Classification by Support Vector Machines

Florian Markowetz
Max-Planck-Institute for Molecular Genetics
– Computational Molecular Biology –
Berlin

Practical DNA Microarray Analysis 2003

Overview

I Large Margin Classifiers

II The Kernel Trick

III Todays practical session

Supervised learning

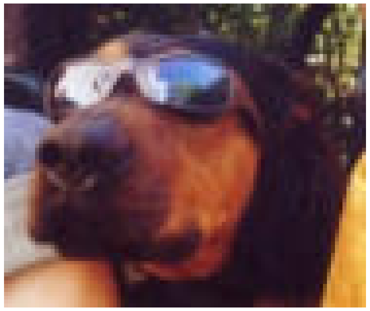
Calvin, I'm still confused about **cats** and **dogs**!

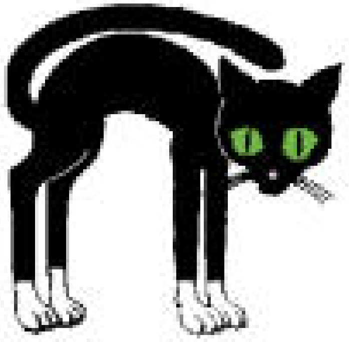
OK, then I will explain it once more ...

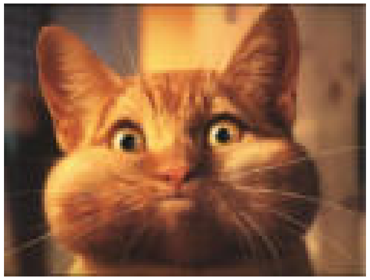
Supervised learning

Calvin, I'm still confused about **cats** and **dogs**!

OK, then I will explain it once more ...







Learning from examples

Gene expression is a complex process we can not describe explicitely.
$\Longrightarrow$ try to learn patterns from examples.

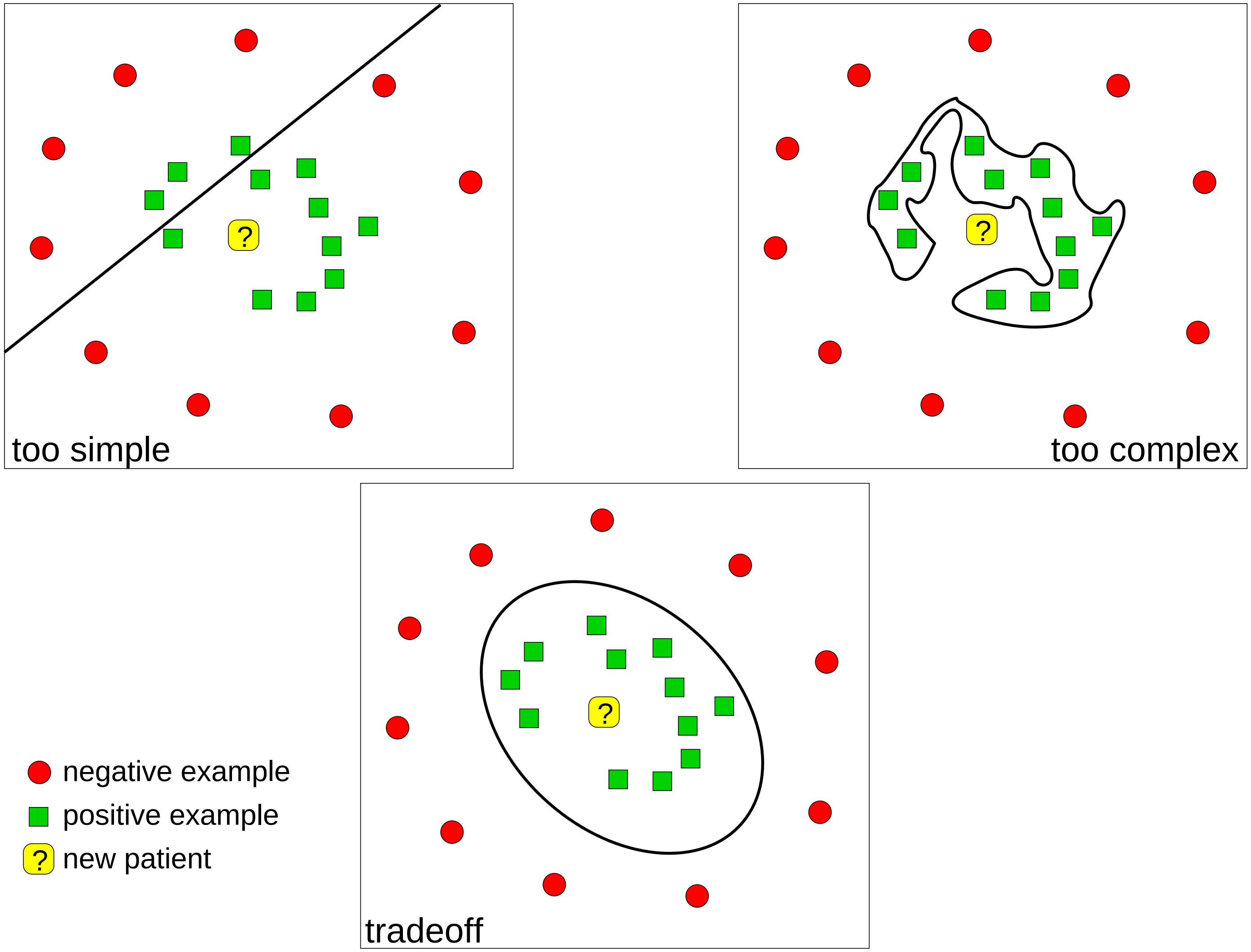
Given:	$\mathcal{X} = \{x_i, y_i\}_{i=1}^{n}$	training set	patients you've already seen
consisting of			
	$x_i \in \mathbb{R}^g$	points	expression profiles
	$y_i \in \{+1, -1\}$	labels	2 types of cancer

