LECTURE 15

THE KERNEL METHOD

- The proof of Grothendieck's Inequality (previous class) is based on the existence of maps $\phi, \psi : \mathbb{R}^d \to \mathbb{R}^N$ s.t.

$$\langle \phi(u), \psi(v) \rangle = \sin\left(\frac{\pi}{2}\langle u, v\rangle\right) \quad \forall u, v \in \mathbb{R}^d \text{ unit vectors.}$$

↑ realizes nonlinearity.

- This idea leads to the kernel trick in Machine Learning (ML)

Our first example of a ML task:

BINARY CLASSIFICATION

- Want: automatic cancer diagnostics

cell $\in \mathbb{R}^d$ → [computer] → diagnosis: 1 = cancer, −1 = no cancer

Each cell = vector of d parameters, e.g. features of a cell (radius, perimeter, texture, symmetry, ...) $\in \mathbb{R}^d$

$\Rightarrow$ Want $f : \mathbb{R}^d \to \{\pm 1\}$.

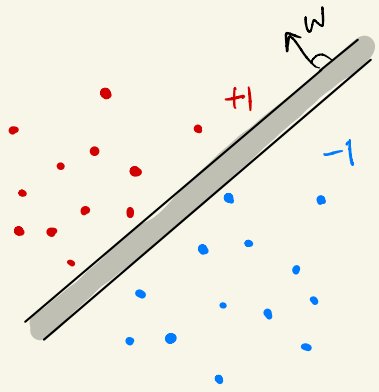
$f(x) = y$

x: test results; y: prediction

- Training data: (x_i, y_i), $i = 1, \dots, n$

= n cells that are already classified by doctors

"Supervised learning"



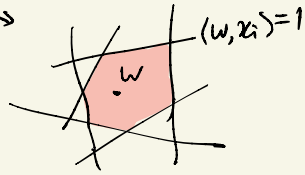
First, assume linear separation:

$$\exists w \in \mathbb{R}^d: \left\{ \begin{array}{ll} \langle w, x_i \rangle > 1 & \text{if } y_i = 1 \\ \langle w, x_i \rangle < -1 & \text{if } y_i = -1 \end{array} \right\} \Leftrightarrow \langle w, x_i \rangle y_i > 1 \ \forall i \quad (*)$$

Learning task: Given data (x_i, y_i), $i = 1, \dots n$ that is linearly separated, find the separator w.

REMARK
Generally, we want to include an offset: $\langle w, x \rangle + b = 0$
but can include it into w:
$\langle w \oplus b, x \oplus 1 \rangle = 0$

$(*)$ = linear constraints on $w \in \mathbb{R}^d$ $\rightarrow$

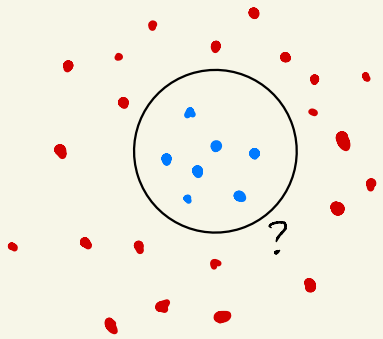


$\Rightarrow$ Linear (feasibility) program.
Tractable. ☺

Prediction: $\forall$ new cell $x \in \mathbb{R}^d$, diagnose it:

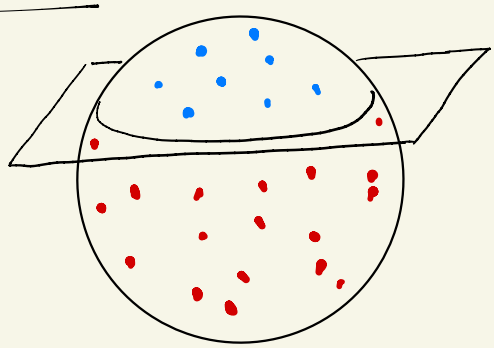
$$f(x) = \left\{ \begin{array}{ll} 1 \text{ (cancer)} & \text{if } \langle w, x \rangle > 1 \\ -1 \text{ (no cancer)} & \text{if } \langle w, x \rangle < -1 \end{array} \right\} = \text{sign} \langle w, x \rangle$$

- "Support Vector Machine" (SVM), "Perceptron"
- What if there is NO linear separation? Transform:



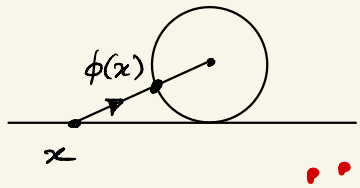
$\mathbb{R}^d$

ϕ
"Feature map"

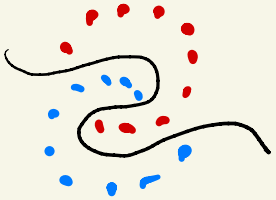


$\mathbb{R}^D$: "feature space"

e.g. stereographic projection



Transformed data is linearly separated.
$\Rightarrow$ Proceed as before.

