize:** $d_n^{(1)} = 1/N$ for all $n = 1 \ldots N$
**Do for** $t = 1, \ldots, T$,

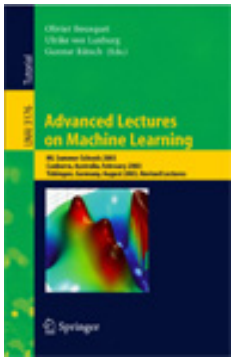
1. Train base learner according to example distribution $\mathbf{d}^{(t)}$ and obtain hypothesis $h_t : \mathbf{x} \mapsto \{\pm 1\}$.
2. compute weighted error $\epsilon_t = \sum_{n=1}^{N} d_n^{(t)} \mathbf{I}(y_n \neq h_t(\mathbf{x}_n))$
3. compute hypothesis weight $\alpha_t = \frac{1}{2} \log \frac{1-\epsilon_t}{\epsilon_t}$
4. update example distribution

$$d_n^{(t+1)} = d_n^{(t)} \exp\left(-\alpha_t y_n h_t(\mathbf{x}_n)\right) / Z_t$$

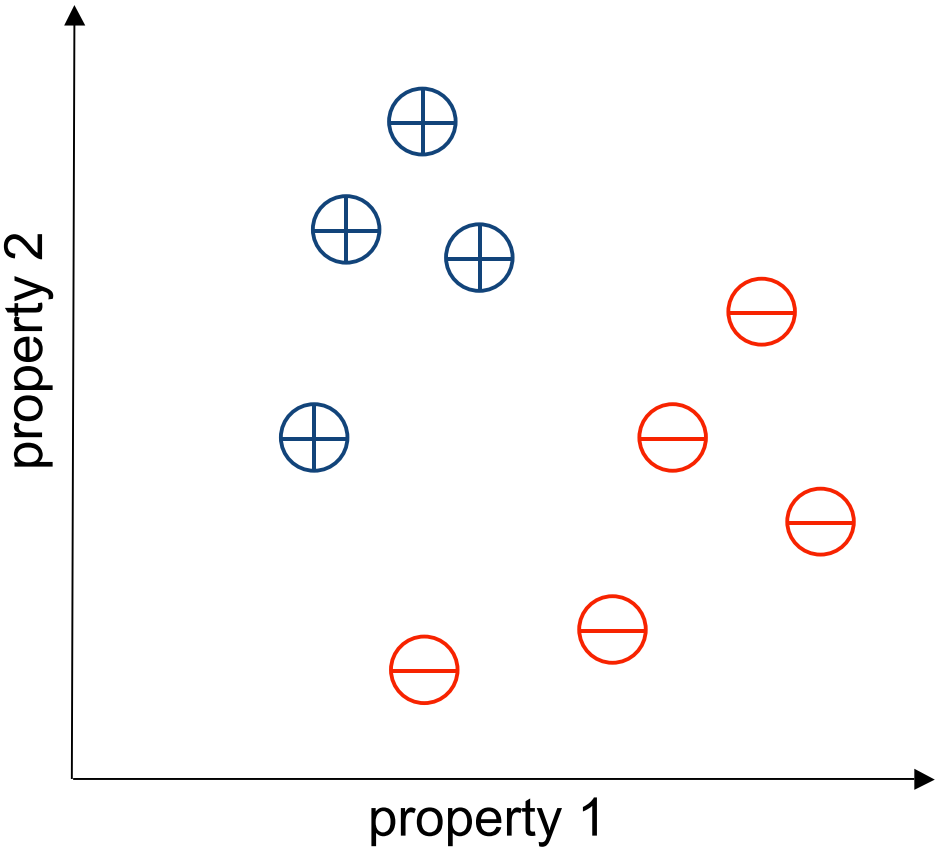
**Output:** final hypothesis $f_{\text{Ens}}(\mathbf{x}) = \sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})$

# AdaBoost algorithm

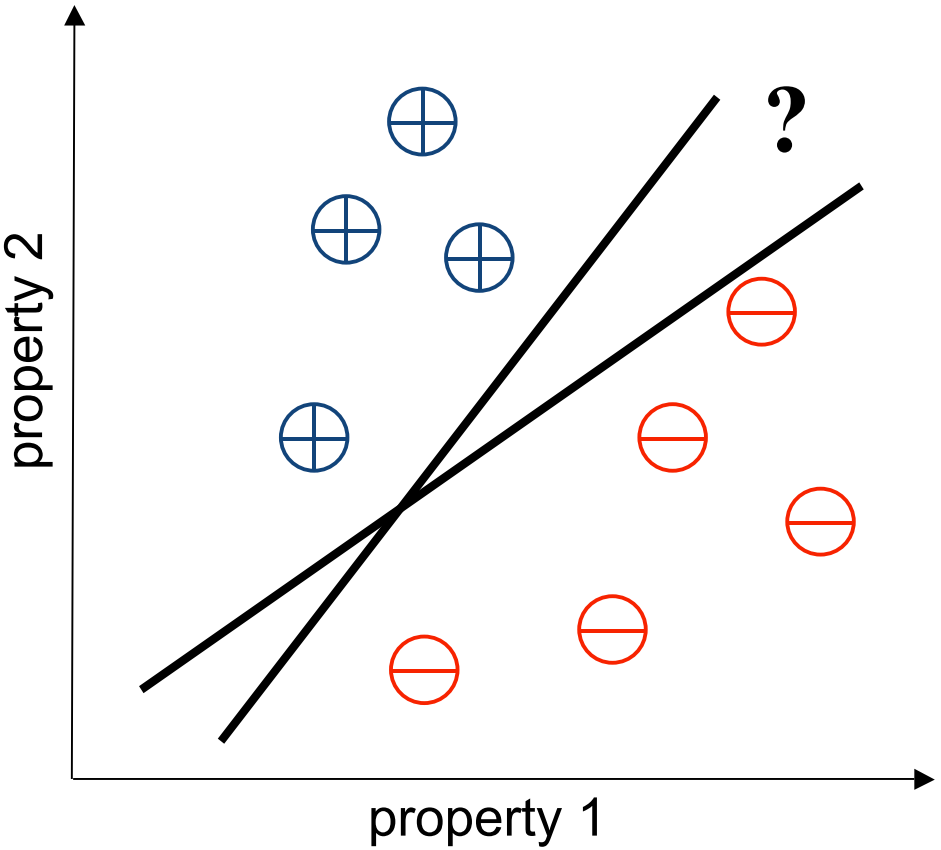
- Combination of
  - Decision stumps/trees
  - Neural networks
  - Heuristic rules
- Further reading
  - http://www.boosting.org
  - http://www.mlss.cc

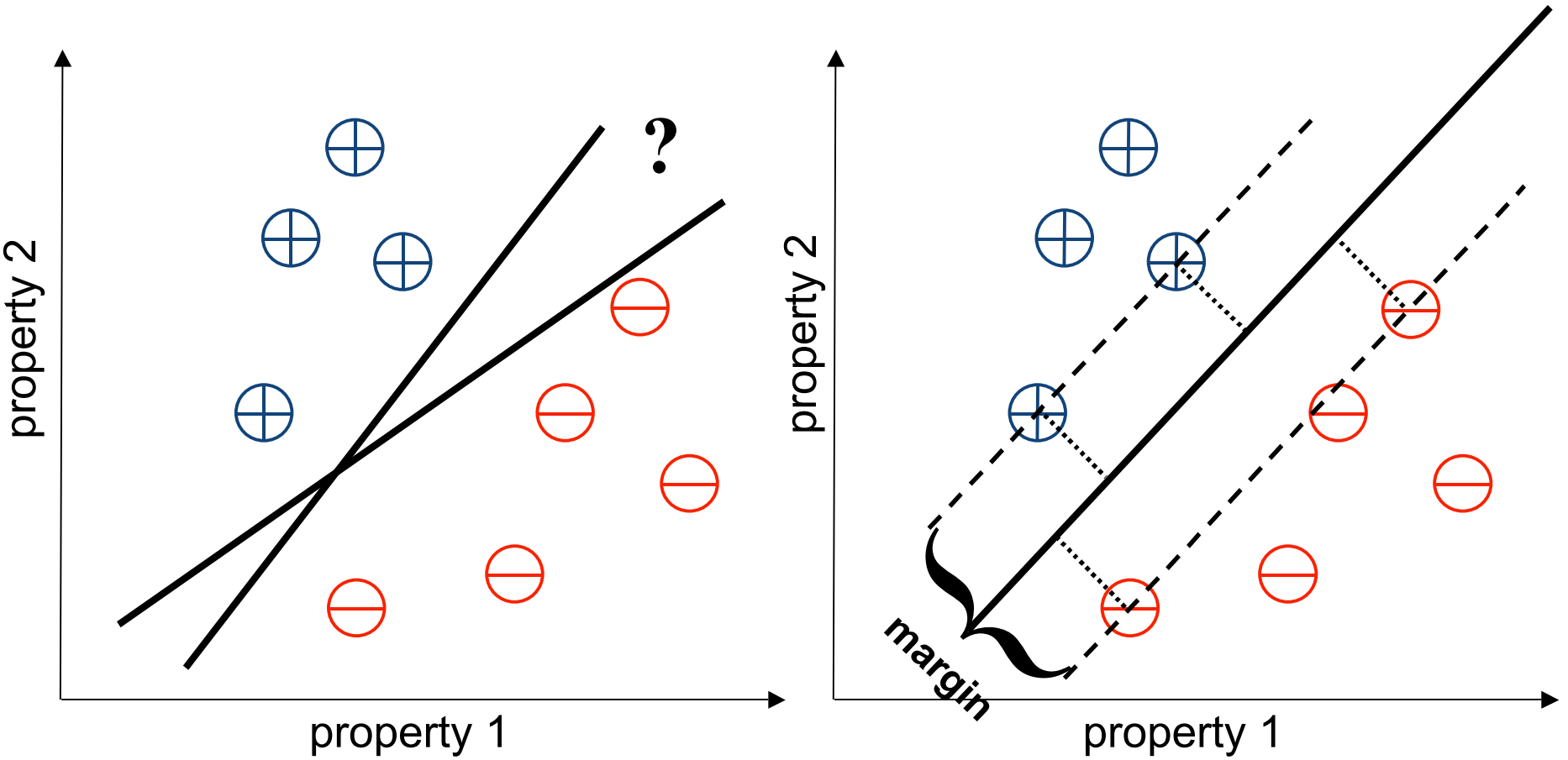
# Linear Separation



# Linear Separation

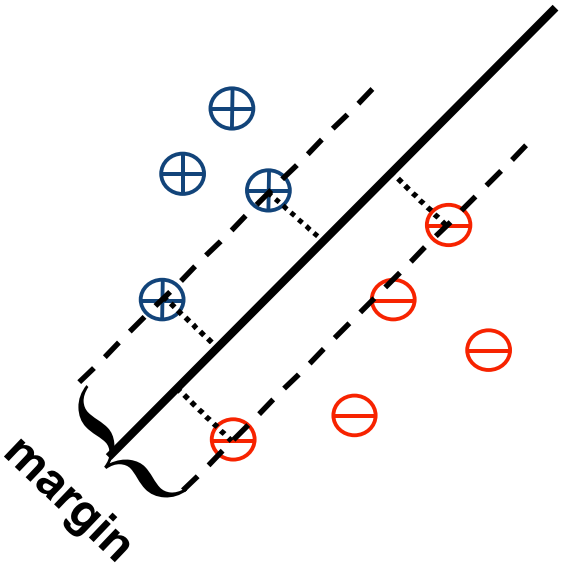


# Linear Separation with Margins




large margin => good generalization

# Large Margin Separation




**Idea:**

- Find hyperplane $f(\mathbf{x}) = (\mathbf{w} \cdot \mathbf{x})$ that maximizes margin

$$f(\mathbf{x}^+) - f(\mathbf{x}^-)$$

(with $\|\mathbf{w}\|_2 = 1$)

