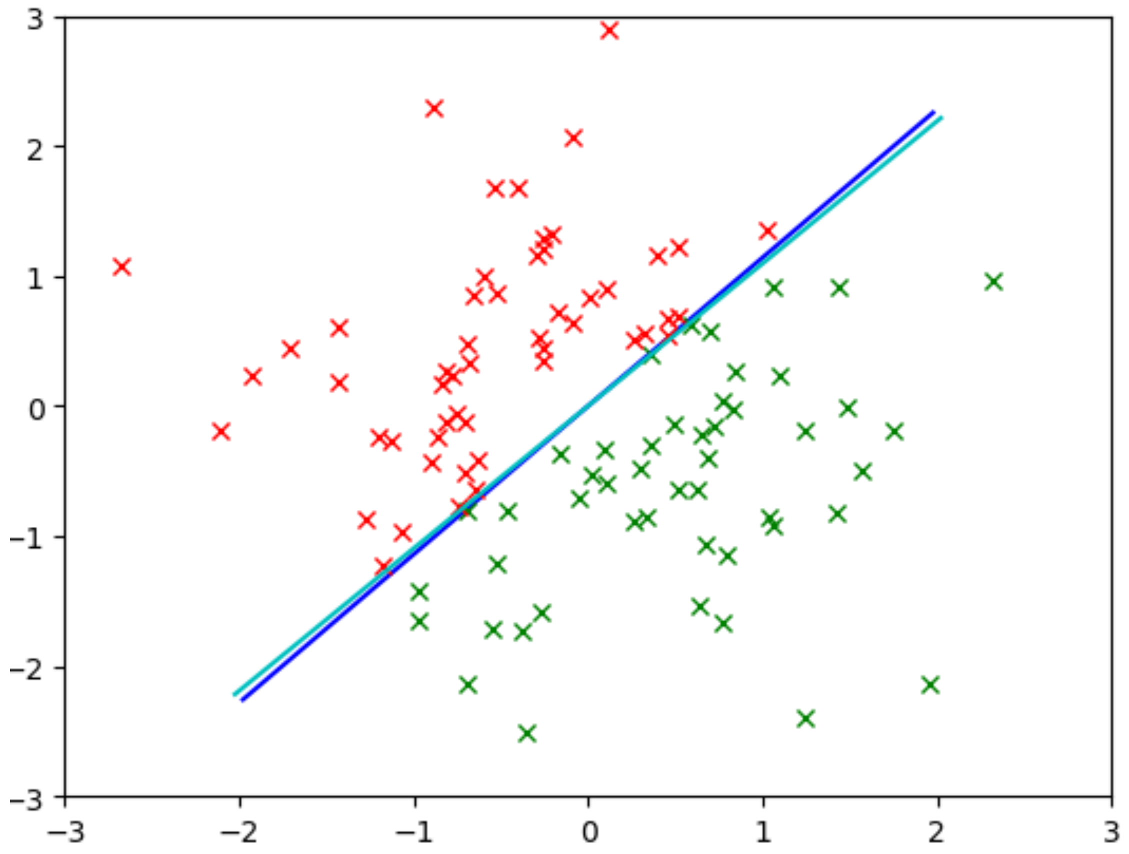
Support Vector Machines

Pascal Fua
IC-CVLab

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Perceptron

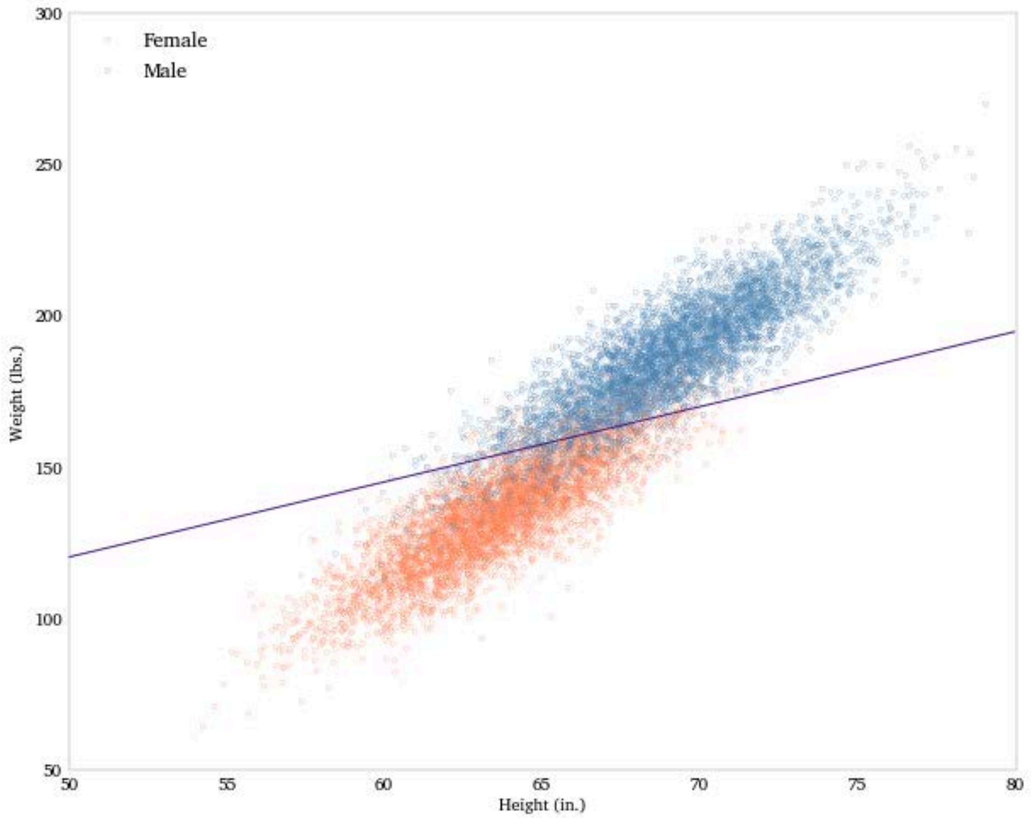


$$y(\mathbf{x};\mathbf{w},w_0) = \begin{cases} 1 & \text{if} \quad \mathbf{w}^T\mathbf{x} + w_0 \geq 0\,, \\ -1 & \text{otherwise.} \end{cases}$$

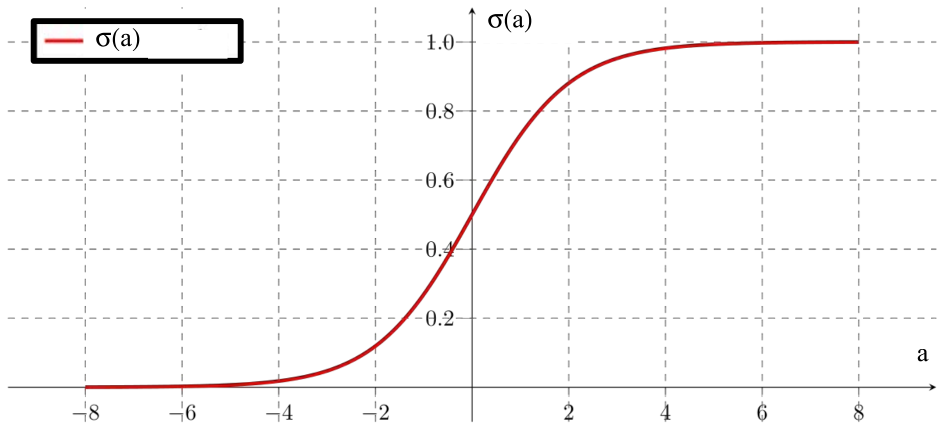
Given the training set $\{(x_n, t_n)_{1 \leq n \leq N}\}$, choose a $\mathbf{w}$ that minimizes

$$E(\mathbf{w}, w_0) = -\sum_{n=1}^{N} (\mathbf{w}^T\mathbf{x}_n + w_0)t_n$$

Logistic Regression



$$y(\mathbf{x};\mathbf{w},w_0) = \sigma(\mathbf{w}^T\mathbf{x}+w_0)$$
$$\approx p(t=1,\mathbf{x})$$

