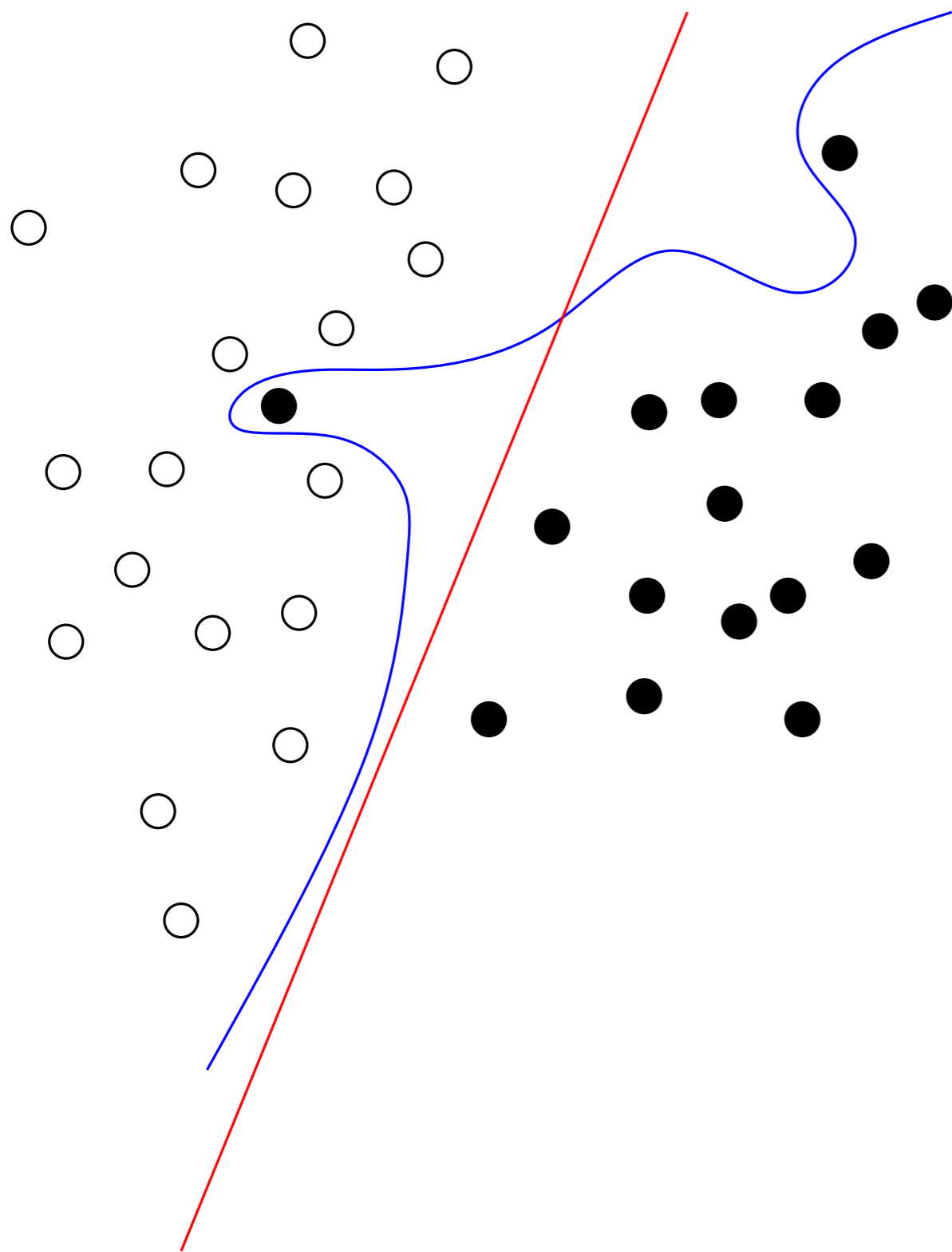


# Plan for the lecture

- introduce another measure to capture the 'complexity' of function classes – the *VC-dimension* (without going into details/proofs  $\Rightarrow$  Kevin might do that)
- extend the linear classifiers as introduced before to cope with 'more complicated' data sets to learn via the so-called 'kernel-trick'
- some examples of successful kernels for certain problems

# Recap of general problem



- we wanted to pick a function  $f$  amongst a class  $\mathcal{F}$  of functions which hopefully not only classifies some given *sample*  $S$  (picked according to some probability distribution) but also future points that come along according to the same prob. dist.
- it seemed reasonable not to pick a very complicated function that fits the sample  $S$  exactly as this might decrease performance on future points that come along
- $\Rightarrow$  need some measure of complicatedness of function class  $\mathcal{F}$  (Jochen: Rademacher complexity)

