Molecular diagnosis, part II

Florian Markowetz
`florian.markowetz@molgen.mpg.de`
Max Planck Institute for Molecular Genetics
Computational Diagnostics Group
Berlin, Germany

IPM workshop
Tehran, 2005 April

Supervised learning

In the first part, I introduced molecular diagnosis as a problem of **classification in high dimensions**.

From given patient expression profiles and labels, we derive **a classifier to predict future patients**.

By the labels we are given a structure in the data. Our task: extract and generalize the structure. This is a problem if **supervised learning**.

It is different from **unsupervised learning**, where we have to find a structure in the data by ourselves: **Clustering, class discovery**.

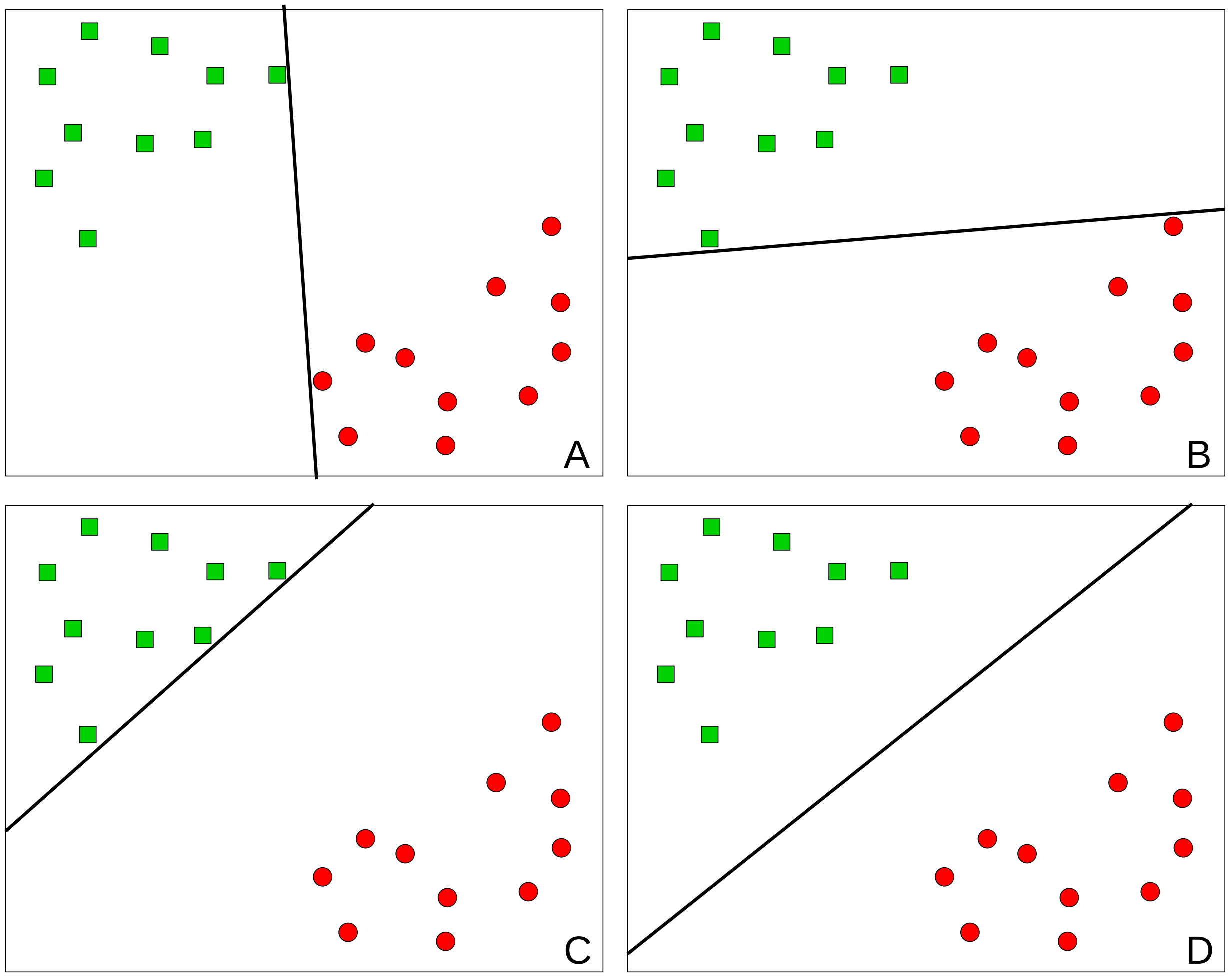
What's to come

This part will deal with

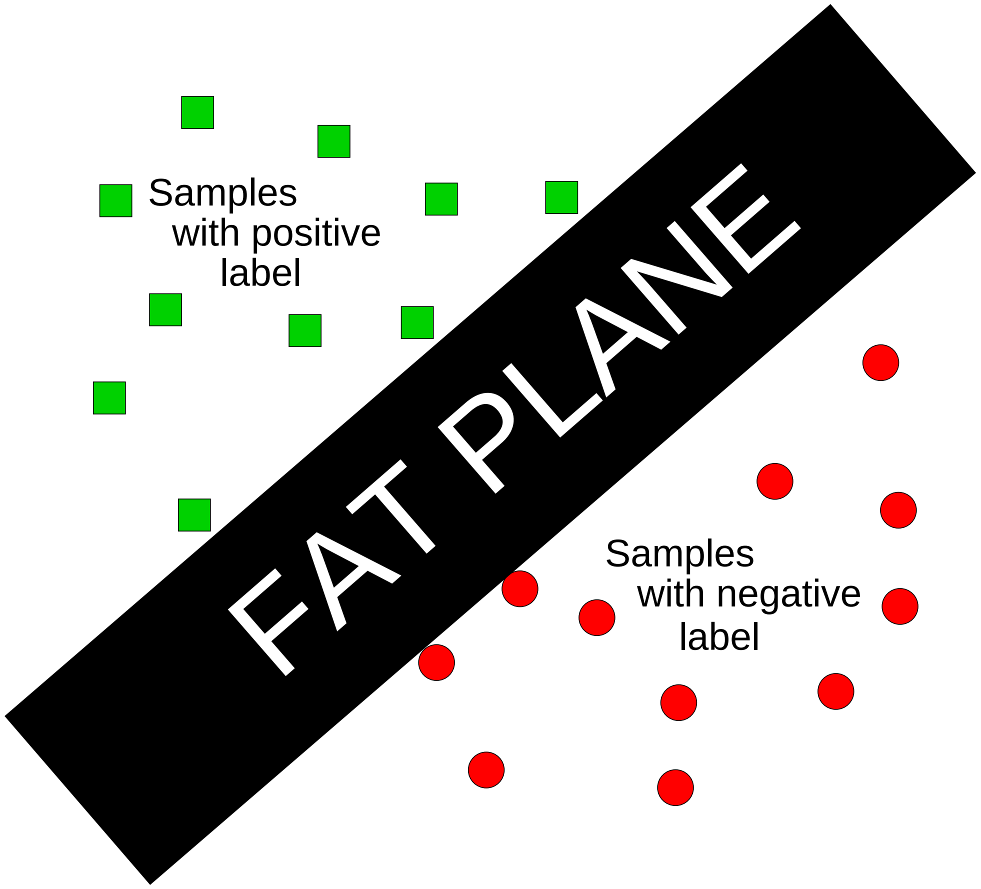
1. **Support vector machines**
   $\longrightarrow$ Maximal margin hyperplanes, non-linear similarity measures
2. **Model selection and assessment**
   $\longrightarrow$ Traps and pitfalls, or: How to cheat.
3. **Interpretation of results**
   $\longrightarrow$ what do classifiers teach us about biology?