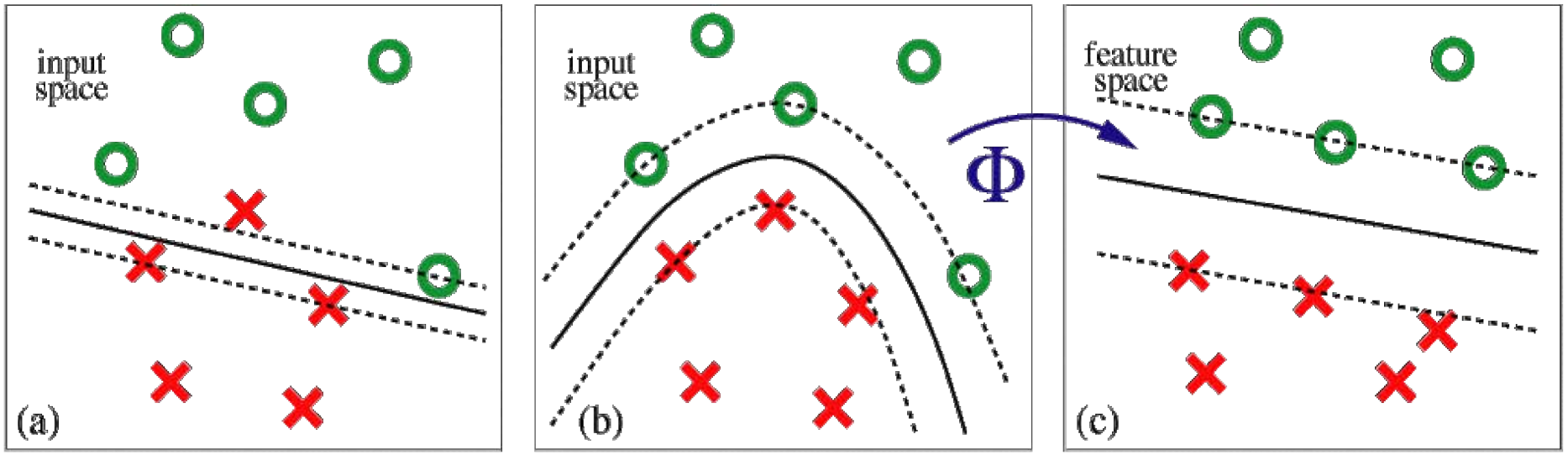
- Use $\mathrm{sgn}(f(\mathbf{x}) + b)$ for prediction

**Solution:**

- Linear combination of examples $\mathbf{w} = \sum_{n=1}^{N} \alpha_n \mathbf{x}_n$
- many $\alpha$'s are zero
- **Support Vector Machines**

  $\Rightarrow$ **Demo**

# Kernel Trick

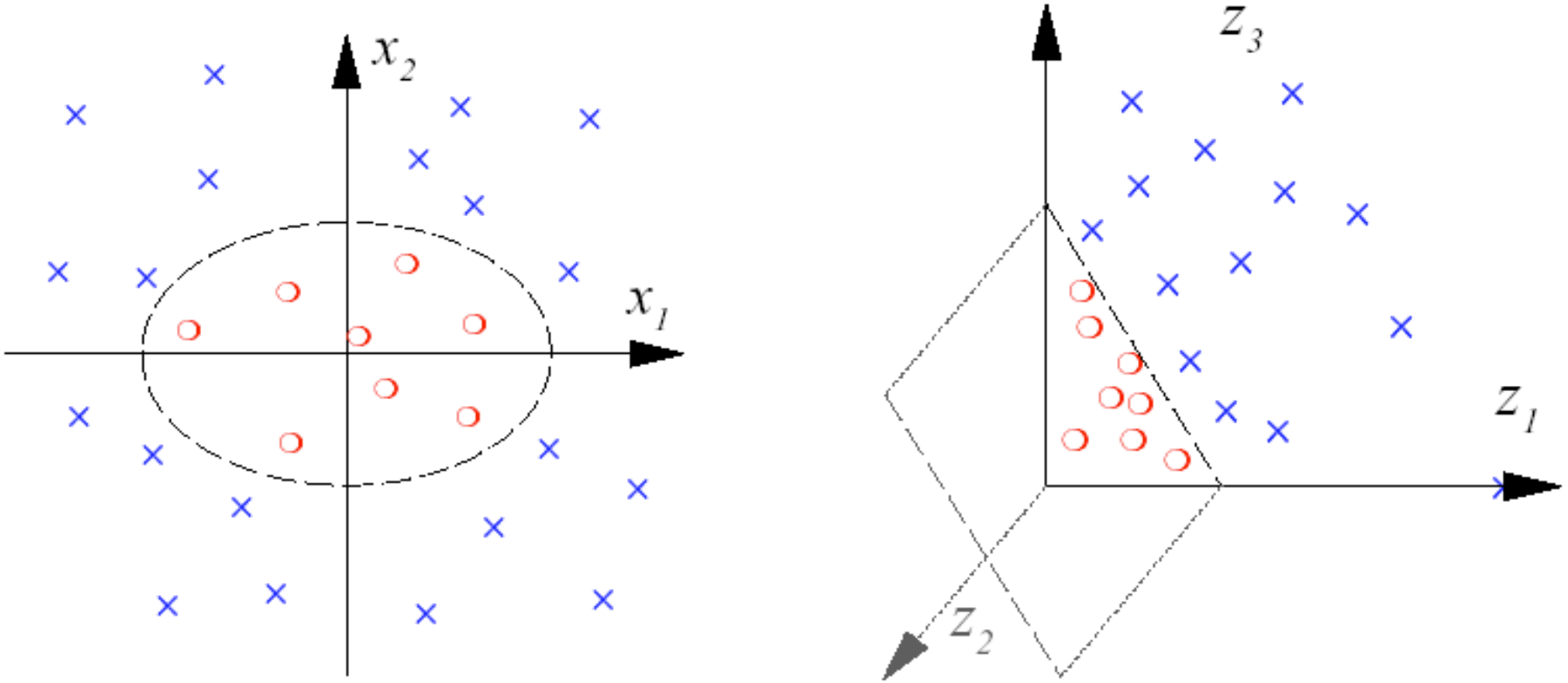



**Linear in input space** **Non-linear in input space** ⇐ **Linear in feature space**

$$k(\mathbf{x}, \mathbf{x}') = (\Phi(\mathbf{x}) \cdot \Phi(\mathbf{x}'))$$

# Example: Polynomial Kernel

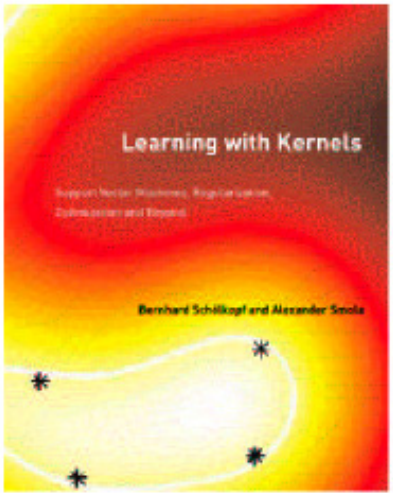
$$\Phi : \mathbb{R}^2 \to \mathbb{R}^3$$
$$(x_1, x_2) \mapsto (z_1, z_2, z_3) := (x_1^2, \sqrt{2}\, x_1 x_2, x_2^2)$$




$$k(\mathbf{x}, \mathbf{x}') = (\Phi(\mathbf{x}) \cdot \Phi(\mathbf{x}')) = (\mathbf{x} \cdot \mathbf{x}')^2$$

# Support Vector Machines

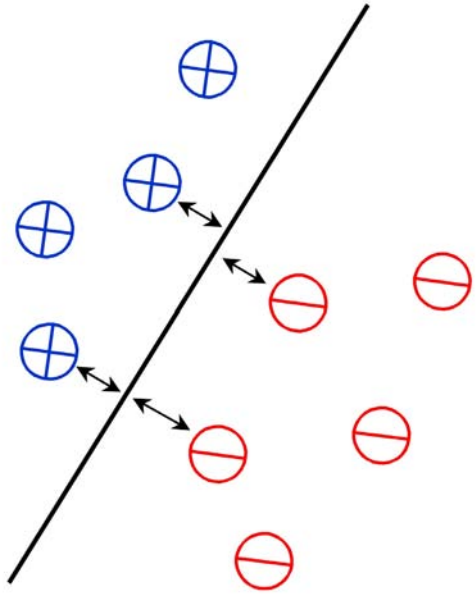
- **Demo:** Gaussian Kernel $k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}'\|}{2\sigma^2}\right)$
- Many other algorithms can use kernels
- Many other application specific kernels




For further information, cf.
`http://www.kernel-machines.org`,
`http://www.learning-with-kernels.org`

# Capabilities of Current Techniques

- Theoretically & algorithmically well understood:
  - **Classification with few classes**
  - Regression (real valued)
  - Novelty Detection

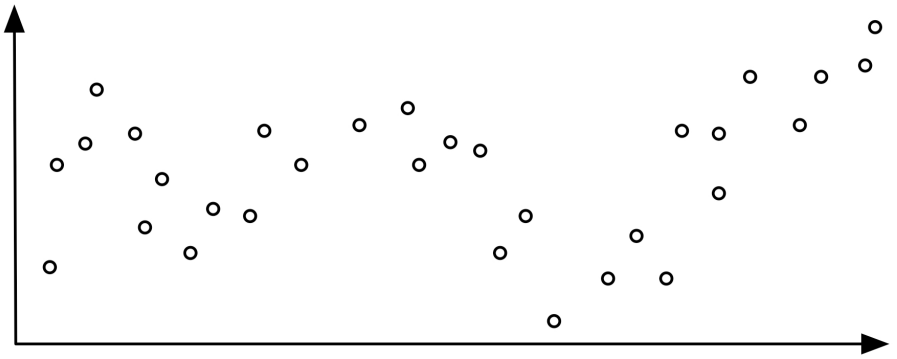


Bottom Line: *Machine Learning works well for relatively simple objects with simple properties*

- Current Research
  - Complex objects
  - Many classes
  - Complex learning setup (active learning)
  - Prediction of complex properties