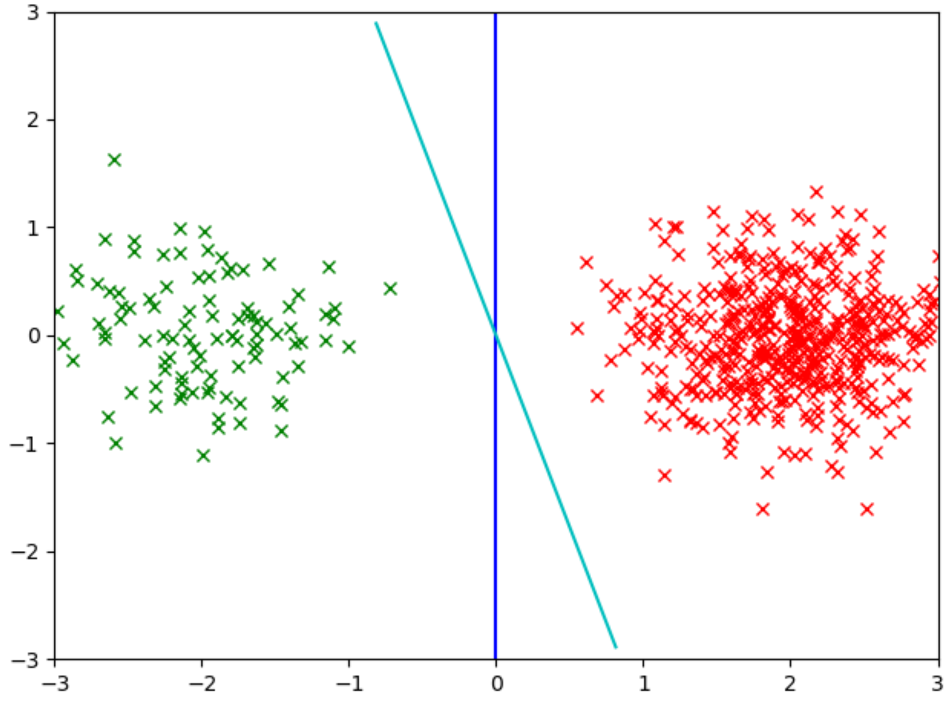


Given the training set $\{(x_n,t_n)_{1\leq n\leq N}\}$, choose a $\mathbf{w}$ that minimizes

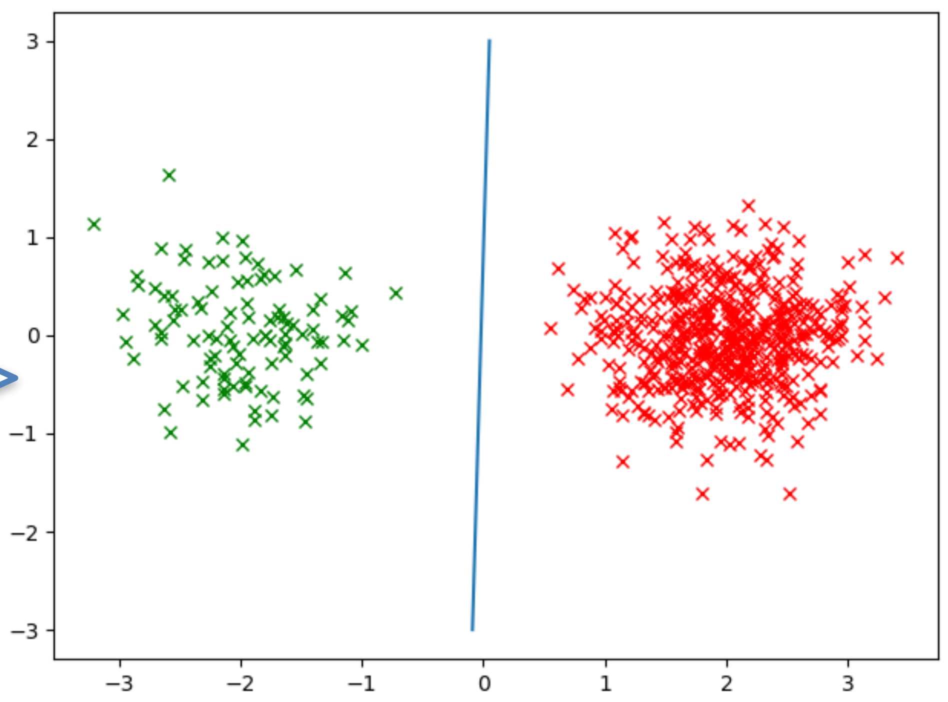
$$E(\mathbf{w},w_0) = -\sum_n \{t_n \ln y_n + (1-t_n)\ln(1-y_n)\} \approx -\ln(p(\mathbf{t}|\mathbf{w},w_0))\ .$$

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Perceptron vs Logistic



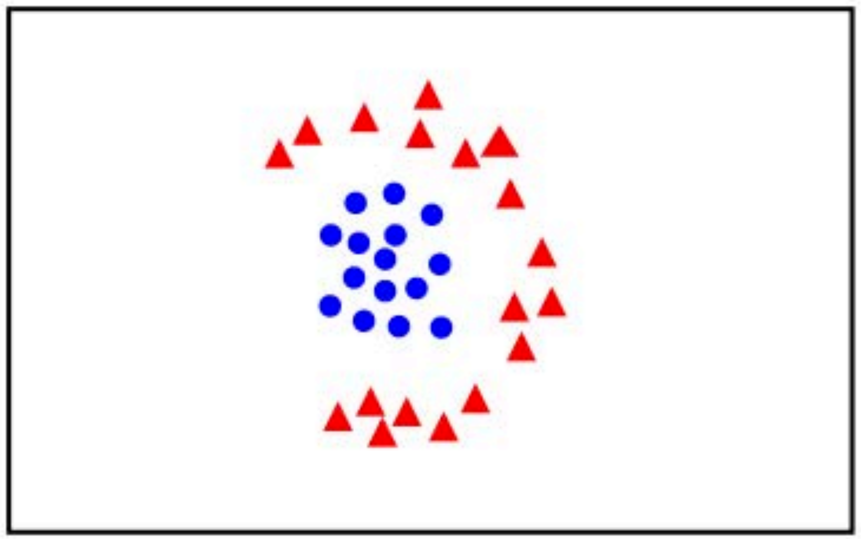
Perceptron



Logistic

But ….

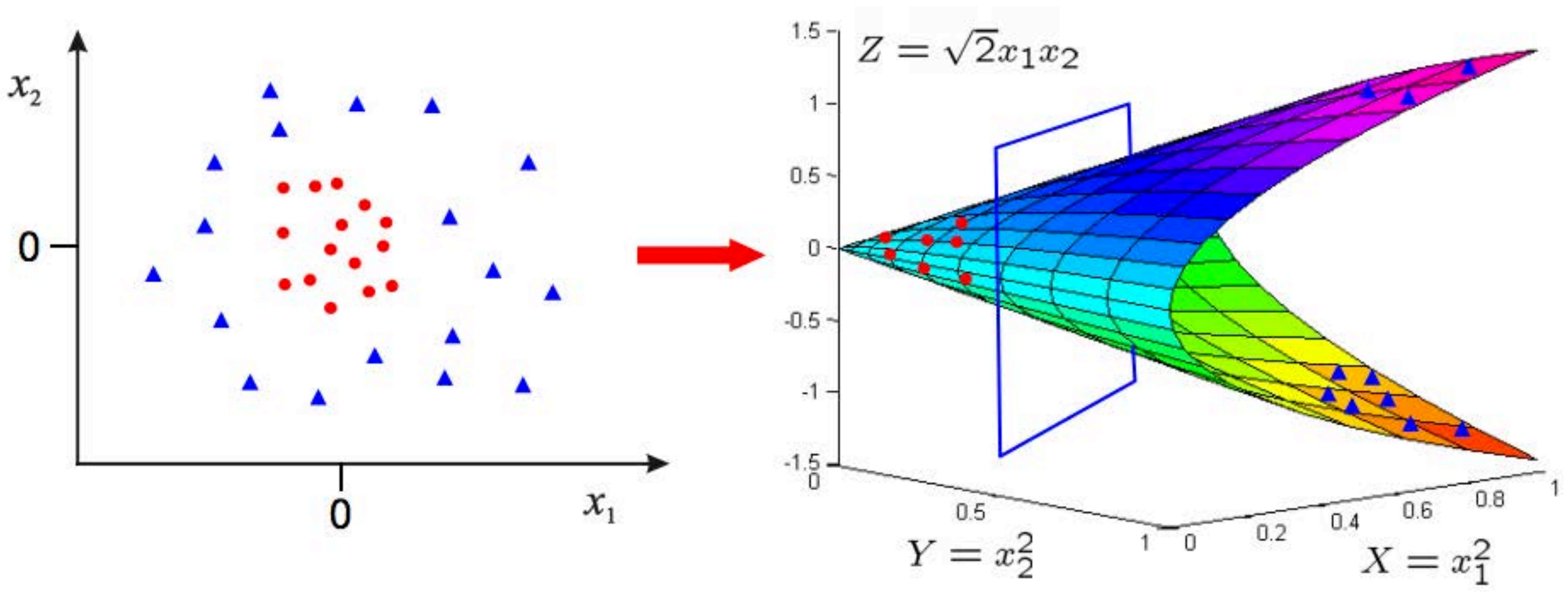
What about Data that is NOT Linearly Separable?



