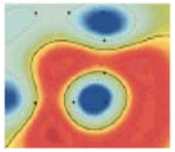
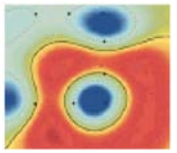


Applied Machine Learning

Timon Schroeter
Konrad Rieck
Soeren Sonnenburg

Intelligent Data Analysis Group
Fraunhofer FIRST
http://ida.first.fhg.de/

Roadmap

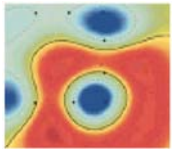


- Some Background
- SVMs & Kernels
- Applications

Rationale: **Let computers learn, to allow humans to**

- **to automate processes**
- **to understand highly complex data**

Example: 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

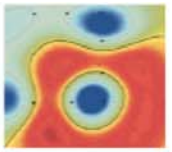
There is considerable synergy for Machine Learning at Santa Cruz:
-New department of Applied Math and Statistics with an emphasis on Bayesian Methods http://www.ams.ucsc.edu/
-- New department of Biomolecular Engineering
http://www.cbse.ucsc.edu/

…

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

Training: analyze previous emails
Application: classify new emails

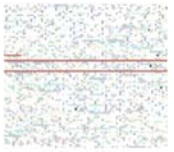
Problem Formulation



The "World":

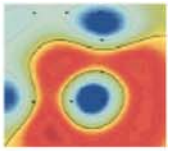
- Data: Pairs (***x***, *y*)
  - Featurevector ***x***
    - Individual features e.g. $x \in R$
    - e.g. Volume, Mass, RGB-Channels
  - Lables $y \in \{+1, -1\}$
- Unknown Target Function $y = f(\boldsymbol{x})$
- Unknown Distribution $x \sim p(\boldsymbol{x})$
- Objective: Given new ***x*** predict *y*

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



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"

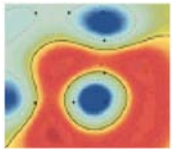
$$h^* = \text{argmin}_{h \in \mathcal{H}} \; L(h) \text{ with } \boxed{L(h) := \mathbf{E}_{\mathbb{X} \times \mathbb{Y}} \; l(\mathbb{Y}, h(\mathbb{X}))}$$

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

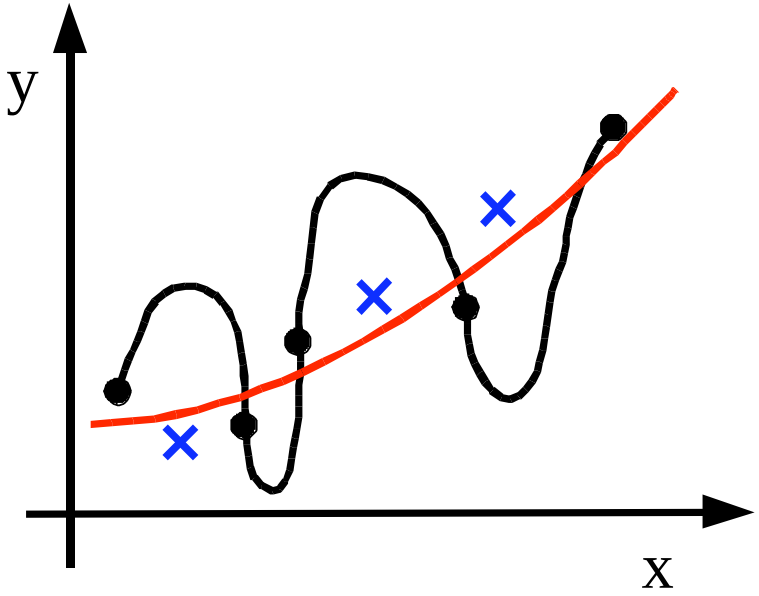
Solution: Find empirical minimizer

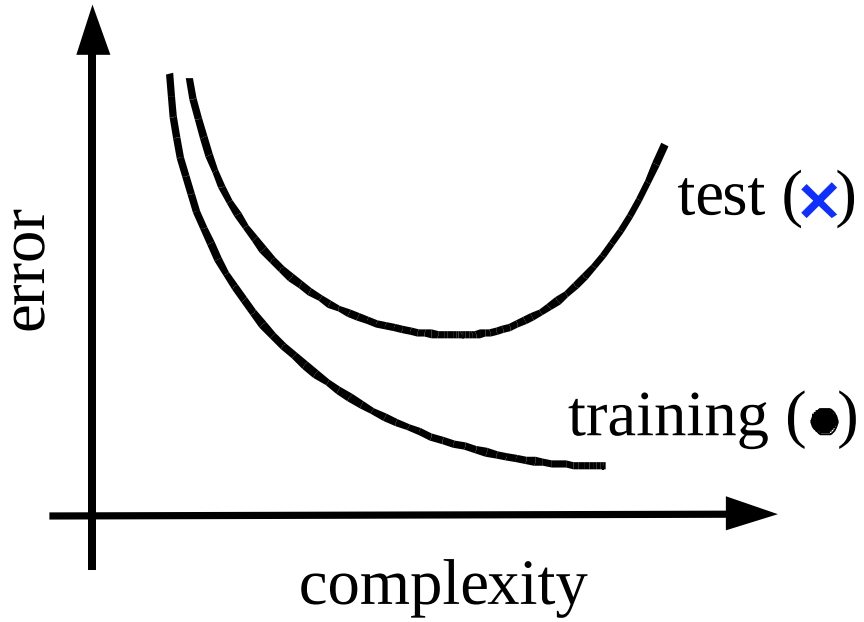
$$\hat{h}_N = \text{argmin}_{h \in \mathcal{H}} \; \frac{1}{N} \sum_{n=1}^{N} l(y_n, h(\mathbf{x}_n))$$

Problem Formulation



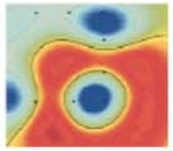
- Want model to generalize
- Need to find a good level of complexity



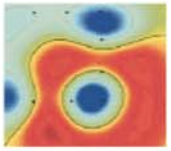


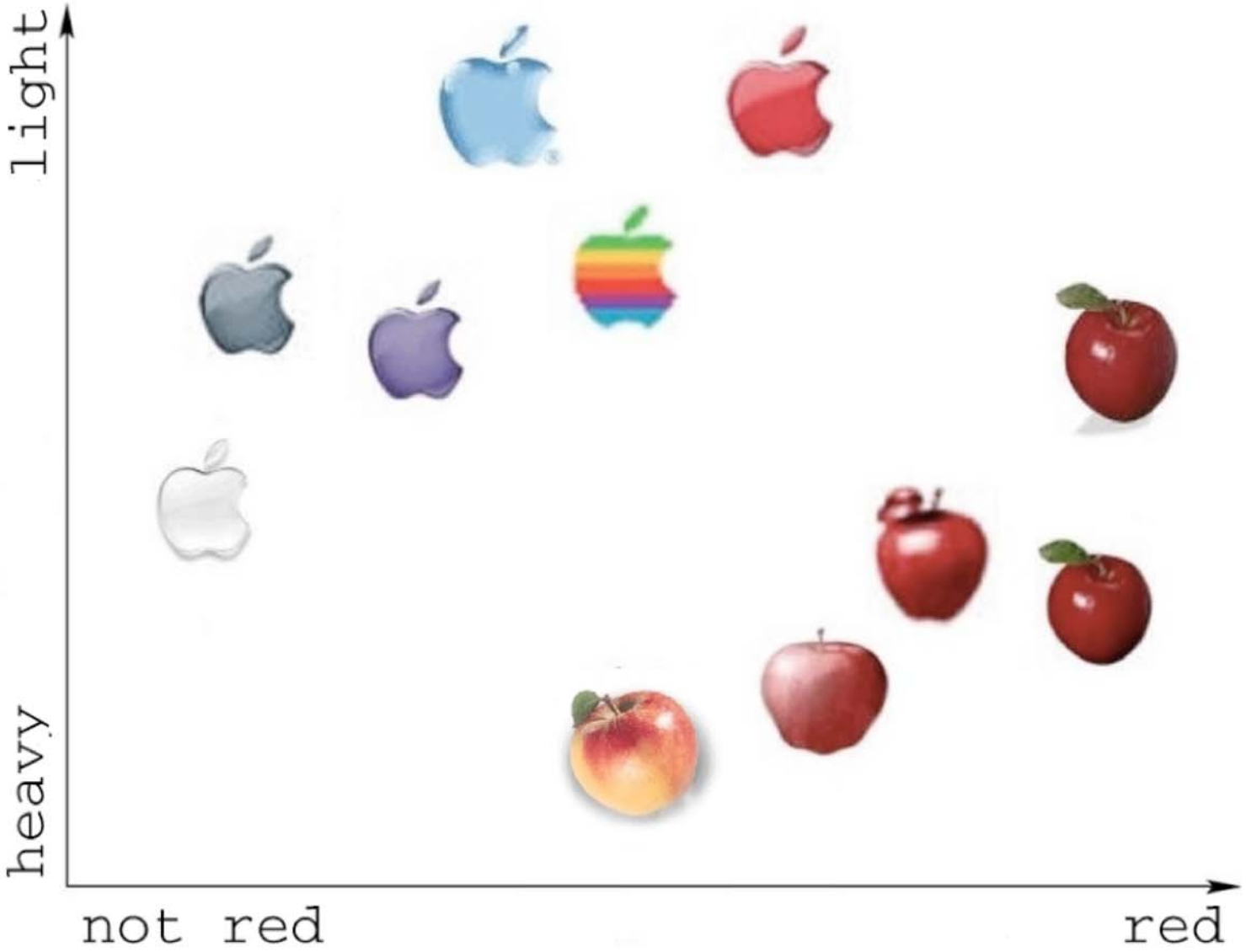
- In practice e.g. model / parameter selection via crossvalidation

