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

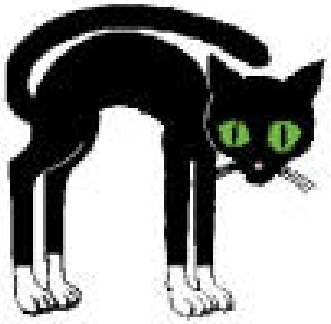
II The Kernel Trick

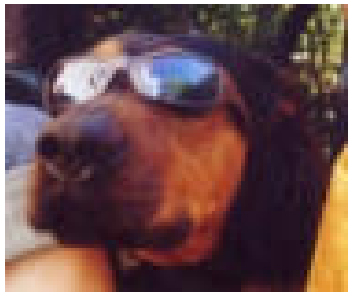
III Todays practical session

Supervised Learning

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

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

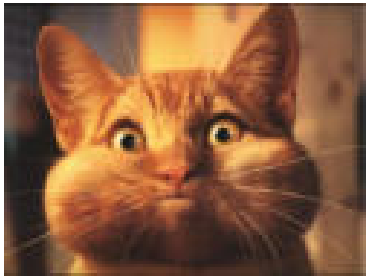


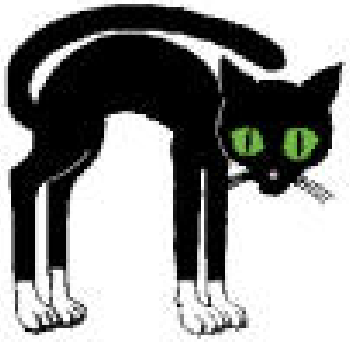


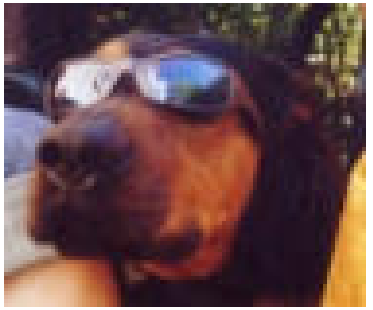
Unsupervised Learning

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

Yeah, me too!







Supervised Learning

Training set: a number of expression profiles with known labels which represent the true population.

> *Difference to clustering: there you don't know the labels, you have to find a structure on your own.*

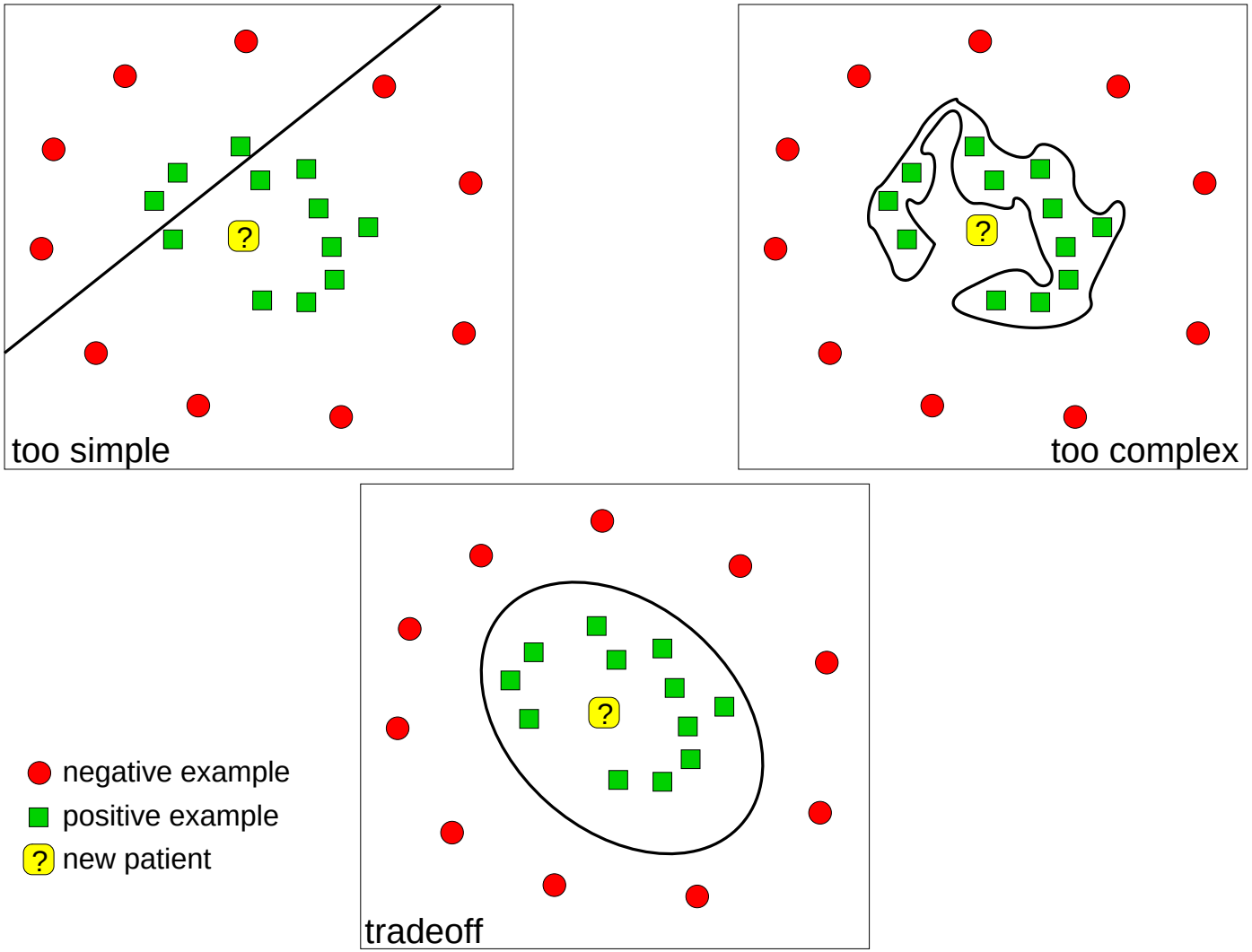
Learning/Training: find a decision rule which explains the training set well.