Map it to a higher dimension!

Mapping Data to a Higher Dimension

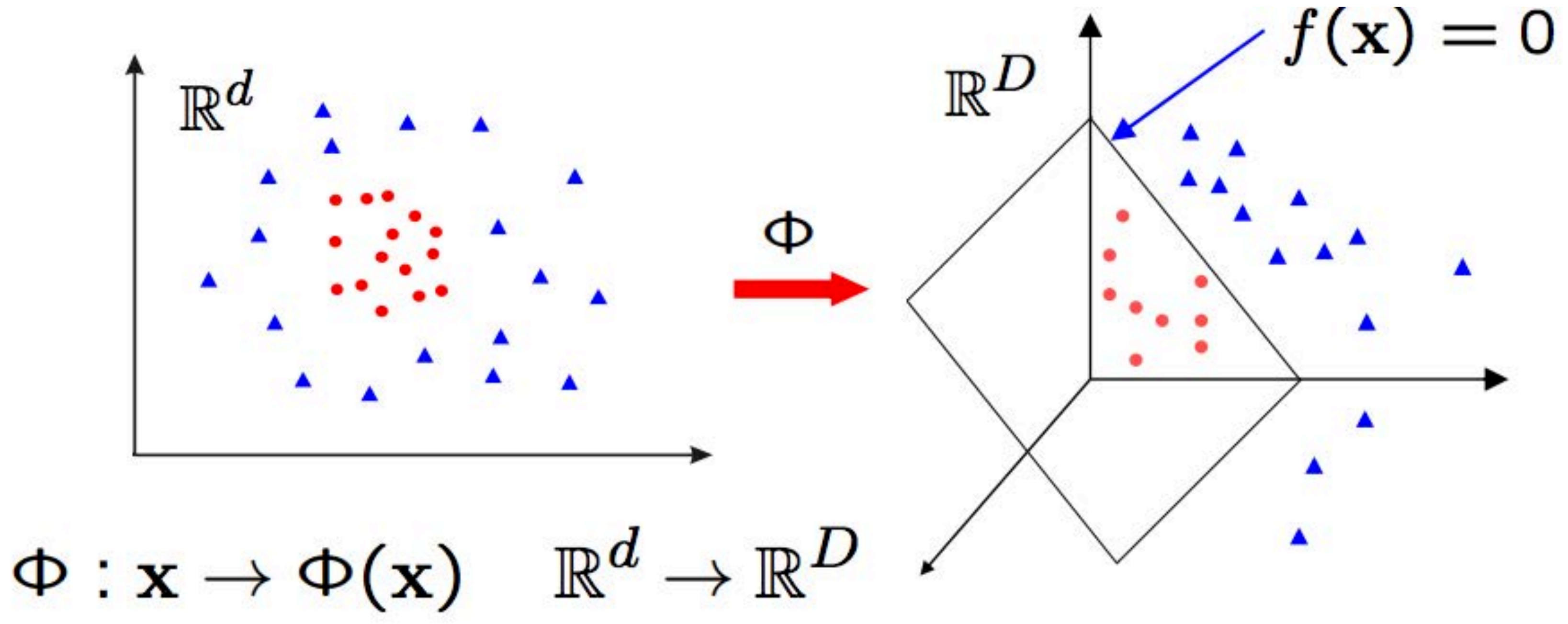
$$\Phi : \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \rightarrow \begin{pmatrix} x_1^2 \\ x_2^2 \\ \sqrt{2}x_1x_2 \end{pmatrix} \qquad \mathbb{R}^2 \rightarrow \mathbb{R}^3$$



- The data is separable in 3D.
- The problem can now be solved using a linear classifier.

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Classification in Feature Space

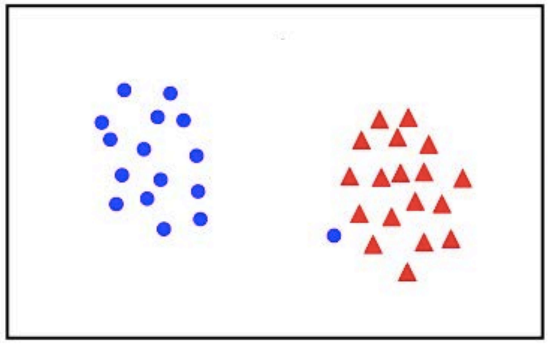

$\mathbb{R}^d$
Φ
$\mathbb{R}^D$
$f(\mathbf{x}) = 0$
$\Phi : \mathbf{x} \rightarrow \Phi(\mathbf{x})$
$\mathbb{R}^d \rightarrow \mathbb{R}^D$

- Map from R^d to R^D
- Learn a linear classifier in R^D

$$f(\mathbf{x}) = \sigma(\mathbf{w}^T \phi(\mathbf{x}) + w_0)$$

$$\phi : \mathbb{R}^d \rightarrow \mathbb{R}^D$$

Potential Exam Questions



(a)

$$y(\mathbf{x};\tilde{\mathbf{w}}) = \begin{cases} 1 & \text{if} \quad \tilde{\mathbf{w}}^T\tilde{\mathbf{x}} \geq 0\,, \\ -1 & \text{otherwise.} \end{cases}$$

$$\tilde{\mathbf{x}} = [1, x_1, ..., x_n]$$

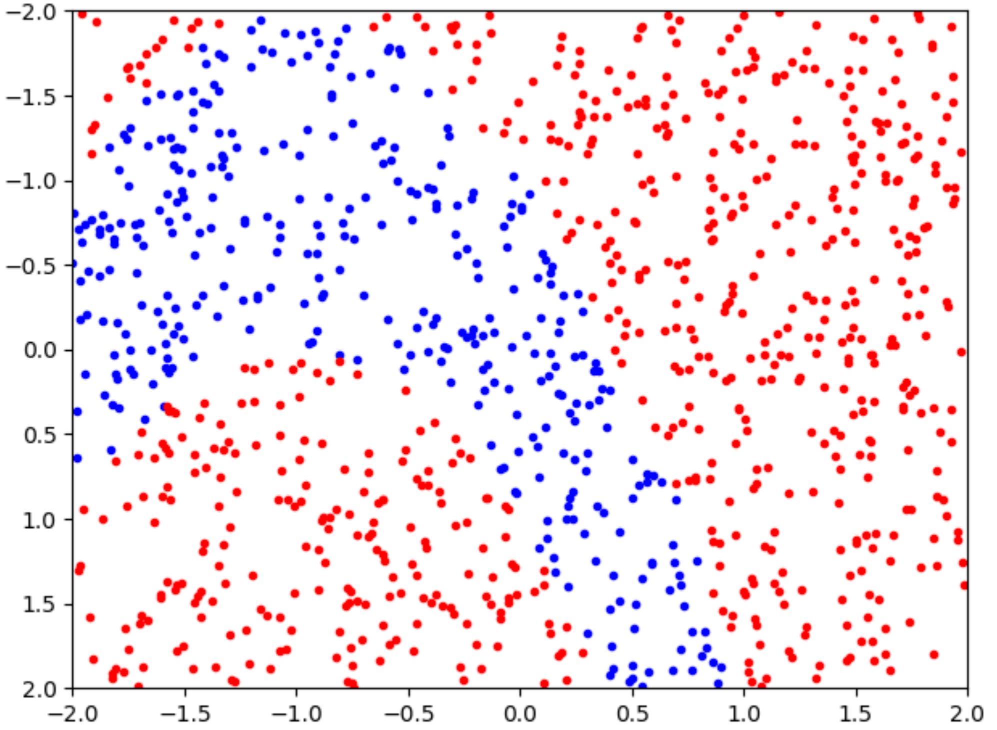
(b)