Example: Natural vs. Plastic Apples

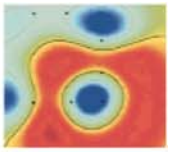


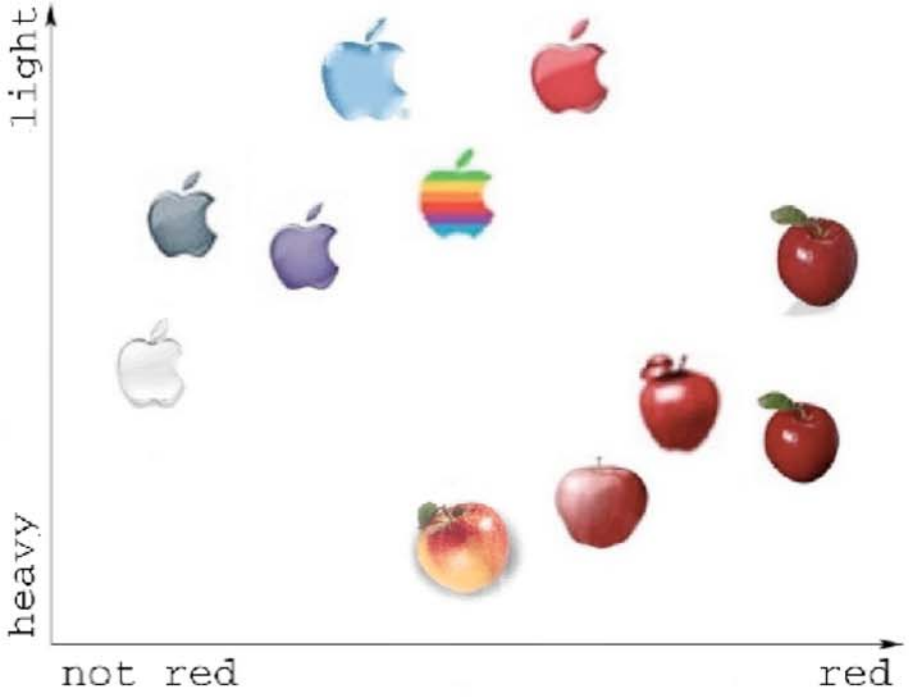
Example: Natural vs. Plastic Apples

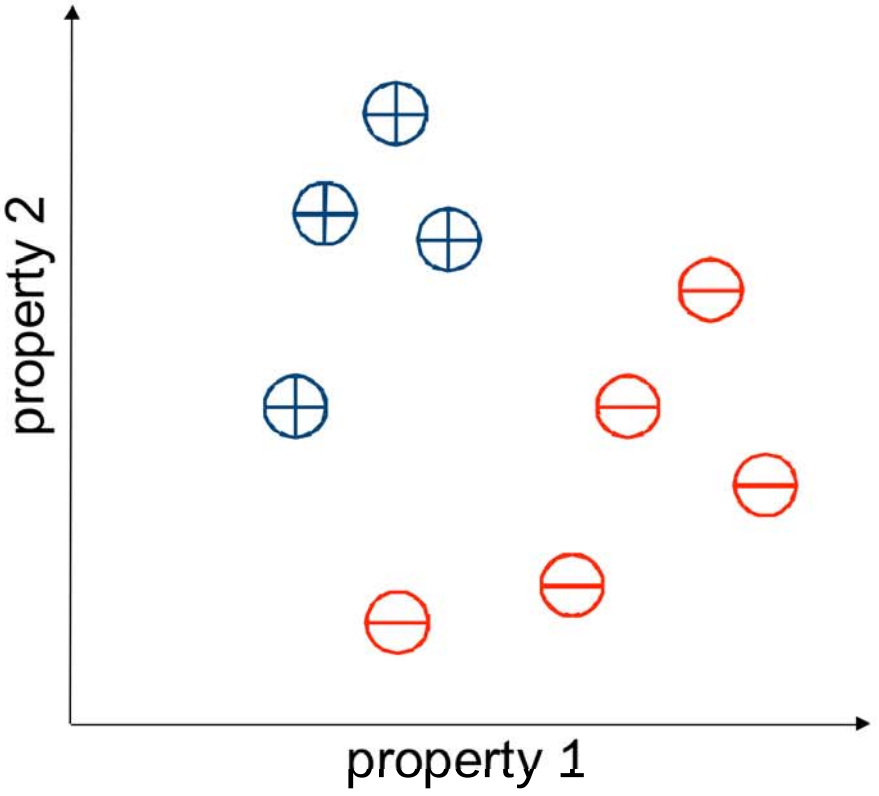




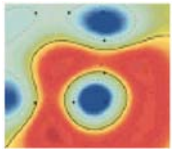
Linear Separation

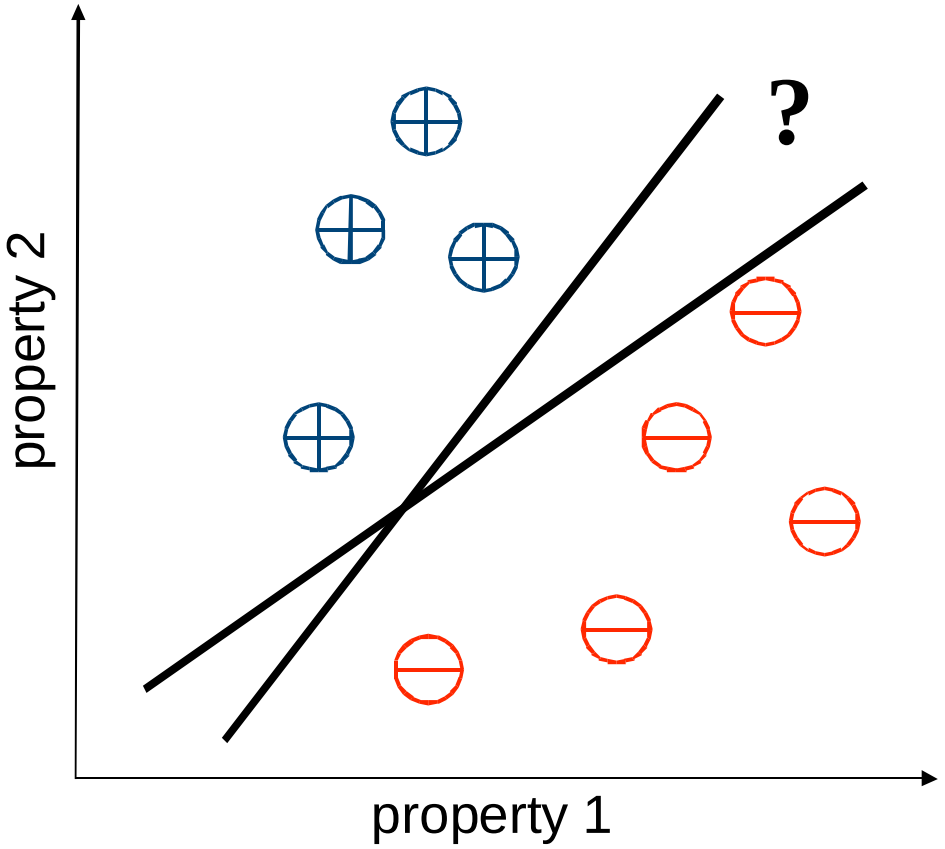




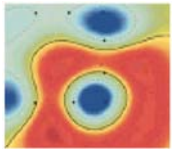


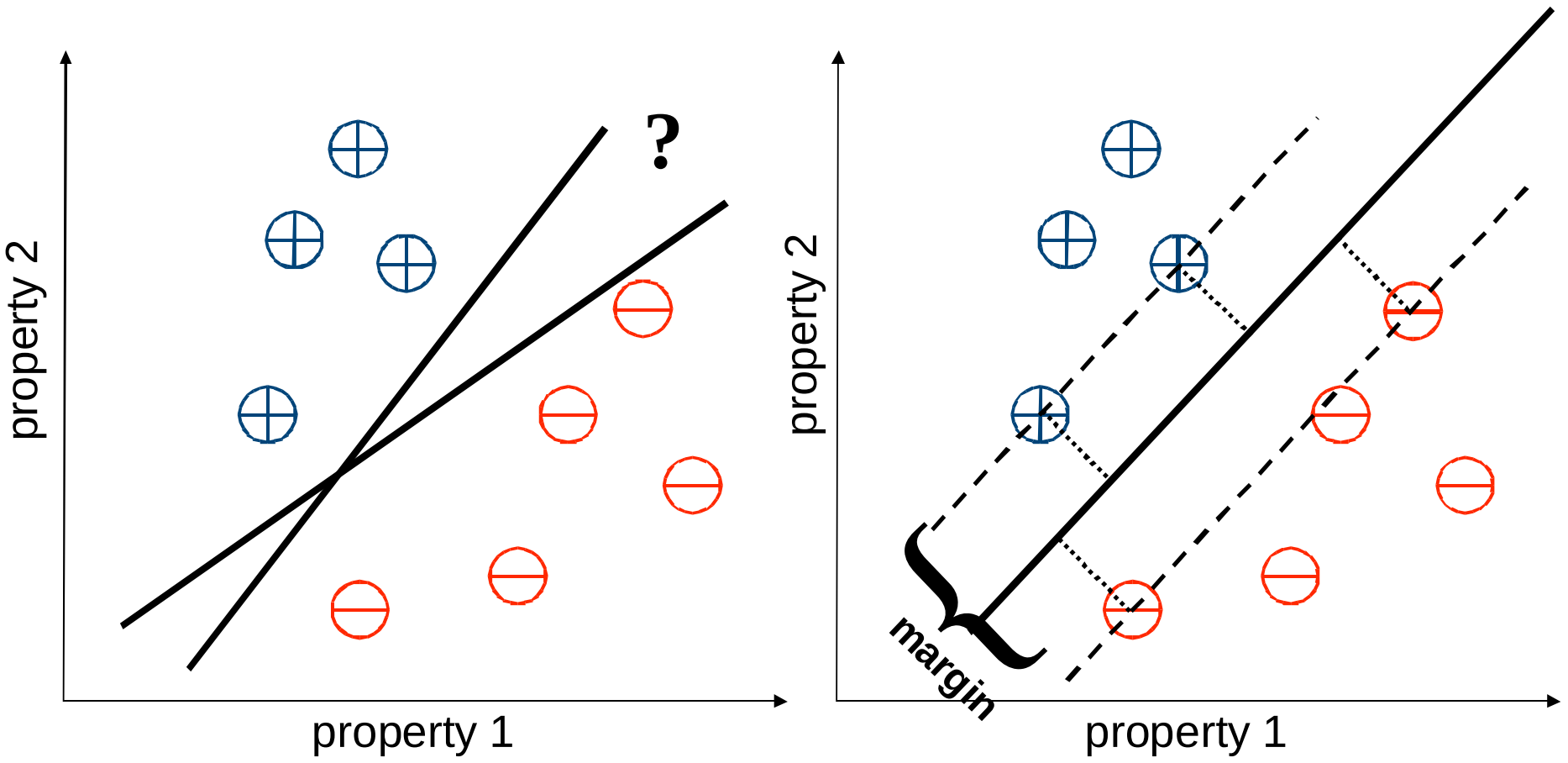
Linear Separation





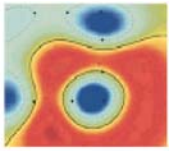
Linear Separation with Margins





large margin => good generalization

Large Margin Separation



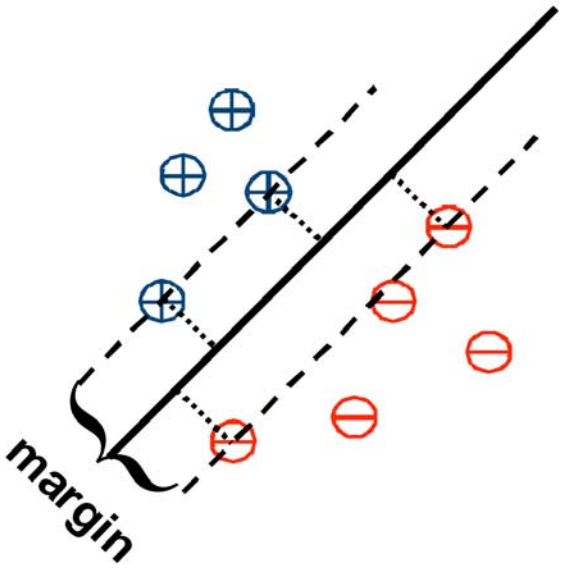
Idea:

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

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

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

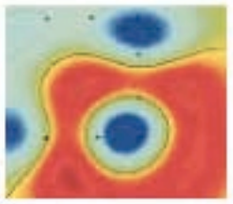
- Use $\mathrm{sgn}(f(\mathbf{x}) + b)$ for prediction

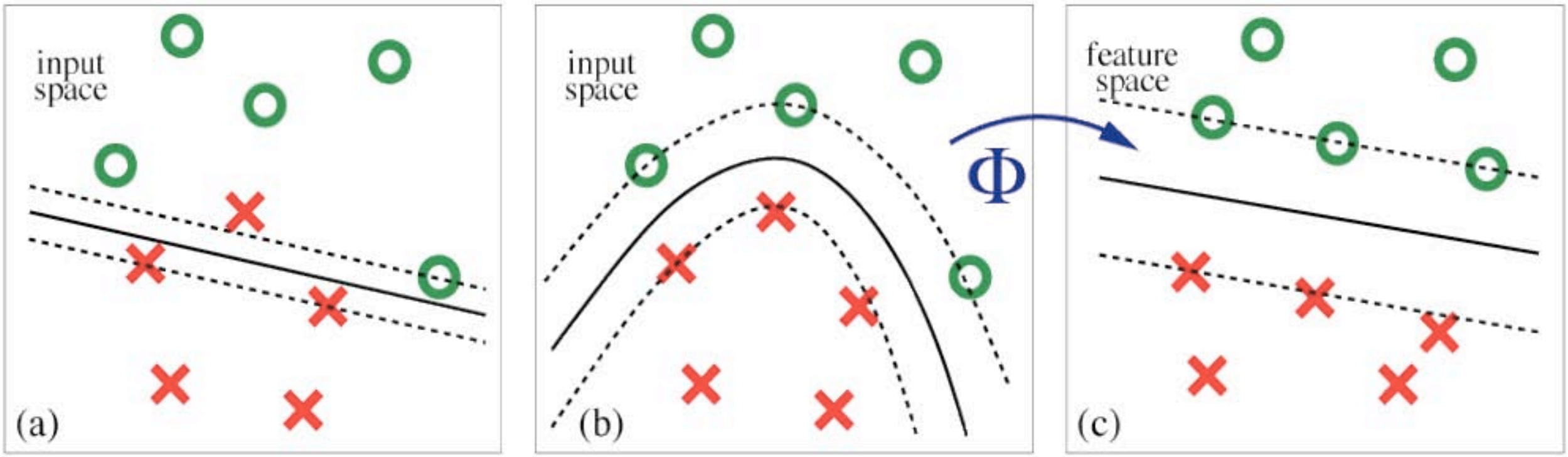


Solution:

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

Kernel Trick

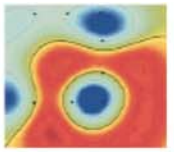




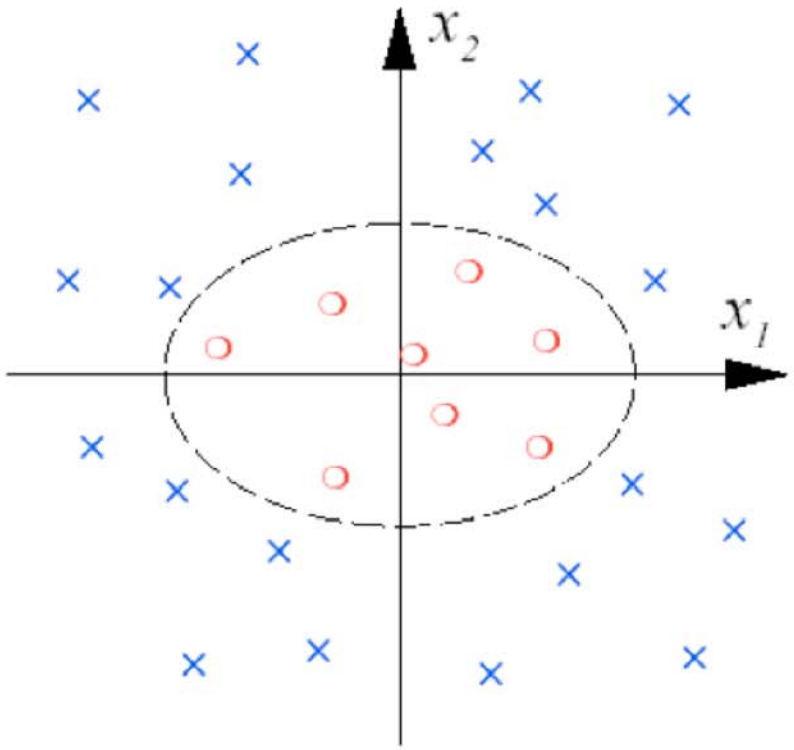
Linear in input space **Non-linear in input space** ⇐ **Linear in feature space**

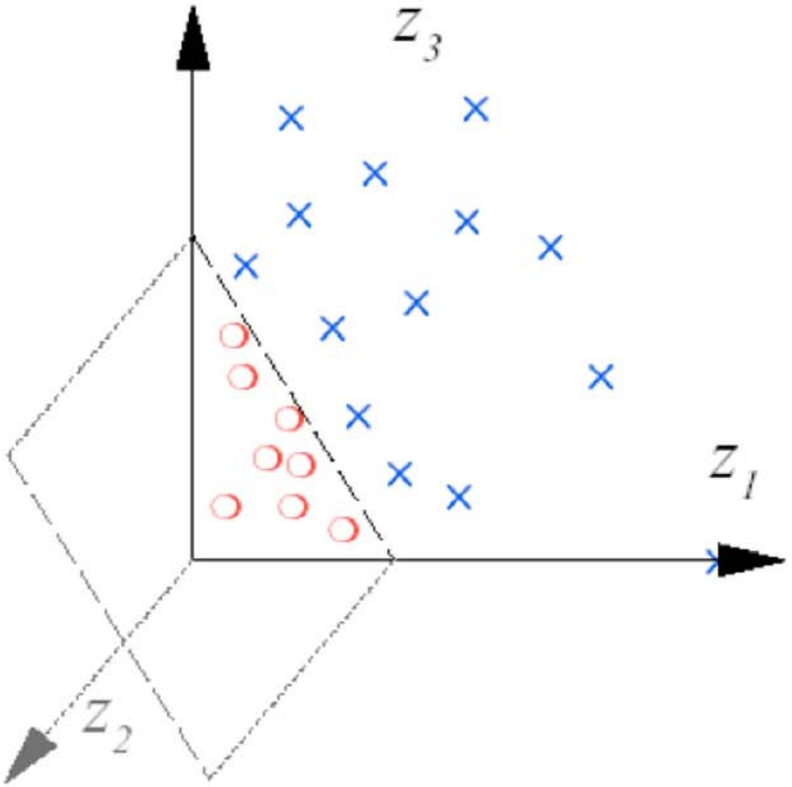
$$k(\mathbf{x}, \mathbf{x}') = (\Phi(\mathbf{x}) \cdot \Phi(\mathbf{x}'))$$

Example: Polynomial Kernel



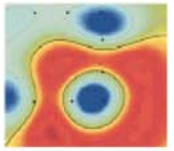
$$\Phi : \mathbb{R}^2 \to \mathbb{R}^3$$
$$(x_1, x_2) \mapsto (z_1, z_2, z_3) := (x_1^2, \sqrt{2}\, x_1 x_2, x_2^2)$$



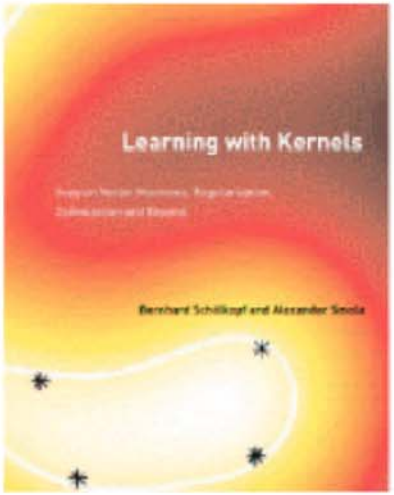


$$k(\mathbf{x}, \mathbf{x}') = (\Phi(\mathbf{x}) \cdot \Phi(\mathbf{x}')) = (\mathbf{x} \cdot \mathbf{x}')^2$$

Support Vector Machines

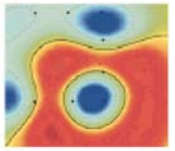


- **Demo:** Gaussian Kernel $k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}'\|}{2\sigma^2}\right)$
- Many other algorithms can use kernels
- Many other application specific kernels

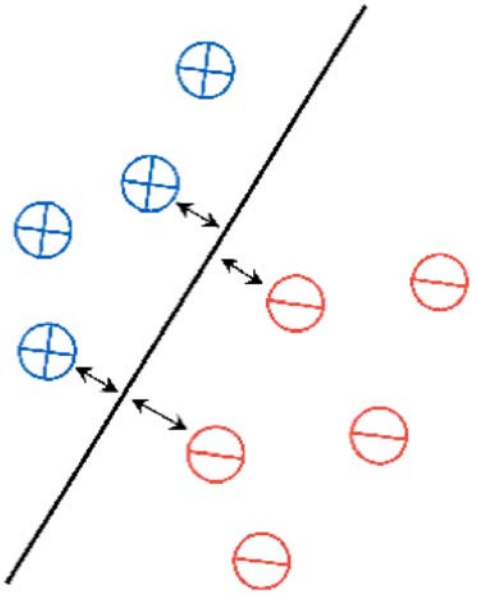


For further information, cf.
http://www.kernel-machines.org,
http://www.learning-with-kernels.org

Capabilities of Current Techniques



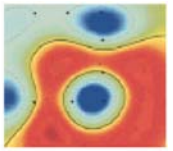
- Theoretically & algorithmically well understood:
  - **Classification with few classes**
  - Regression (real valued)
  - Novelty / Anomaly Detection



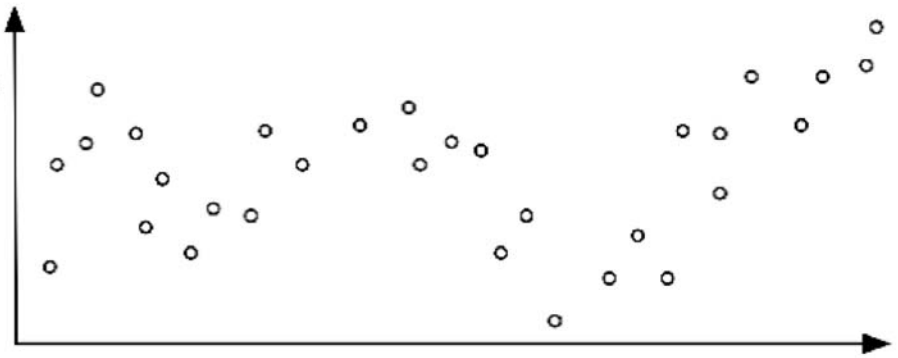
Bottom Line: *Machine Learning works well for relatively simple objects with simple properties*