# The VC-dimension

- VC stands for *Vapnik Chervonenkis* (they apparently came up with this)
- we consider a family of functions  $\mathcal{F} = \{f_\alpha\}$  ( $\alpha$  parametrizes each function)
- assume  $f_\alpha : X \rightarrow \{-1, 1\}$
- if for a given set of  $l$  points and for all possible labellings of these  $l$  points  $(-1, +1, \dots - 1)$  there exists some  $f_\alpha$  which produces exactly this labelling, we say,  $\mathcal{F}$  *shatters* this set of points
- The VC dimension for a function class  $\mathcal{F}$  is the maximum number of points that can be shattered by it
- Important: if  $\mathcal{F}$  has VC dimension  $h$  this does *not* mean that any set of  $h$  points can be shattered by  $\mathcal{F}$

# VC-Dimension: Examples

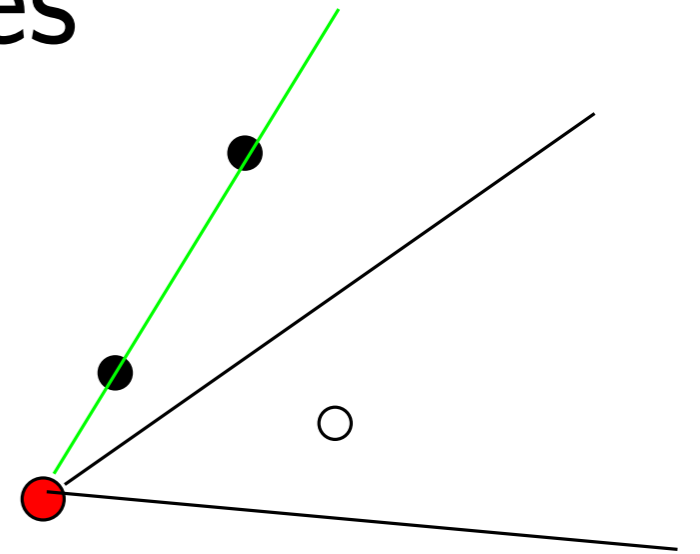
What's the VC dimension  $h$  of:

- $\mathcal{F} = \{f_{s,m}(x,y) = s \cdot \text{sign}'(m \cdot x - y)\}$  ?  
 $\text{sign}'(v) = +1$  if  $v > 0$ ,  $-1$  o.w.  
 $s \in \{-1, +1\}$ ,  $m \in \mathbb{R}$   
 $\Rightarrow h = 2$

but cannot shatter every set of 2 points

- $\mathcal{F} = \{f_{s,m,c}(x,y) = s \cdot \text{sign}'(m \cdot x + c - y)\}$  ?  
 $c \in \mathbb{R}$   
 $\Rightarrow h = 3$

can shatter every set of 2 points but not every set of 3 points



The latter function family can be easily generalized to oriented hyperplanes in  $\mathbb{R}^d$ .

**Theorem:** The set of oriented hyperplanes in  $\mathbb{R}^d$  has VC-dimension  $d + 1$ .

**Proof:** Exercise :-)

# VC-Dimension continued

- VC dimension seems to capture power of function families
- it looks as if  $\#$  of parameters = VC-Dimension
- unfortunately not true; there are very simple one-parameter function families which have even infinite VC-dimension
- example  $\mathcal{F} = \{f_\alpha(x) = \text{sign}'(\sin(\alpha x))\}$ 
  - Set of points  $x_i = 10^{-i}$  can always be shattered by choosing
$$\alpha = \pi \left( 1 + \sum \frac{(1-y_i)10^i}{2} \right)$$
$$y_i \in \{-1, 1\}$$
  - 4 equally spaced points cannot be shattered, though (0010)  
 $\Rightarrow$  exercise

# A bound on the quality of a trained machine

- $R(f_\alpha)$  ... expected error of the learned function/machine
- then we have with probability at least  $1 - \delta$

$$R(f_\alpha) \leq \text{avg. error on test set} + \sqrt{\frac{h(\log(2l/h) + 1) - \log(\delta/4)}{l}}$$