Support Vector Machines

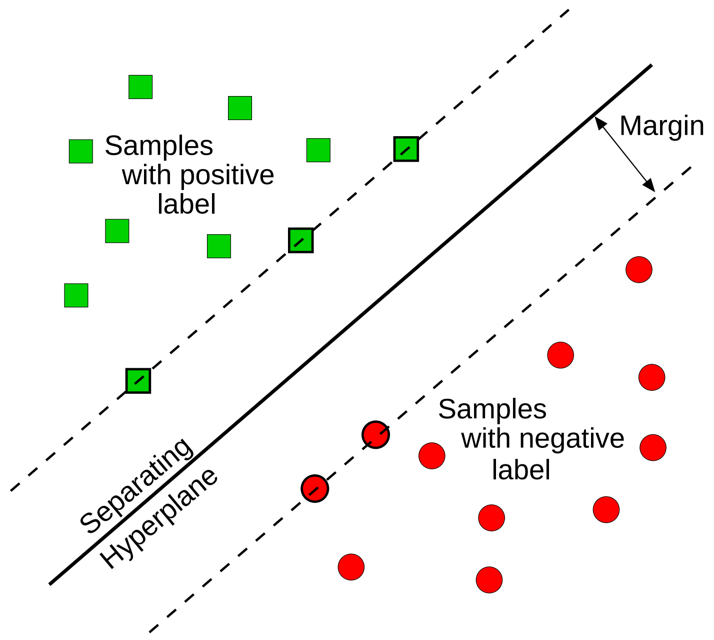
Which hyperplane is the best?



No sharp knive, but a fat plane



Separate the training set with maximal margin



A hyperplane is a set of points $\mathbf{x}$ satisfying

$$\langle \mathbf{w}, \mathbf{x} \rangle + b = 0$$

corresponding to a decision function

$$c(\mathbf{x}) = sign(\langle \mathbf{w}, \mathbf{x} \rangle + b).$$

There exists a unique **maximal margin hyperplane** solving

$$\underset{\mathbf{w},b}{\text{maximize}} \min\{\|\mathbf{x} - \mathbf{x}^{(i)}\| : \ \mathbf{x} \in \mathbb{R}^p, \ \langle \mathbf{w}, \mathbf{x} \rangle + b = 0, \ i = 1, \ldots, N\}$$

Hard margin SVM

First we scale $(\mathbf{w}, b)$ with respect to $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}$ such that

$$\min_i |\langle \mathbf{w}, \mathbf{x}^{(i)} \rangle + b| = 1.$$

The points closest to the hyperplane now have a distance of $1/\|\mathbf{w}\|$.

Hard margin SVM

First we scale $(\mathbf{w}, b)$ with respect to $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}$ such that

$$\min_i |\langle \mathbf{w}, \mathbf{x}^{(i)} \rangle + b| = 1.$$

The points closest to the hyperplane now have a distance of $1/\|\mathbf{w}\|$.

Then the maximal margin hyperplane is the solution of the **primal optimization problem**

$$\begin{array}{ll} \underset{\mathbf{w},b}{\text{minimize}} & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{subject to} & y_i(\langle \mathbf{x}^{(i)}, \mathbf{w} \rangle + b) \geq 1, \quad \text{for all } i = 1, \ldots, N \end{array}$$

The Lagrangian

To solve the problem, introduce the Lagrangian

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{N} \alpha_i (y_i(\langle \mathbf{x}^{(i)}, \mathbf{w} \rangle + b) - 1).$$

It must be **maximized w.r.t.** α and **minimized w.r.t** $\mathbf{w}$ **and** b, *i.e.* a saddle point has to be found.

The Lagrangian

To solve the problem, introduce the Lagrangian

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{N} \alpha_i (y_i(\langle \mathbf{x}^{(i)}, \mathbf{w}\rangle + b) - 1).$$

It must be **maximized w.r.t.** α and **minimized w.r.t** $\mathbf{w}$ **and** b, *i.e.* a saddle point has to be found.

KKT conditions: for all i

$$\alpha_i (y_i(\langle \mathbf{x}^{(i)}, \mathbf{w}\rangle + b) - 1) = 0$$

The Lagrangian *cont'd*

Derivatives w.r.t primal variables must vanish:

$$\frac{\partial}{\partial b} L(\mathbf{w}, b, \alpha) = 0 \quad \text{and} \quad \frac{\partial}{\partial \mathbf{w}} L(\mathbf{w}, b, \alpha) = 0,$$

which leads to

$$\sum_i \alpha_i y_i = 0 \quad \text{and} \quad \mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}^{(i)}.$$

The dual optimization problem

Substituting the conditions for the extremum into the Lagrangian, we arrive at the dual optimization problem:

$$\underset{\alpha}{\text{maximize}} \quad \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}^{(i)}, \mathbf{x}^{(j)} \rangle,$$

$$\text{subject to} \quad \alpha_i \geq 0 \quad \text{and} \quad \sum_{i=1}^{N} \alpha_i y_i = 0.$$

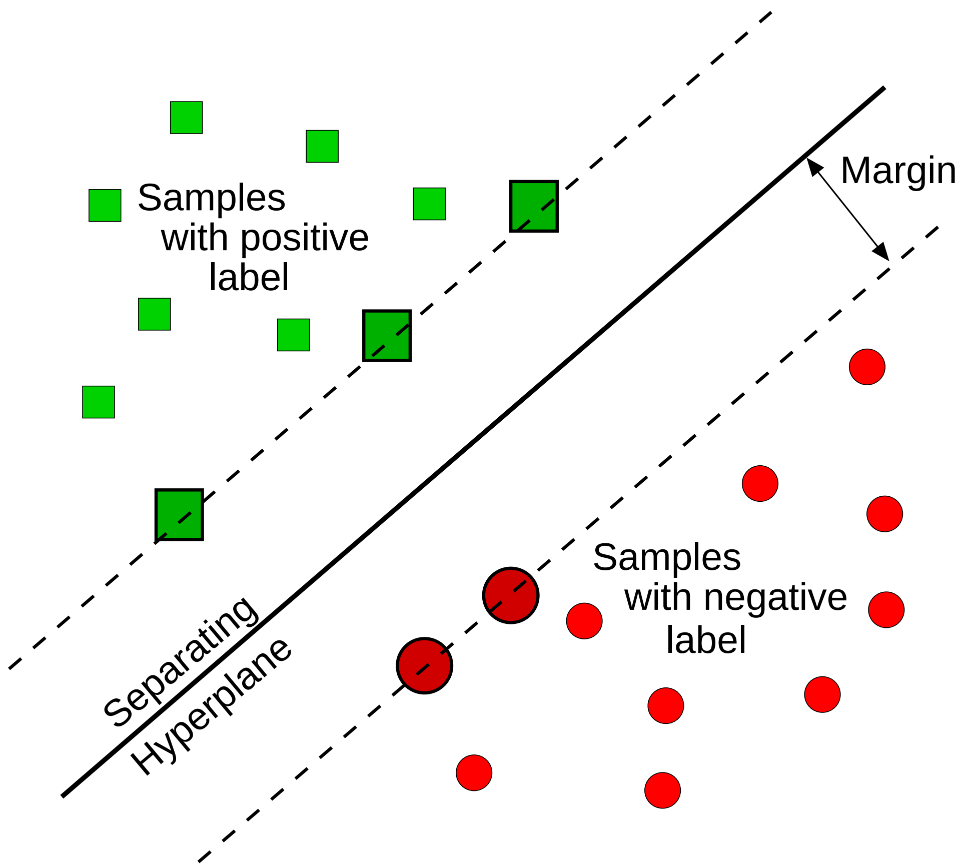
What are Support Vectors?

By the KKT conditions, the points with $\alpha_i > 0$ satisfy

$$y_i(\langle \mathbf{x}^{(i)}, \mathbf{w} \rangle + b) = 1$$

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

The expansion of the $\mathbf{w}$ only depends on them.



Maximal margin hyperplanes

Capacity decreases with increasing margin!