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

Capabilities of Current Techniques



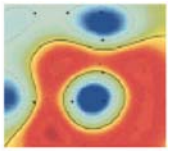
- Theoretically & algorithmically well understood:
  - Classification with few classes
  - **Regression (real valued)**
  - Novelty / Anomaly Detection



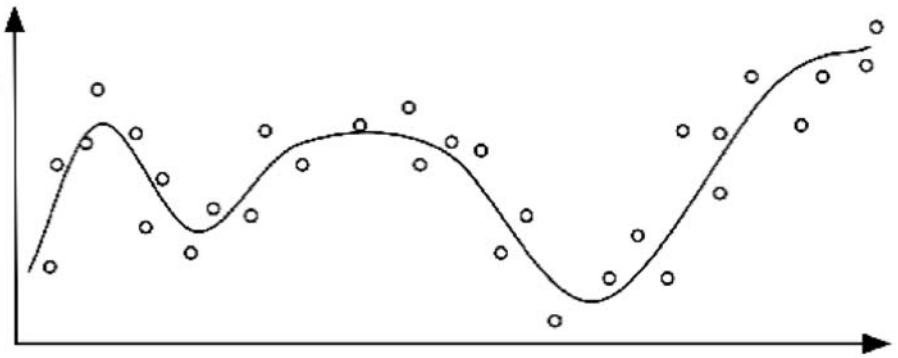
Bottom Line: *Machine Learning works well for relatively simple objects with simple properties*

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

Capabilities of Current Techniques



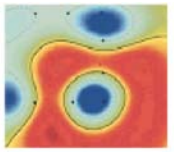
- Theoretically & algorithmically well understood:
  - Classification with few classes
  - **Regression (real valued)**
  - Novelty / Anomaly Detection



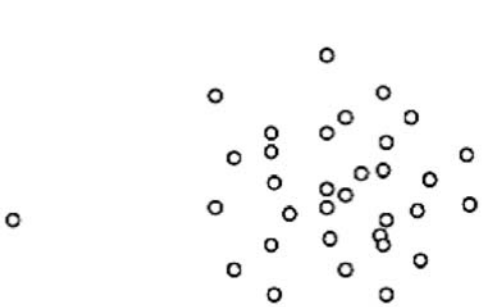
Bottom Line: *Machine Learning works well for relatively simple objects with simple properties*

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

Capabilities of Current Techniques



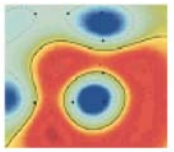
- Theoretically & algorithmically well understood:
  - Classification with few classes
  - Regression (real valued)
  - **Novelty / Anomaly Detection**



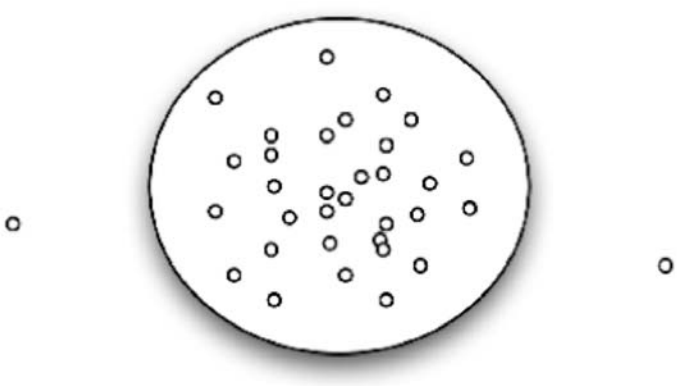
Bottom Line: *Machine Learning works well for relatively simple objects with simple properties*

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

Capabilities of Current Techniques



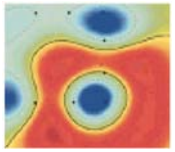
- Theoretically & algorithmically well understood:
  - Classification with few classes
  - Regression (real valued)
  - **Novelty / Anomaly Detection**



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

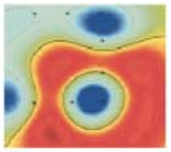
- Current Research
  - Complex objects
  - Many classes
  - Complex learning setup (active learning)
  - Prediction of complex properties

Many Applications



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

Will discuss in more Detail:

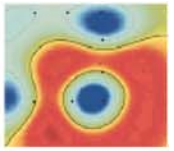


- Handwritten Letter/Digit recognition
- Drug Discovery
- Fun examples
- Gene Finding
- Brain-Computer Interfacing

Want to try this at home?

- Libsvm (C++) http://www.csie.ntu.edu.tw/~cjlin/libsvm/
- Torch (Java, C++) http://torch.ch
- Numarray (Python) http://sourceforge.net/projects/numpy

MNIST Benchmark

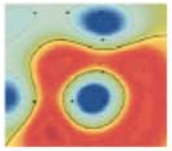


handwritten character benchmark (60000 training & 10000 test examples, 28×28)

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

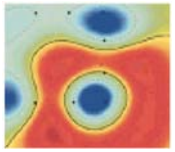
(considers d-th order correlations of pixels)

MNIST Error Rates



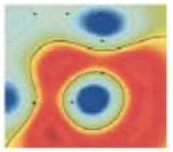
Classifier	test error	reference
linear classifier	8.4%	*Bottou et al. (1994)*
3-nearest-neighbour	2.4%	*Bottou et al. (1994)*
SVM	1.4%	*Burges and Schölkopf (1997)*
Tangent distance	1.1%	*Simard et al. (1993)*
LeNet4	1.1%	*LeCun et al. (1998)*
Boosted LeNet4	0.7%	*LeCun et al. (1998)*
Translation invariant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