> *This is the easy part, because we know the labels of the training set!*

Generalisation ability: how does the decision rule learned from the training set generalize to new specimen?

Goal: find a decision rule with high generalisation ability.

Underfitting and Overfitting

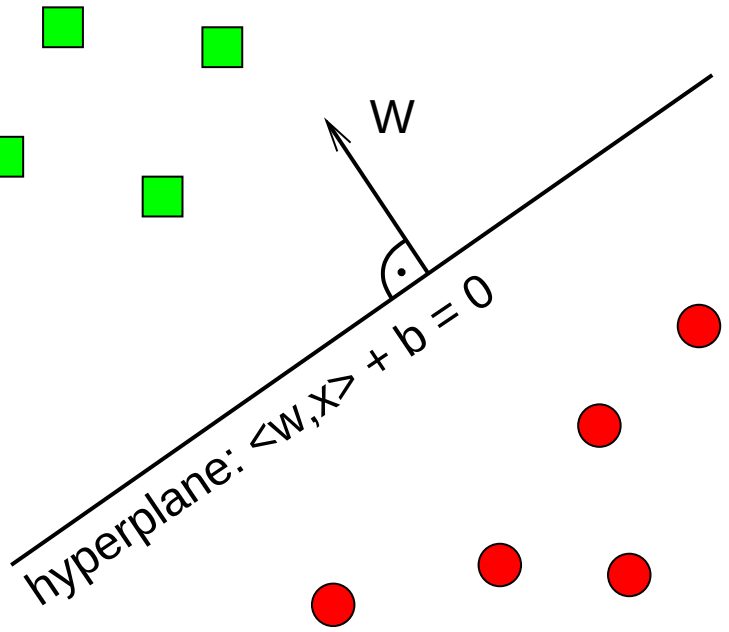


Linear separation of the training set

We start with linear separation and add complexity in a second step by using kernel functions.

A **separating hyperplane** is defined by
- the normal vector w and
- the offset b:

$$\text{hyperplane} = \{\, x \mid \langle w, x \rangle + b = 0 \}$$



$\langle \cdot, \cdot \rangle$ is called *inner product*, *scalar product* or *dot product*.

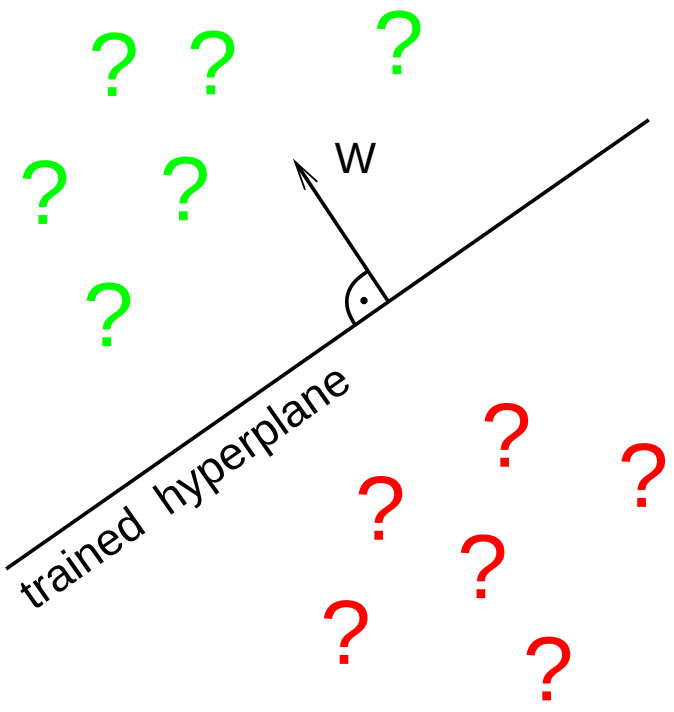
Training: Choose w and b from the labeled examples in the training set.