Goal: Learn a decision function that describes the data well.

$$f_{\mathcal{X}} : \quad \mathbb{R}^g \mapsto \{+1, -1\}$$

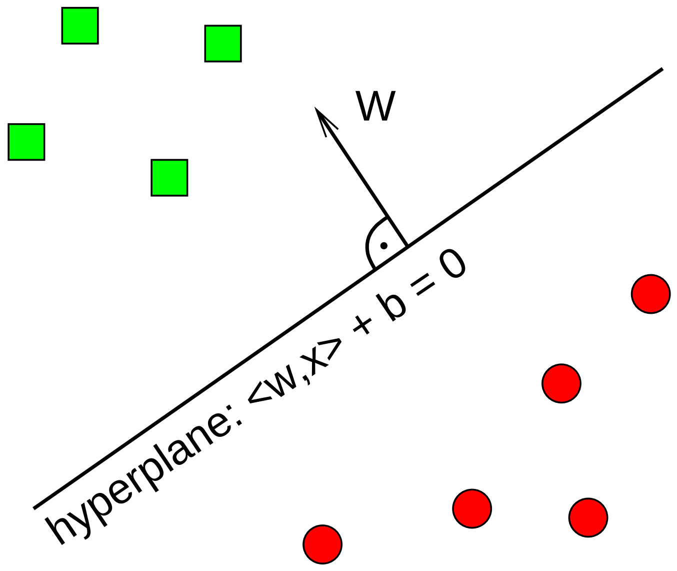
$$\text{Diagnosis} = f_{\mathcal{X}}(new\ patient)$$

Problems of learning



Linear separation

Most easy case: data set is linearly separable.



We need only a very simple classifier:

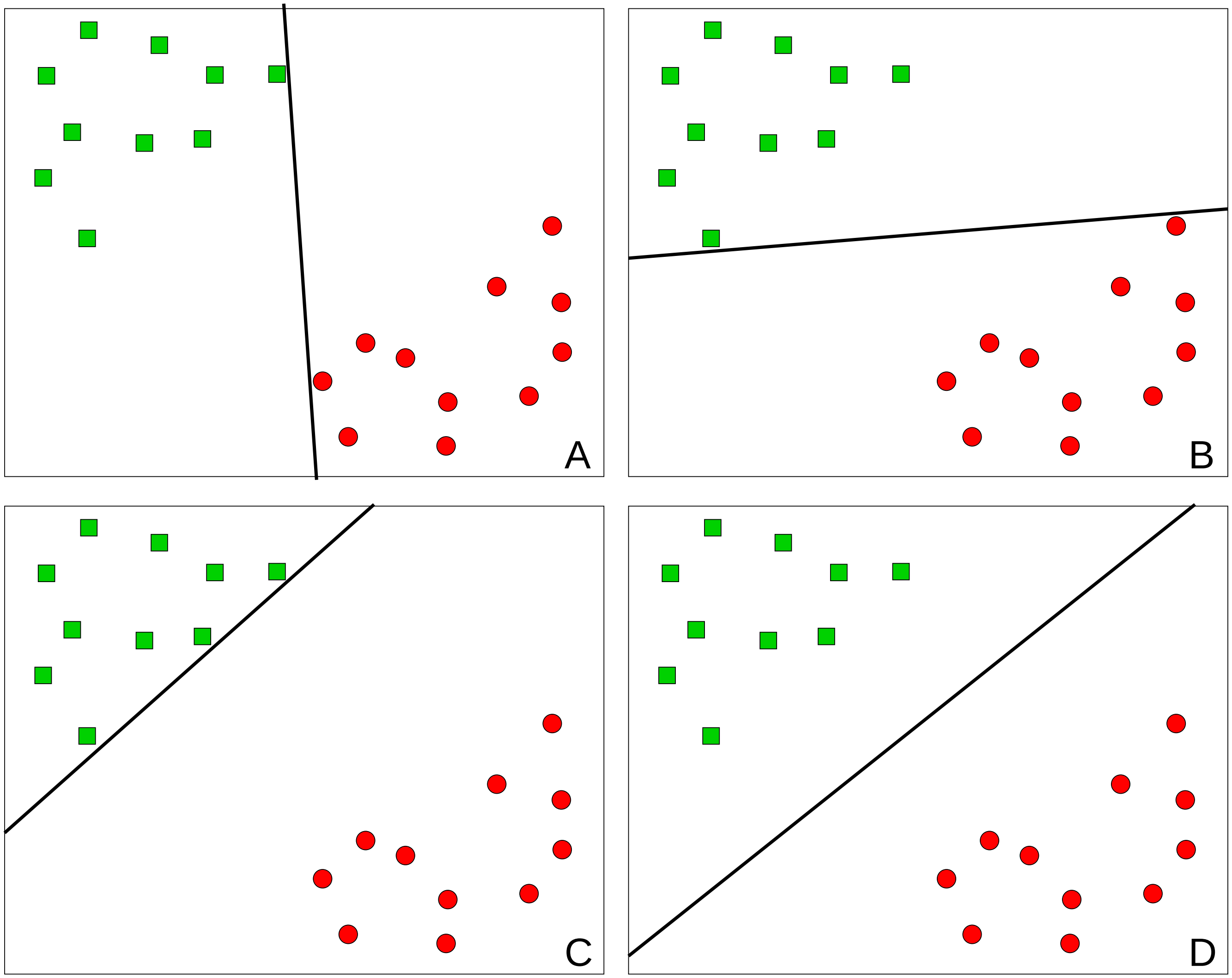
$$\mathcal{S} = \{\ x \mid \langle w, x \rangle + b = 0\}$$

Choose w and b from the trainingset $\mathcal{X}$.

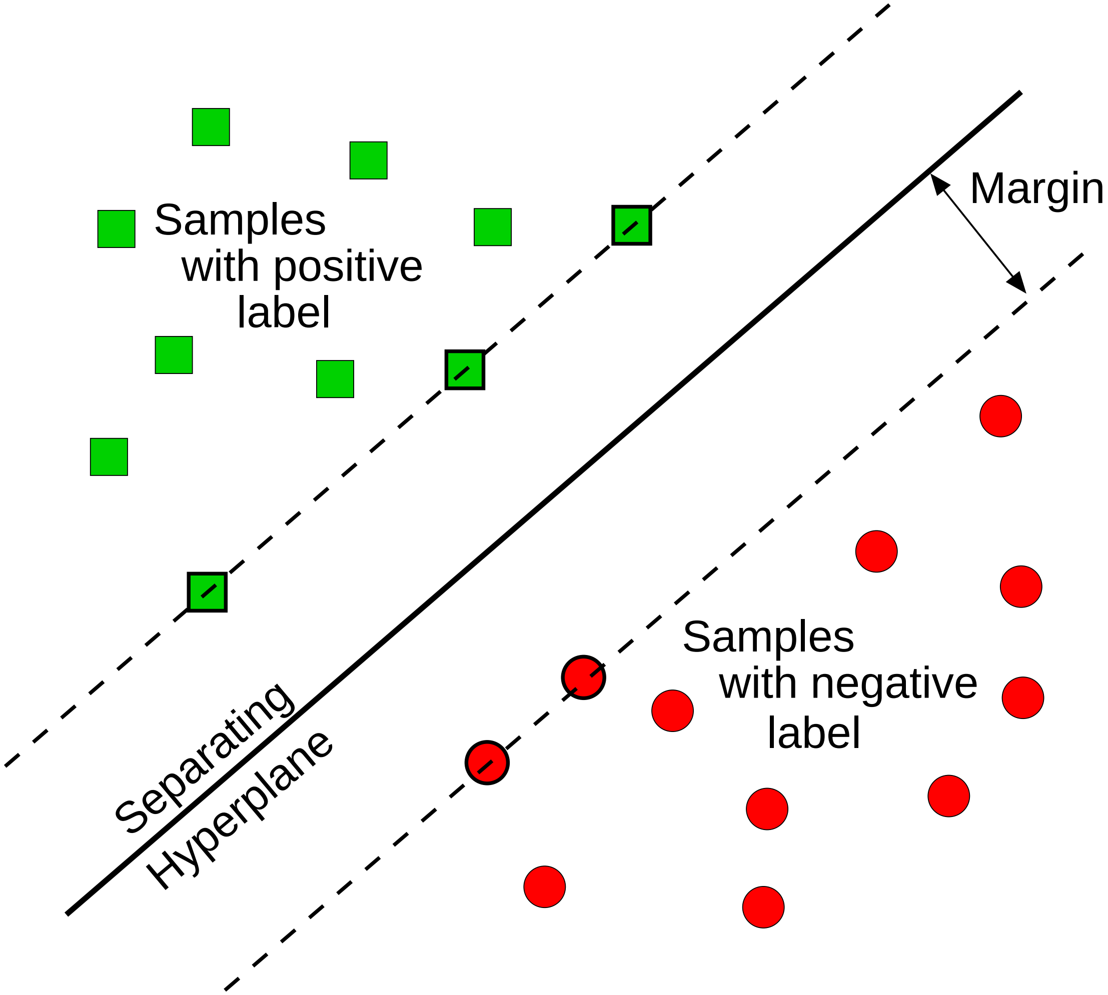
Prediction: On which side of the hyperplane does the new point lie?

