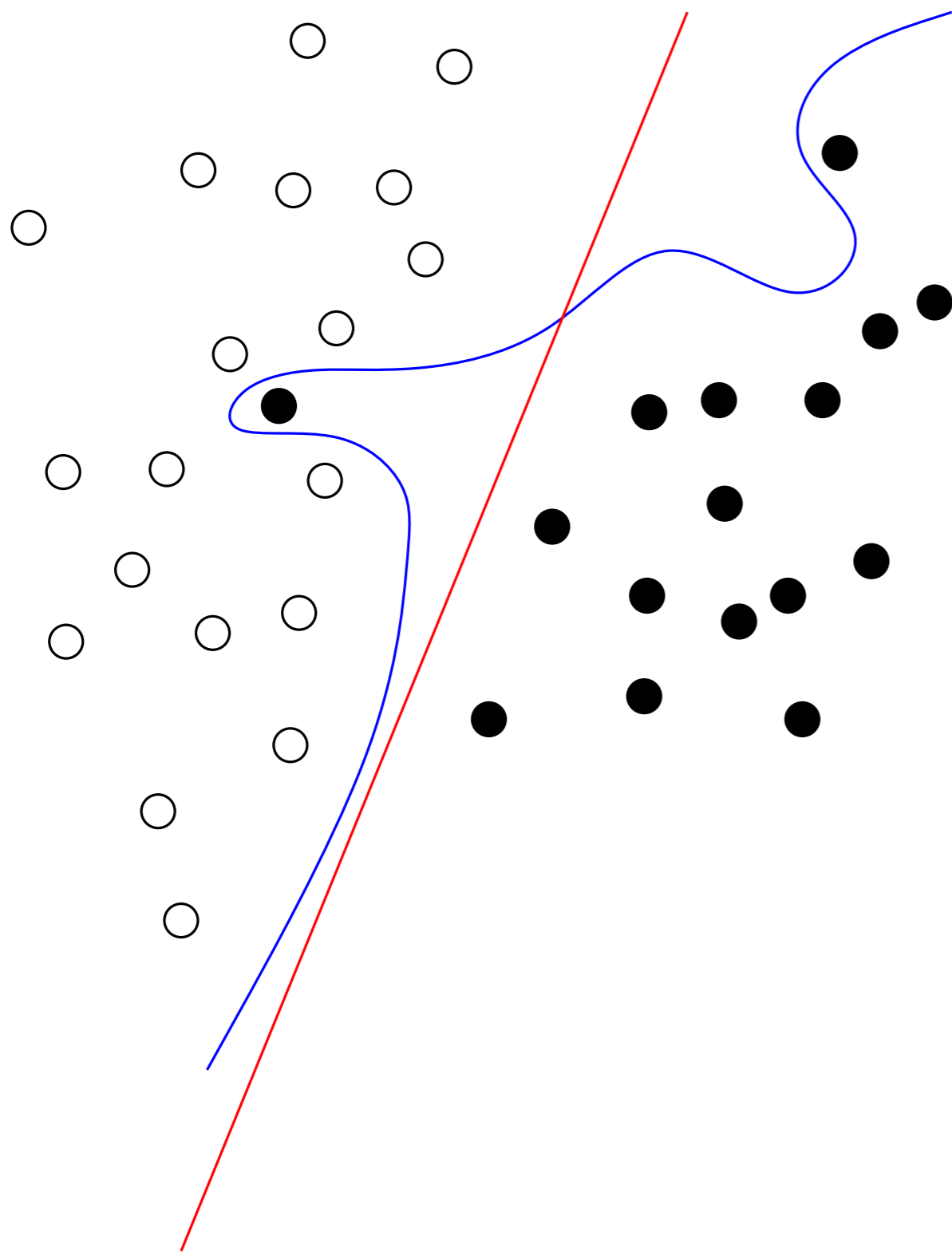


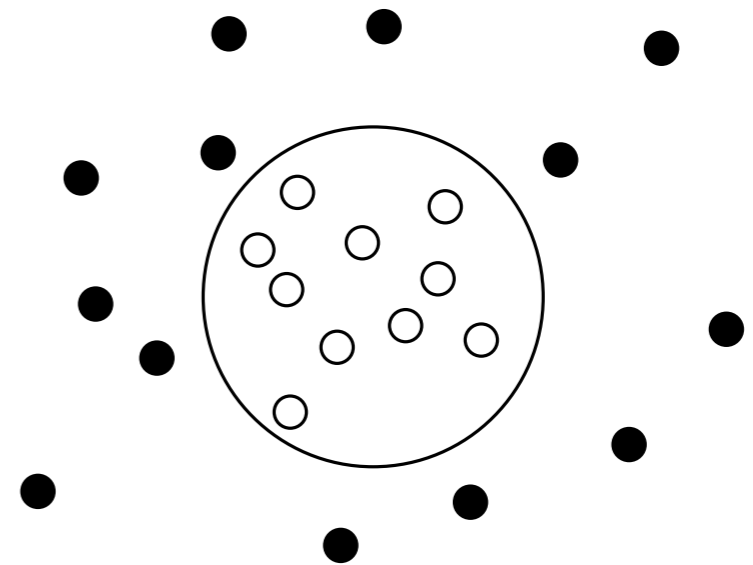
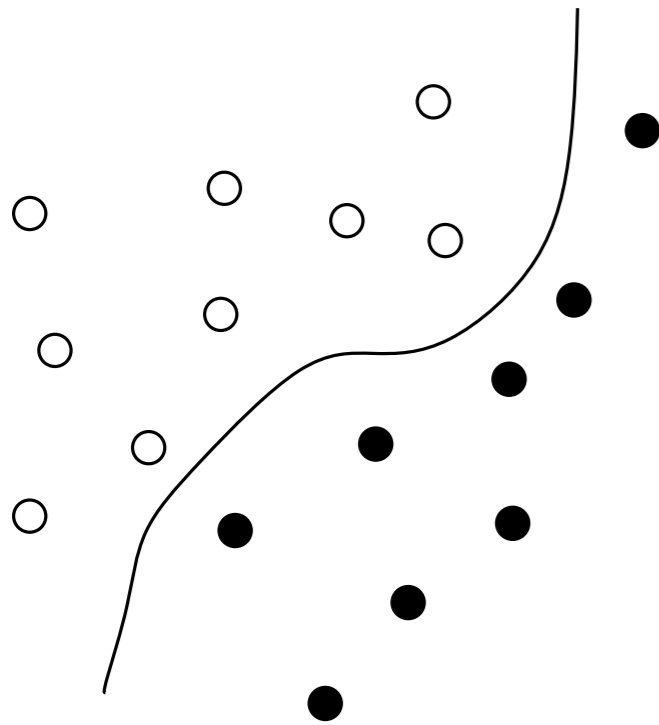
Recap of general problem



- 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)

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 ϕ 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 $k(x, y) = \tanh(x^T y + c)$

Linear Support Vector Machines

- we want to determine a hyperplane of the form $\mathbf{w} \cdot \mathbf{x} + \mathbf{b} = 0$ which separates the $+1$ from the -1 points, maximizing the 'margin', i.e.