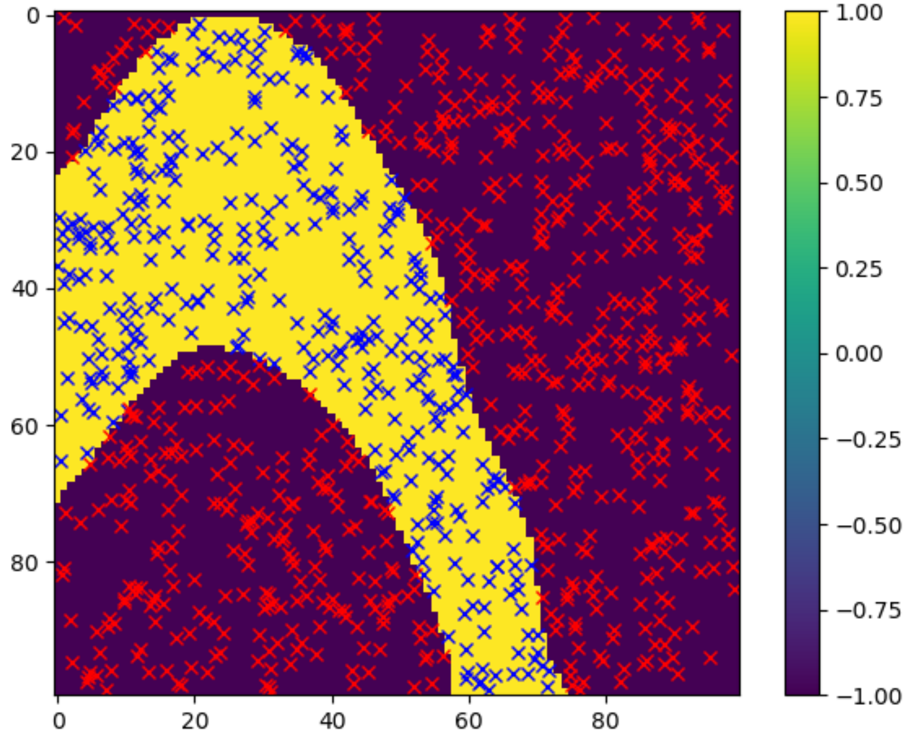
- Would the perceptron work in the case of Fig(a)? Why or why not? What other algorithm could you use? What would be the advantage?
- What's the meaning of the first 1 in the definition of $\tilde{\mathbf{x}}$ in (b)? Why is it needed?

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Classification as Surface Approximation

Rosenbrock:





$$r(x,y) = 100 * (y - x^2)^2 + (1-x)^2$$

$$f(x,y) = \begin{cases} \text{-1} & \text{if } r(x,y) < \text{T} \\ 1 & \text{otherwise} \end{cases}$$

$$f(\mathbf{x}) = \sigma(\mathbf{w}^T \phi(\mathbf{x}) + w_0)$$

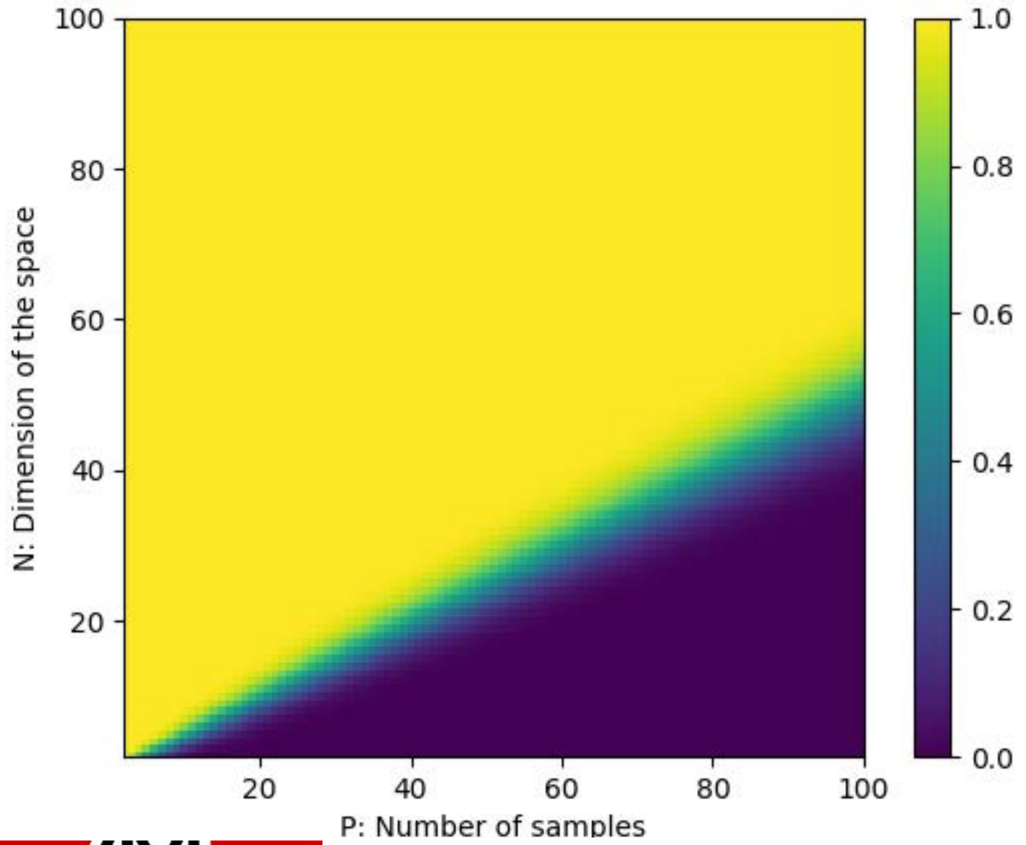
$$\phi : \mathbb{R}^d \rightarrow \mathbb{R}^D$$

- How large should D be? Does it even exist?
- How should we choose ϕ?

Cover’s Theorem

A complex pattern-classification problem, cast in a high-dimensional space nonlinearly, is more likely to be linearly separable than in a low-dimensional space, provided that the space is not densely populated.

Geometrical and Statistical properties of systems of linear inequalities with applications,1965



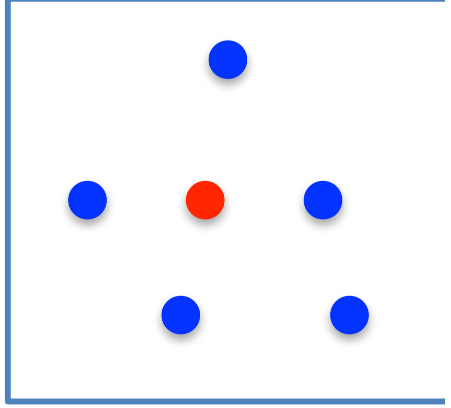
N : Dimension of space

p : Number of samples

$\frac{C(p,N)}{2^p}$: Percentage of separable partitions

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Proof Sketch



$$
\begin{aligned}
C(p+1,N) &= C(p,N) + C(p,N-1) \\
&= C(p-1,N) + 2C(p-1,N-1) + C(p-1,N-2) \\
&= C(p-2,N) + 3C(p-2,N-1) + 3C(p-2,N-2) + C(p-2,N-3) \\
&= \dots \\
&= \binom{p}{0} C(1,N) + \binom{p}{1} C(1,N-1) + \dots + \binom{p}{p} C(1,N-p)
\end{aligned}
$$