# Capabilities of Current Techniques

- Theoretically & algorithmically well understood:
  - Classification with few classes
  - **Regression (real valued)**
  - Novelty Detection

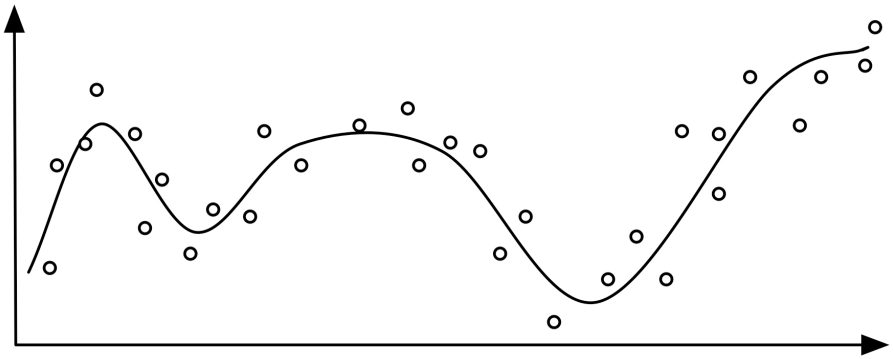


Bottom Line: *Machine Learning works well for relatively simple objects with simple properties*

- Current Research
  - Complex objects
  - Many classes
  - Complex learning setup (active learning)
  - Prediction of complex properties

# Capabilities of Current Techniques

- Theoretically & algorithmically well understood:
  - Classification with few classes
  - **Regression (real valued)**
  - Novelty Detection

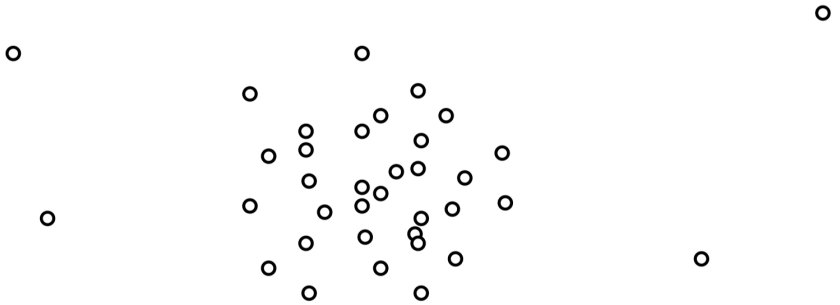


Bottom Line: *Machine Learning works well for relatively simple objects with simple properties*

- Current Research
  - Complex objects
  - Many classes
  - Complex learning setup (active learning)
  - Prediction of complex properties

# Capabilities of Current Techniques

- Theoretically & algorithmically well understood:
  - Classification with few classes
  - Regression (real valued)
  - **Novelty Detection**

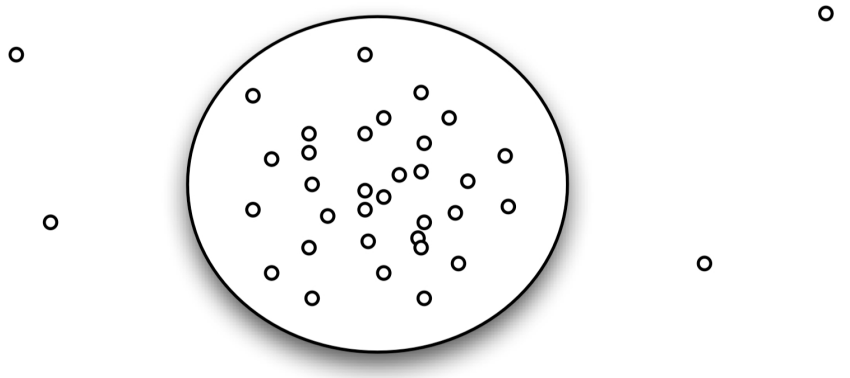


Bottom Line: *Machine Learning works well for relatively simple objects with simple properties*

- Current Research
  - Complex objects
  - Many classes
  - Complex learning setup (active learning)
  - Prediction of complex properties

# Capabilities of Current Techniques

- Theoretically & algorithmically well understood:
  - Classification with few classes
  - Regression (real valued)
  - **Novelty Detection**



Bottom Line: *Machine Learning works well for relatively simple objects with simple properties*

- Current Research
  - Complex objects
  - Many classes
  - Complex learning setup (active learning)
  - Prediction of complex properties

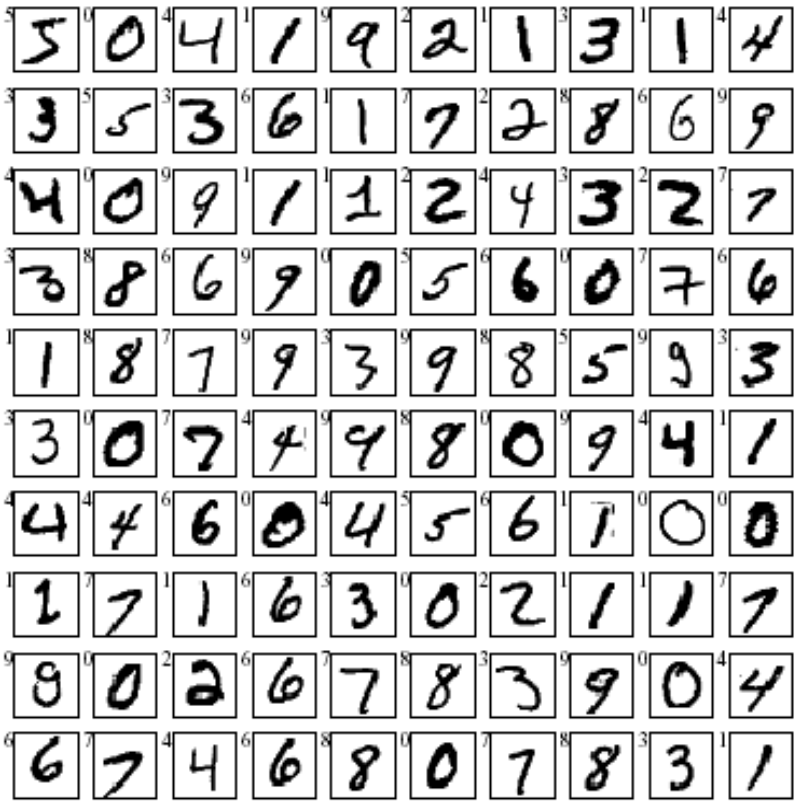
# Many Applications

- Handwritten Letter/Digit recognition
- Face/Object detection in natural scenes
- Brain-Computer Interfacing
- Gene Finding
- Drug Discovery
- Intrusion Detection Systems (unsupervised)
- Document Classification (by topic, spam mails)
- Non-Intrusive Load Monitoring of electric appliances
- Company Fraud Detection (Questionaires)
- Fake Interviewer identification in social studies
- Optimized Disk caching strategies
- Optimal Disk-Spin-Down prediction
- …

# MNIST Benchmark



handwritten character benchmark (60000 training & 10000 test examples, $28 \times 28$)



SVM with polynomial kernel $k(\mathbf{x}, \mathbf{x}') = (\mathbf{x} \cdot \mathbf{x}')^d$

(considers d-th order correlations of pixels)

# MNIST Error Rates

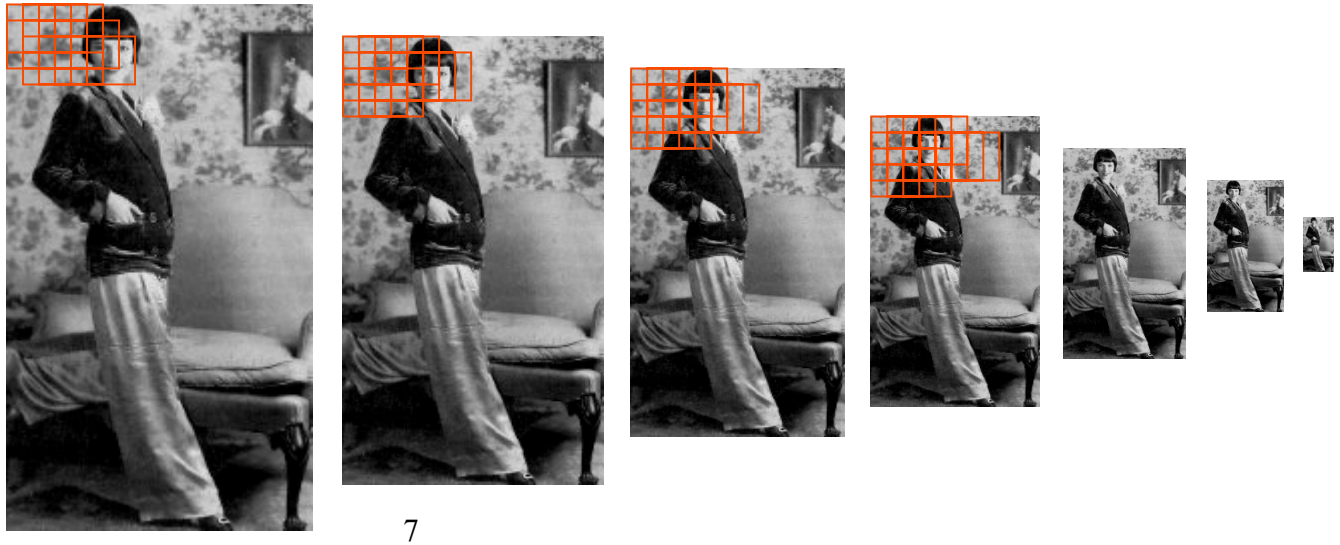
| Classifier | test error | reference |
|---|---|---|
| linear classifier | 8.4% | *Bottou et al. (1994)* |
| 3-nearest-neighbour | 2.4% | *Bottou et al. (1994)* |
| SVM | 1.4% | *Burges and Schölkopf (1997)* |
| Tangent distance | 1.1% | *Simard et al. (1993)* |
| LeNet4 | 1.1% | *LeCun et al. (1998)* |
| Boosted LeNet4 | 0.7% | *LeCun et al. (1998)* |
| Translation invariant SVM | 0.56% | *DeCoste and Schölkopf (2002)* |

# Face Detection

1.

Classifier → face / non-face

2. Search



$$\sum_{l=1}^{7} 600 \cdot 450 \cdot 0.7^{2(l-1)}$$

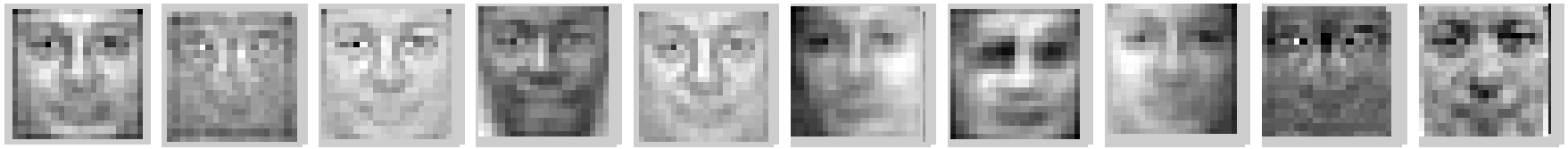
$\Rightarrow$ 525,820 patches

# Fast Face Detection

Note: for "easy" patches, a quick and inaccurate classification is sufficient.