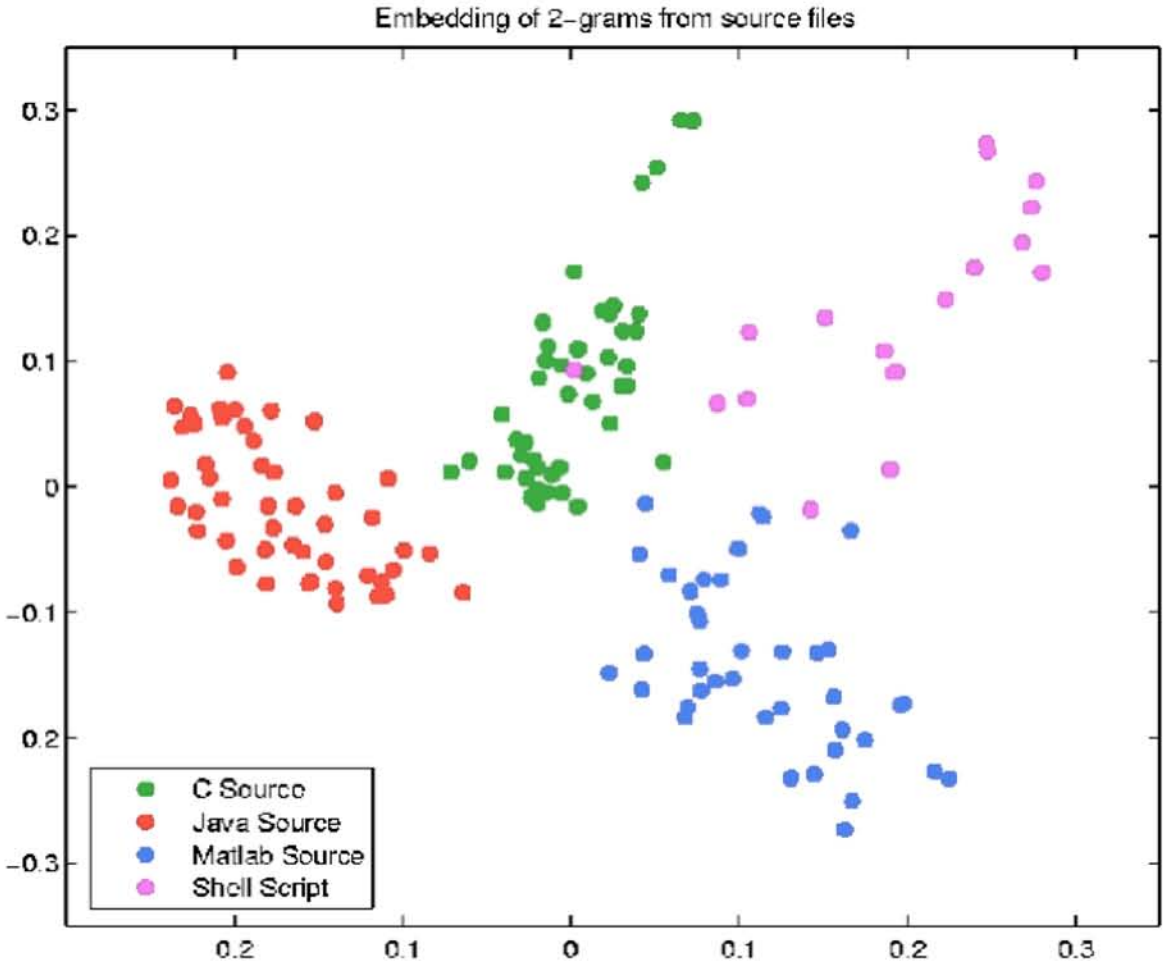
Drug Discovery / PCADMET



- To be inserted later

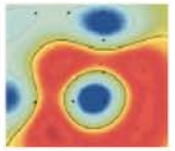
File Analysis: Sourcecode

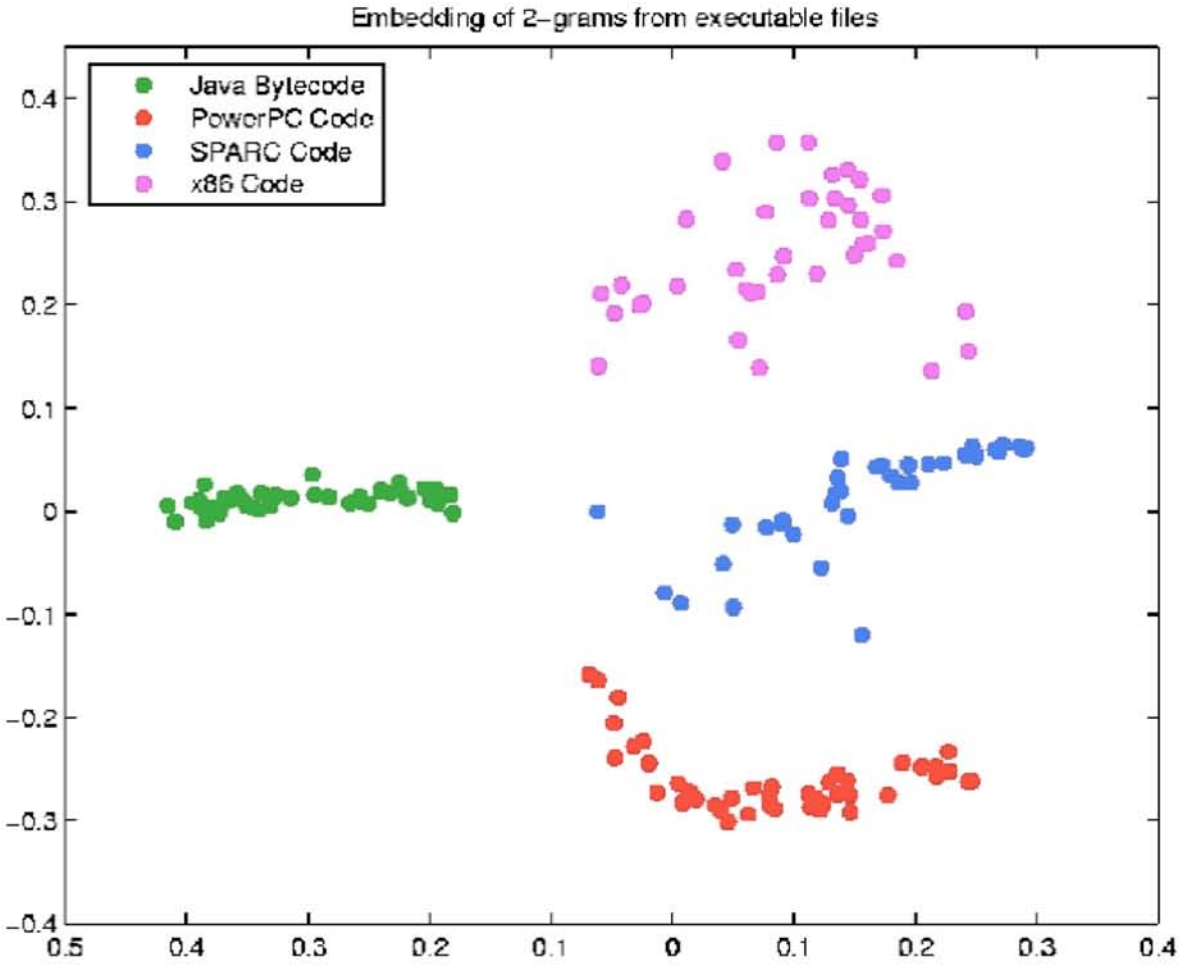




- Pseudocode for Visualisation
- Determine distances between all (pairs of) files
  - Find and count all n-Grams in each file (gives histograms)
  - Distance meaure for histograms of n-grams is the Canberra-distance
- Calculate kernel matrix
- Calculate eigenvalues and eigenvectors of kernel matrix (PCA)
- Plot the two PCA components with largest variance

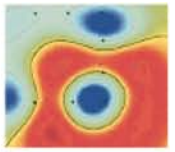
File Analysis: Binary Code

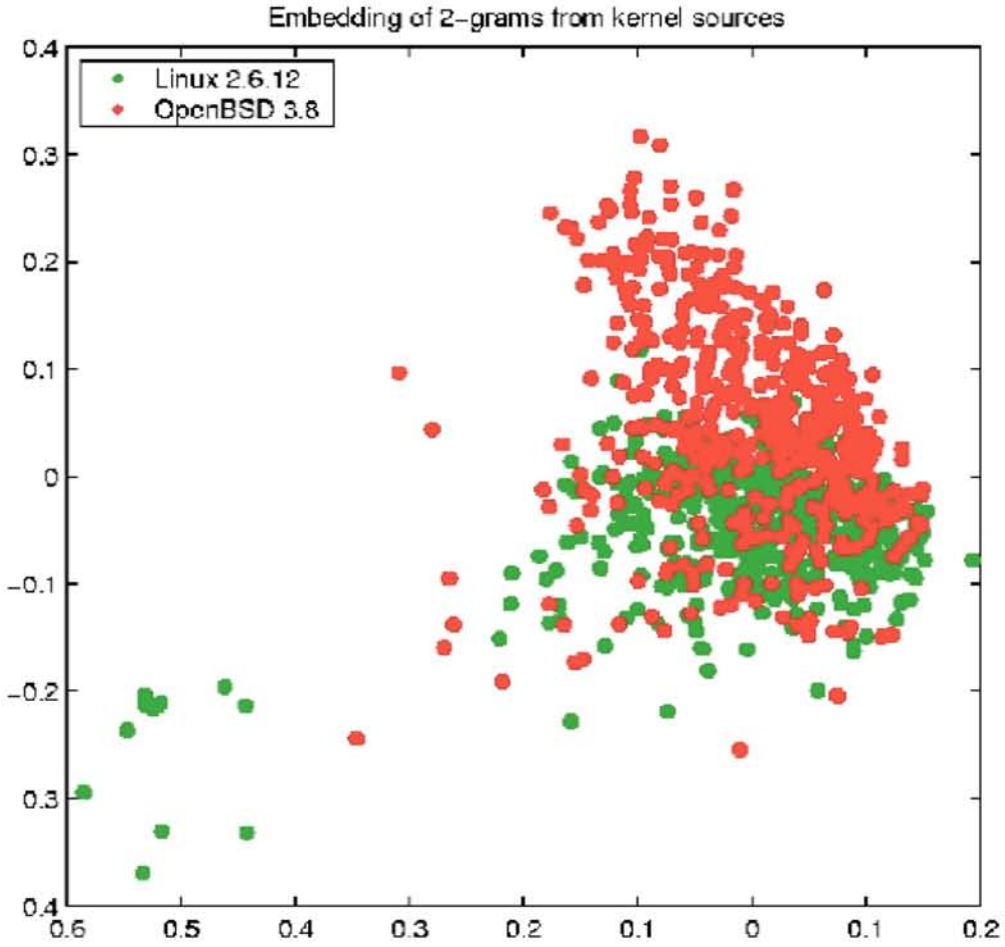




- Pseudocode for Visualisation
- Determine distances between all (pairs of) files
  - Find and count all n-Grams in each file (gives histograms)
  - Distance meaure for histograms of n-grams is the Canberra-distance
- Calculate kernel matrix
- Calculate eigenvalues and eigenvectors of kernel matrix (PCA)
- Plot the two PCA components with largest variance

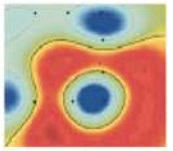
Fun Examples: Linux vs. OpenBSD

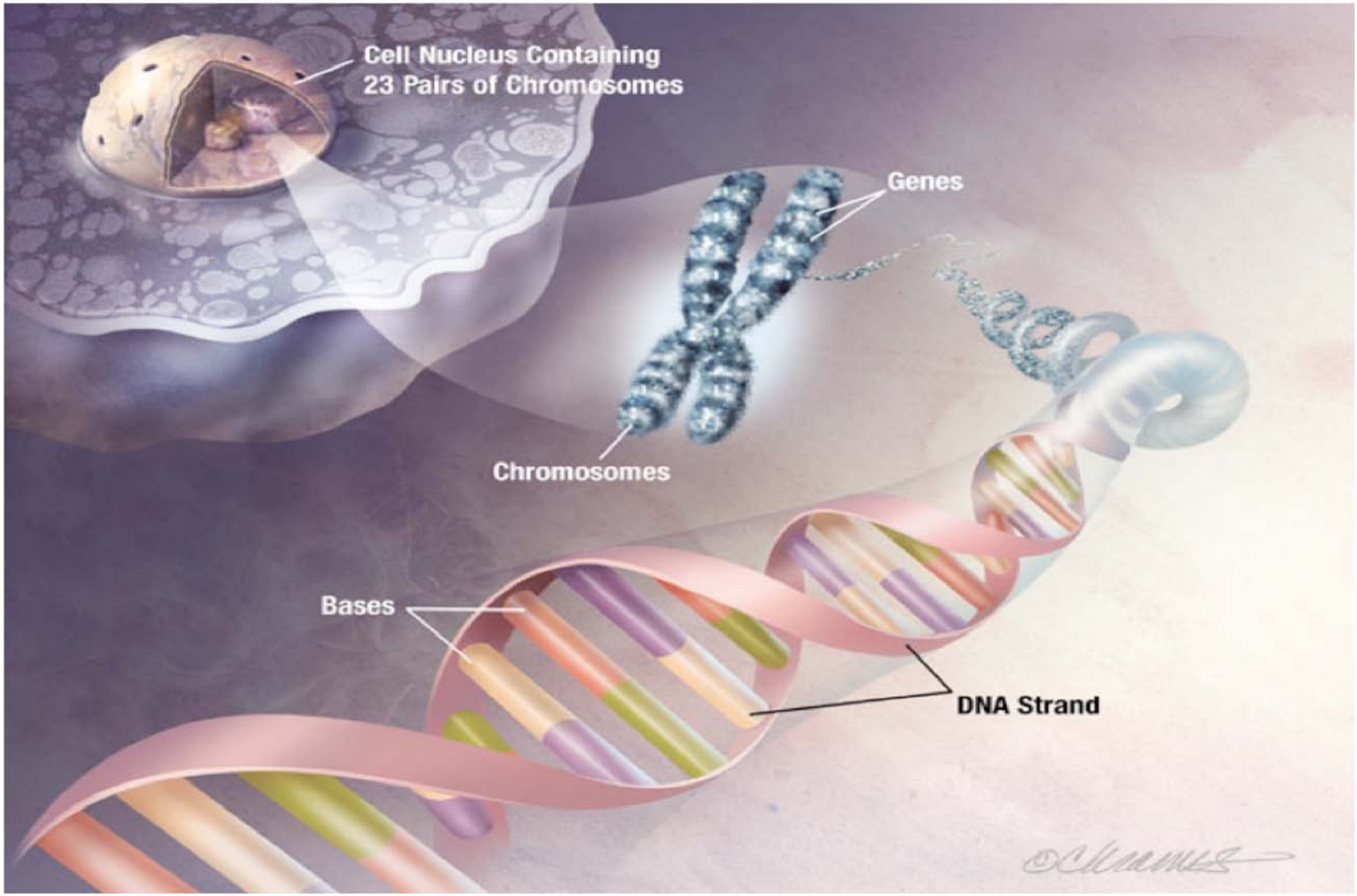




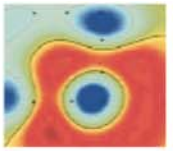
- Visuell, 2 Dimensions
  - 2 / 3 correct?
- SVM, 2 Dimensions
  - 73 % korrekt
- SVM, 50 Dimensions
  - 95 % korrekt