$$\forall n \ , \ C(1,n) = 2$$

$$\forall p \ , \ C(p,1) = p+1$$

$$\Rightarrow C(p+1,N) = 2\sum_{i=0}^{N-1} \binom{p}{i}$$

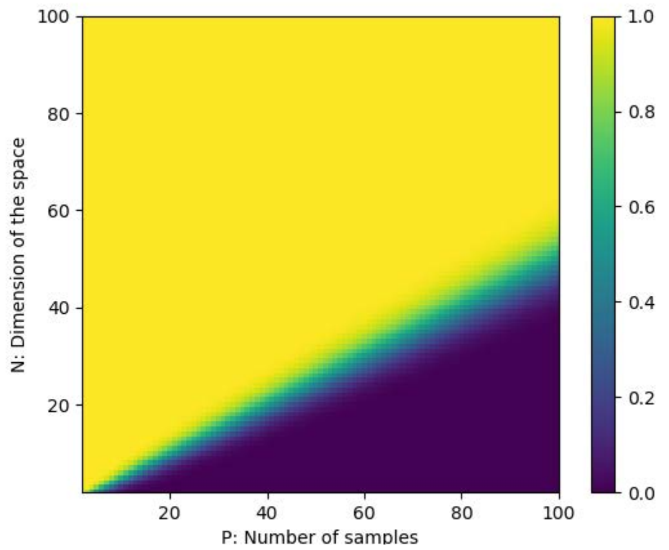
http://www.cns.nyu.edu/~eorhan/notes/covers-theorem.pdf

Recursive Implementation

p\n	N=1	N=2	N=3
p=1	2	2	2
p=2	3	4	4
p=3	4	7	8
p=4	5	11	15

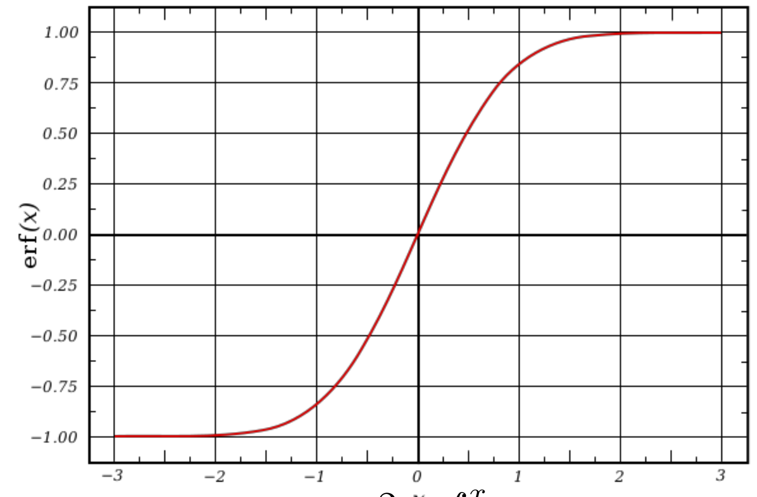
$$\forall n \;,\; C(1,n) = 2$$
$$\forall p \;,\; C(p,1) = p+1$$