Method: sequential approximation of the classifier in a Hilbert space

Result: a set of face detection filters



*Romdhani, Blake, Schölkopf, & Torr, 2001*

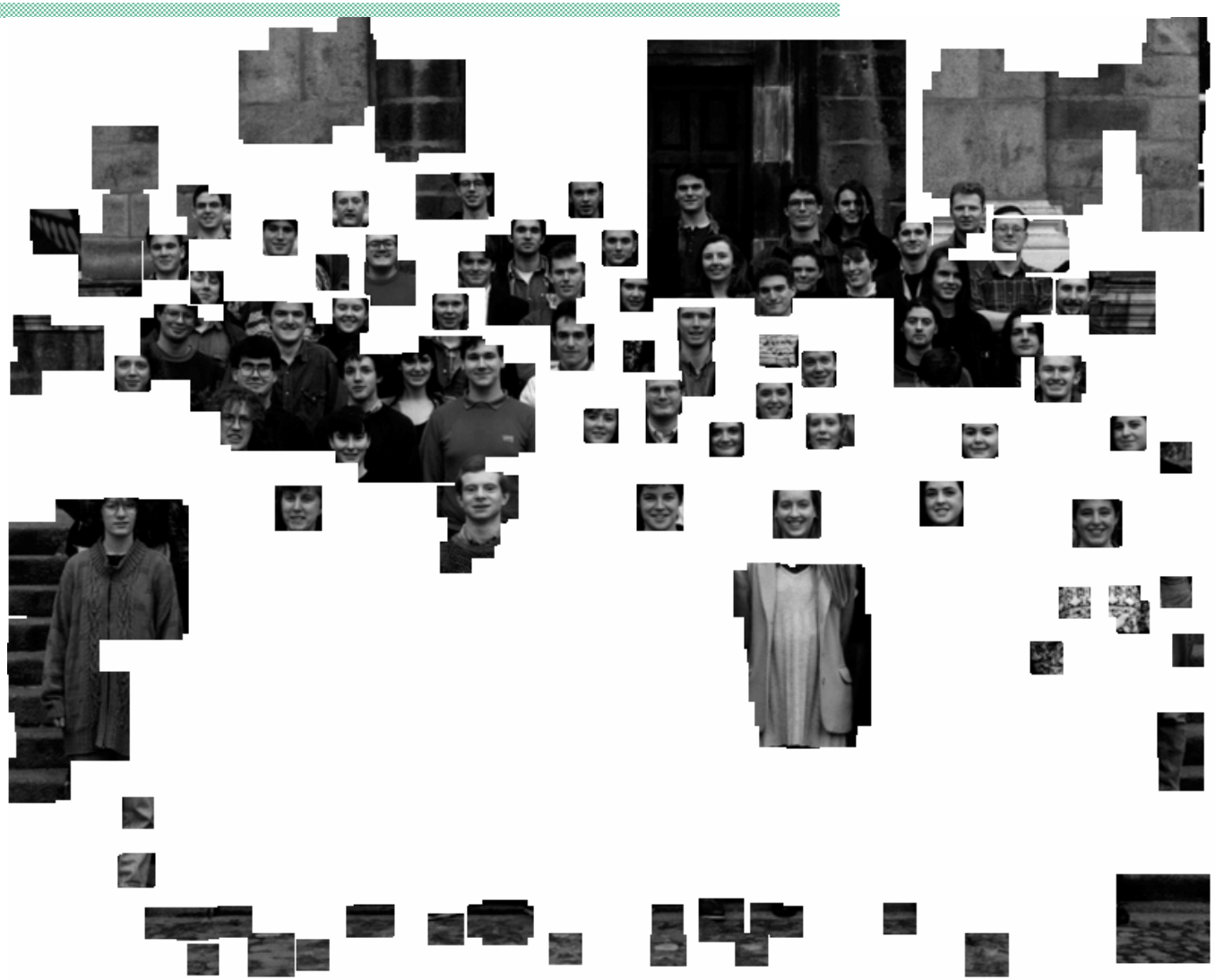
# Example: 1280x1024 Image

1 Filter, 19.8% patches left

# Example: 1280x1024 Image

10 Filters, 0.74% Patches left

# Example: 1280x1024 Image

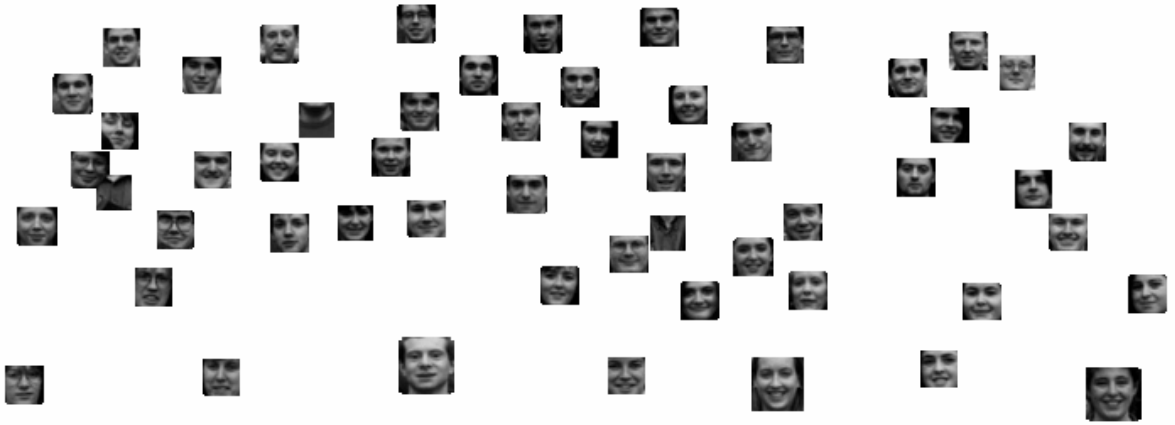


20 Filters, 0.06% Patches left

# Example: 1280x1024 Image

30 Filters, 0.01% Patches left

# Example: 1280x1024 Image



70 Filters, 0.007 % patches left

# Single Trial Analysis of EEG:towards BCI

Gabriel Curio

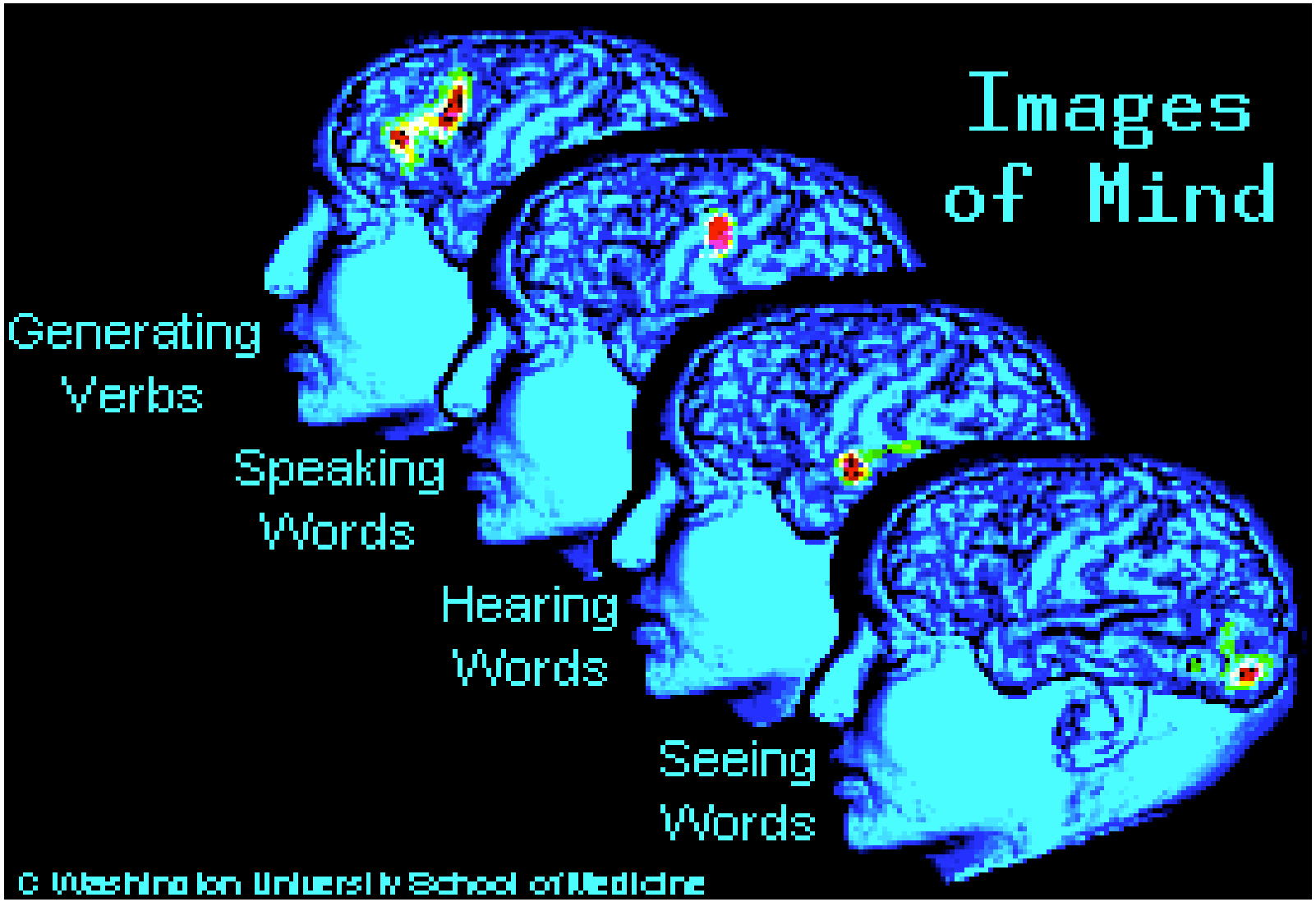
Benjamin Blankertz

Klaus-Robert Müller

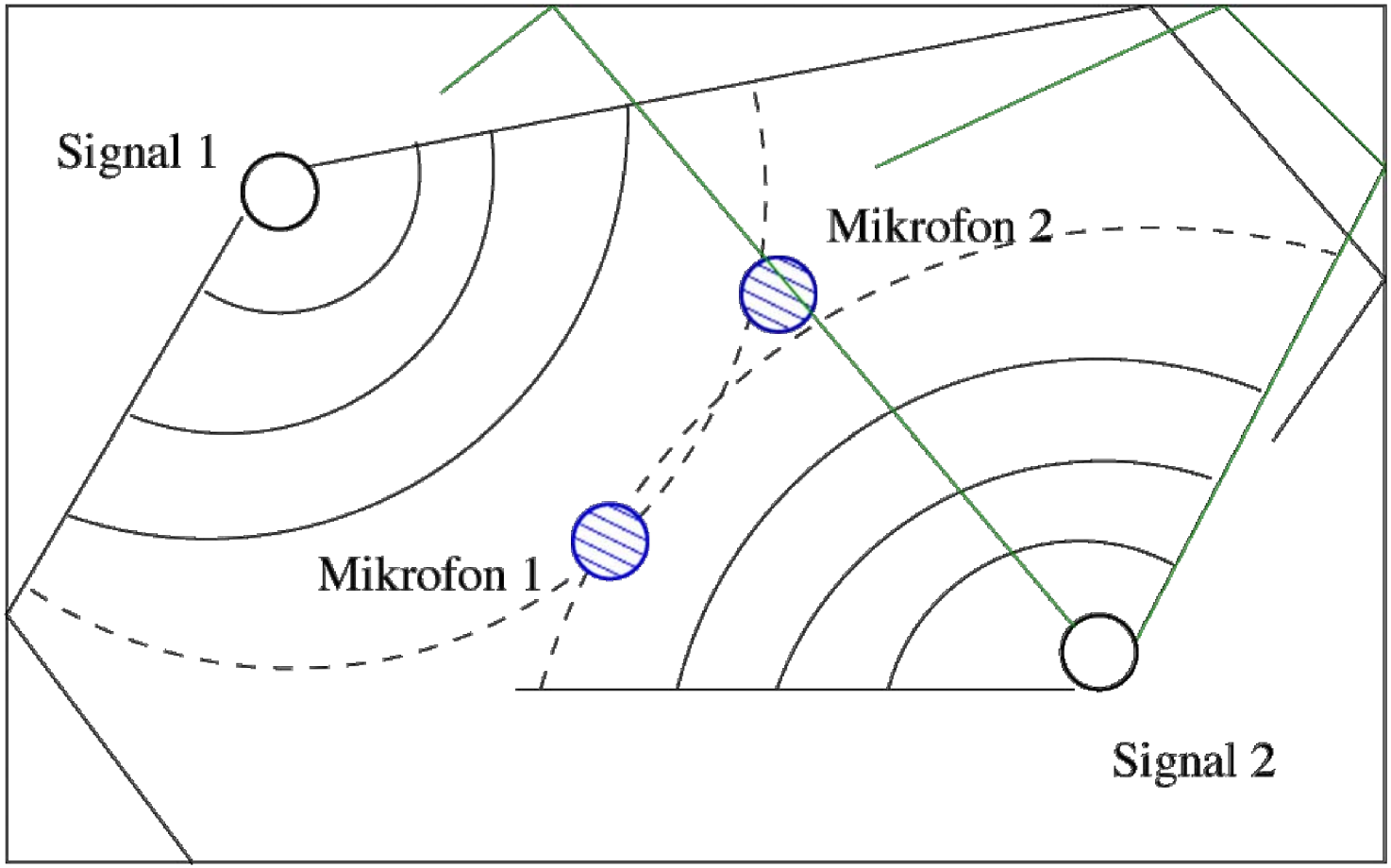
Neurophysics Group
Dept. of Neurology
Klinikum Benjamin Franklin
Freie Universität
Berlin, Germany

Intelligent Data Analysis Group, Fraunhofer-FIRST
Berlin, Germany

# Cerebral Cocktail Party Problem



# The Cocktail Party Problem




How to decompose **superimposed** signals?

**Analogous** Signal Processing problem as for cocktail party problem

# The Cocktail Party Problem

- input: 3 mixed signals
- algorithm: enforce *independence* ("independent component analysis") via temporal de-correlation