$$x_i \cdot \mathbf{w} + \mathbf{b} \geq 1 \text{ for } y_i = +1$$

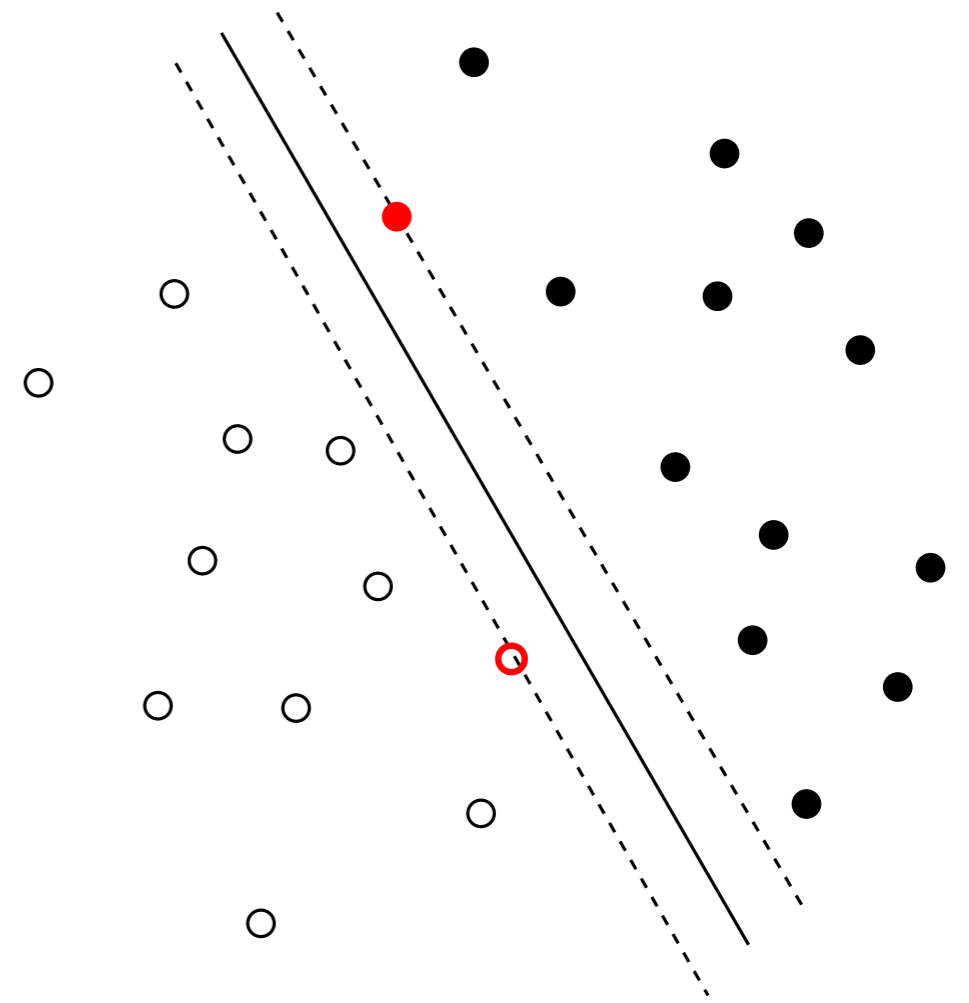
$$x_i \cdot \mathbf{w} + \mathbf{b} \leq -1 \text{ for } y_i = -1$$

\Leftrightarrow

$$\min \|\mathbf{w}\|^2 \text{ such that}$$

$$y_i(x_i \cdot \mathbf{w} + \mathbf{b}) - 1 \geq 0$$

- for x_i with tight constraints, distance to origin is $\frac{|1-b|}{\|\mathbf{w}\|}$, $\frac{|-1-b|}{\|\mathbf{w}\|}$ respectively, so the margin is $2/\|\mathbf{w}\|$



Lagrangian Formulation

- We replace the original problem

$$\min \|\mathbf{w}\|^2 \text{ s.t.} \\ y_i(x_i \cdot \mathbf{w} + \mathbf{b}) - 1 \geq 0$$

(penalizing violations of the constraint) by

- its Lagrangian formulation

$$L_P \equiv \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^l \alpha_i y_i (x_i \cdot \mathbf{w} + b) + \sum_{i=1}^l \alpha_i$$

which needs to be minimized (wrt \mathbf{w}, \mathbf{b} ; requiring that the derivatives wrt α_i vanish and $\alpha_i \geq 0$)

- in the respective dual we *maximize* L_P s.t. the gradients wrt to w, b vanish and $\alpha_i \geq 0$; hence the respective dual is

$$L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \text{ s.t. } \sum \alpha_i y_i = 0, \alpha_i \geq 0$$