Decision function: $f_{\mathcal{X}}(x_{\text{new}}) = sign\left(\langle w, x_{\text{new}} \rangle + b\right)$

Which hyperplane is the best?

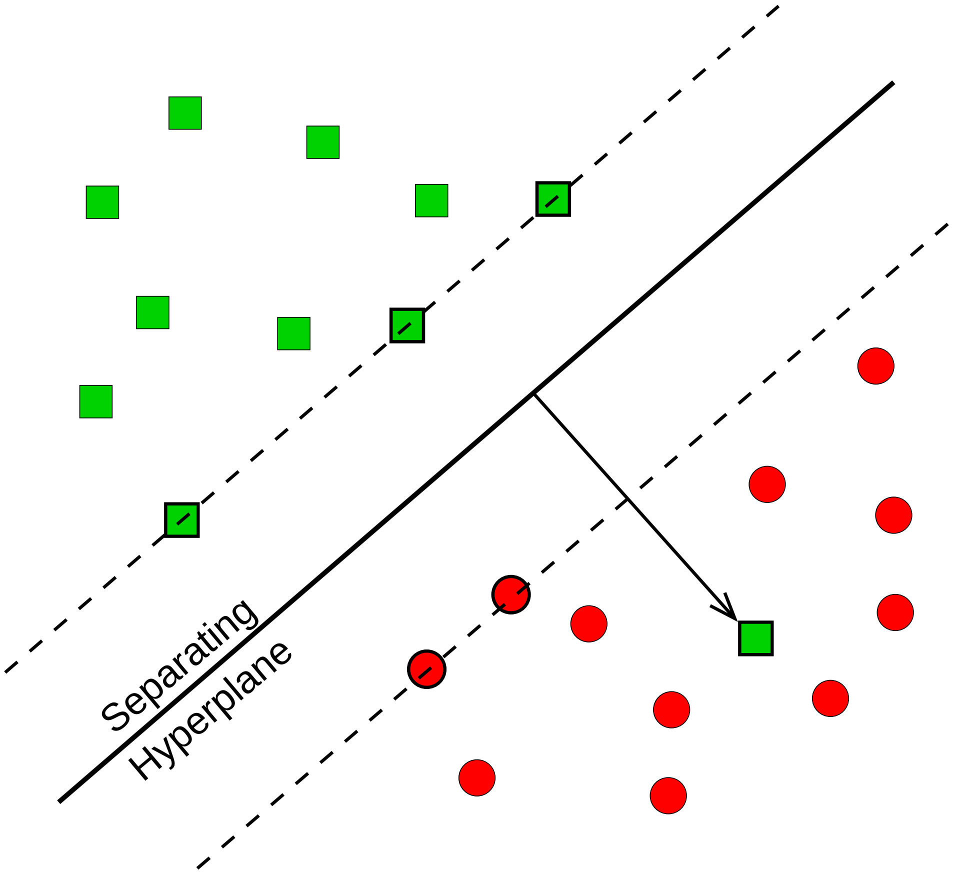


Separate the training set with maximal margin



Non-separable training sets

Use linear separation, but admit training errors.



Penalty of error: distance to hyperplane multiplied by *error cost* C.

Construction of the maximal margin hyperplane

Maximizing the margin is a problem of **constrained optimisation**, which can be solved by **Lagrange Method**.

Each training point x_i is described by a Lagrange multiplier α_i:

$\alpha_i = 0 \quad \Rightarrow \quad$ x_i has no influence on the hyperplane

$\alpha_i > 0 \quad \Rightarrow \quad$ x_i determines the sep. hyperplane
These points are called **Support Vectors**.
They lie nearest to the hyperplane.