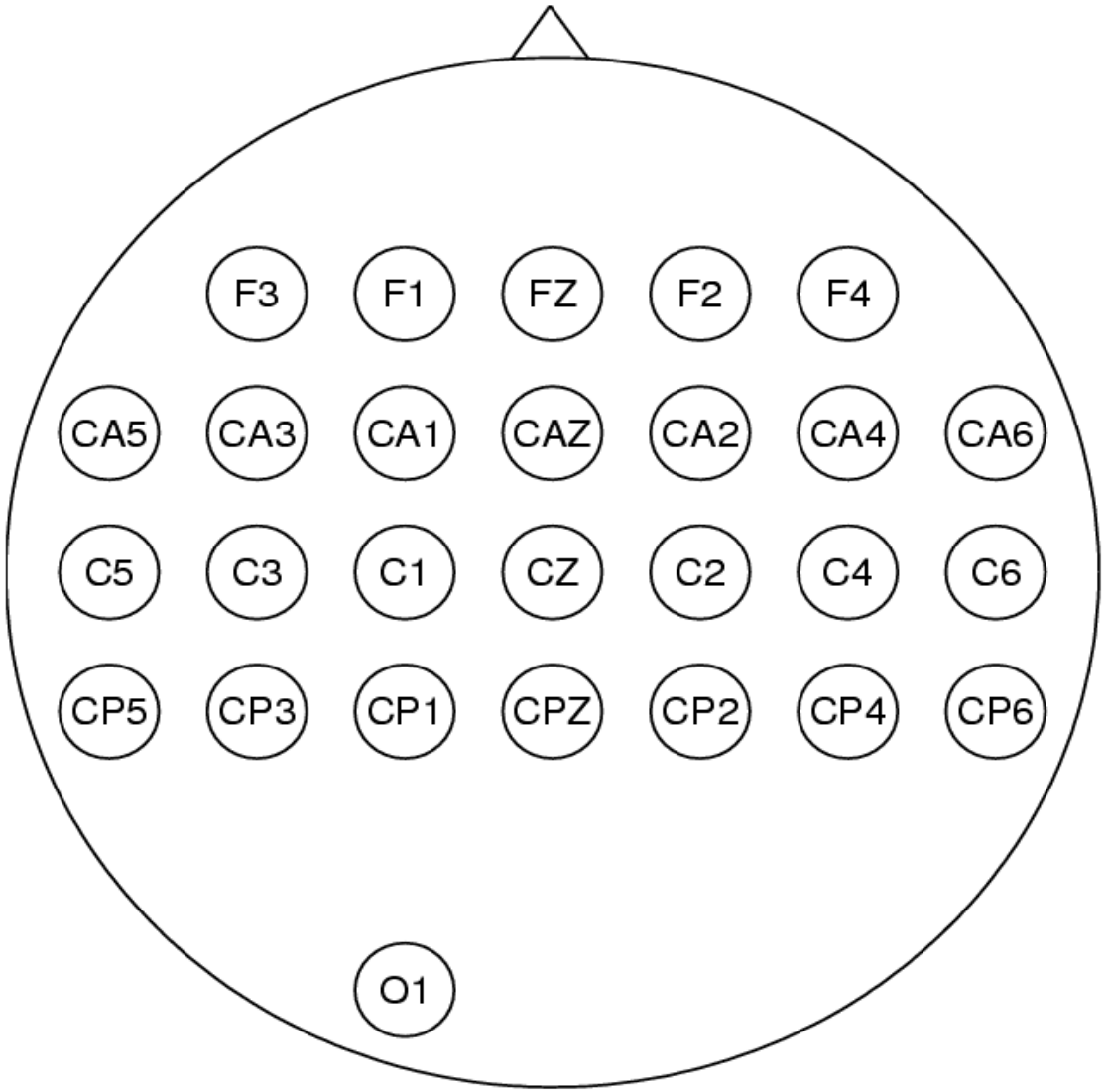
- output: 3 separated signals

"Imagine that you are on the edge of a lake and a friend challenges you to play a game. The game is this: Your friend digs two narrow channels up from the side of the lake […]. Halfway up each one, your friend stretches a handkerchief and fastens it to the sides of the channel. As waves reach the side of the lake they travel up the channels and cause the two handkerchiefs to go into motion. You are allowed to look only at the handkerchiefs and from their motions to answer a series of questions: How many boats are there on the lake and where are they? Which is the most powerful one? Which one is closer? Is the wind blowing?" (**Auditory Scene Analysis**, *A. Bregman* )

*(Demo: Andreas Ziehe, Fraunhofer FIRST, Berlin)*

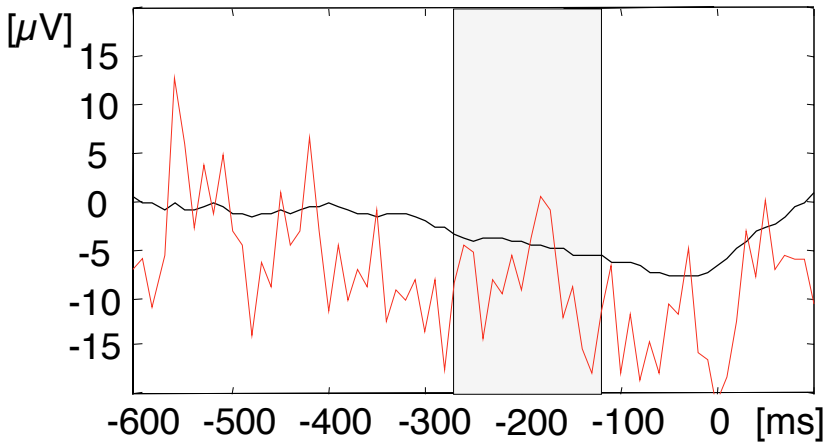
# Minimal Electrode Configuration

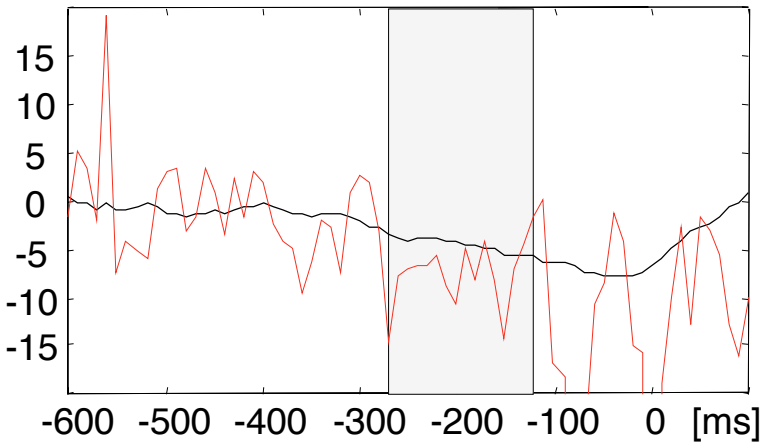
- coverage: bilateral primary sensorimotor cortices
- 27 scalp electrodes
- reference: nose
- bandpass: 0.05 Hz - 200 Hz
- ADC 1 kHz
- downsampling to 100 Hz
- EMG (forearms bilaterally): m. flexor digitorum
- EOG
- event channel: keystroke timing (ms precision)