Predict the label of a new point

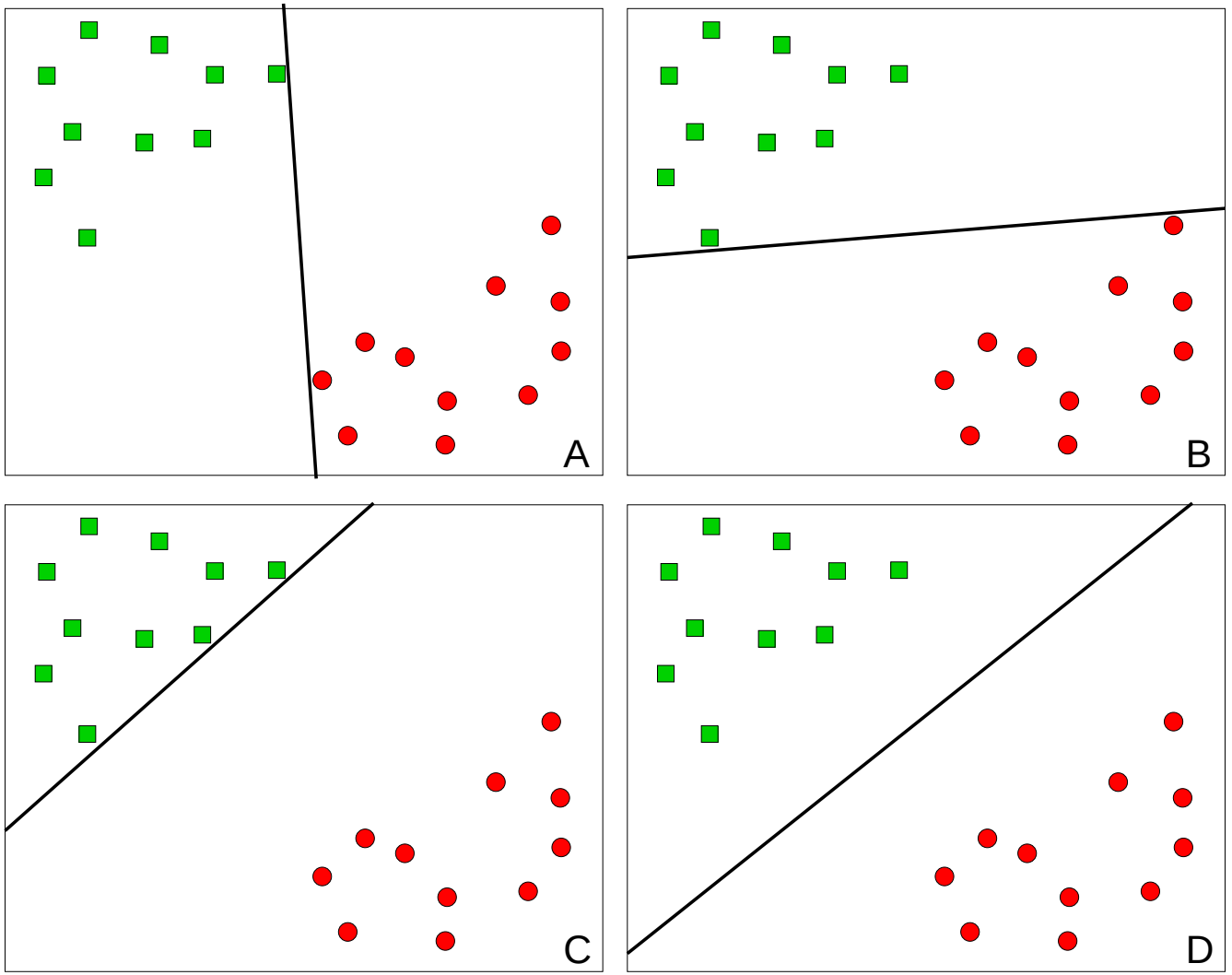
Prediction: On which side of the hyperplane does the new point lie?

Points in the direction of the normal vector are classified as **POSITIVE**.

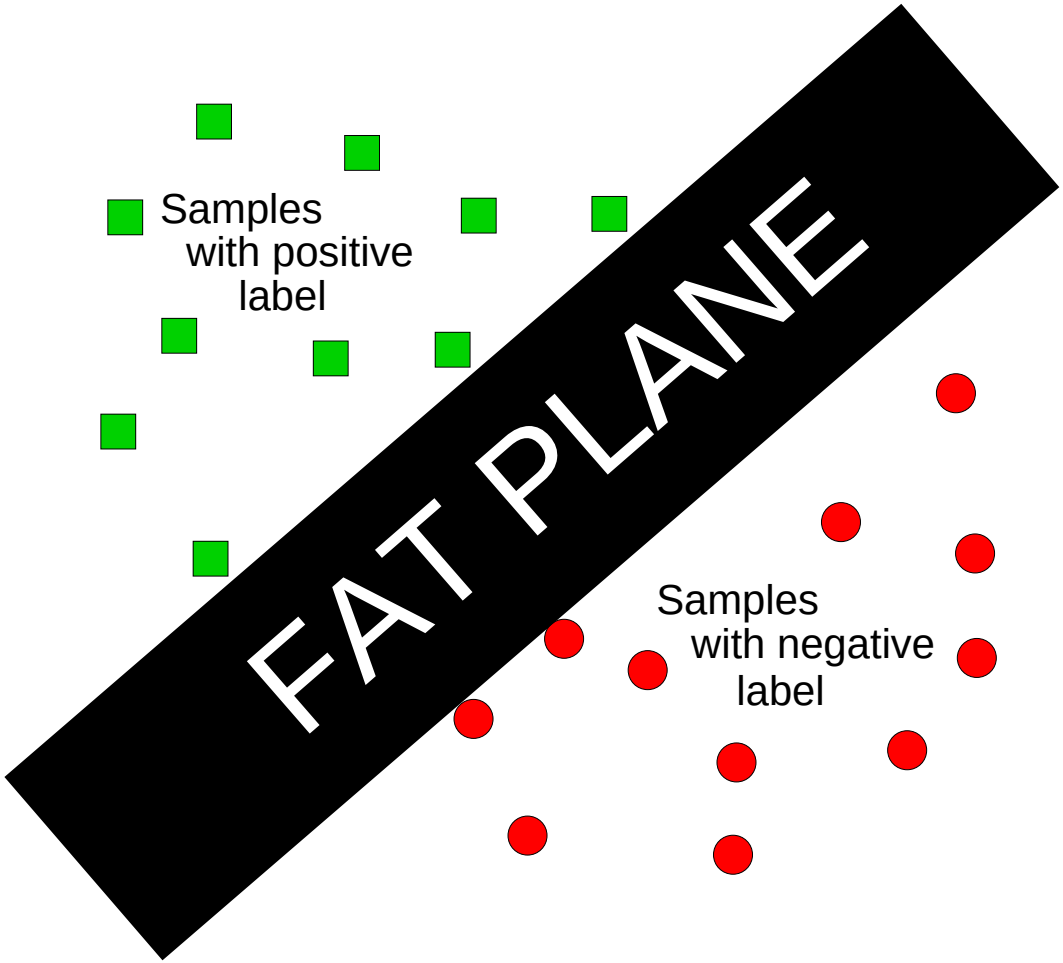
Points in the opposite direction are classified as **NEGATIVE**.