Solution

Solution: $$w = \sum_{i=1}^{\#SV} \alpha_i y_i x_i^{sv}$$

Diagnosis: $$f(x_{\text{new}}) = sign\left(\sum_{i=1}^{\#SV} \alpha_i y_i \langle x_i^{sv}, x_{\text{new}} \rangle + b\right)$$

The decision function only depends on the Support Vectors.

They are the critical elements of the training set.

All other points could be removed without changing the solution.

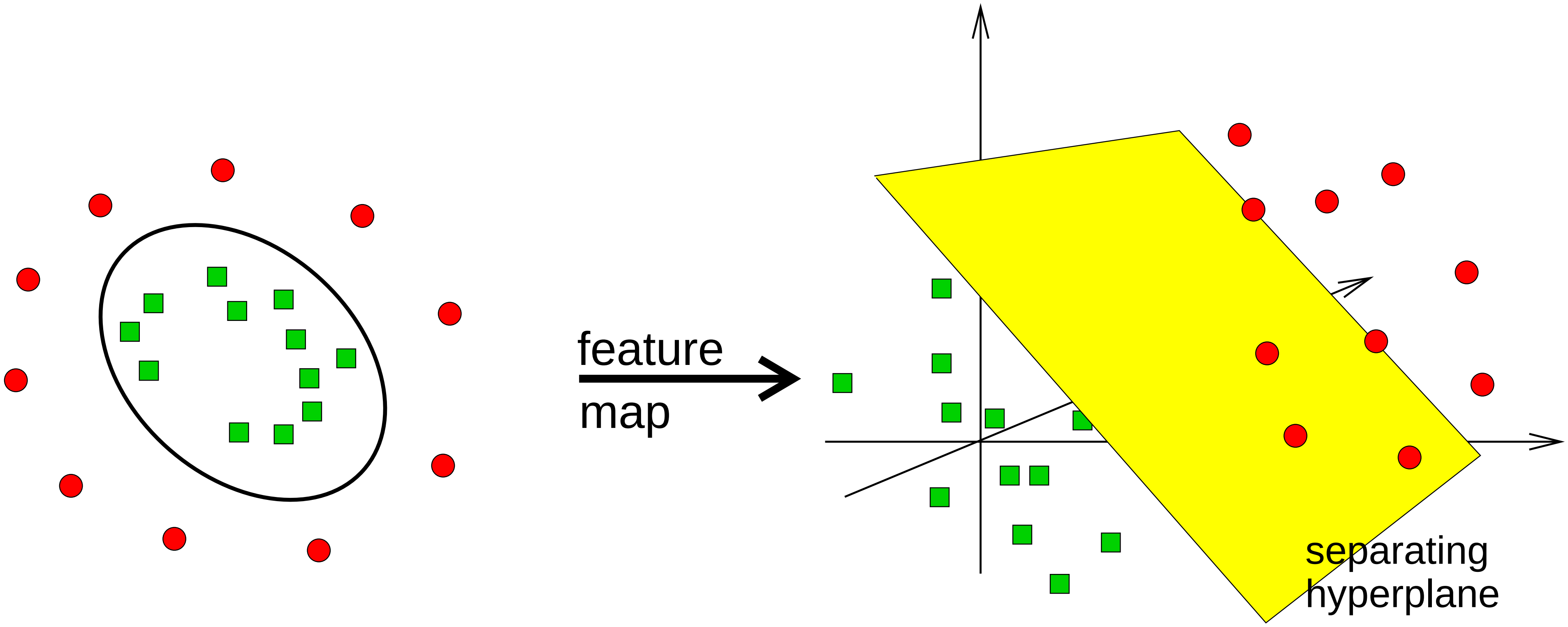
What's next?

I Large Margin Classifiers

II **The Kernel Trick**

III Todays practical session

Separation may be easier in higher dimensions



complex in low dimensions

simple in higher dimensions

The kernel trick

Classification is easier in high dimensions.

In the construction of the maximal margin hyperplane, we have to evaluate high dimensional inner products of the form

$$\langle \Phi(x_1), \Phi(x_2) \rangle_{\mathcal{H}}$$

where $\Phi : \mathcal{L} \rightarrow \mathcal{H}$ is the feature map from a low to a high dimensional space.

Problem: Computationally expensive!

Idea: do the feature map **implicitely**!