A Bioinformatics Application



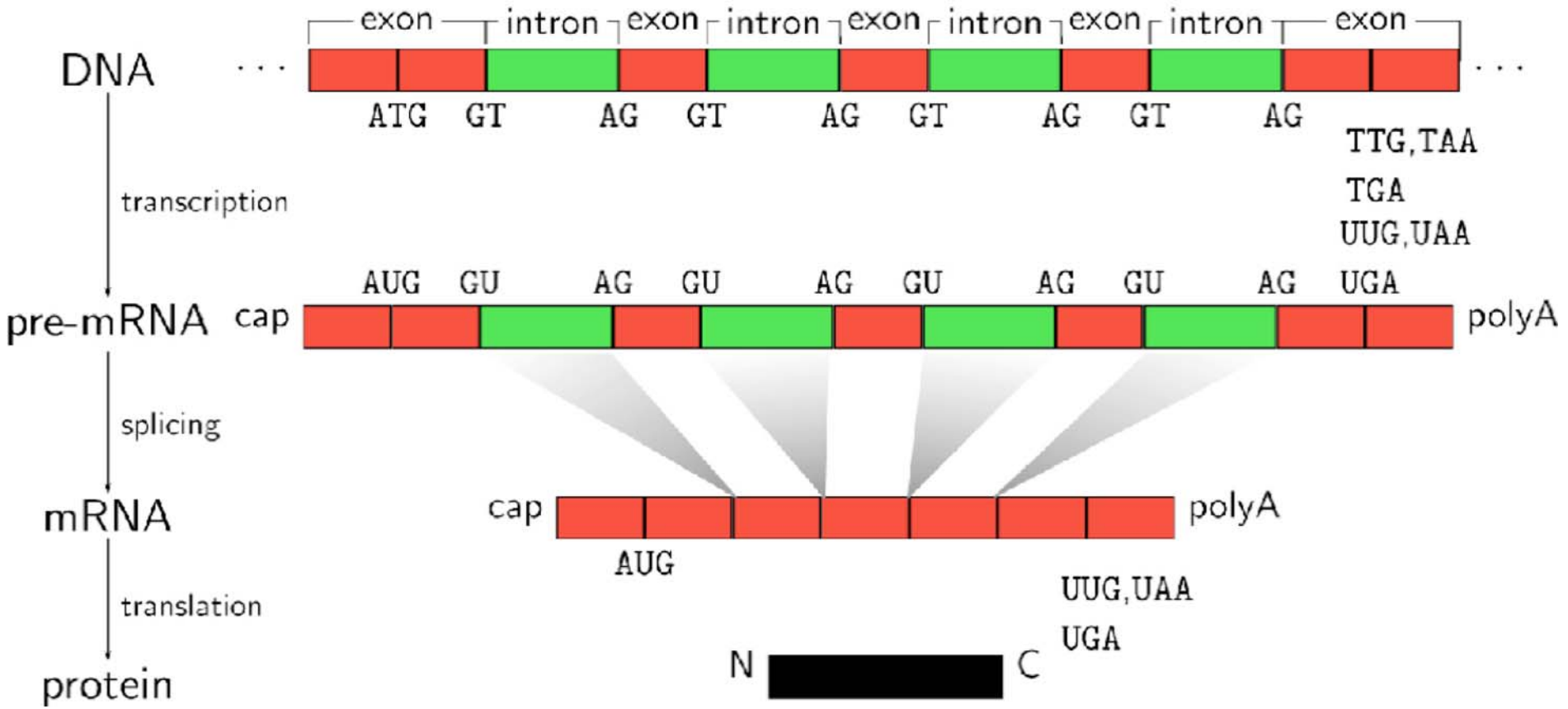


Finding Genes on Genomic DNA

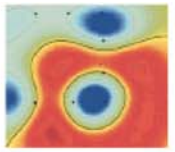


Splice Sites: on the boundary

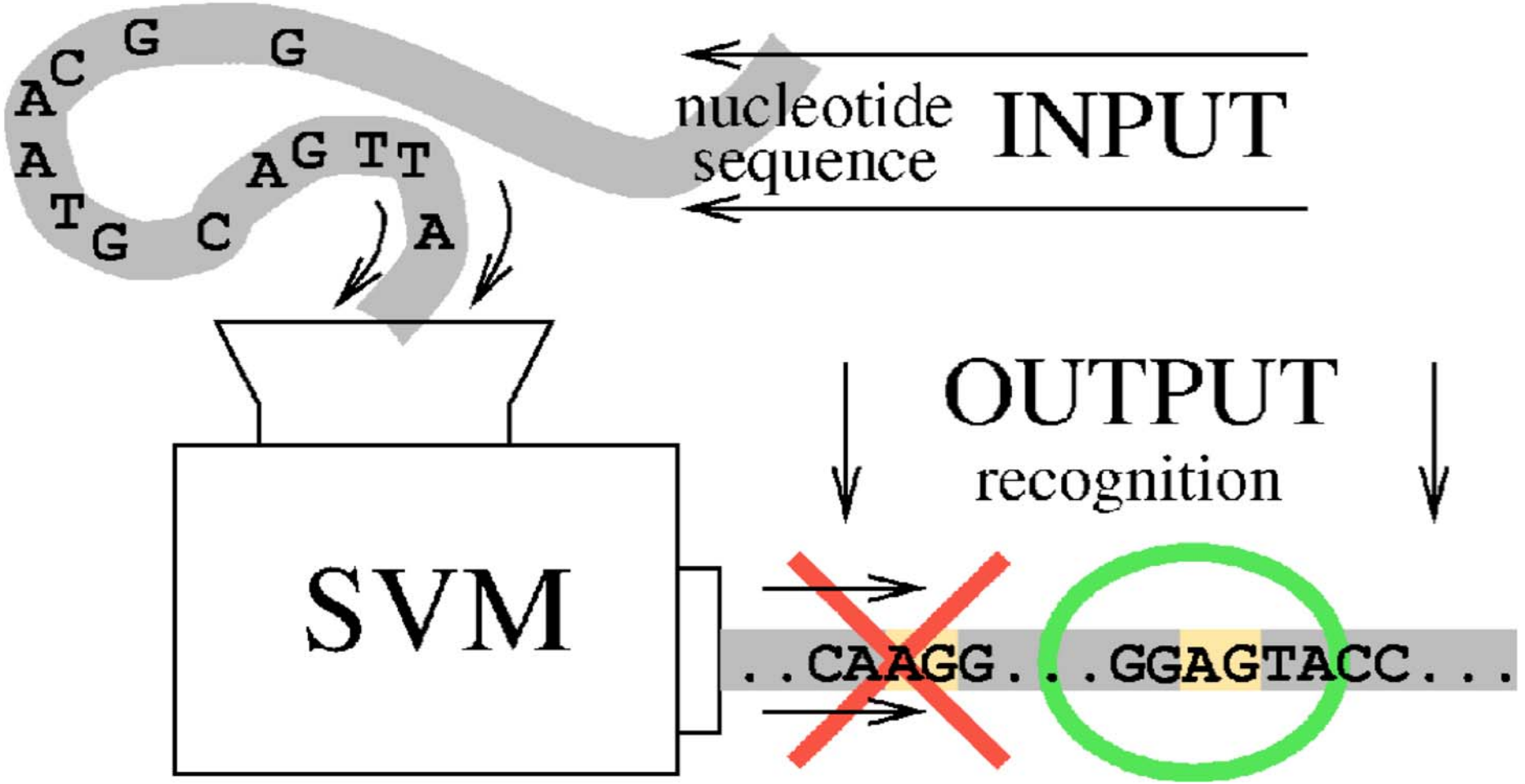
- **Exons (may code for protein)**
- **Introns (noncoding)**



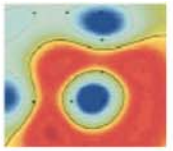
Application: Splice Site Detection



Engineering Support Vector Machine (SVM) Kernels That Recognize Splice Sites



2-class Splice Site Detection



Window of 150nt

CT...GTAGAG TGTA..GAAGCT AG GAGCGC..ACCGT ACGCGT...GA

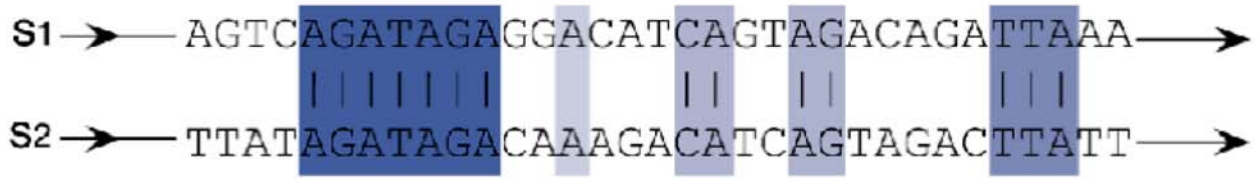
around known splice sites

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

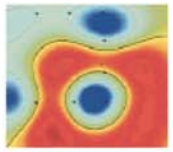
```
AAACAAATAAGTAACTAATCTTTTAGGAAGAACGTTTCAACCATTTTGAG
AAGATTAAAAAAAAACAAATTTTTAGCATTACAGATATAATAATCTAATT
CACTCCCCAAATCAACGATATTTTAGTTCACTAACACATCCGTCTGTGCC
TTAATTTCACTTCCACATACTTCCAGATCATCAATCTCCAAAACCAACAC
```