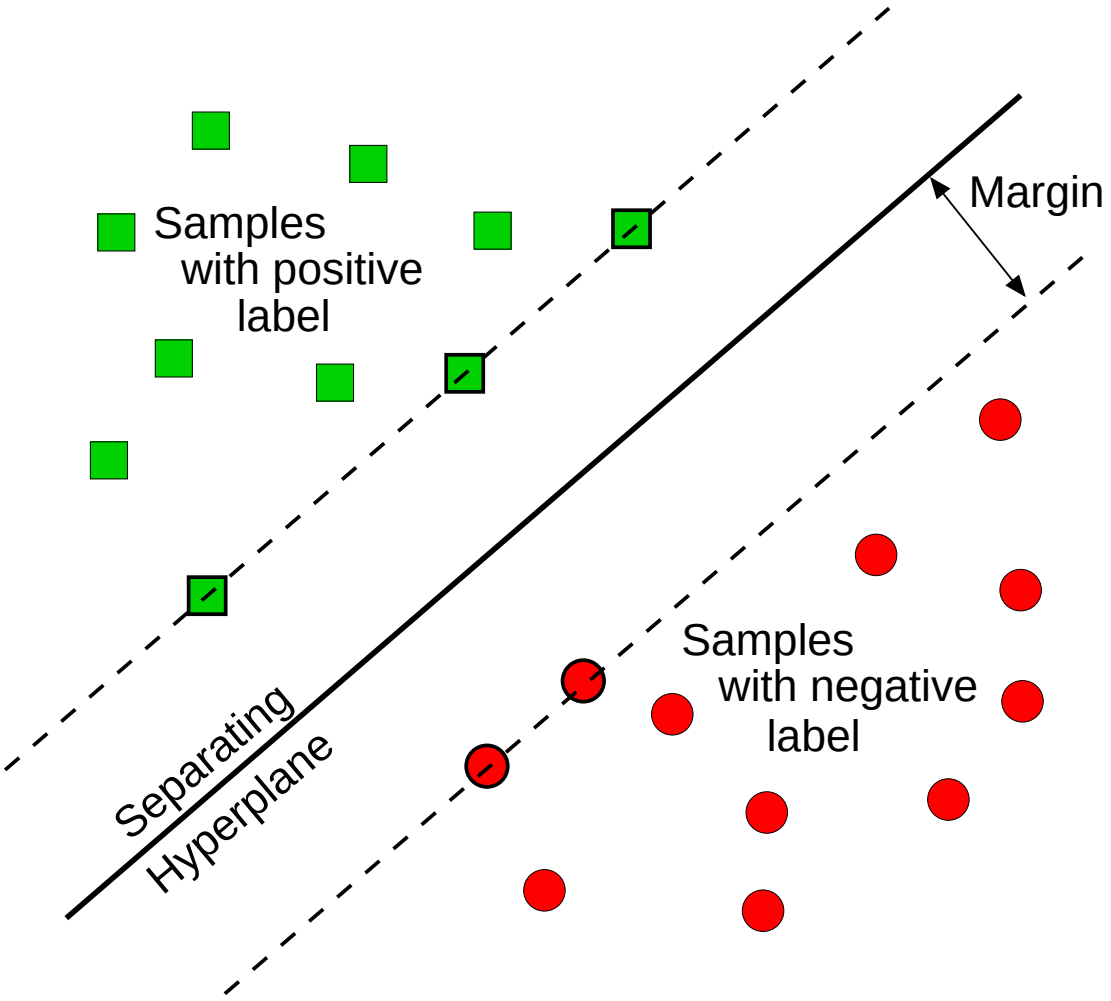
Which hyperplane is the best?



No sharp knive, but a fat plane



Separate the training set with maximal margin

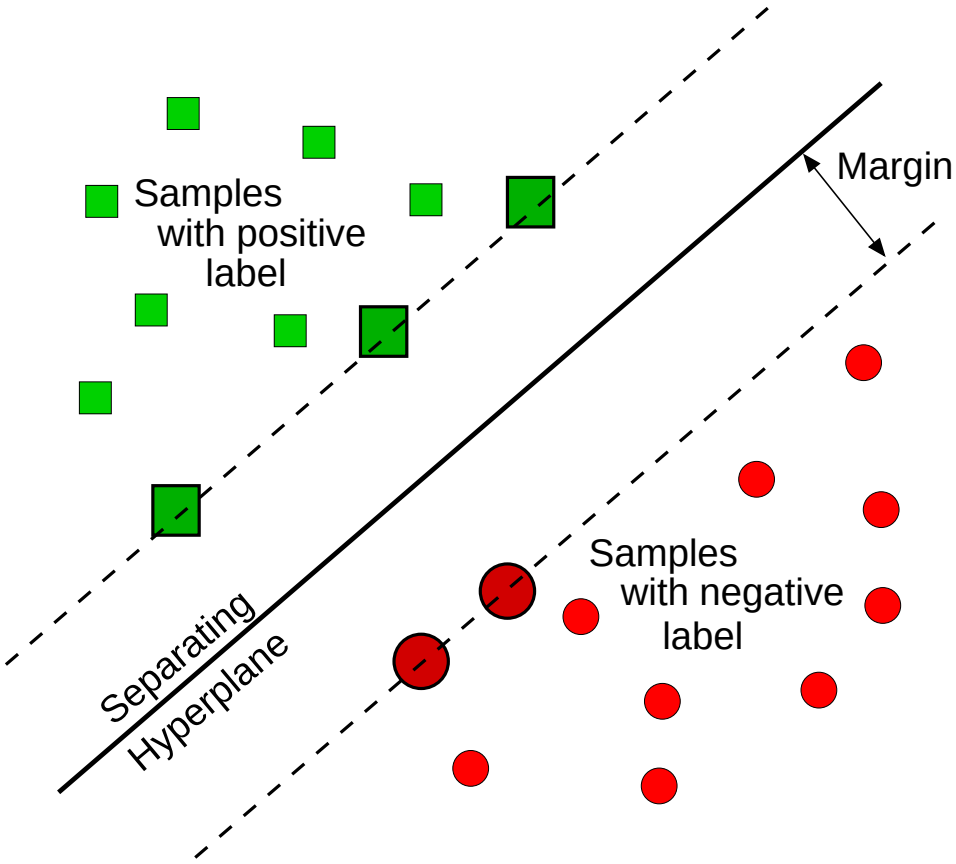


What are Support Vectors?