$$C(p,N) = C(p-1,N) + C(p-1,N-1)$$

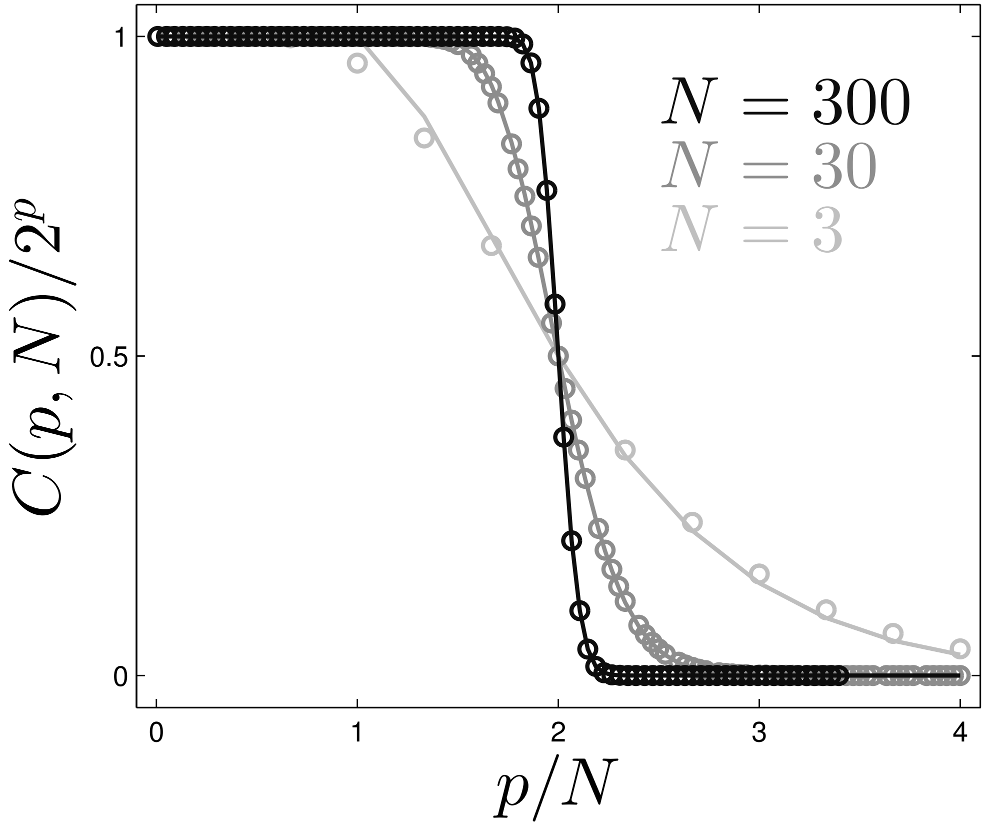


ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Numerical Approximation



$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-t^2)dt$$

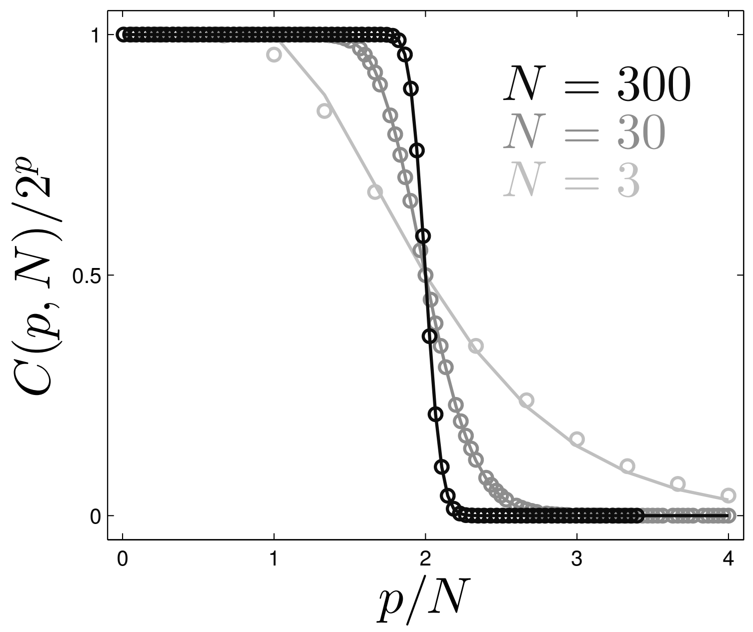


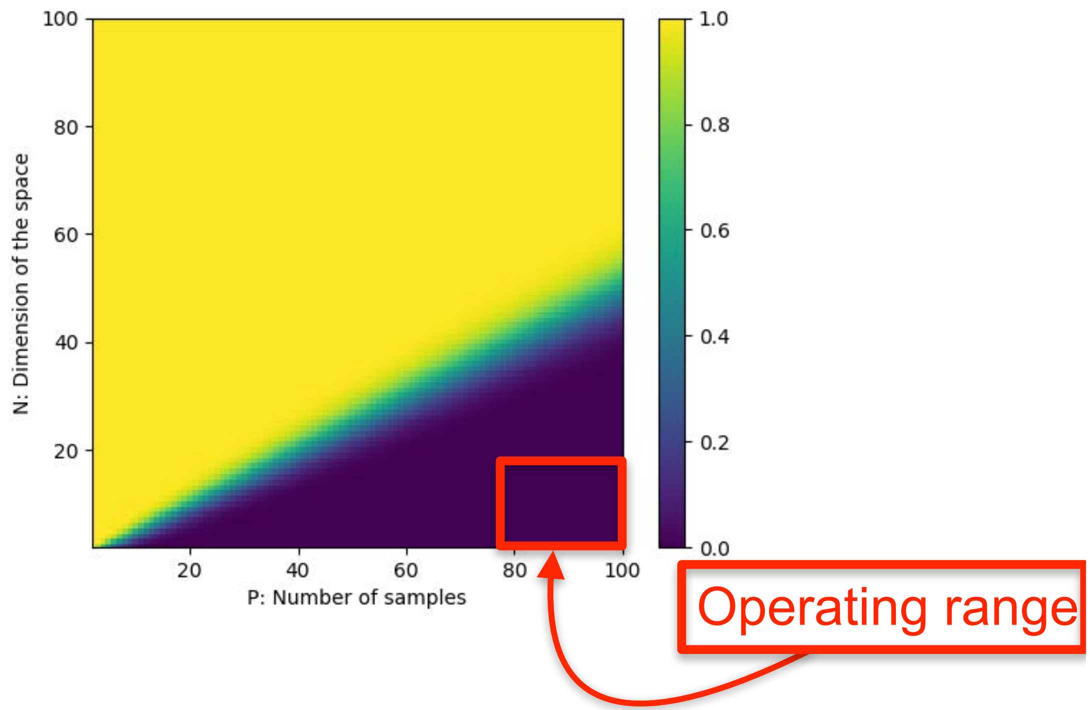
$$C(p+1,N) = 2\sum_{i=0}^{N-1} \left(\begin{array}{c} p \\ i \end{array} \right)$$

$$\frac{C(p,N)}{2^p} \approx 0.5 * (1 + \mathrm{erf}(n\sqrt(2/p) - \sqrt(p/2)))) \text{ when } N \text{ is large.}$$

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

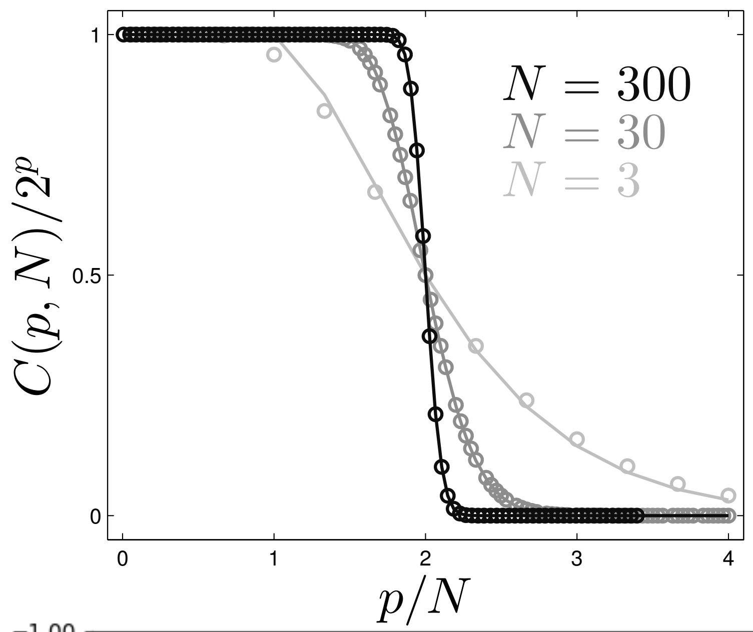
Problem Solved?

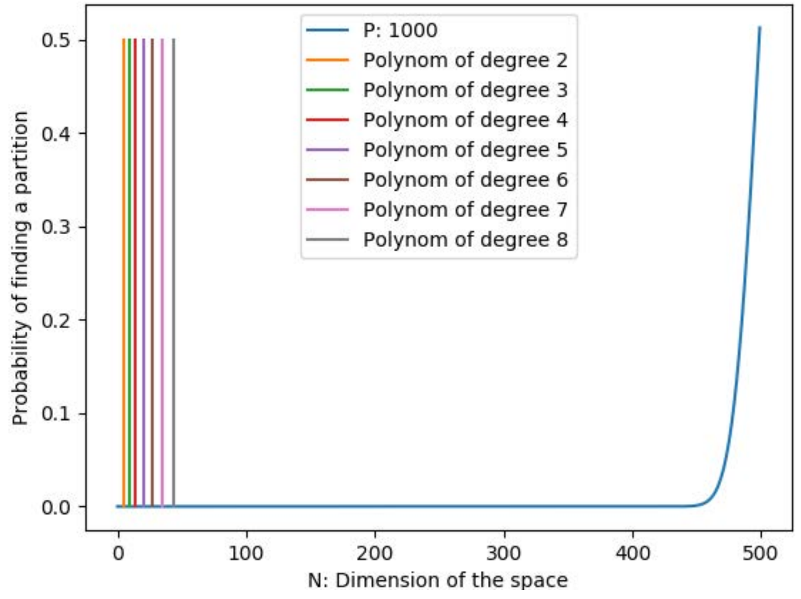


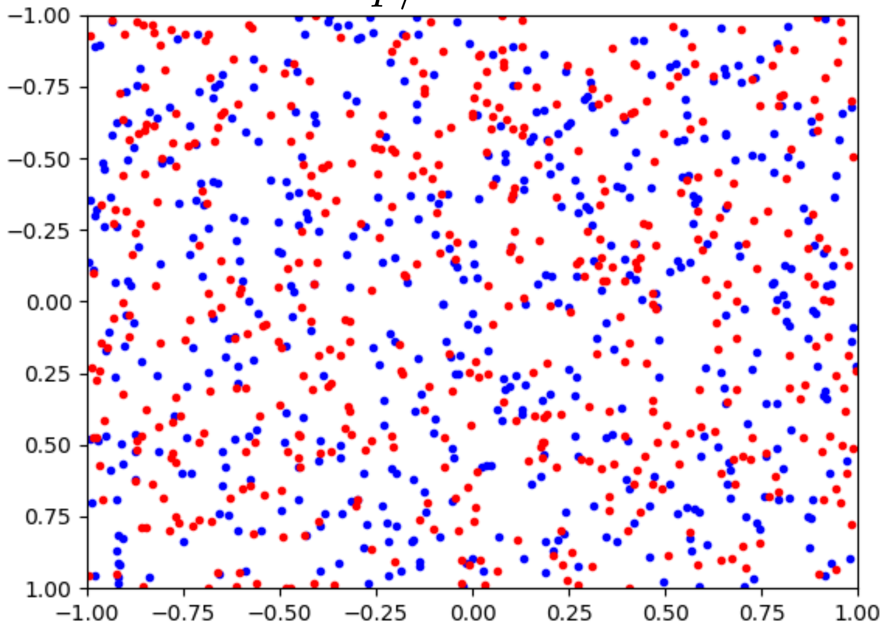


- Facebook or Google deal with BILLIONS of images.
- p and therefore N should be of that magnitude.
- Dealing with matrices of dimension NxN is impractical.

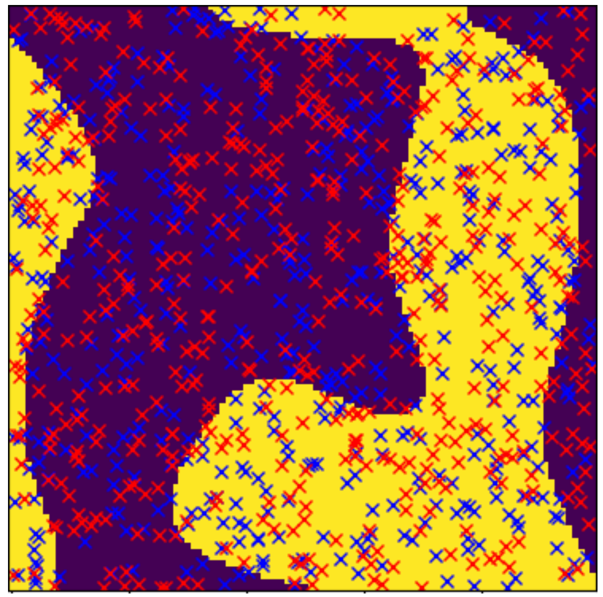
Hopeless Problem?