- Lagrange multiplier α_i for each training point x_i ; $\alpha_i > 0 \Leftrightarrow x_i$ is support vector
- \mathbf{w} can be computed as $\sum \alpha_i y_i x_i$
- KKT conditions guarantee equality of primal and dual solution and allow to compute b

Non-linear SVMs

- Recall we need to maximize the following:

$$L_D = \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \text{ s.t. } \sum \alpha_i y_i = 0, \alpha_i \geq 0$$

- only operation on the training points is dot product
- we could replace the dot product by any other function $k(x_i, x_j)$ that is "justifiable", e.g.
 $\exists \phi : \mathbb{R}^d \rightarrow \mathcal{H}$ s.t.
 $k(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$
i.e. the result of $k(x_i, x_j)$ equals dot product in some other space \mathcal{H} (feature space)
- we don't even need to know about Φ or \mathcal{H} provided we know that such exists
- from VC-dimension we know that linear classifiers become more powerful in higher dimensions \Rightarrow this kernel trick might allow to separate more complicated data

Using the SVM

- the separating hyperplane unfortunately also lives in that highdimensional space \mathcal{H} !
- there need not exist some \mathbf{w}' living in the original space that maps via Φ to the above
- but we can evaluate the learned SVM using
$$f(x) = \mathbf{w} \cdot \Phi(x) = \sum \alpha_i y_i \Phi(s_i) \cdot \Phi(x) + b = \sum \alpha_i y_i k(s_i, x) + b$$
so no need to compute $\Phi(x)$ explicitly

Characterization of valid kernel functions

- let us first consider a kernel function for which we can explicitly construct the mapping Φ
- data in \mathbb{R}^2 , $k(x, x') = (x \cdot x')^2$
- we need to find $\Phi : \mathbb{R}^2 \rightarrow \mathcal{H}$ such that $(x \cdot y)^2 = \Phi(x) \cdot \Phi(y)$
- let's choose $\mathcal{H} = \mathbb{R}^3$ and $\Phi(x) = \begin{pmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{pmatrix}$
- \mathcal{H} and $\Phi(x)$ are not unique for $k(\cdot)$
- $\Phi(x) = \begin{pmatrix} x_1^2 \\ x_1x_2 \\ x_1x_2 \\ x_2^2 \end{pmatrix}$, $\mathcal{H} = \mathbb{R}^4$
would have worked as well

Mercer's Theorem

- characterizes for which kernel functions \mathcal{H}, Φ exist.
- for $k(x, y)$ there exists a mapping Φ such that
$$k(x, y) = \sum \Phi(x)_i \Phi(y)_i$$
if and only if for any $g(x)$ with
$$\int g(x)^2 dx$$
 finitewe have that $\int k(x, y)g(x)g(y)dx dy \geq 0$
- again example for $k(x, y) = (x \cdot y)^2$ in \mathbb{R}^2
- generalizes to $k(x, y) = (x \cdot y)^p$
- Mercer's Theorem does not reveal anything about \mathcal{H} or Φ

Discussion

- e.g. 16×16 images ($d = 256$) and a degree $p = 4$ polynomial, the dimension of \mathcal{H} is $> 180000000!$
- there are kernels for which Mercer's Theorem does not hold, still they seem to work in practice
- due to the often high dimension of \mathcal{H} , the VC dimension 'justification' seems pointless; generalization performance should be very bad
- some handwaving arguments why things still work out nicely:
 - mapped surface still lives in some sort of low-dimensional subspace
 - looking for the maximum margin hyperplane also seems to help
- in general there is no theory which guarantees good generalization properties of SVMs!

Examples: Polynomial Classifiers

- $k(x, y) = (x \cdot y + 1)^p$
- kernel functions with implicit high-dimensional \mathcal{H} don't seem to perform too badly on linearly separable examples (no overfitting observed)
really high dimension leads to overfitting, though
- high-dimensional kernel really necessary to separate more difficult data

Examples: Radial Basis functions

- $k(x, y) = e^{-\|x-y\|^2/2\sigma^2}$

Examples: Sigmoidal Neural Network

- $k(x, y) = \tanh(\kappa x \cdot y - \delta)$