The points nearest to the separating hyperplane are called Support Vectors.

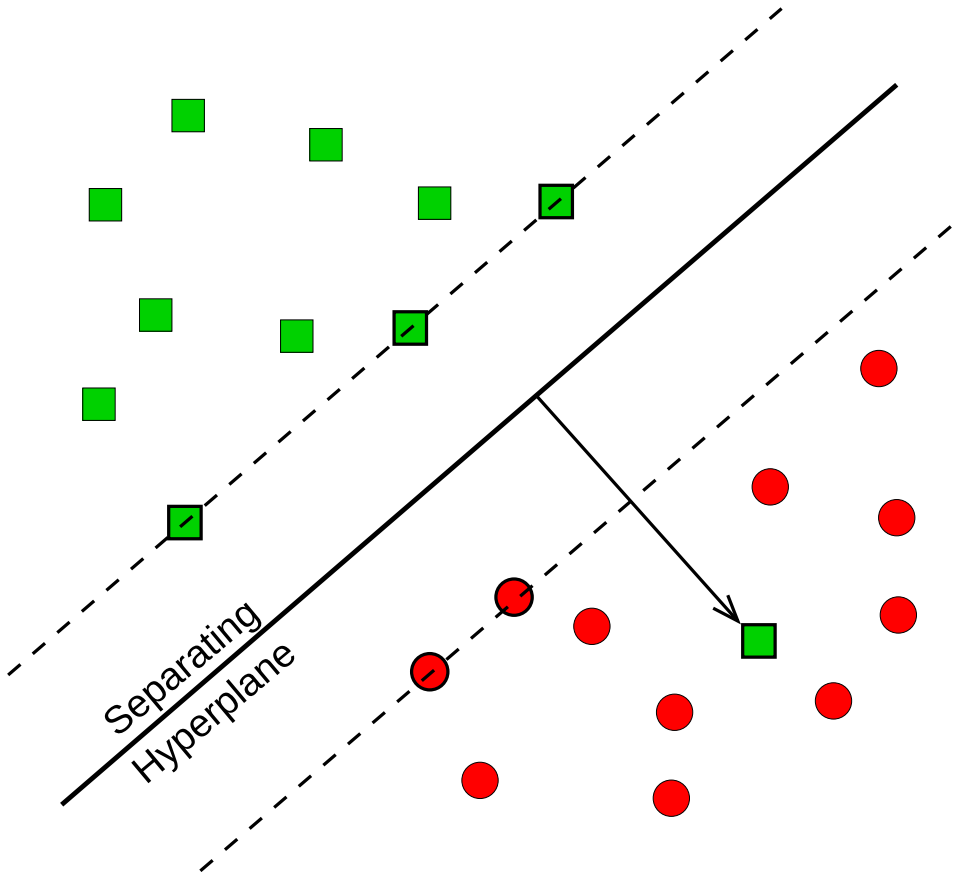
Only they determine the position of the hyperplane. **All other points have no influence!**

Mathematically: the weighted sum of the Support Vectors is the normal vector of the hyperplane.



Non-separable training sets

Use linear separation, but admit training errors.



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

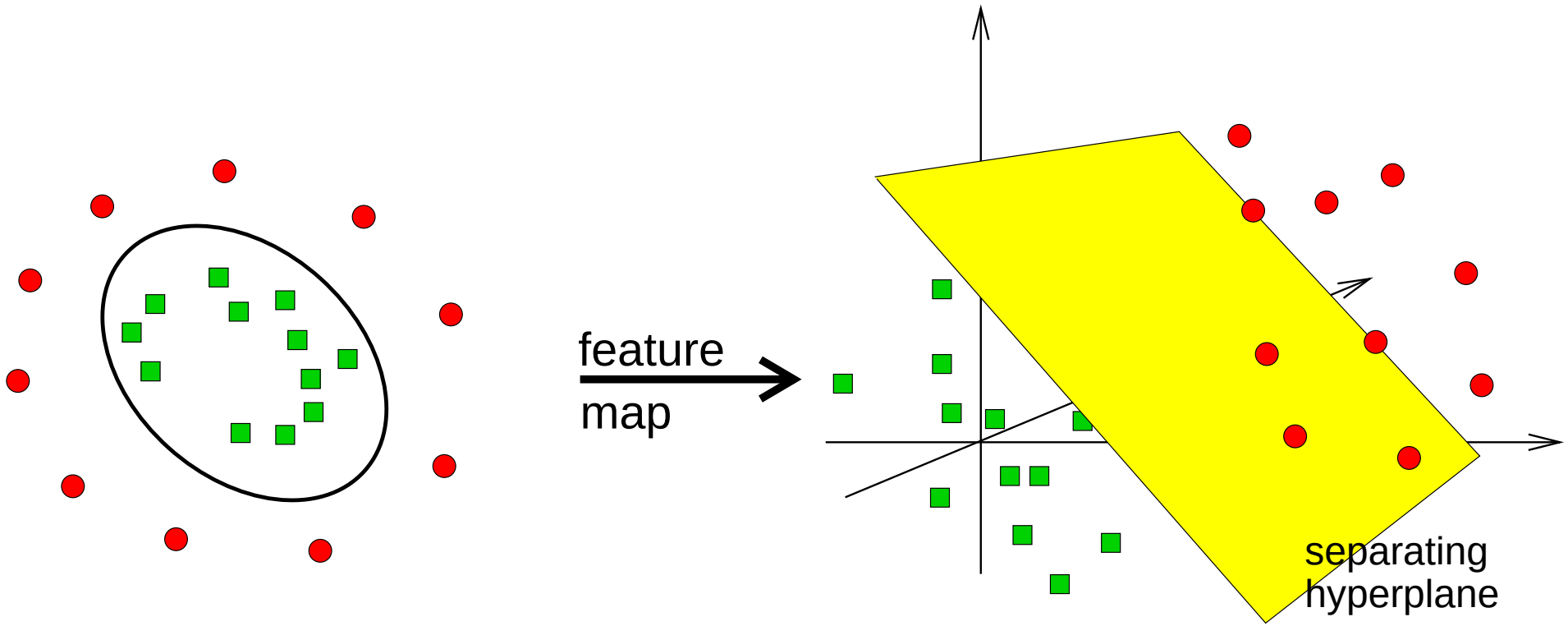
What's next?