Kernel Mapping

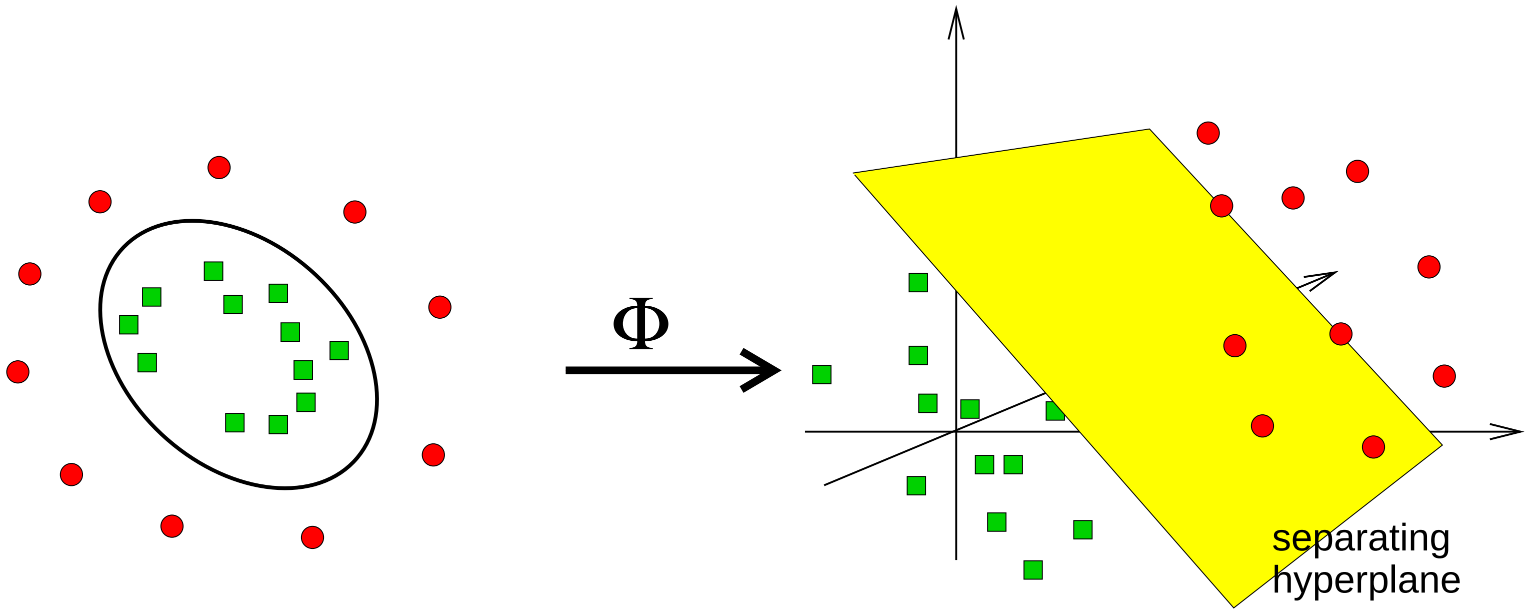
Mercer Theorem:
Under some conditions on $\mathcal{K}$ there exists an inner product $\langle\, \cdot\, ,\, \cdot\, \rangle_{\mathcal{H}}$ and a mapping $\Phi : \mathcal{L} \longrightarrow \mathcal{H}$ such that

$$\langle \Phi(x_1), \Phi(x_2) \rangle_{\mathcal{H}} \,=\, \mathcal{K}(x_1, x_2)$$

Using this kernel the decision function becomes

$$f(x_{\text{new}}) = sign\left(\sum_{i=1}^{\#SV} \alpha_i y_i \mathcal{K}(x_i, x_{\text{new}}) + b \right)$$