N = 2, p=1000



N = 44, p=1000

Neither Solved nor Hopeless

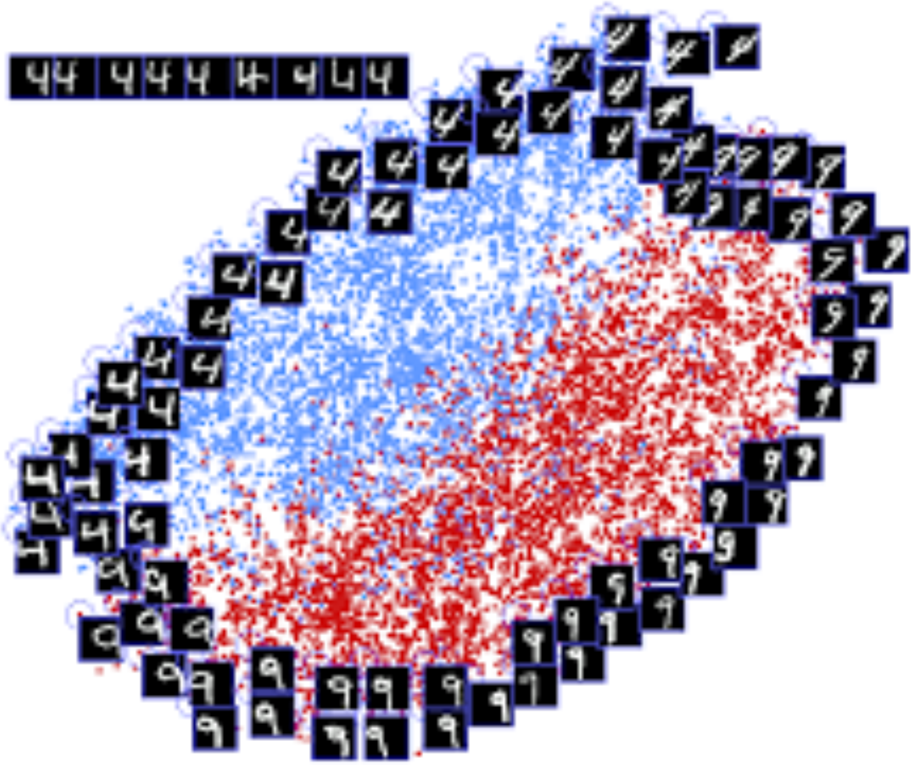
Bad news:

- The ratio of the number of points to the dimension must be less than 2.
- The dimension must be huge for large databases.
- As the dimension increases, the boundaries become increasingly irregular and sensitive to noise.

Good news:

- The world is structured and the points we want to classify are NOT randomly distributed.
- We can compute feature vectors that are “close” for objects that belong to the same class.

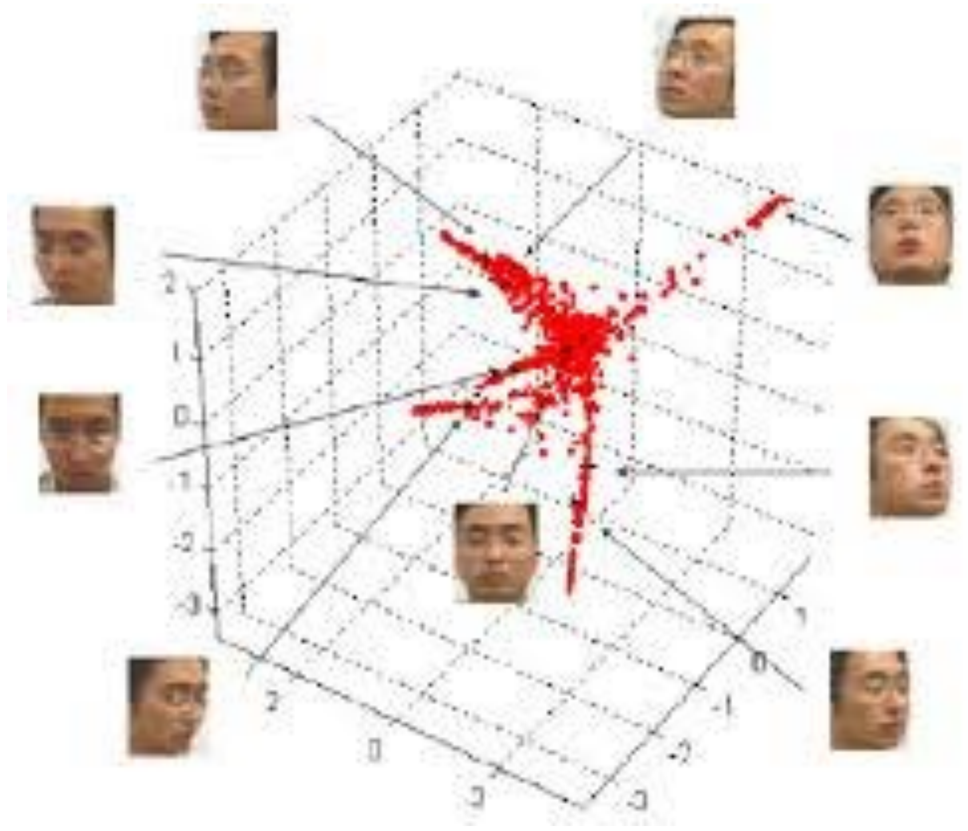
Dimensionality Reduction



- The MNIST images are 28x28 arrays.
- They are **not** uniformly distributed in R^{784}.
- In fact they exist on a low dimensional manifold.

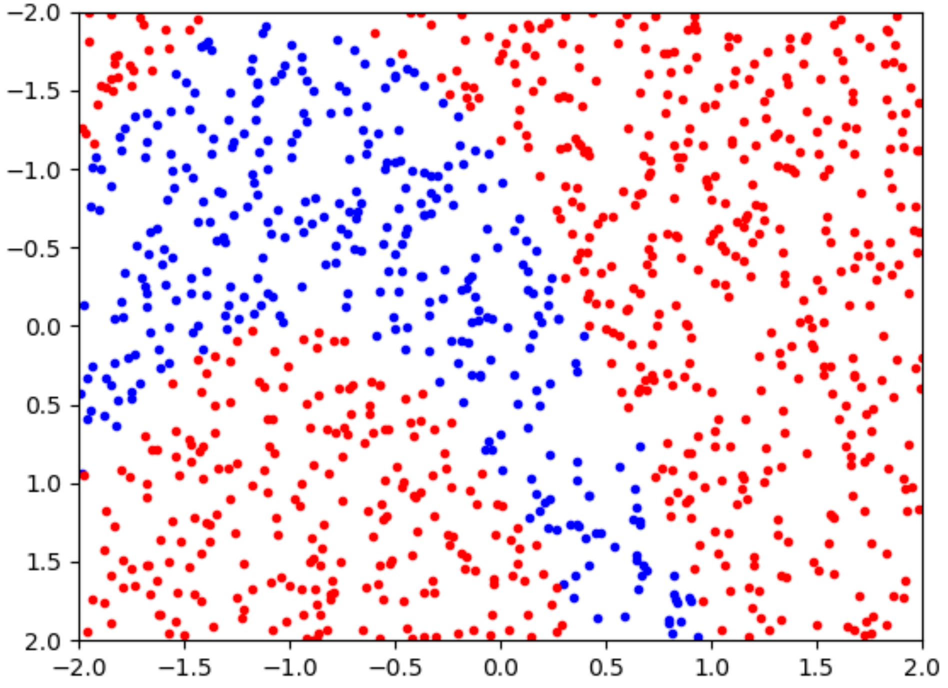
ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Face Images

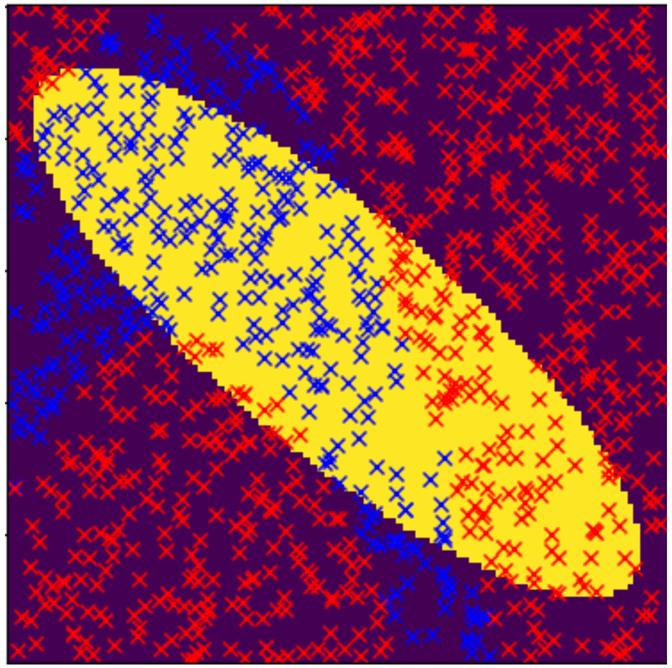


- The same can be said about face images.
- And of many other things.