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

Maximal margin hyperplanes in feature space

If classification is easier in a high-dimenisonal feature space, we would like to build a maximal margin hyperplane there.

The construction depends on inner products $\Rightarrow$ we will have to evaluate inner products in the feature space.

This can be computationally intractable, if the dimensions become too large!

Loophole

Use a kernel function that lives in low dimensions, but behaves like an inner product in high dimensions.

Kernel functions

Expression profiles $p = (p_1, p_2, \ldots, p_g) \in \mathbb{R}^g$
and $q = (q_1, q_2, \ldots, q_g) \in \mathbb{R}^g$.

Similarity in gene space: INNER PRODUCT

$$\langle p, q \rangle = p_1 q_1 + p_2 q_2 + \ldots + p_g q_g$$

Similarity in feature space: KERNEL FUNCTION

$$\mathcal{K}(p, q) = \text{polynomial, radial basis, ...}$$

Examples of Kernels

linear $\mathcal{K}(p,q) = \langle p,q \rangle$

polynomial $\mathcal{K}(p,q) = (\gamma \langle p,q \rangle + c_0)^d$

radial basis function $\mathcal{K}(p,q) = \exp\left(-\gamma \|p-q\|^2\right)$

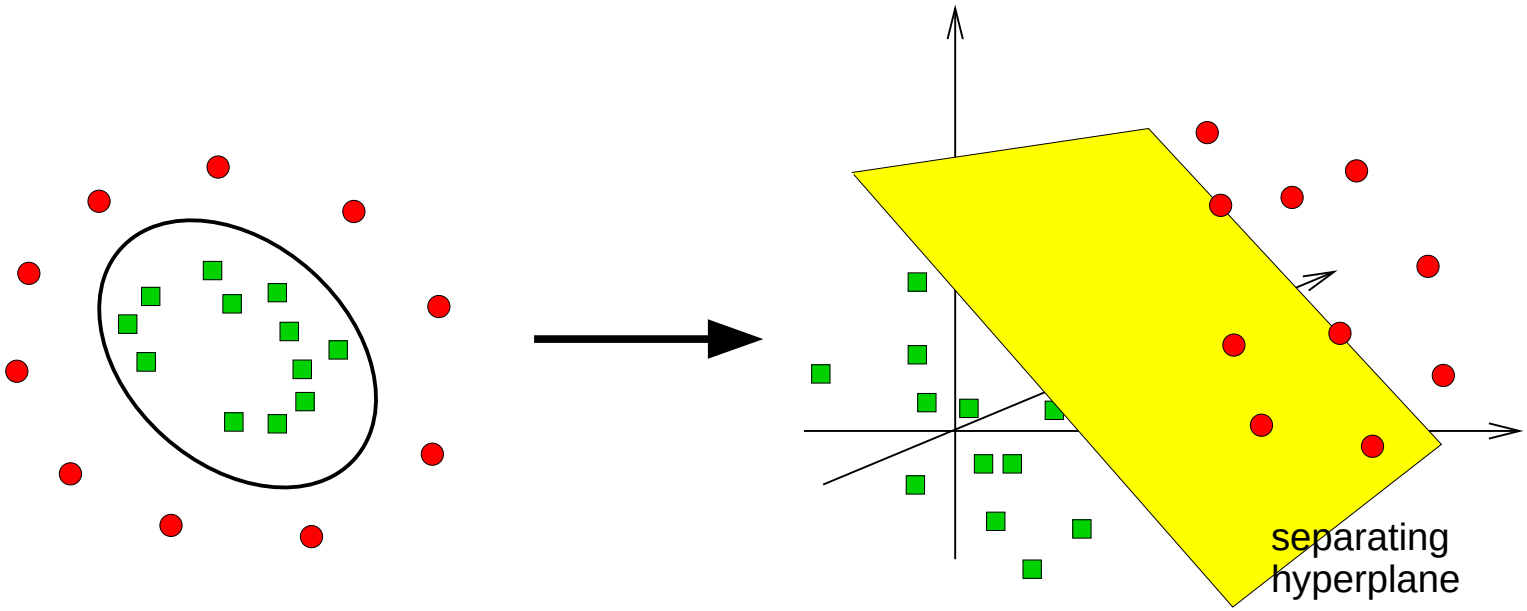
Why is it a trick?

We do not need to know, how the feature space really looks like, we just need the kernel function as a measure of similarity.

This is kind of black magic: we do not know what happens inside the kernel, we just get the output.

Still, we have the geometric interpretation of the maximal margin hyperplane, so SVMs are more transparent than e. g. Artificial Neural Networks.

The kernel trick: summary



Non-linear separation
between vectors
in gene space
using kernel functions

=

Linear separation
between vectors
in feature space
using inner product

Support Vector Machines

A Support Vector Machine is

a maximal margin hyperplane in feature space

built by using a kernel function in gene space.