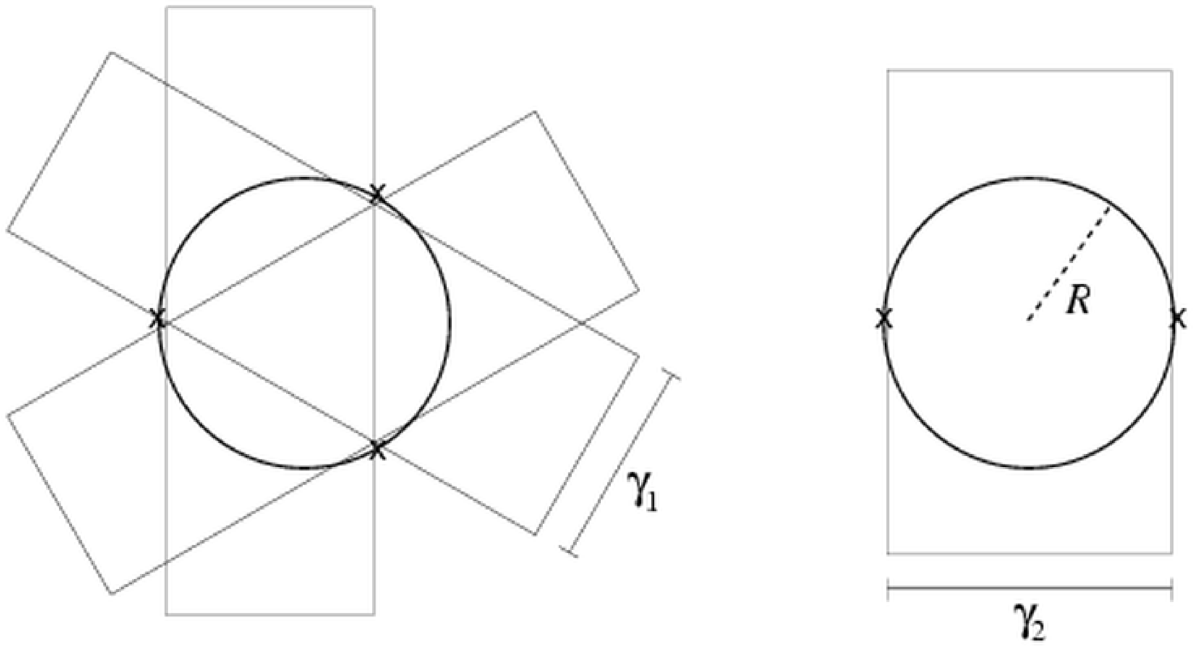
Consider hyperplanes $\langle \mathbf{w}, \mathbf{x} \rangle = 0$, where $\mathbf{w}$ is normalized such that $\min_i |\langle \mathbf{w}, \mathbf{x_i} \rangle| = 1$ for $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$.

The set of decision functions $f_{\mathbf{w}} = sign(\langle \mathbf{w}, \mathbf{x} \rangle)$ defined on $\mathcal{X}$ satisfying $\|\mathbf{w}\| \leq \Lambda$, has a VC dimension h satisfying

$$\boxed{h \leq R^2 \Lambda^2}$$

Here, R is the radius of the smallest sphere centered at the origin and containing the training data [8].

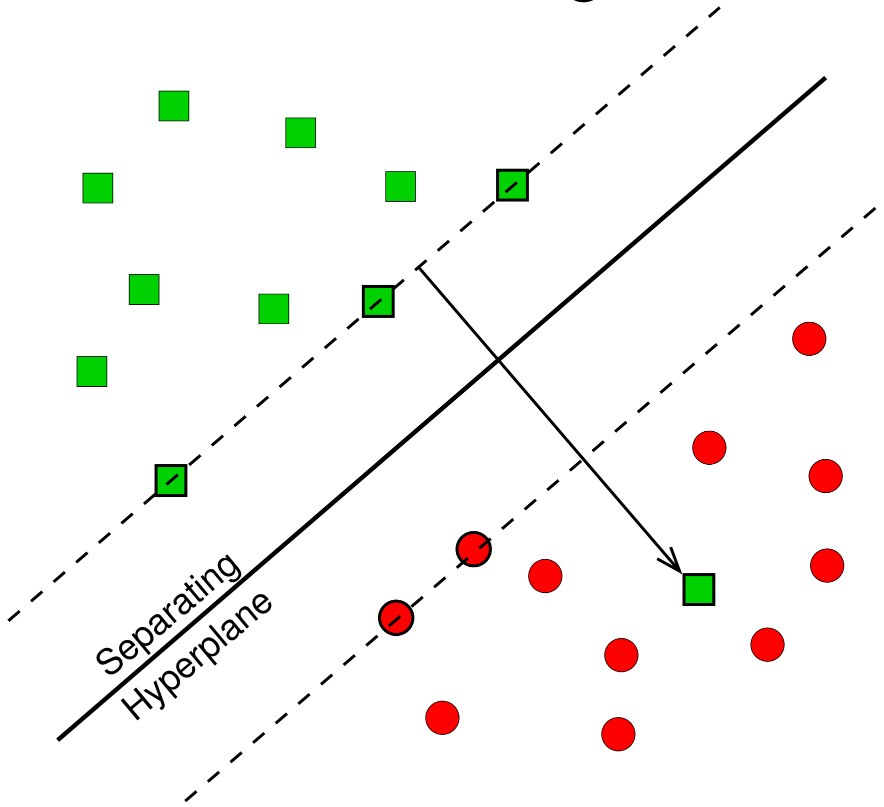
Maximal margin hyperplanes



With margin γ_1 we separate 3 points, with margin γ_2 only two.

Non-separable training sets

Use linear separation, but admit training errors and margin violations.



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

Soft margin primal problem

We relax the **separation constraints** to

$$y_i(\langle \mathbf{x}^{(i)}, \mathbf{w} \rangle + b) \geq 1 - \xi_i$$

and minimize over $\mathbf{w}$ and b the **objective function**

$$\frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{N} \xi_i.$$

Writing down the Lagrangian, computing derivatives w.r.t primal variables, substituting them back into the objective function . . .

Soft margin dual problem

... gives the dual problem

$$\underset{\alpha}{\text{maximize}} \quad \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}^{(i)}, \mathbf{x}^{(j)} \rangle,$$

$$\text{subject to} \quad 0 \leq \alpha_i \leq C \quad \text{and} \quad \sum_{i=1}^{N} \alpha_i y_i = 0.$$

It differs from the **hard margin dual problem** only in an upper bound on α_i, which limits the influence of single points.

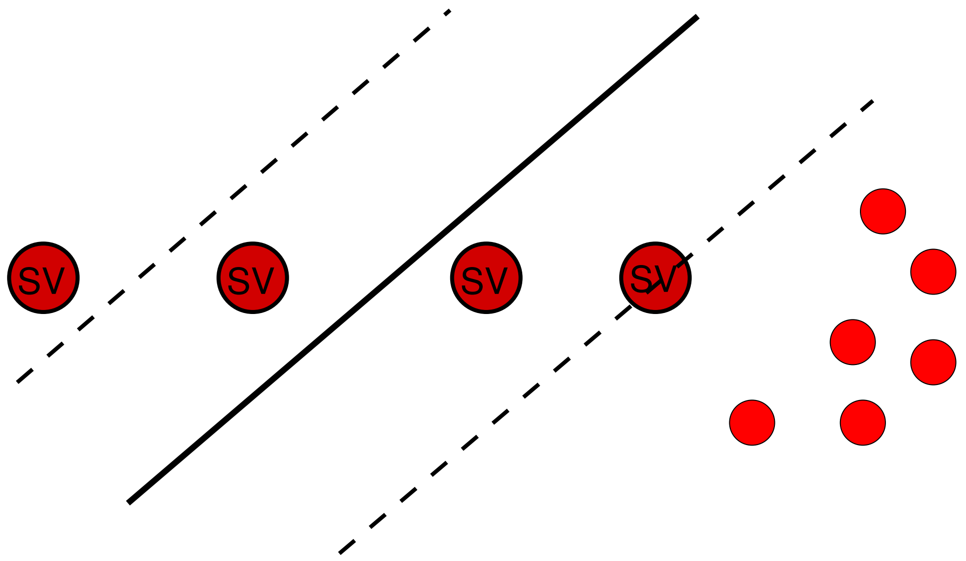
Support vectors revisited

There are three kinds of support vectors in soft margin SVMs:

1. points on the boundary,

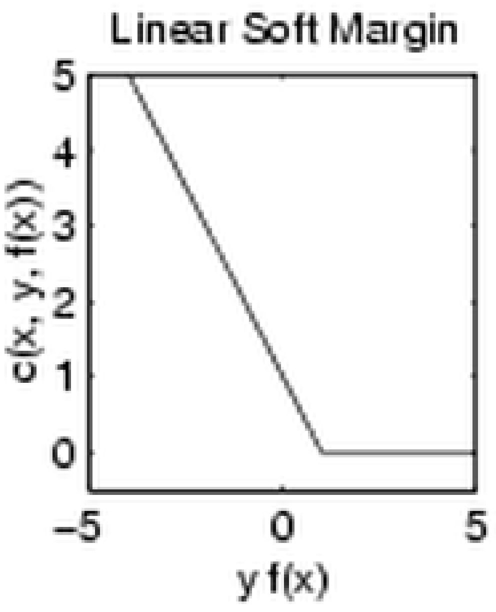
2. margin violations,

3. training errors.