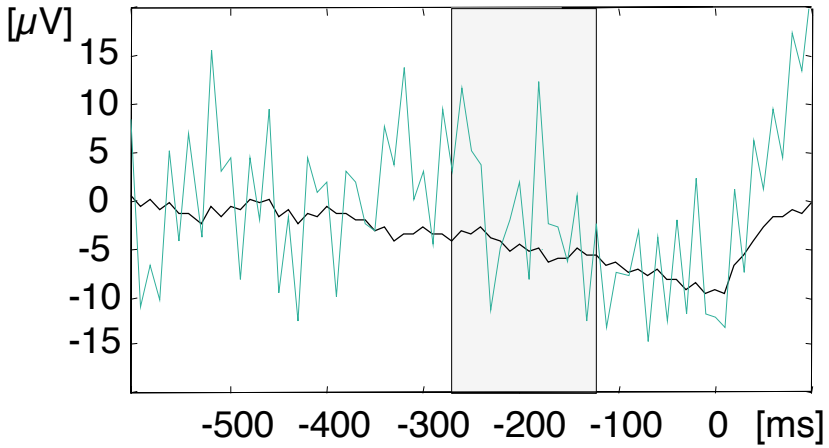
# Single Trial vs. Averaging

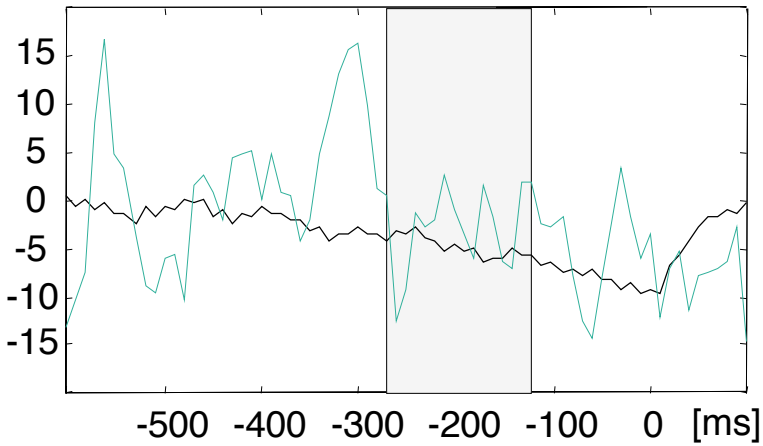
LEFT
hand
(ch. C4)



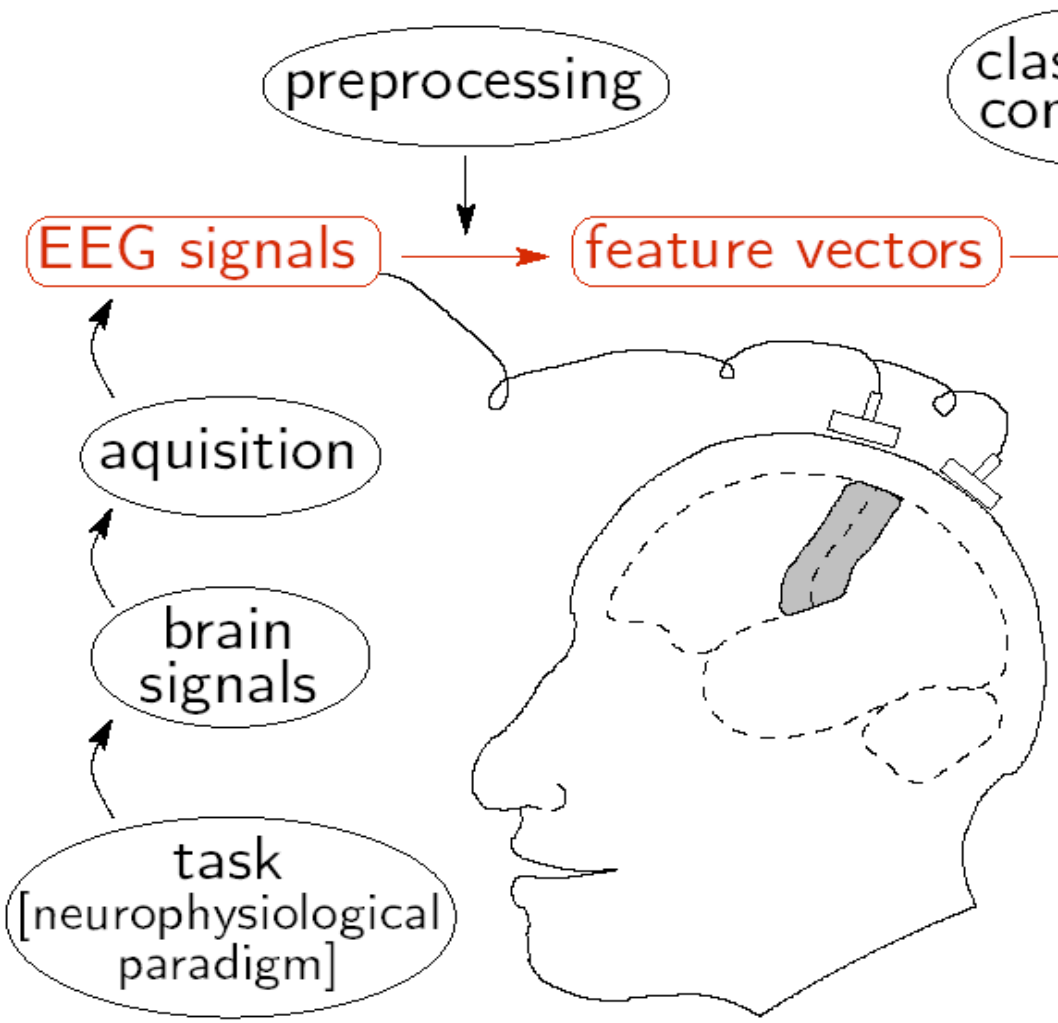




**RIGHT**
hand
(ch. **C3**)



# BCI Setup




ACQUISITION modes:
- few single electrodes
- 32-128 channel electrode caps
- subdural macroelectrodes
- intracortical multi-single-units

EEG parameters:
- slow cortical potentials
- $\mu$/ $\beta$ amplitude modulations
- Bereitschafts-/motor-potential

TASK alternatives:
- feedback control
- imagined movements
- movement (preparation)
- mental state diversity


NIPS 2003
5

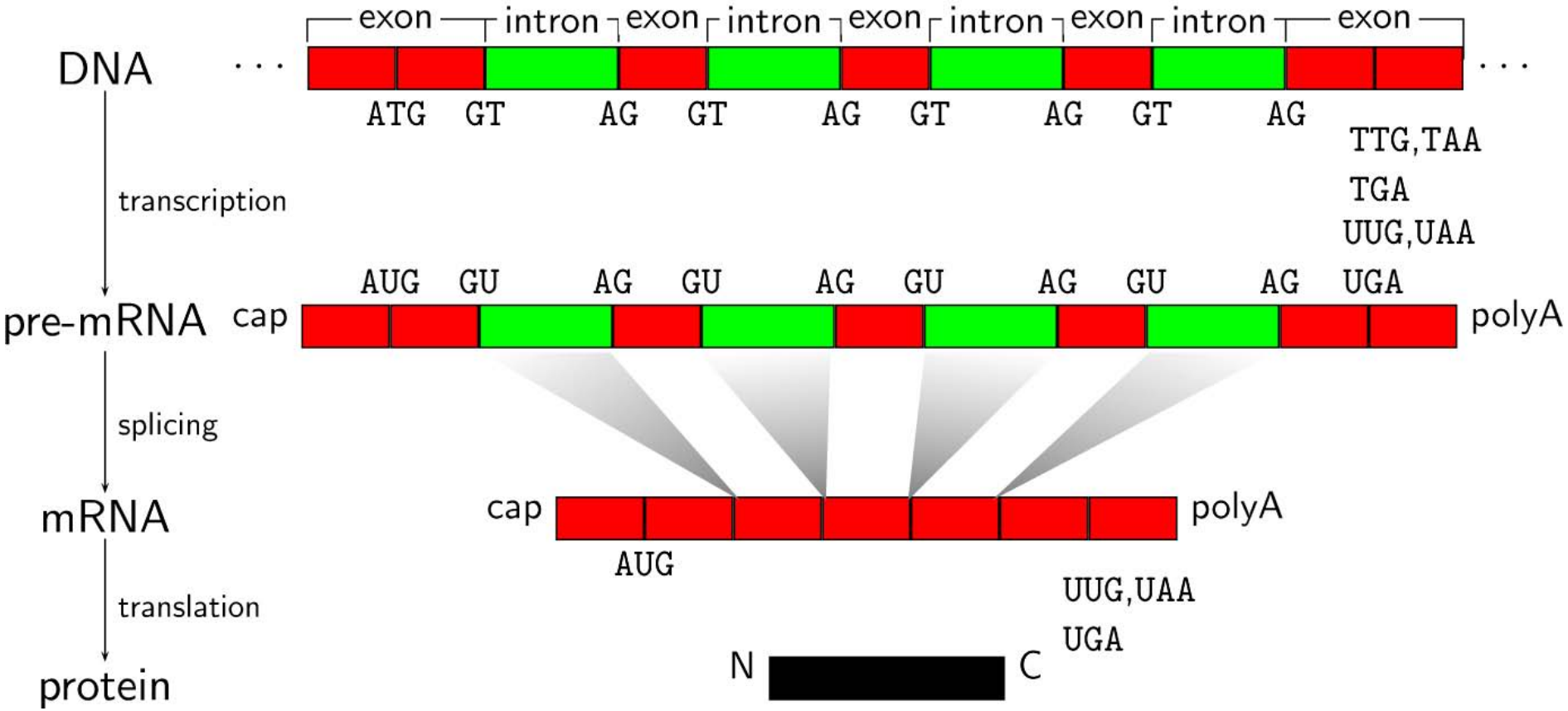
# Finding Genes on Genomic DNA



**Splice Sites: on the boundary**

- **Exons (may code for protein)**
- **Introns (noncoding)**

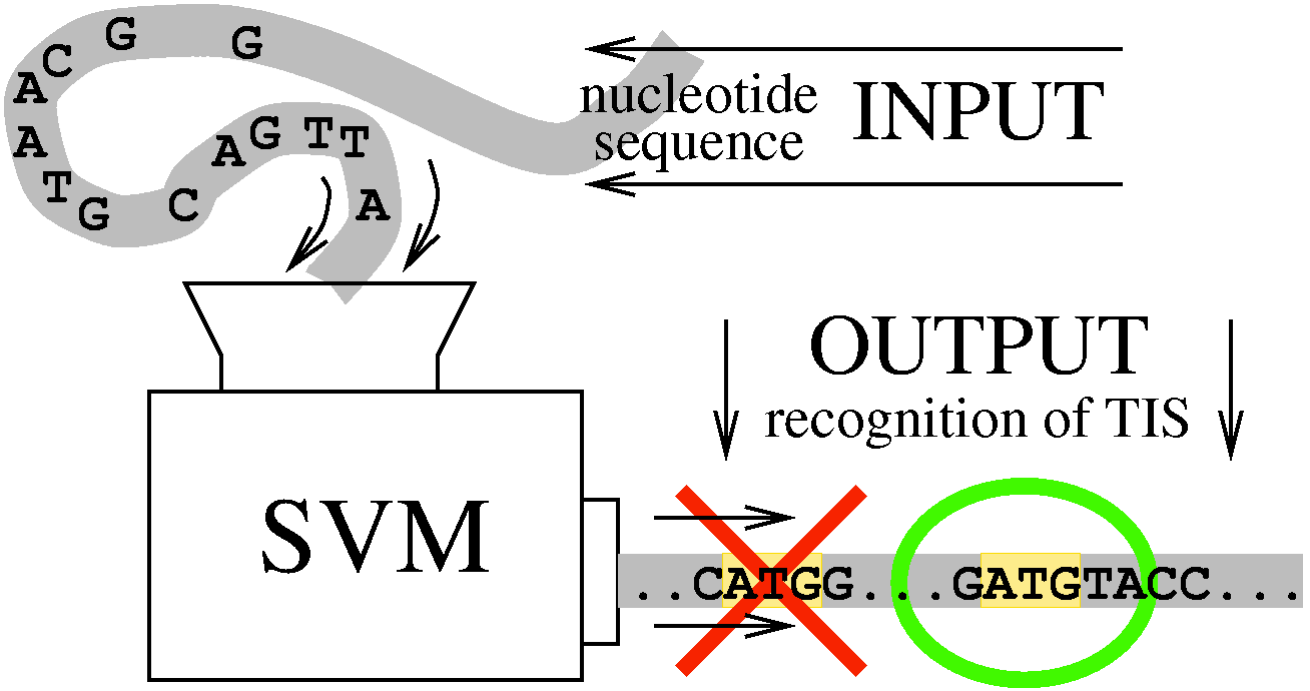
**Coding region starts with Translation Initiation Site (TIS: "ATG")**