Design of new Support Vector Kernel

$$k(s_1,s_2) = w_7 + w_1 + w_2 + w_2 + w_3$$



Single Trial Analysis of EEG:towards BCI



Gabriel Curio

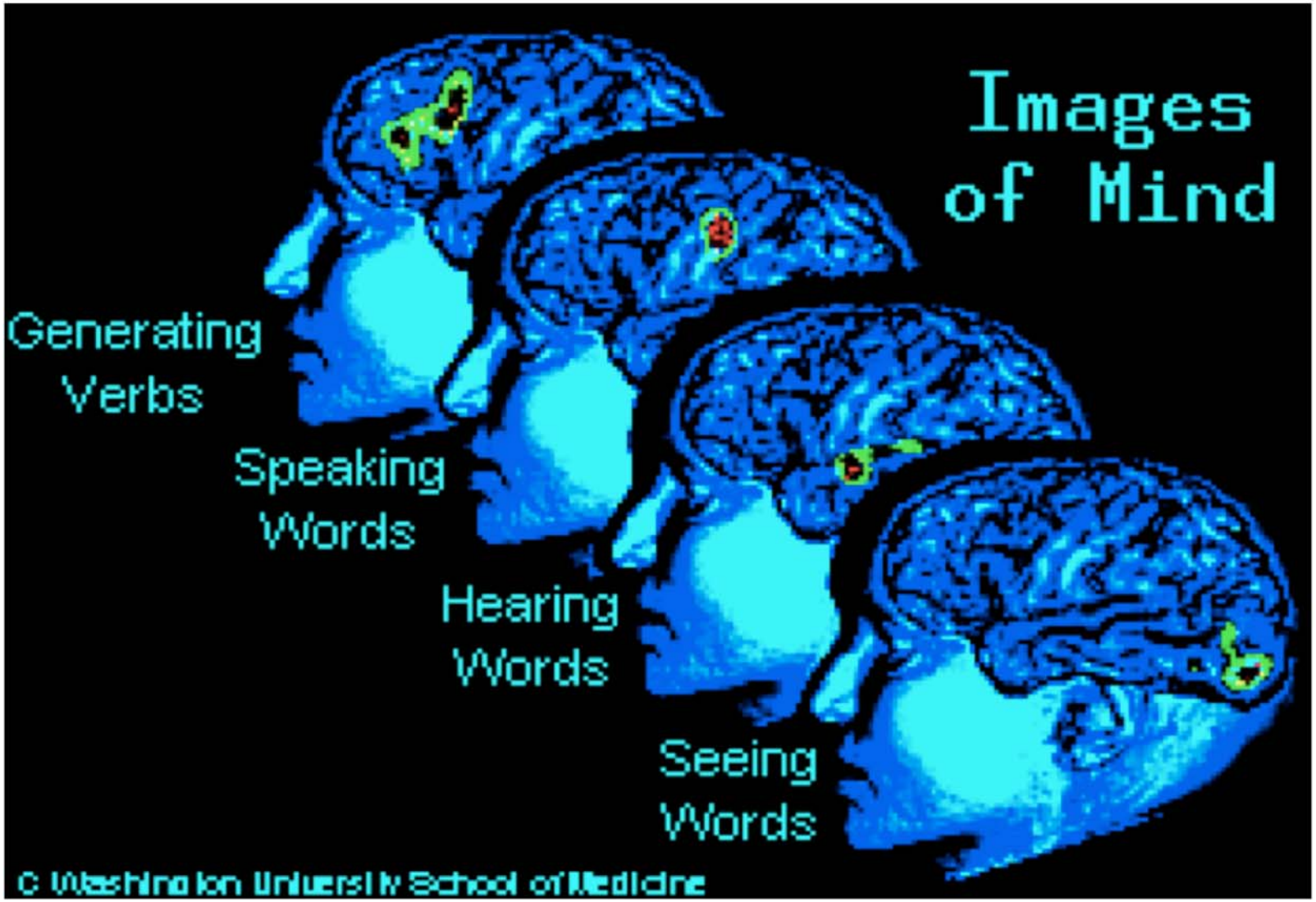
Neurophysics Group
Dept. of Neurology
Klinikum Benjamin Franklin
Freie Universität
Berlin, Germany

Benjamin Blankertz

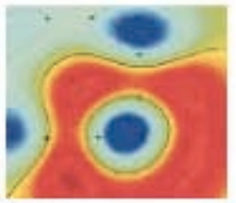
Klaus-Robert Müller

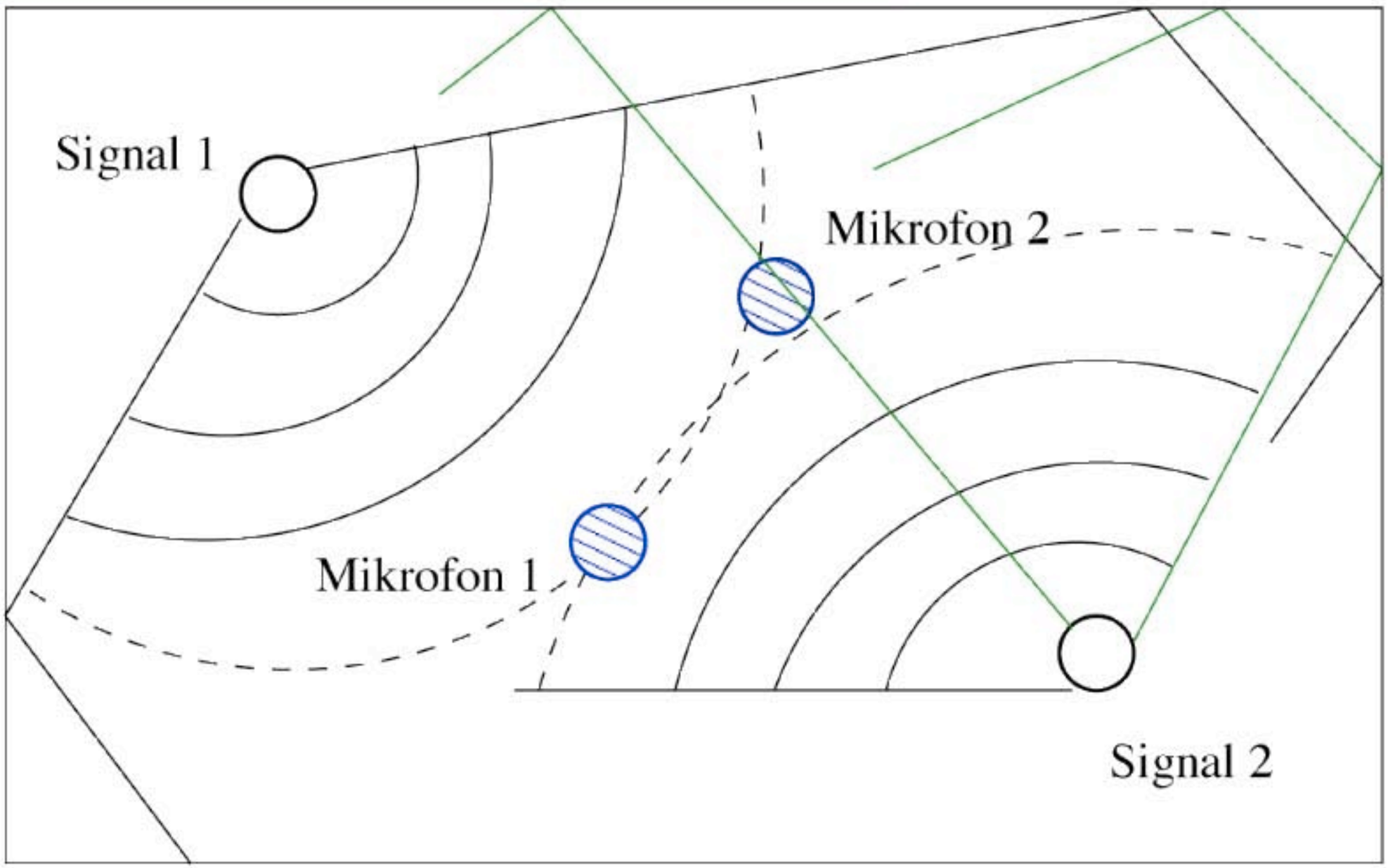
Intelligent Data Analysis Group, Fraunhofer-FIRST
Berlin, Germany

Cerebral Cocktail Party Problem



The Cocktail Party Problem

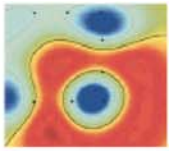




How to decompose **superimposed** signals?

Analogous Signal Processing problem as for cocktail party problem

The Cocktail Party Problem



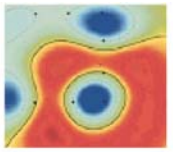
- input: 3 mixed signals
- algorithm: enforce *independence* ("independent component analysis") via temporal de-correlation

- output: 3 separated signals

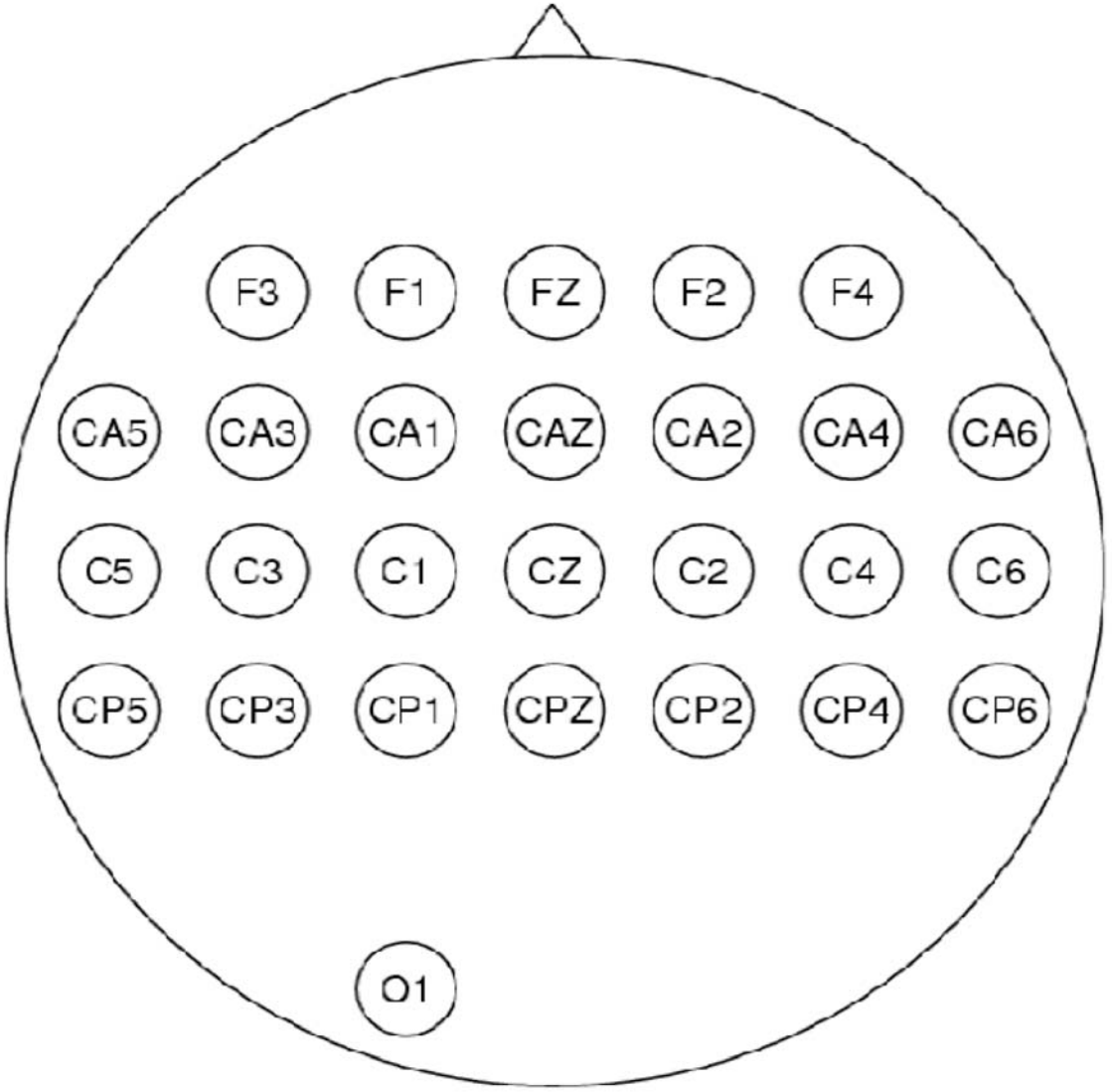
"Imagine that you are on the edge of a lake and a friend challenges you to play a game. The game is this: Your friend digs two narrow channels up from the side of the lake […]. Halfway up each one, your friend stretches a handkerchief and fastens it to the sides of the channel. As waves reach the side 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)