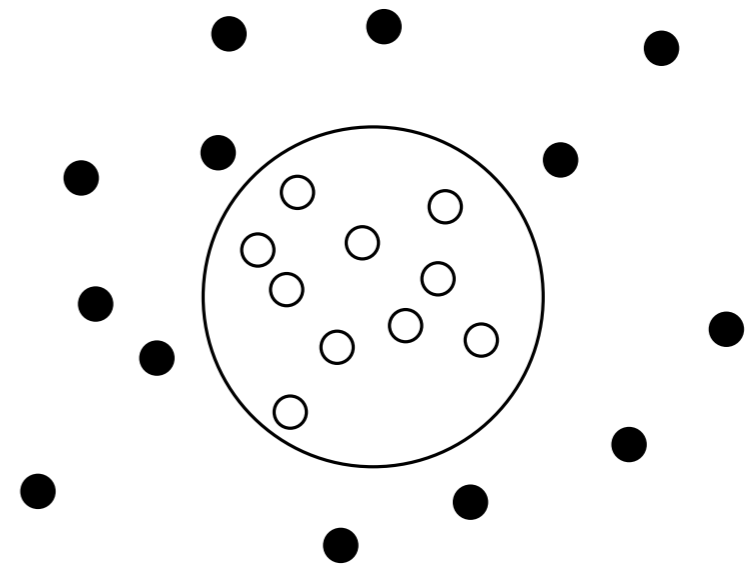
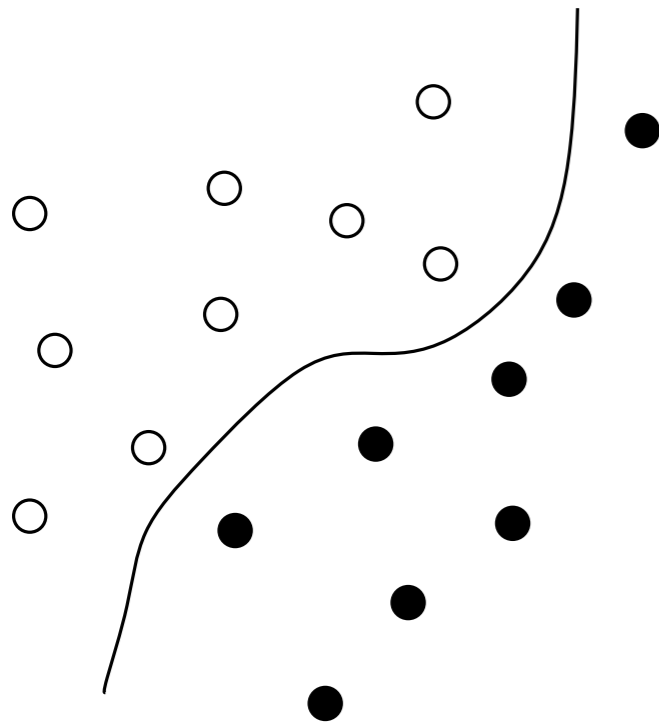
$l$  ... size of the test set

$h$  ... VC dimension of the function class

- an analogue to the bound Jochen presented wrt to the Rademacher complexity of the function family (there was a more subtle dependency on the sample set, though)
- suggests to learn a function which performs reasonably well on the test set but is not too 'complicated' balancing out the two terms above
- VC dimension does not provide full explanation, though; e.g. nearest neighbor classifier has infinite VC dimension but still apparently performs reasonably well in practice

# What to do with data not separable linearly ?

- Some data does not look linearly separable at all



- idea: apply some mapping to the points before trying the linear classifier; e.g. for points separated by  $f : x \mapsto x^3$ , use mapping  $(x, y) \rightarrow (x, y^{1/3})$
- mapping might go to some higher dimension where linear classifiers are also more powerful (in terms of VC dimension) e.g.  $(x, y) \rightarrow (x, y, x^2 + y^2)$
- Kernel methods: do not apply the mapping explicitly, but instead of applying dot products using some *kernel function*

# The kernel method

- we are interested in functions  $k(x, y)$  that replace  $x \cdot y$
- Mercer's theorem characterizes functions  $k(x, y)$  for which there exists a mapping  $\phi$  to some strange space such that  $k(x, y) = \phi(x) \cdot \phi(y)$
- typical kernel methods that can be realized as dot products in some higher dimensional space are
  - polynomials of degree  $d'$ :  $k(x, y) = (x^T y + c)^{d'}$
  - Gaussian functions:  $k(x, y) = e^{-c\|x-y\|^2}$
  - Sigmoid functions  $k(x, y) = \tanh(x^T y + c)$