Regularized Risk

How do SVMs fit in the risk framework?



In constructing support vector machines we minimize the empirical risk with **soft margin loss** under the additional constrain of **maximizing the margin**.

This is called a regularized risk [8].

We minimize the risk over a class of functions characterized by big margins (and thus, low capacity).

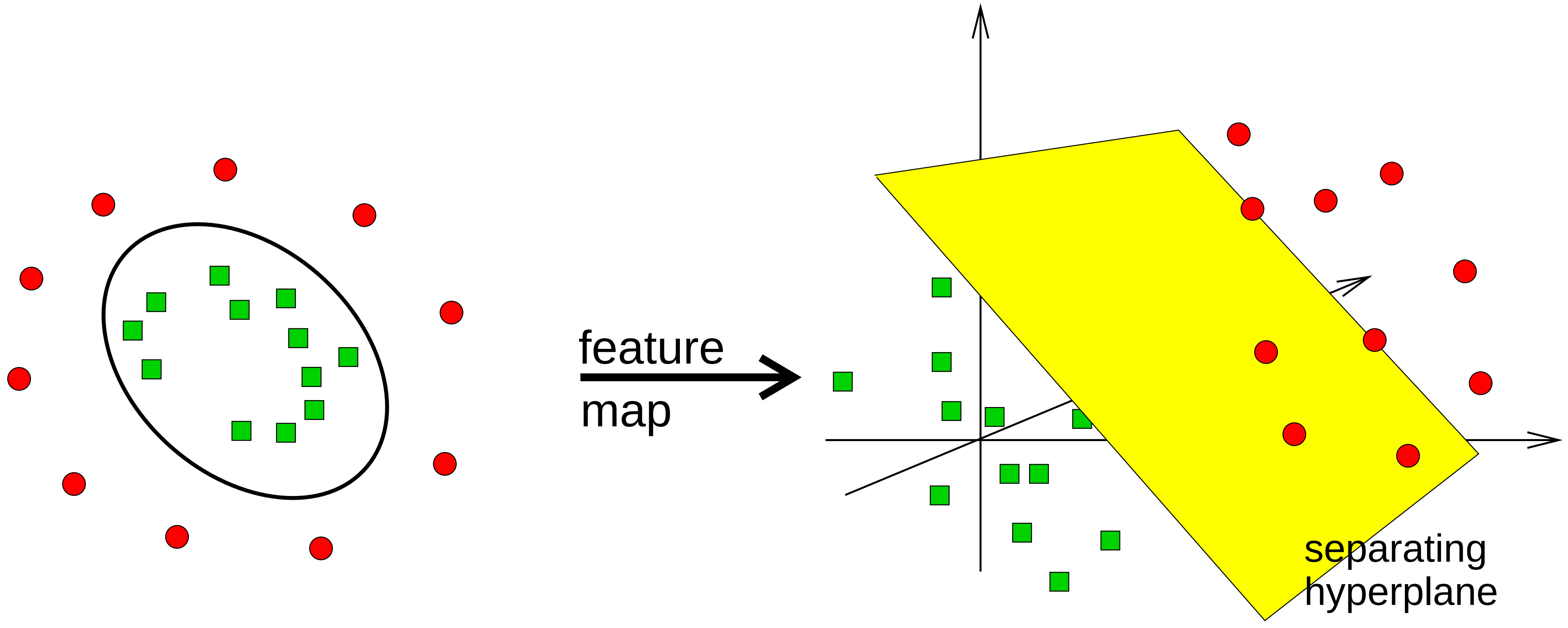
The end?

What we learned so far is

1. how to construct maximal margin hyperplanes (with soft margin),
2. capacity decreases with increasing margin,
3. Maximal margin hyperplanes minimize the regularized risk (and not the empirical risk).

For **microarray data**, you will seldom need more than a maximal margin hyperplane. This is the most simple example of a **support vector machine**. What is missing for a full SVM is a concept of **nonlinear similarity measures** called kernels.

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!

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

Kernels

A kernel is a **(non)linear similarity measure** defined on some set $\mathcal{X}$, which needs not to be an inner product space. (For microarray data, of course $\mathcal{X} = \mathbb{R}^p$)

$$k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$$

Kernels

A kernel is a **(non)linear similarity measure** defined on some set $\mathcal{X}$, which needs not to be an inner product space. (For microarray data, of course $\mathcal{X} = \mathbb{R}^p$)

$$k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$$

Kernels are defined by

1. mapping the data into some inner product space $\mathcal{H}$ and

2. then computing the inner product there:

$$k(x, x') = \langle \Phi(x), \Phi(x') \rangle, \quad \text{with } \Phi : \mathcal{X} \to \mathcal{H}$$

Examples of Kernels

In classification mostly used are $ldots$

linear $k(x, x') = \langle \mathbf{x}, \mathbf{x}' \rangle$

polynomial $k(x, x') = (\gamma \langle \mathbf{x}, \mathbf{x}' \rangle + c_0)^d$

radial basis function $k(x, x') = \exp\left(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2\right)$

... and there are many others tailored to specific purposes.

Why use kernels?

1. Being able to compute dot products amounts to being able to carry out all geometric constructions that can be formulated in terms of angles, lengths, and distances.
2. in $\mathcal{H}$ we can use linear algebra and analytic geometry and have simple interpretations,
3. freedom to choose kernel map Φ enables us to design a large variety of similarity measures and learning algorithms,
4. Choice of kernel (and kernel parameters) controls capacity of classifier.

Support vector machines

A support vector machine is a marriage between
a **maximal margin hyperplane** and a **kernel function**.

We saw how to construct a maximal margin hyperplane using inner products like $\langle \mathbf{w}, \mathbf{x} \rangle$.

Just exchange each inner product by a kernel $k(\cdot, \cdot)$ and you get a full SVM.

The maximal margin hyperplane is constructed in feature space $\mathcal{H}$, not in input space $\mathcal{X}$.

Model assessment

Model selection and assessment

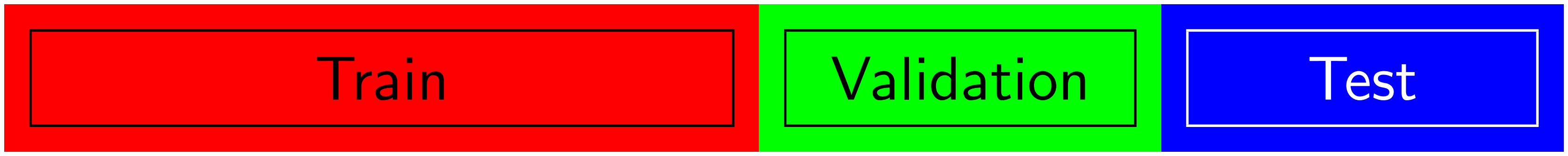
We have to distinguish two different objectives:

Model selection: Estimating the performance of different models in order to choose the (approximate) best one.

Model assessment: Having chosen a final model, estimating its prediction error (generalization error) on new data.