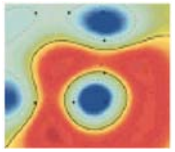
Minimal Electrode Configuration



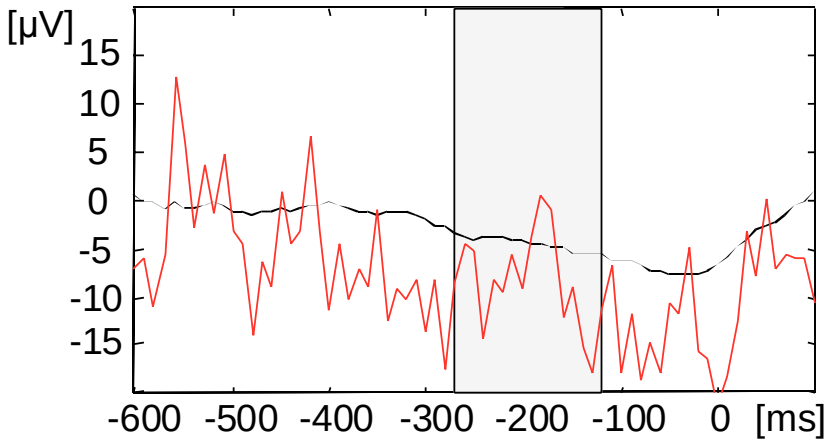
- coverage: bilateral primary sensorimotor cortices
- 27 scalp electrodes
- reference: nose
- bandpass: 0.05 Hz - 200 Hz
- ADC 1 kHz
- downsampling to 100 Hz
- EMG (forearms bilaterally): m. flexor digitorum
- EOG
- event channel: keystroke timing (ms precision)

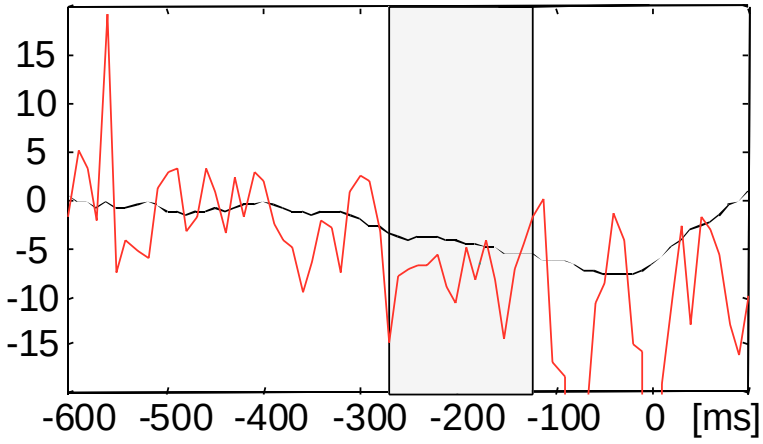


Single Trial vs. Averaging

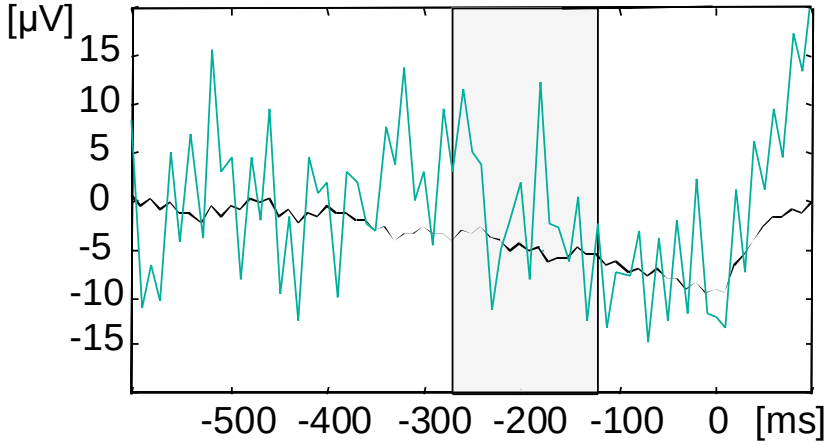


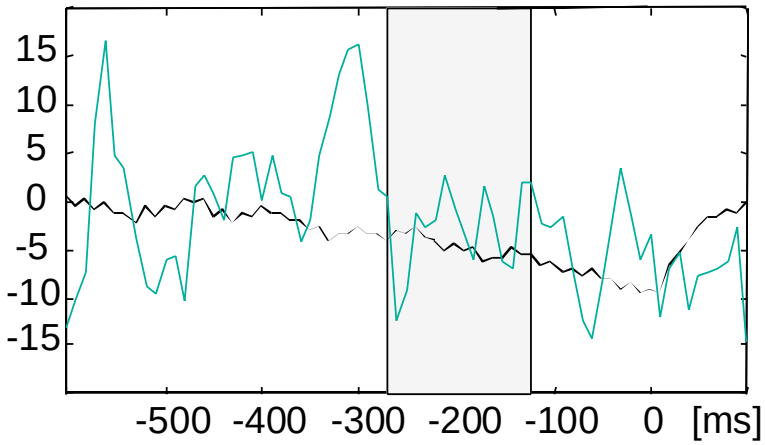
LEFT hand (ch. C4)



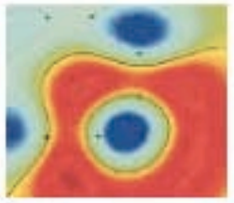


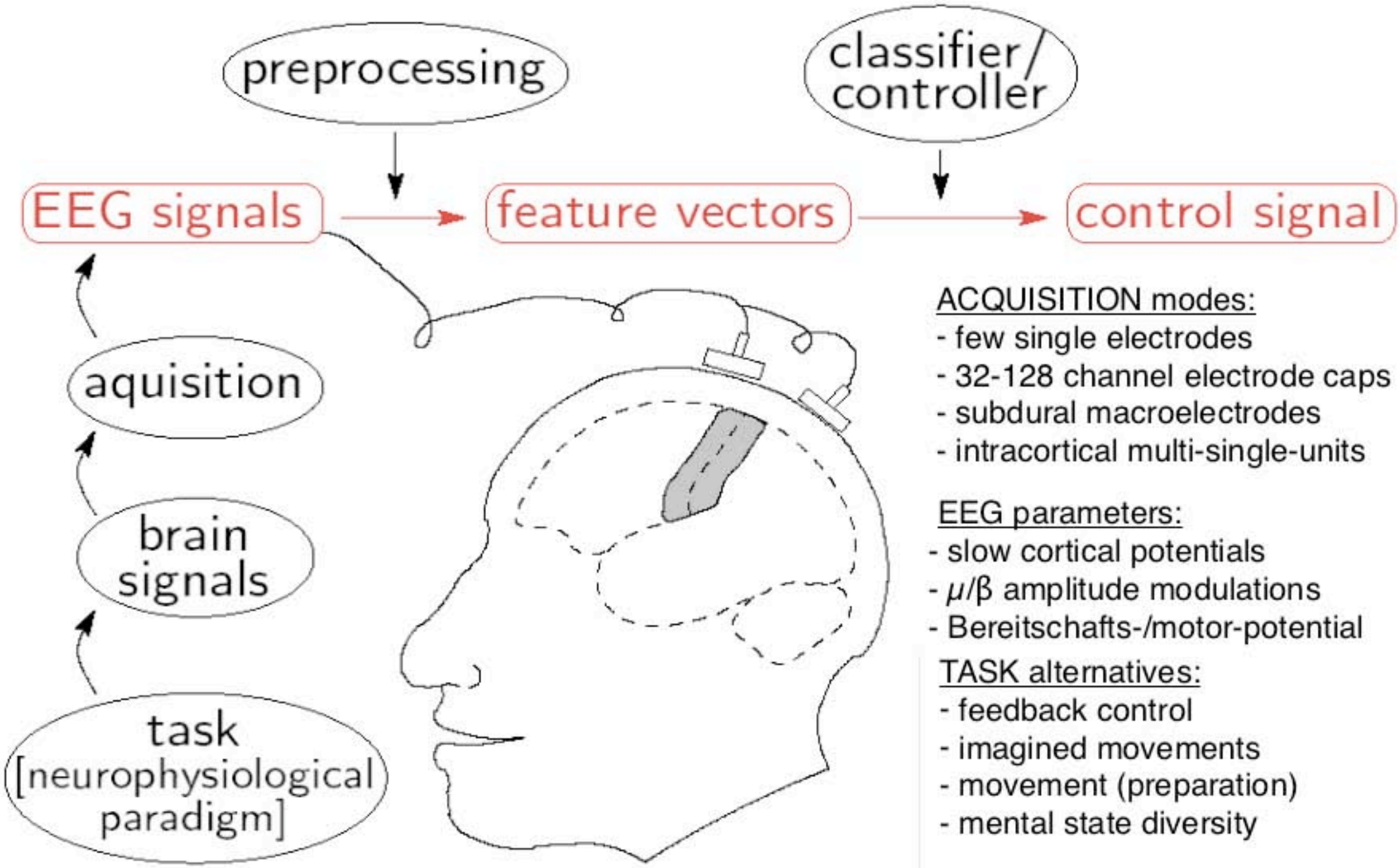
RIGHT hand (ch. **C3**)





BCI Setup





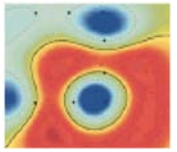
ACQUISITION modes:
- few single electrodes
- 32-128 channel electrode caps
- subdural macroelectrodes
- intracortical multi-single-units

EEG parameters:
- slow cortical potentials
- μ/β amplitude modulations
- Bereitschafts-/motor-potential

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

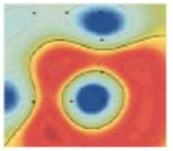
BCI Demo: BrainPong

BCI Demo: BrainPong



- Video 1 Player
- Video 2 Player

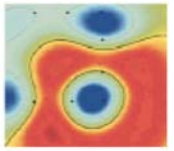
Concluding Remarks



- 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 is computationally expensive)
  - Need computer clusters and Job scheduling systems (pbs, gridengine)
  - Often use MATLAB (to be replaced by python ?!)

- 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.**
- Always looking for motivated students, Ph.D. Students, post-docs

Thanks for Your Attention!



Speakers at 22c3: Timon Schroeter, Konrad Rieck, Sören Sonnenburg
[timon, rieck, sonne]@first.fhg.de, http://ida.first.fhg.de

Contributors / Coworkers: Klaus-Robert Müller, Jens Kohlmorgen, Benjamin Blankertz, Alex Zien, Motoaki Kawanabe, Pavel Laskov, Gilles Blanchard, Bernhard Schoelkopf, Anton Schwaighofer, Guido Nolte, Florin Popescu, Stefan Harmeling, Julian Laub, Andreas Ziehe, Steven Lemm, Christin Schäfer, Guido Dornhege, Frank Meinecke, Matthias Krauledat, Patrick Düssel,

Special Thanks: Gunnar Rätsch (speaker at 21c3, slides)

FIRST
Fraunhofer Institut Rechnerarchitektur und Softwaretechnik

MAX-PLANCK-GESELLSCHAFT