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

SVM@work: low complexity

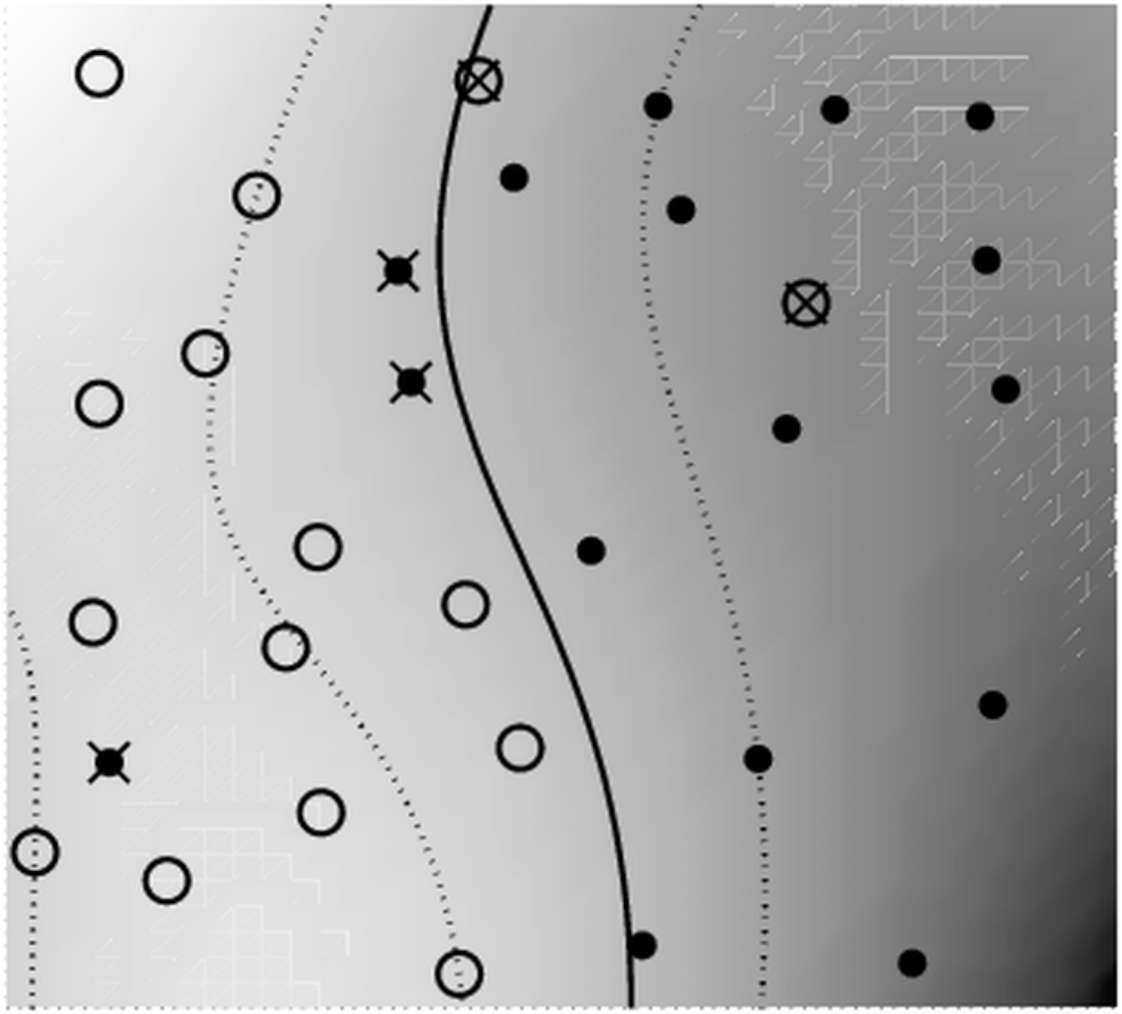


Figure taken from SCHÖLKOPF and SMOLA, *Learning with Kernels*, MIT Press 2002, p217