# Application: TIS Finding



**Engineering Support Vector Machine (SVM) Kernels That Recognize Translation Initiation Sites (TIS)**



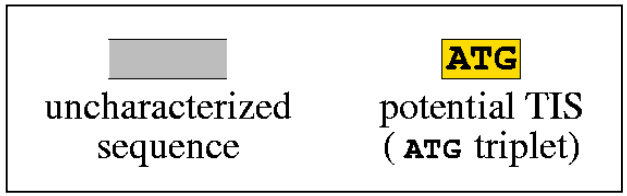

GMD.SCAI
Institute for Algorithms and Scientific Computing

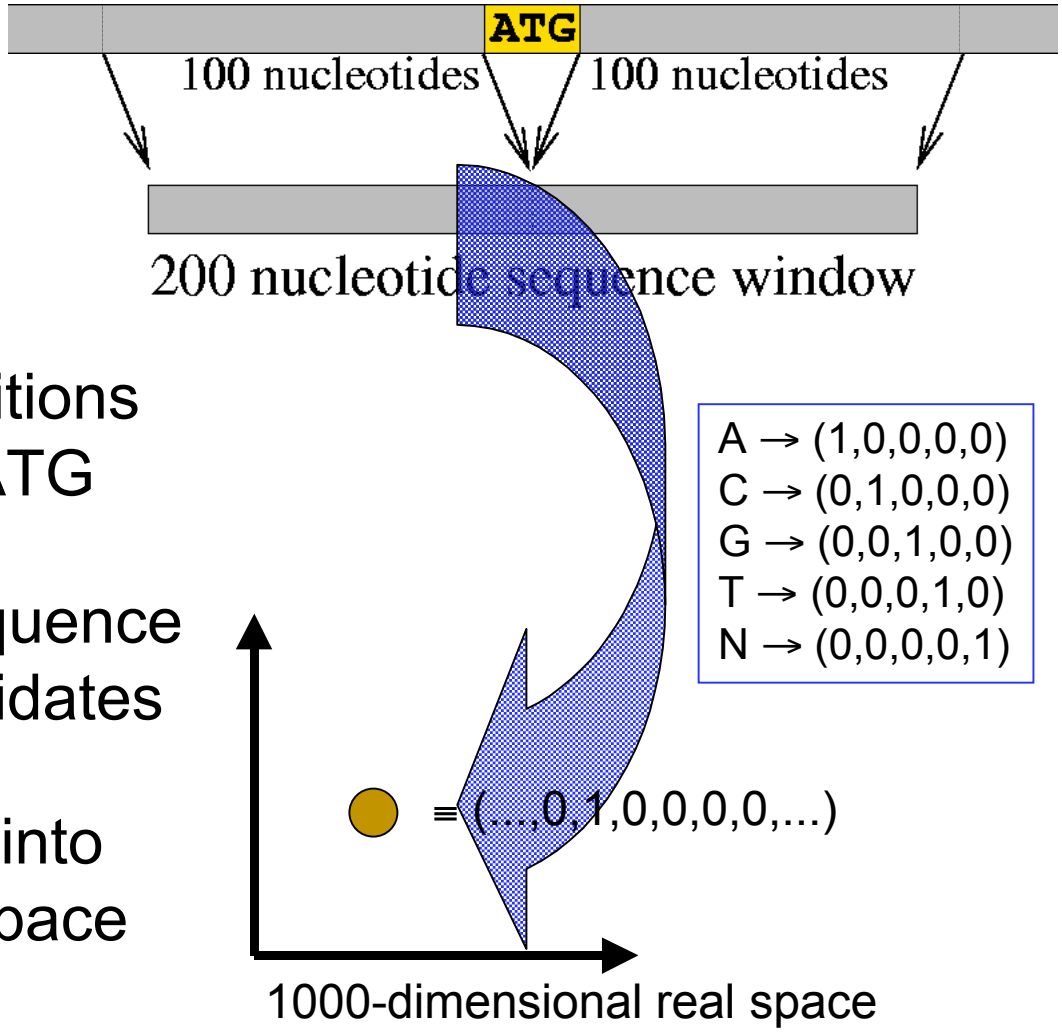
**Alexander Zien**
Thomas Lengauer

GMD.FIRST
Institute for Computer Architecture and Software Technology

**Gunnar Rätsch**
Sebastian Mika
Bernhard Schölkopf
Klaus-Robert Müller

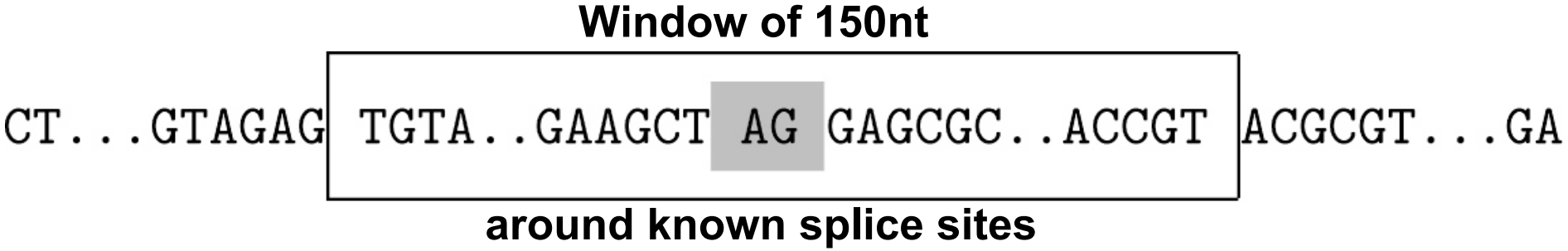
# TIS Finding: Classification Problem



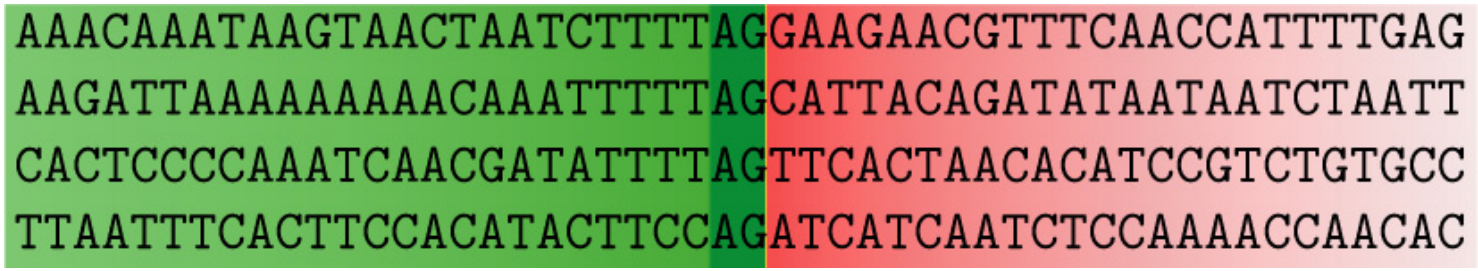




- Select candidate positions for TIS by looking for ATG
- Build fixed-length sequence representation of candidates
- Transform sequence into representaion in real space