- What if  ? How do we find ϕ?

NO NEED!

KERNEL TRICK

- Express everything in terms of the inner products $\langle x_i, x_j \rangle$:

WLOG $w = \sum_{j=1}^n \alpha_j x_j \Rightarrow \langle w, x_i \rangle = \sum_{j=1}^n \alpha_j \langle x_i, x_j \rangle$

	$\xrightarrow{\phi}$ transform	
Learn: find $(\alpha_j)_{j=1}^n$: $\sum_{j=1}^n y_i \alpha_j \langle x_i, x_j \rangle > 0 \ \forall i$		Learn: find $(\alpha_j)_{j=1}^n$: $\sum_{j=1}^n y_i \alpha_j \langle \phi(x_i), \phi(x_j) \rangle > 0 \ \forall i$
Predict: $f(x) = \text{sign}\left(\sum_{j=1}^n \alpha_j \langle x, x_j \rangle \right)$		Predict: $f(x) = \text{sign}\left(\sum_{j=1}^n \alpha_j \langle \phi(x), \phi(x_j) \rangle \right)$
← Linear program		still a linear program ↑ ☺

- Denote $K(x,y) := \langle \phi(x), \phi(y) \rangle$ "Kernel" ⇒

Learn: find $(\alpha_j)_{j=1}^n$: $\sum_j y_j \alpha_j K(x_i, x_j) > 0 \quad \forall i$
Predict: $f(x) = \text{sign}\left(\sum_j \alpha_j K(x, x_j) \right)$

(*)

(still a linear program).

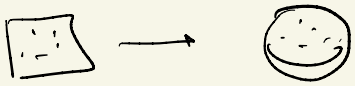
- Forget ϕ. Just choose a suitable kernel $K(\cdot,\cdot)$ and solve (*). "Kernel SVM"

- Ex: $K(x,y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2} \right)$ "Radial Basis Function" Kernel (RBF)

‖ nontrivial

$\langle \phi(x), \phi(y) \rangle$ but we don't need to know ϕ! Just use in (*)

$\|\phi(x)\|_2^2 = K(x,x) = \exp(0) = 1 \Rightarrow$ maps $\mathbb{R}^d \to$ sphere.

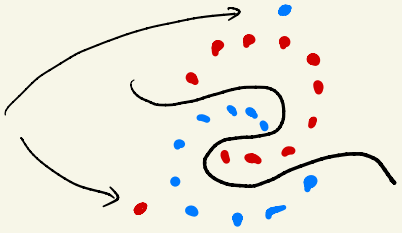


- "Kernelizing" a ML algorithm:

1. Express it in terms of inner products of data $\langle x_i, x_j \rangle$
2. Replace all instances $\langle x, y \rangle \mapsto K(x,y)$ for a suitable choice of the kernel K.

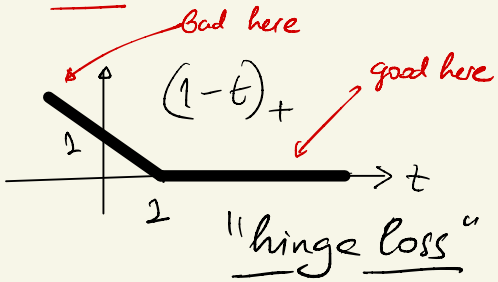
WHAT IF THE DATA IS NOT PERFECTLY SEPARABLE?

Outliers:



?

- **Penalize**: pay penalty $1-\langle w, x_i\rangle y_i$ for each bad pt:



$\Rightarrow$ **loss** $\ell(w) := \sum_{i=1}^{n} (1-\langle w, x_i\rangle y_i)_+$ where

Minimize $\ell(w)$ over $w \in \mathbb{R}^n$ (convex program)

- Furthermore, regularize (for robustness):

loss: $\ell(w) = \sum_{i=1}^{n} (1-\langle w, x_i\rangle y_i)_+ + \lambda \|w\|_2^2$

"Soft-margin SVM"

- And, since data may not be linearly separated, **kernelize** as before:

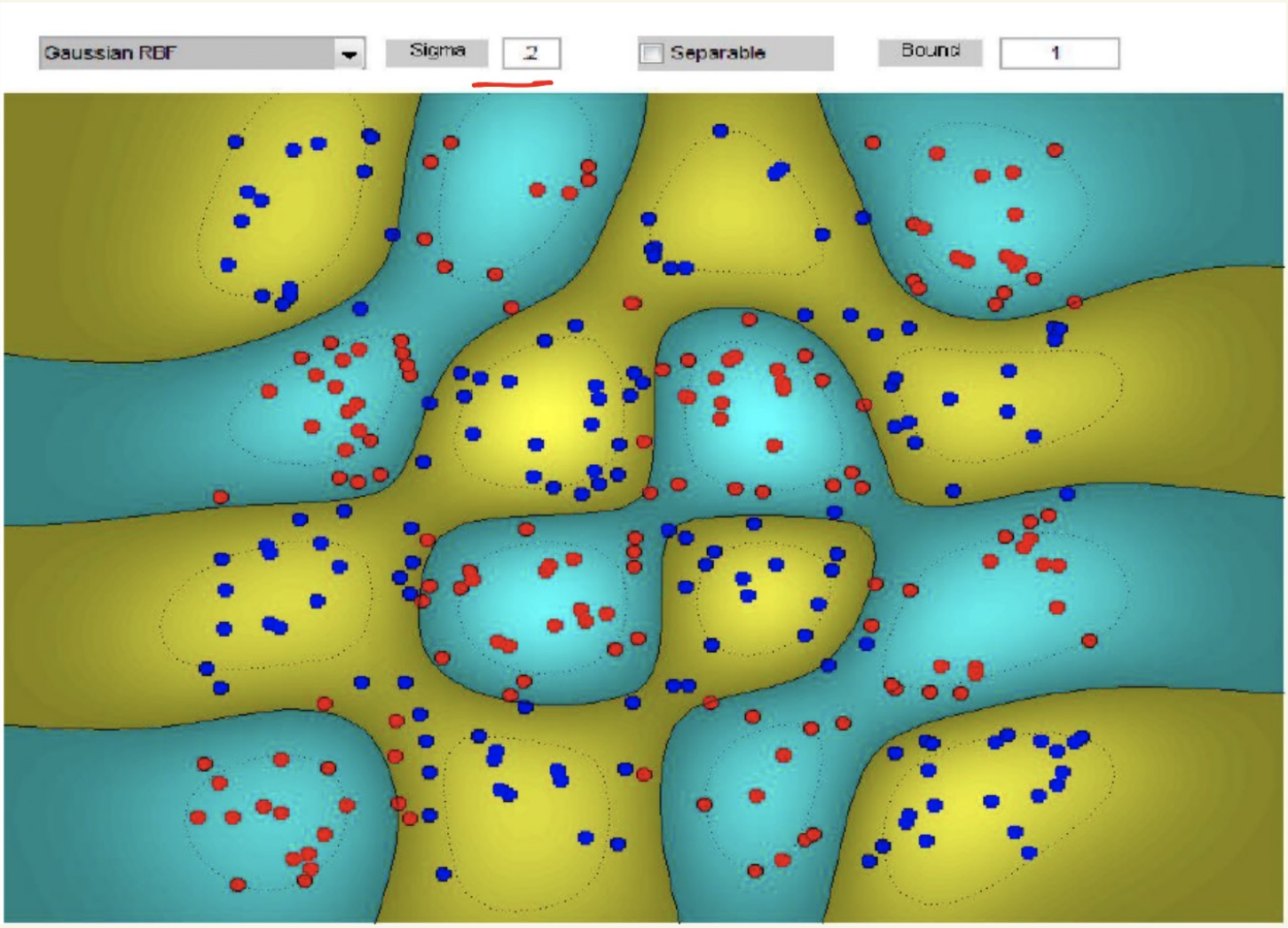
loss: $\ell(\alpha) := \sum_{i=1}^{n} \left(1 - \sum_{j=1}^{n} y_j \alpha_j K(x_i, x_j)\right)_+ + \lambda \sum_{i,j=1}^{n} \alpha_i \alpha_j K(x_i, x_j)$

Learn: find $\alpha = (\alpha_1, \ldots, \alpha_n)$ that minimizes $\ell(\alpha)$.

Predict: for any point x, output $f(x) = \text{sign}\left(\sum \alpha_j K(x_i, x_j)\right)$

"Kernel SVM"

Example: RBF kernel $K(x,y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)$ ↘


Gaussian RBF
Sigma
.2
Separable
Bound
1