The kernel trick: summary



Non−linear separation
between vectors
using kernel function

=

Linear separation
between mapped vectors
using inner product

$K(X1,X2)$

$< \Phi(X1) , \Phi(X2) >$

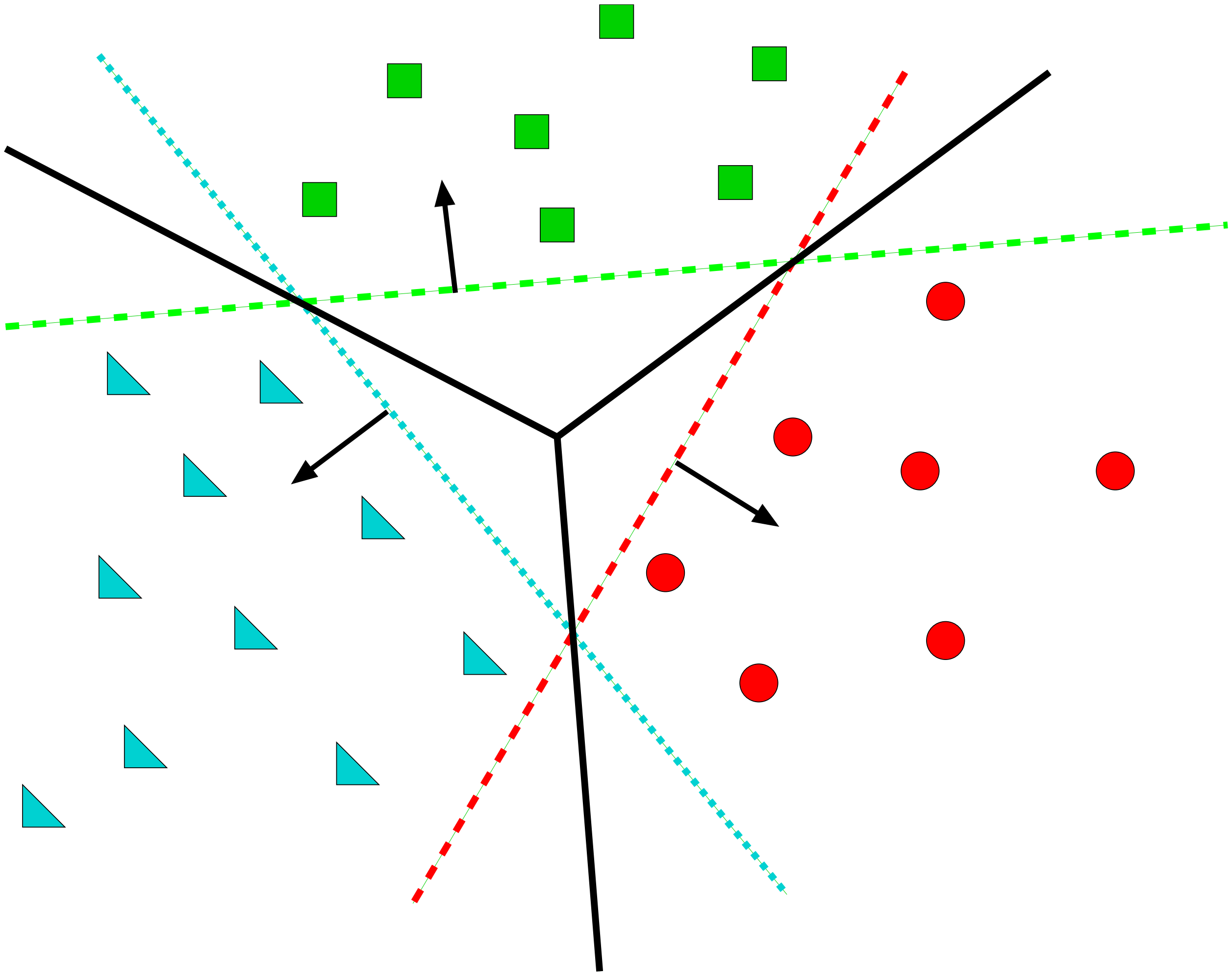
Examples of Kernels

$$\begin{aligned} \textbf{linear} \quad \mathcal{K}(x_1, x_2) &= \langle x_1, x_2 \rangle \\ \textbf{polynomial} \quad \mathcal{K}(x_1, x_2) &= (\gamma \langle x_1, x_2 \rangle + c_0)^d \\ \textbf{radial basis function} \quad \mathcal{K}(x_1, x_2) &= \exp\left(-\gamma \|x_1 - x_2\|^2\right) \end{aligned}$$