SVM@work: medium complexity

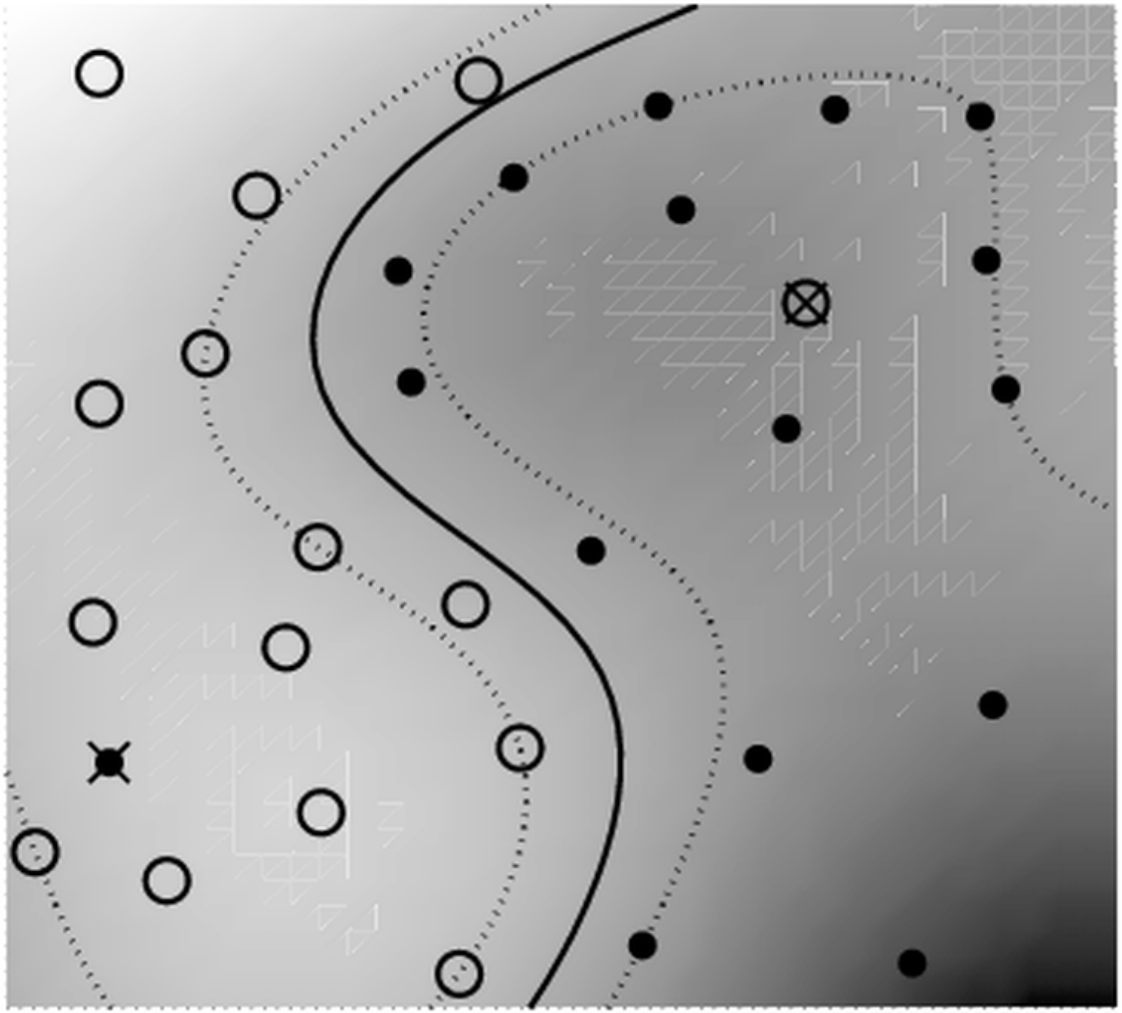


Figure taken from SCHÖLKOPF and SMOLA, *Learning with Kernels*, MIT Press 2002, p217

SVM@work: high complexity

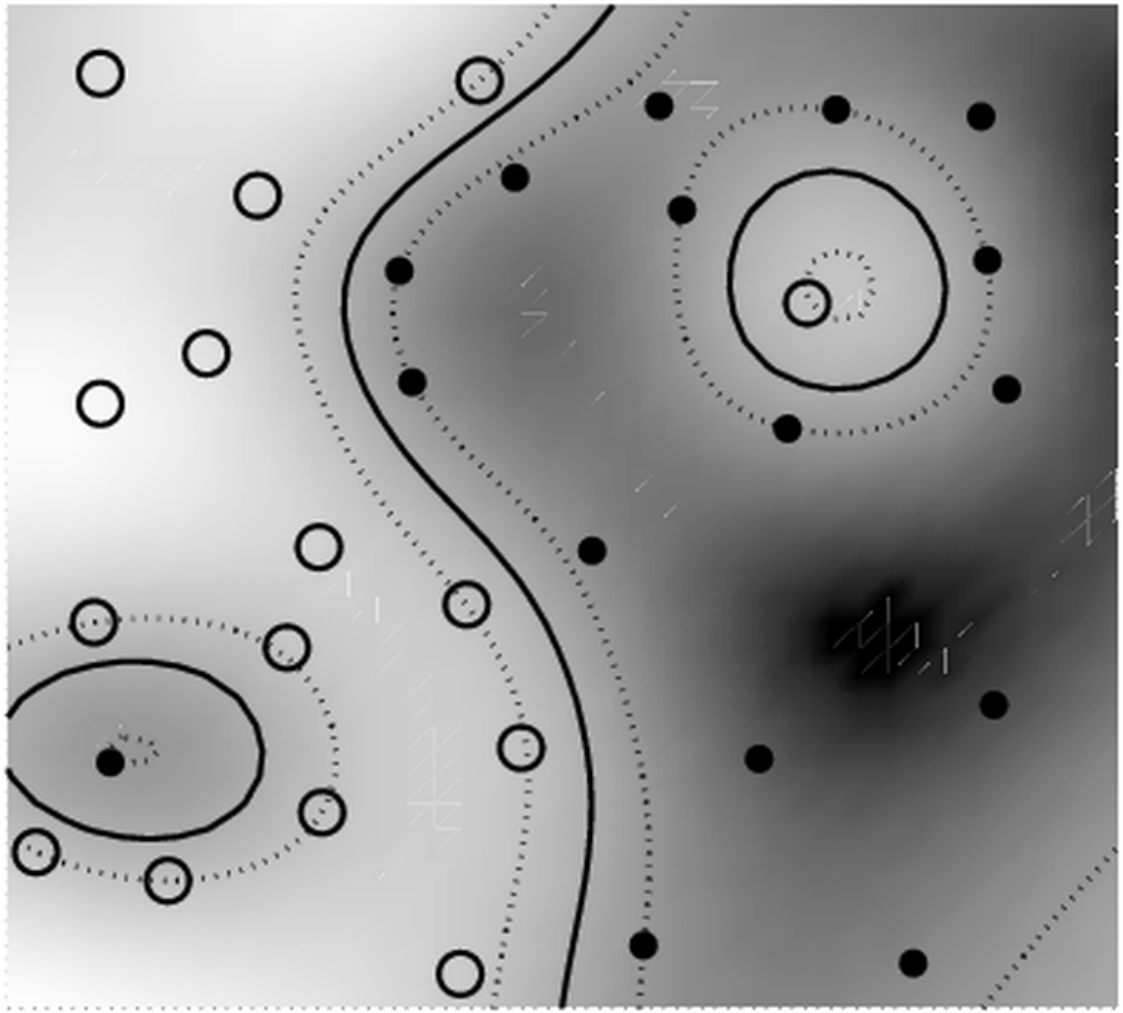


Figure taken from SCHÖLKOPF and SMOLA, *Learning with Kernels*, MIT Press 2002, p217

Literature on SVM

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

- 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.

- 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

What's next?

I Large Margin Classifiers

II The Kernel Trick

III **Todays practical session**

Practical session on classification

Learn to classify tumor samples

by Support Vector Machines

and Nearest Shrunken Centroids.

SVM and PAMR

http://cran.r-project.org/

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

You can also download **pamr** here. See the authors webpage for some more information `http://www-stat.stanford.edu/~tibs/PAM/`

Computational Diagnosis

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 SVM for good generalization ability (training error, cross validation error). Select informative genes.

TESTING:

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