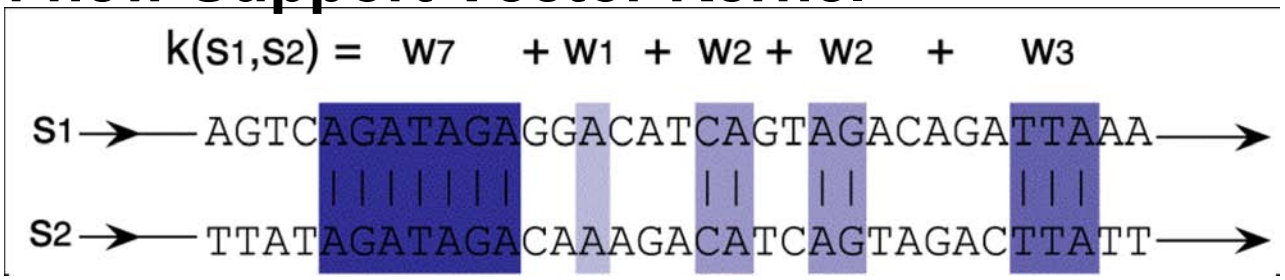
# 2-class Splice Site Detection



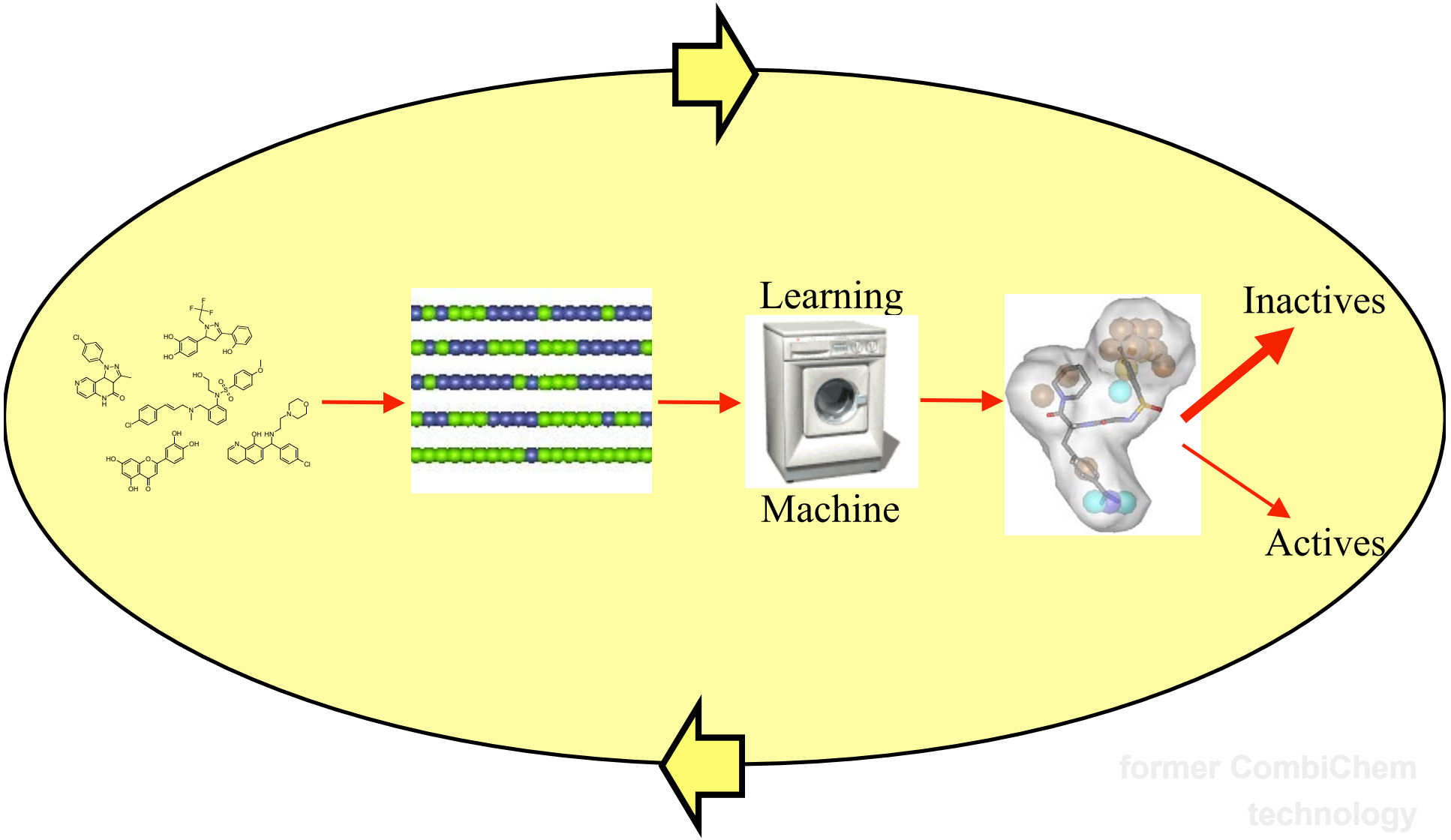

**Positive examples:** fixed window around a true splice site
**Negative examples:** generated by shifting the window




## Design of new Support Vector Kernel

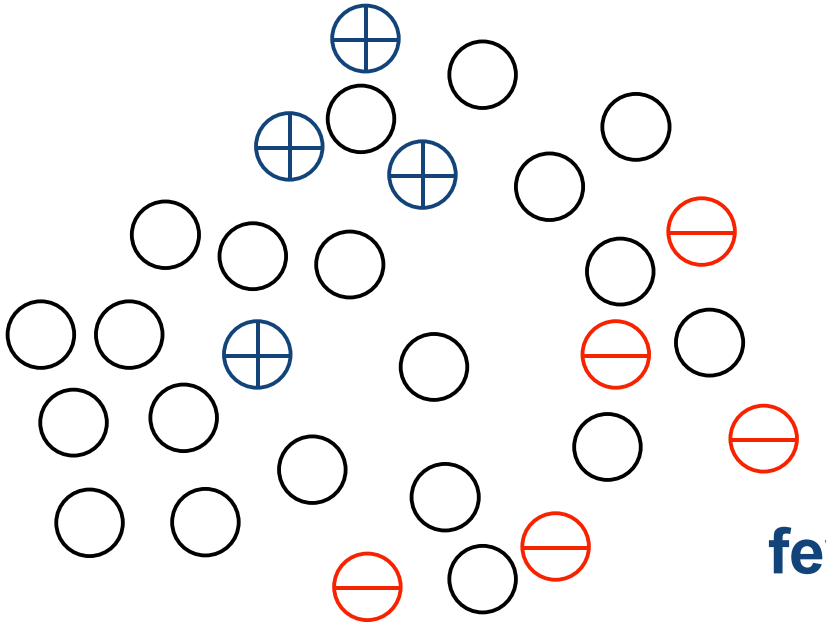


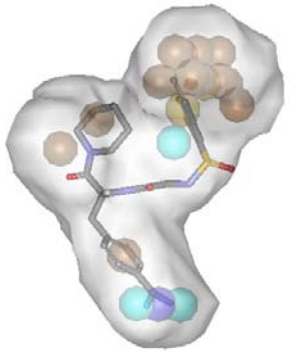
# The Drug Design Cycle




former CombiChem technology

# Three types of Compounds/Points



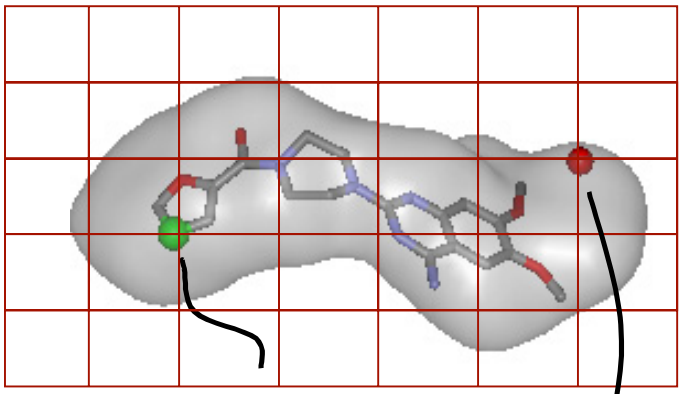


**few ⊕ actives**

**more ⊖ inactives**

**plenty ○ untested**

# Shape/Feature Descriptor



bit number
254230

**bit** = { Shape, Feature type, Feature location }

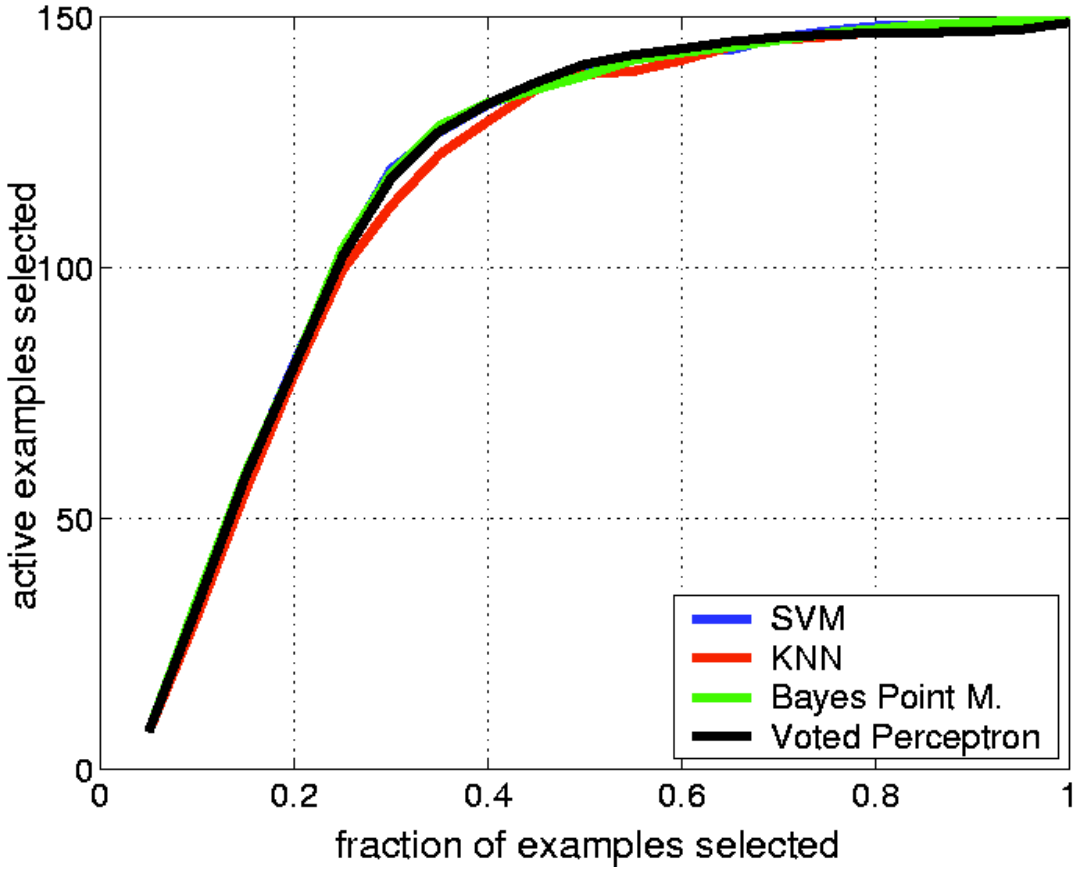

~$10^5$ bits

S. Putta, *A Novel Shape/Feature Descriptor*, 2001

# Maximizing the Number of Hits

**Total number of active examples selected after each batch**

**On Thrombin dataset**




***Largest* Selection Strategy**

# Concluding Remarks

- Computational Challenges
  - Algorithms can work with 100.000's of examples (need $\mathcal{O}(N^2)$ - $\mathcal{O}(N^3)$ operations
  - Usually model parameters to be tuned (cross-validation ⇒ computationally expensive)
  - Need computer clusters and Job scheduling systems (pbs, gridengine)
  - Often use MATLAB (to be replaced by python: help!)
- Machine learning is an exciting research area …
- … involving Computer Science, Statistics & Mathematics
- … with…
  - **a large number of present and future applications (in all situations where data is available, but explicit knowledge is scarce)…**
  - **an elegant underlying theory…**
  - **and an abundance of questions to study.**

**New computational biology group in Tübingen: looking for people to hire**

# Thanks for Your Attention!

**Gunnar Rätsch**

http://www.tuebingen.mpg.de/~raetsch

Gunnar.Raetsch@tuebingen.mpg.de

**Collegues & Contributors:** K. Bennett, **G. Dornhege**, A. Jagota, M. Kawanabe, J. Kohlmorgen, S. Lemm, C. Lemmen, **P. Laskov**, J. Liao, T. Lengauer, R. Meir, S. Mika, **K-R. Müller**, T. Onoda, A. Smola, **C. Schäfer**, **B. Schölkopf**, R. Sommer, **S. Sonnenburg**, J. Srinivasan, K. Tsuda, **M. Warmuth**, J. Weston, A. Zien

**Special Thanks:** Nora Toussaint, Julia Lüning, Matthias Noll

THE AUSTRALIAN NATIONAL UNIVERSITY

idalab intelligent data analysis

FIRST Fraunhofer Institut Rechnerarchitektur und Softwaretechnik

UC SANTA CRUZ

MAX-PLANCK-GESELLSCHAFT