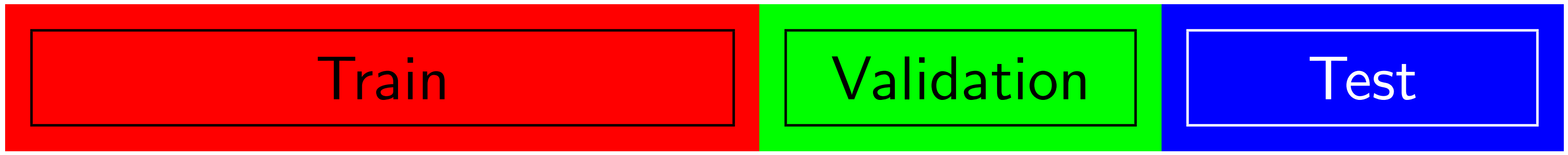
Model selection

Best of all worlds

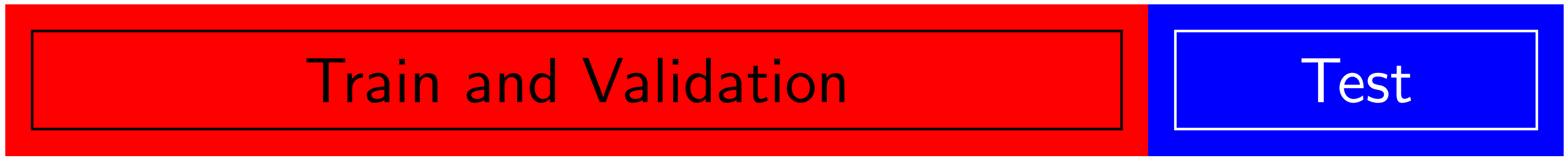


Model selection

Best of all worlds

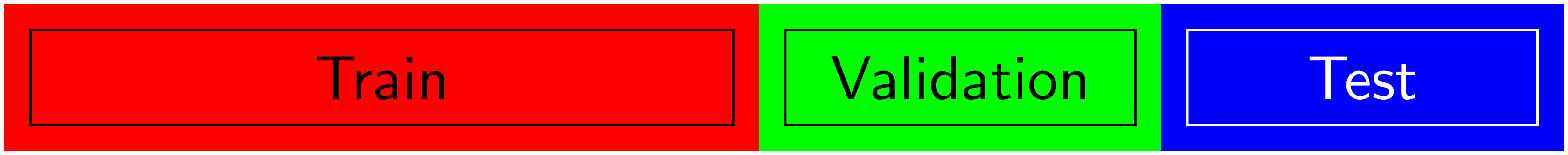


Also OK

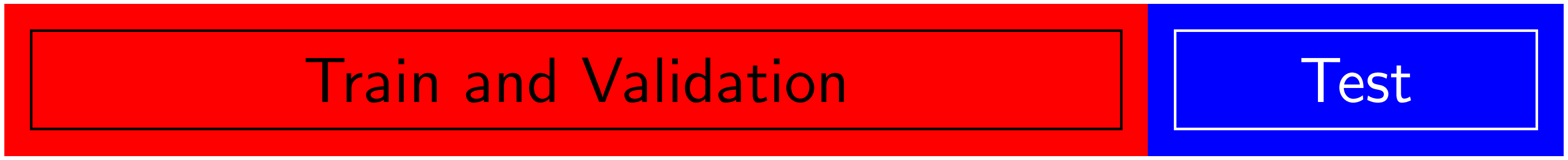


Model selection

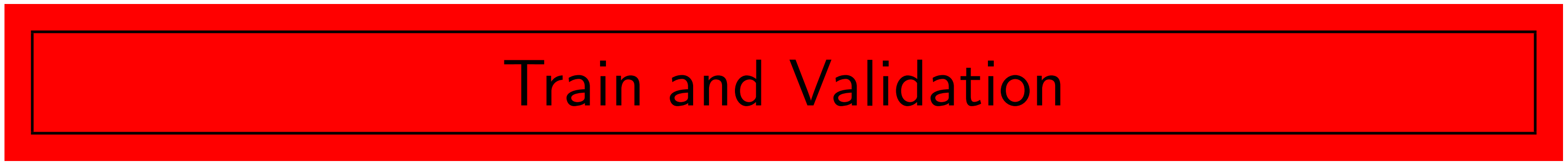
Best of all worlds



Also OK

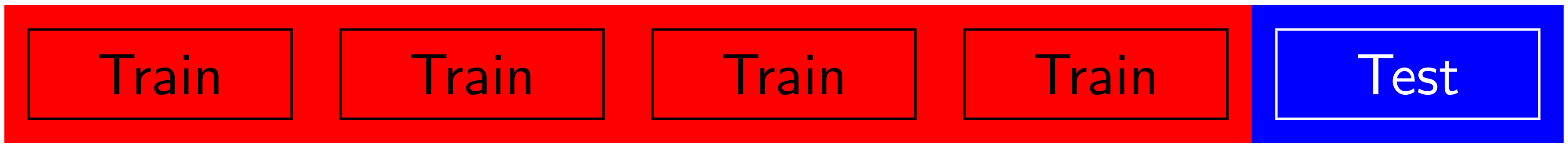


The world we (usually) live in



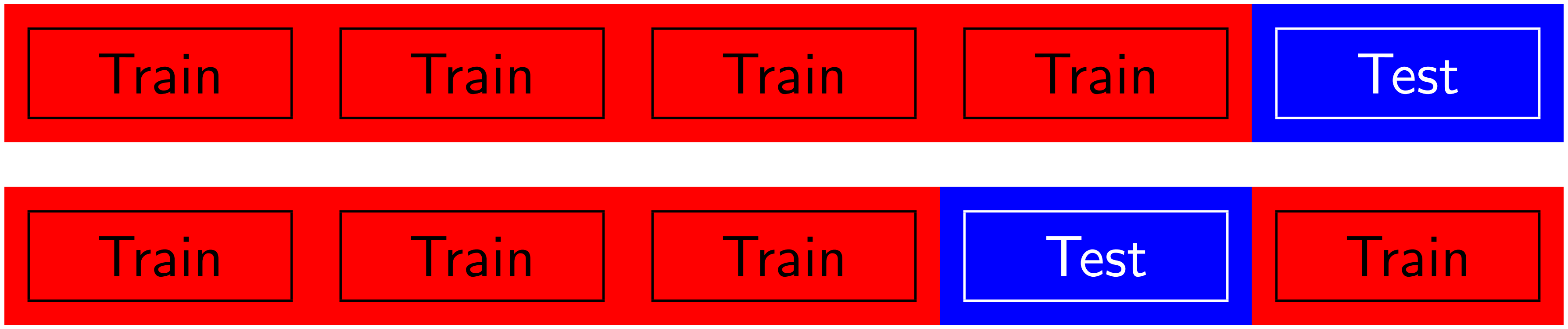
Cross-validation

Efficient way to estimate the error rate:



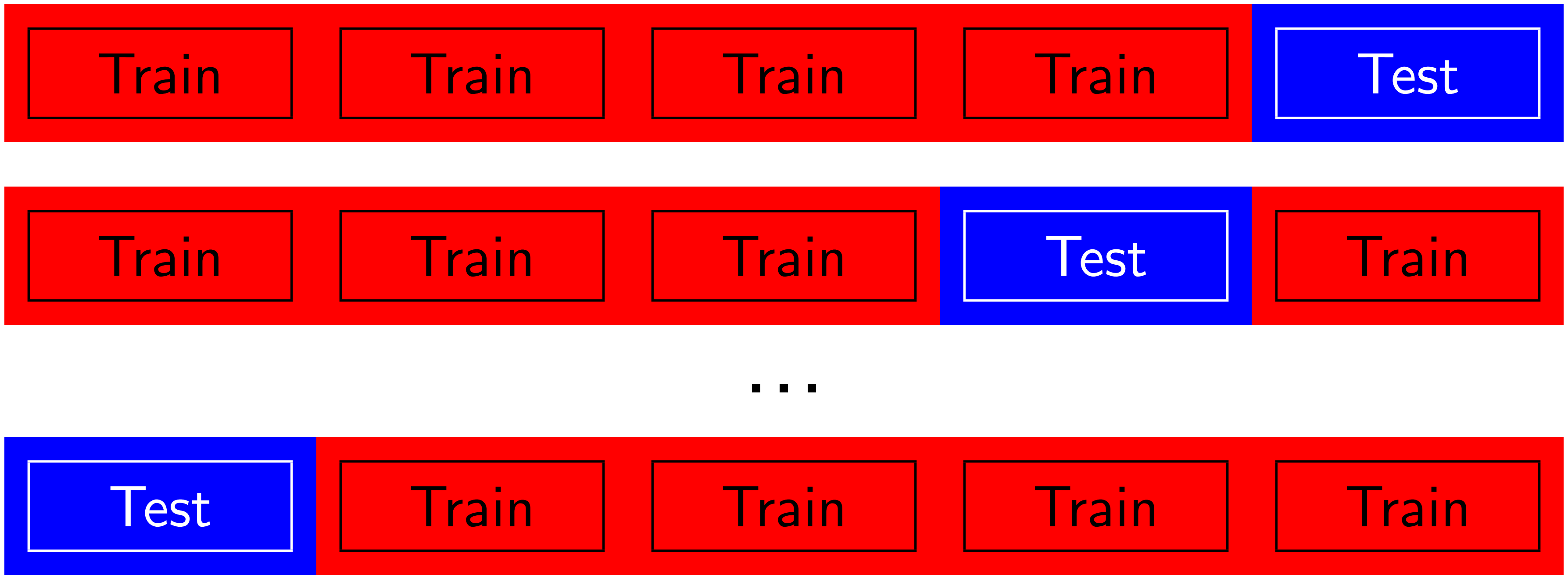
Cross-validation

Efficient way to estimate the error rate:



Cross-validation

Efficient way to estimate the error rate:



K-fold cross-validation

1. Given: a training set $\mathcal{D}$ of size N
2. Divide $\mathcal{D}$ into K disjoint subsets $\mathcal{D}_1, \dots, \mathcal{D}_K$ of equal size N/K
3. For each $\mathcal{D}_i$:
   Train a classifier on $\mathcal{D}$ without $\mathcal{D}_i$
   Compute prediction error on $\mathcal{D}_i$
4. Output the average error

Cross validation estimate of risk

Indexing function $\kappa : \{1, \ldots, N\} \mapsto \{1, \ldots, K\}$

Let $c^{-k}(x)$ be classifier fitted with k-th part of data removed.

The **cross validation estimate** $R_{cv}[c]$ **of risk** $R[c]$ is defined by

$$R_{cv}[c] \;=\; \frac{1}{N}\sum_{i=1}^{N} l(\,\mathbf{x}^{(i)},\; c^{-\kappa(i)}(\mathbf{x}^{(i)}),\; y_i).$$

Cross validation estimate of risk

Indexing function $\kappa : \{1, \ldots, N\} \mapsto \{1, \ldots, K\}$

Let $c^{-k}(x)$ be classifier fitted with k-th part of data removed.

The **cross validation estimate** $R_{cv}[c]$ **of risk** $R[c]$ is defined by

$$R_{cv}[c] = \frac{1}{N} \sum_{i=1}^{N} l(\,\mathbf{x}^{(i)},\; c^{-\kappa(i)}(\mathbf{x}^{(i)}),\; y_i).$$

$$R_{emp}[x] = \frac{1}{N} \sum_{i=1}^{N} l(\,\mathbf{x}^{(i)},\; c(\mathbf{x}^{(i)}),\; y_i)$$

A pitfall in model selection

Very optimistic cross-validation results are achieved by

1. selecting the most discriminative genes on the whole dataset,
2. performing cross-validation on reduced profiles.

What goes wrong?

A pitfall in model selection

Very optimistic cross-validation results are achieved by

1. selecting the most discriminative genes on the whole dataset,

2. performing cross-validation on reduced profiles.

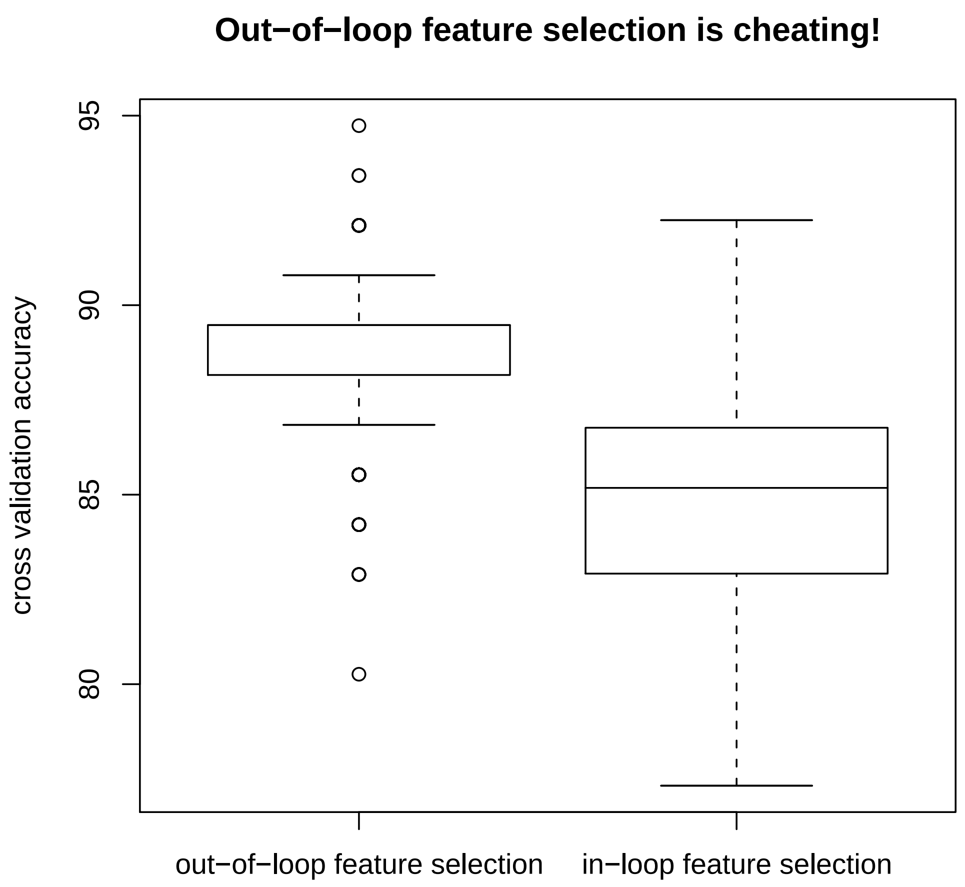
What goes wrong?

For honest error estimates, the test sets in cross-validation have to remain untouched.

But here **test sets were already used for feature selection!**

This makes the error estimate overoptimistic [9, 1].

In-loop versus out-of-loop



One more complication

To select between different models we do 10-fold cross validation with in-loop feature selection. We choose the best model.

Is the CV performance of this model a **honest estimate of generalization performance** for model assessment?