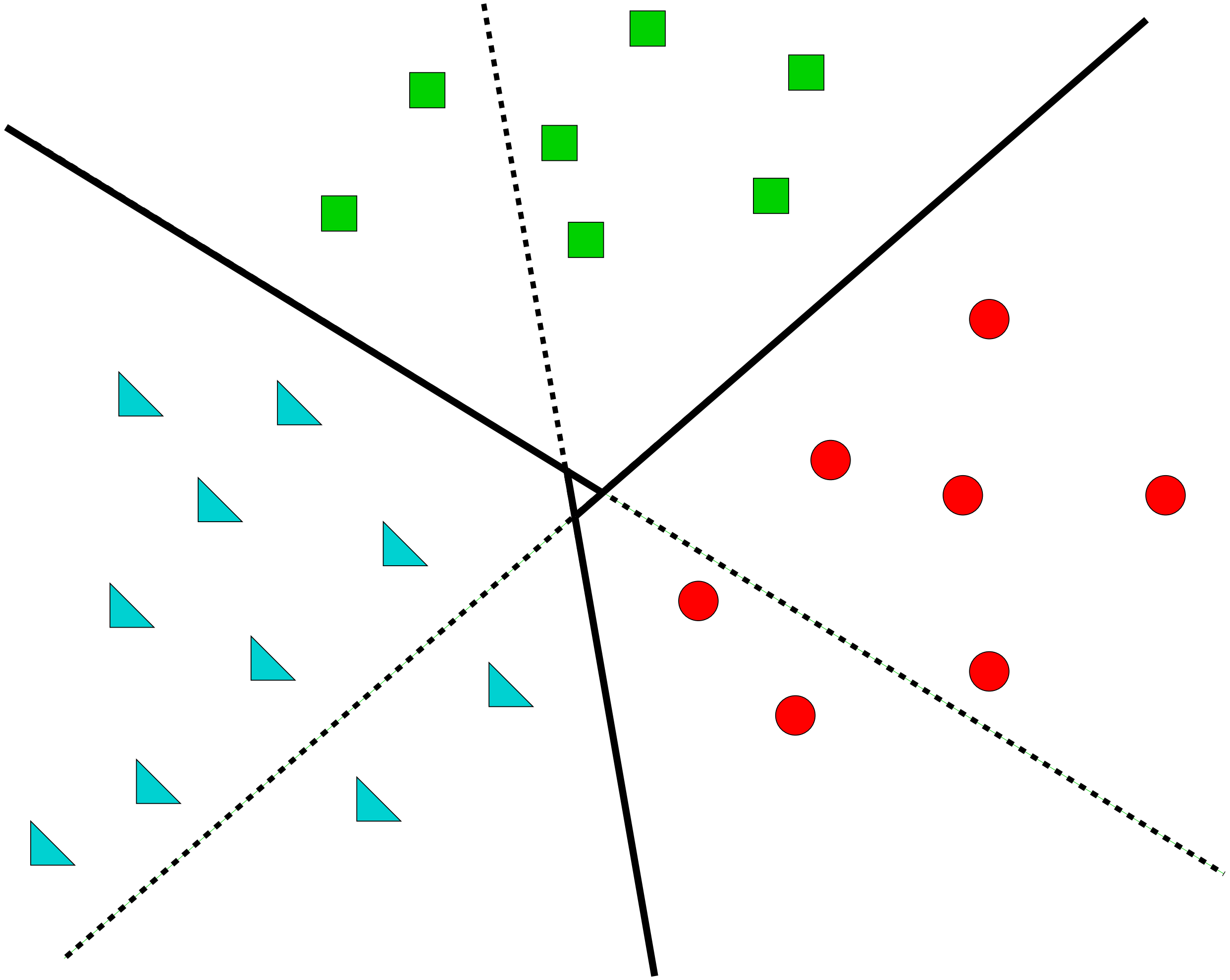
Parameters of SVM

Kernel Parameters	γ:	width of rbf coeff. in polynomial ($= 1$)
	d:	degree of polynomial
	c_0	additive constant in polynomial ($= 0$)
Error weight	C:	influence of training errors

More than 2 classes: *ONE-versus-ALL*



More than 2 classes: *ONE-versus-ONE*



Literature on SVM

- `http://www.kernel-machines.org`

- Vladimir Vapnik.
  Statistical Learning Theory. Wiley, NY, 1998.
  The comprehensive treatment of statistical learning theory, including a large amount of material on SVMs

  The Nature of Statistical Learning Theory. Springer, NY, 1995.
  An overview of statistical learning theory, containing no proofs, but most of the crucial theorems and milestones of learning theory. With a detailed chapter on SVMs for pattern recognition and regression

- Bernhard Schölkopf and Alex Smola.
  Learning with Kernels. MIT Press, Cambridge, MA, 2002.
  An introduction and overview over SVMs. A free sample of one third of the chapters (Introduction, Kernels, Loss Functions, Optimization, Learning Theory Part I, and Classification) is available on the book website.

What's next?

I Large Margin Classifiers

II The Kernel Trick

III **Todays practical session**

SVM and PAMR

SVM

SVMs are part of the R package e1071 (called after the TU Vienna statistics department).

Source + Reference Manual: `http://cran.r-project.org/`

PAMR

PAMR is not a regular R package. It has to be obtained from the author directly. There seems to be an EXCEL plug-in coming soon.

Source + Reference Manual: `http://www-stat.stanford.edu/~tibs/PAM/`

Diagnosis by SVMs

TASK:

For 3 new patients in your hospital, decide which kind of breast cancer they suffer from (ER+ or ER-) using their expression profiles.

IDEA:

Learn the difference between the cancer types from an archive of 46 expression profiles, which were analyzed and classified by an expert.

Training ... tuning ... testing

TRAINING:

```
svm.doctor <- svm(data="46 profiles", labels="by an expert",
                  kernel="..", parameters="..")
```

TUNING:

Now tune the kernel and the parameters to achieve a good generalization ability - measured by training error, cross validation error. Select informative genes.

TESTING:

```
svm.diagnosis <- predict(svm.doctor, new.patients)
```

Thank you!