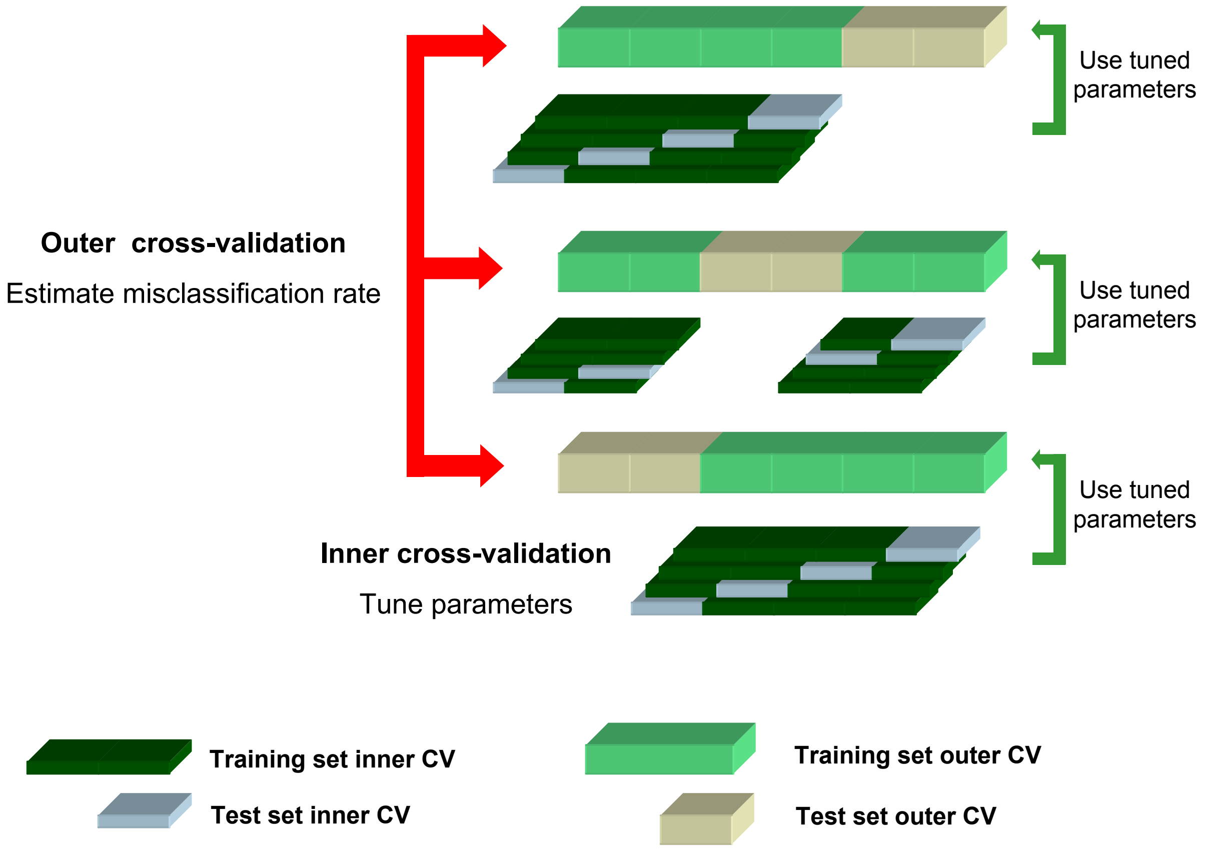
One more complication

To select between different models we do 10-fold cross validation with in-loop feature selection. We choose the best model.

Is the CV performance of this model a **honest estimate of generalization performance** for model assessment?

No, it will be overoptimistic,
because we optimized over all models.

Nested-loop cross validation



[3, 7]

Clever methods of overfitting [5]

General overfitting:

over-representing the performance of systems.

Traditional overfitting: Train a complex predictor on too-few examples.

Clever methods of overfitting [5]

General overfitting:

over-representing the performance of systems.

Traditional overfitting: Train a complex predictor on too-few examples.

Parameter tweak overfitting: Use a learning algorithm with many parameters. Choose the parameters based on the test set performance. For example, choosing the features so as to optimize test set performance can achieve this.

Clever methods of overfitting [5]

Human-loop overfitting: Use a human as part of a learning algorithm and don't take into account overfitting by the entire human/computer interaction.

Clever methods of overfitting [5]

Human-loop overfitting: Use a human as part of a learning algorithm and don't take into account overfitting by the entire human/computer interaction.

Data set selection: Chose to report results on some subset of datasets where your algorithm performs well.

Clever methods of overfitting [5]

Human-loop overfitting: Use a human as part of a learning algorithm and don't take into account overfitting by the entire human/computer interaction.

Data set selection: Chose to report results on some subset of datasets where your algorithm performs well.

Old datasets: Create an algorithm for the purpose of improving performance on old datasets.

Clever methods of overfitting [5]

Human-loop overfitting: Use a human as part of a learning algorithm and don't take into account overfitting by the entire human/computer interaction.

Data set selection: Chose to report results on some subset of datasets where your algorithm performs well.

Old datasets: Create an algorithm for the purpose of improving performance on old datasets.

Overfitting by review: 10 people submit a paper to a conference. The one with the best result is accepted.

Interpretation of results

Is the predictive signature unique?

Typical scenario:

1. You select a number of genes (from all the genes on the microarray) and find that they support a well generalizing classifier.
2. You ask your favorite biologist to make a story out of the gene list.
3. Usually some interesting genes are found.

Is the predictive signature unique?

Typical scenario:

1. You select a number of genes (from all the genes on the microarray) and find that they support a well generalizing classifier.
2. You ask your favorite biologist to make a story out of the gene list.
3. Usually some interesting genes are found.
4. **Is this gene set unique?**
   Are there other sets working as well?
   Do the genes tell us something about the disease causes?

An experiment by Ein-Dor et al. [2]

Data from single experiment (**van't Veer *et al.*, 2002**) on breast cancer patients. Consists of 96 samples with 5852 genes. Van't Veer *et al.* randomly split the patients into training set (77) and test set (19).

They found the **70 genes most highly correlated** with disease outcome to form a **predictive signature**.

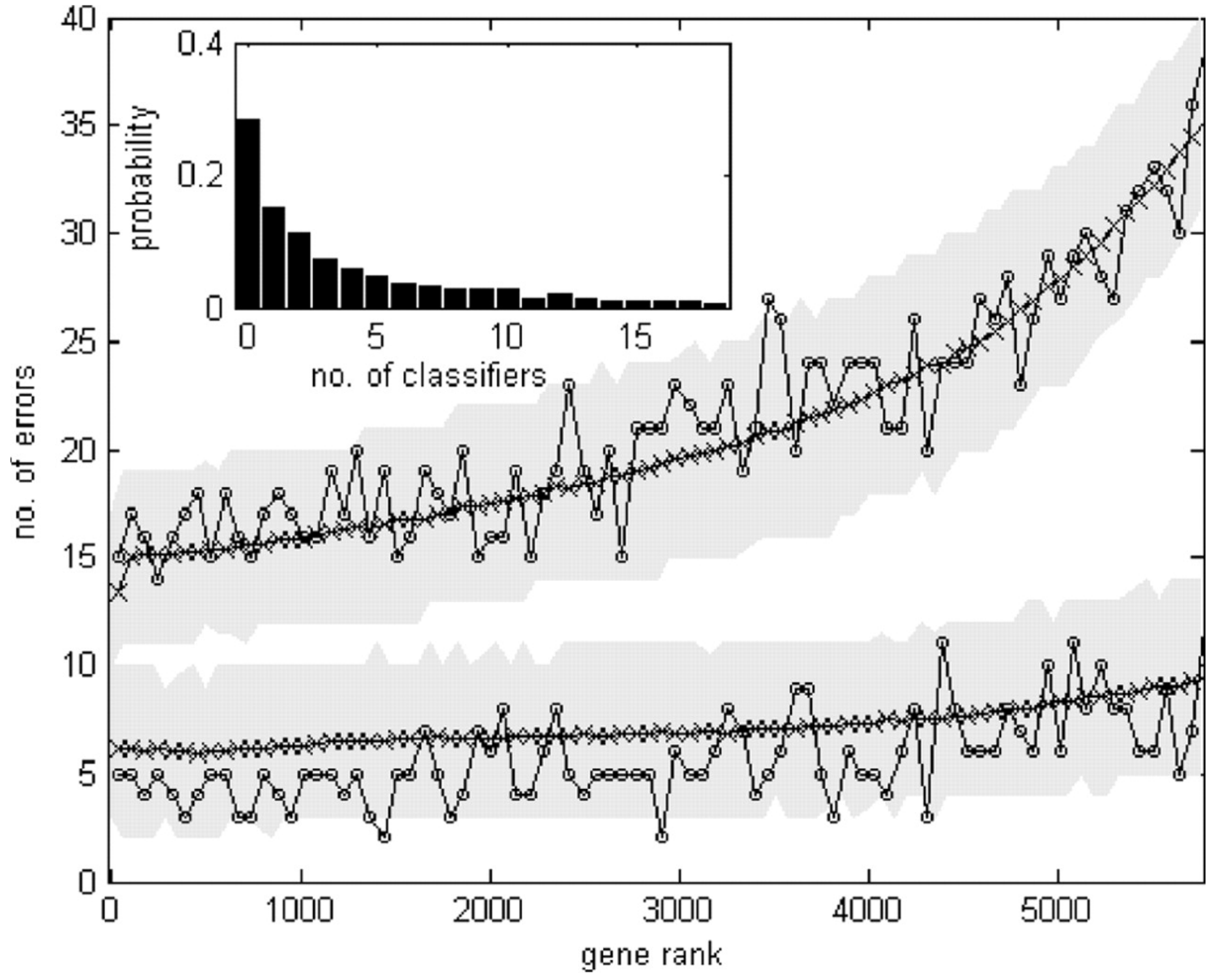
An experiment by Ein-Dor et al. [2]

Data from single experiment (**van't Veer *et al.*, 2002**) on breast cancer patients. Consists of 96 samples with 5852 genes. Van't Veer *et al.* randomly split the patients into training set (77) and test set (19).

They found the **70 genes most highly correlated** with disease outcome to form a **predictive signature**.

Ein-Dor *et al.* build a set of **classifiers on consecutive groups of 70 genes** found on 1000 random partitionings of the data.

Many predictive gene sets



[2]

The message

Why is there no overlap between predictive gene sets?

Lack of agreement could be attributed to different chips, different methods of sample preparation, mRNA extraction, analysis of data, genuine differences between patients (tumor grade, stage, ...).

But even without these sources of variations, **the biological signal is widely spread!**

There is no golden needle hidden!

Interpreting gene lists

Why NOT to do it:

1. to find new insights into biology

2. to find the cause of the disease

For these tasks, do testing! Which has it's own problems: see the talk by Stephane Robin on *Finding differential genes and FDR*.

Why to do it:

Additional reassurance that the model makes biological sense.

Top-down and bottom-up

Message: **Don't hope for top-down approaches to work!**

To get an interpretable classifier, better try **bottom-up approaches**: Select genes from biological knowledge and build classifiers on them.

Example: Nearest Shrunken Centroids on Gene Ontology hierarchy by Lottaz and Spang [6].

Summary

1. **Classification in high dimensions**

   $\longrightarrow$ a fight against overfitting

2. **Discriminant Analysis**

   $\longrightarrow$ Gaussian assumption, feature selection

3. **Support vector machines**

   $\longrightarrow$ Maximal margin hyperplanes, non-linear similarity measures

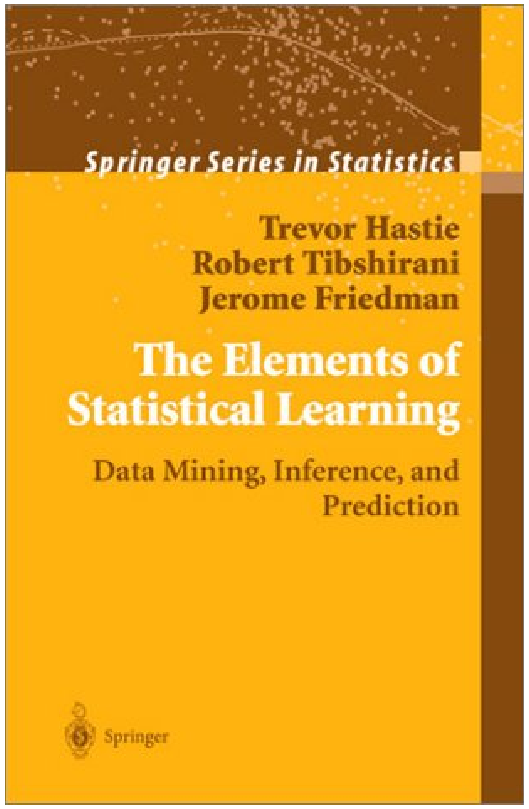
4. **Model selection and assessment**

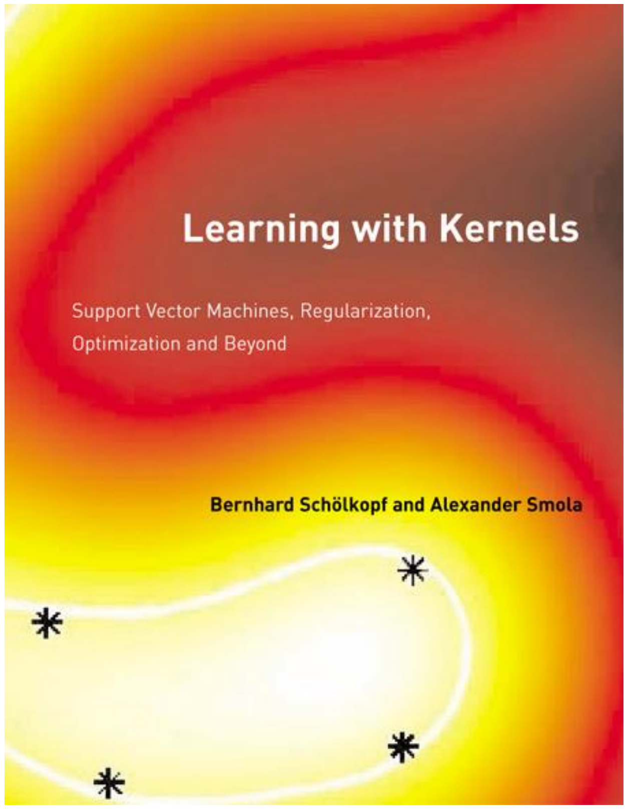
   $\longrightarrow$ Traps and pitfalls, or: How to cheat.

5. **Interpretation of results**

   $\longrightarrow$ what do classifiers teach us about biology?

Recommendations


Springer Series in Statistics
Trevor Hastie
Robert Tibshirani
Jerome Friedman
The Elements of Statistical Learning
Data Mining, Inference, and Prediction
Springer


Learning with Kernels
Support Vector Machines, Regularization, Optimization and Beyond
Bernhard Schölkopf and Alexander Smola

Software for microarray analysis

www.R-project.org
R is a language and environment for statistical computing and graphics. Free software!

www.bioconductor.org
Bioconductor is open source and open development software project for the analysis and comprehension of genomic data.

Courses in Practical Microarray Analysis

Regularly held courses teach basic techniques of practical gene expression data analysis. For infos go to:

http://compdiag.molgen.mpg.de/ngfn

Topics: Quality control, Data preprocessing and normalization, Identification of differentially expressed genes, Clustering, Classification and molecular diagnosis, Computer lab classes.

Courses are free!

Acknowledgements

Thanks to MIT Press and the authors for making the figures from *Learning with Kernels* available at
http://www.learning-with-kernels.org.

Thanks to Springer and the authors for making the figures from *The Elements of Statistical Learning* available at
http://www-stat-class.stanford.edu/∼tibs/ElemStatLearn/.

Acknowledgements

Thanks to MIT Press and the authors for making the figures from *Learning with Kernels* available at http://www.learning-with-kernels.org.

Thanks to Springer and the authors for making the figures from *The Elements of Statistical Learning* available at http://www-stat-class.stanford.edu/∼tibs/ElemStatLearn/.

Thank you! Questions?

References

[1] Christophe Ambroise and Geoffrey J McLachlan. Selection bias in gene extraction on the basis of microarray gene-expression data. *Proc Natl Acad Sci U S A*, 99(10):6562–6, May 2002.

[2] Liat Ein-Dor, Itai Kela, Gad Getz, David Givol, and Eytan Domany. Outcome signature genes in breast cancer: is there a unique set? *Bioinformatics*, 21(2):171–8, Jan 2005.

[3] S Geisser. The predictive sample reuse method with applications. *Journal of the American Statistical Association*, 70(350):320–328, 1975.

[4] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer, 2001.

[5] John Langford. Clever methods of overfitting, Feb 2005. http://hunch.net/index.php?p=22.

[6] Claudio Lottaz and Rainer Spang. Molecular decomposition of complex clinical phenotypes using biologically structured analysis of microarray data. *Bioinformatics*, 2005. to appear.

[7] Markus Ruschhaupt, Wolfgang Huber, Annemarie Poustka, and Ulrich Mansmann. A compendium to ensure computational reproducibility in high-dimensional classification tasks. *Statistical Applications in Genetics and Molecular Biology*, 3(1):37, 2004.

[8] Bernhard Schölkopf and Alexander J. Smola. *Learning with kernels*. The MIT Press, Cambridge, MA, 2002.

[9] Richard Simon, Michael D Radmacher, Kevin Dobbin, and Lisa M McShane. Pitfalls in the use of DNA microarray data for diagnostic and prognostic classification. *J Natl Cancer Inst*, 95(1):14–8, Jan 2003.