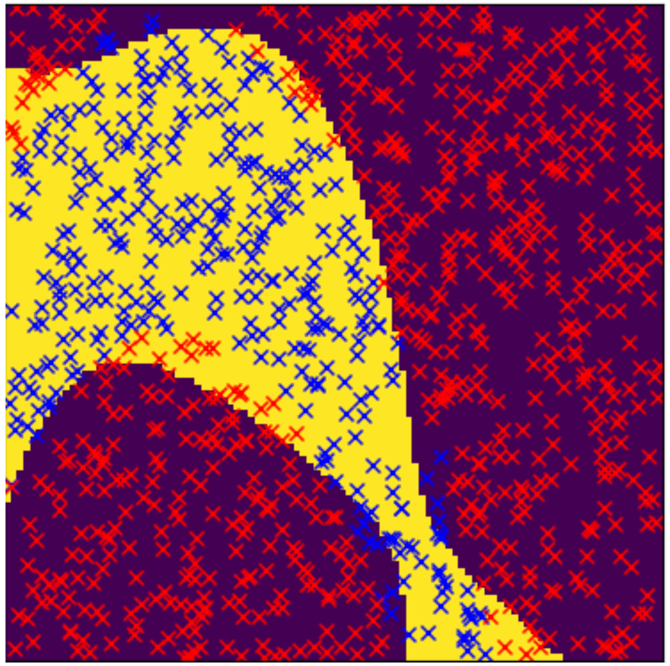
—> Non linear classification is a practical proposition.



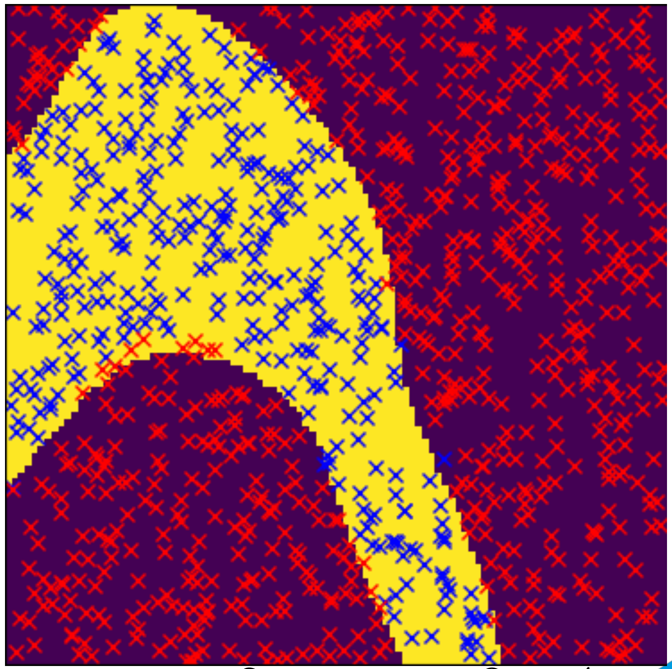
Rosenbrock:
$$r(x,y) = 100 * (y - x^2)^2 + (1-x)^2$$
$$f(x,y) = \begin{cases} \text{-1} & \text{if } r(x,y) < \text{T} \\ 1 & \text{otherwise} \end{cases}$$



$[x, y, xy]$

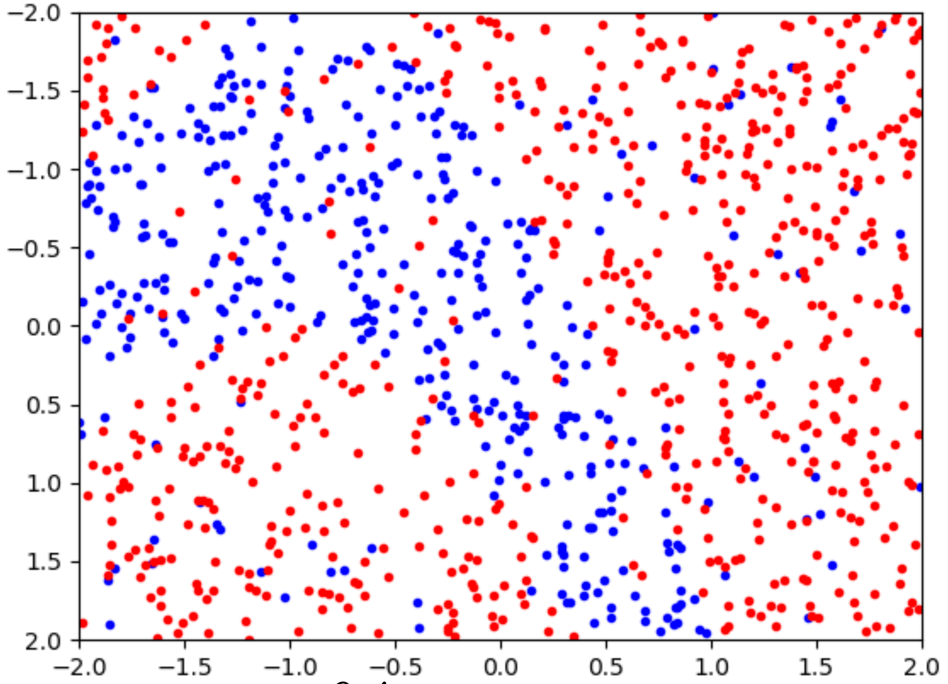


$[x, y, x^2, \ldots, xy^2, y^3]$

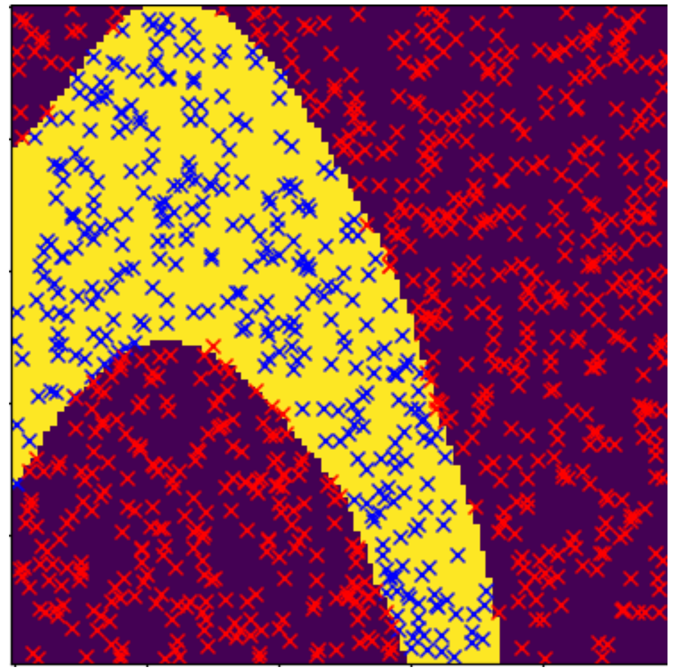


$[x, y, x^2, \ldots, xy^3, y^4]$

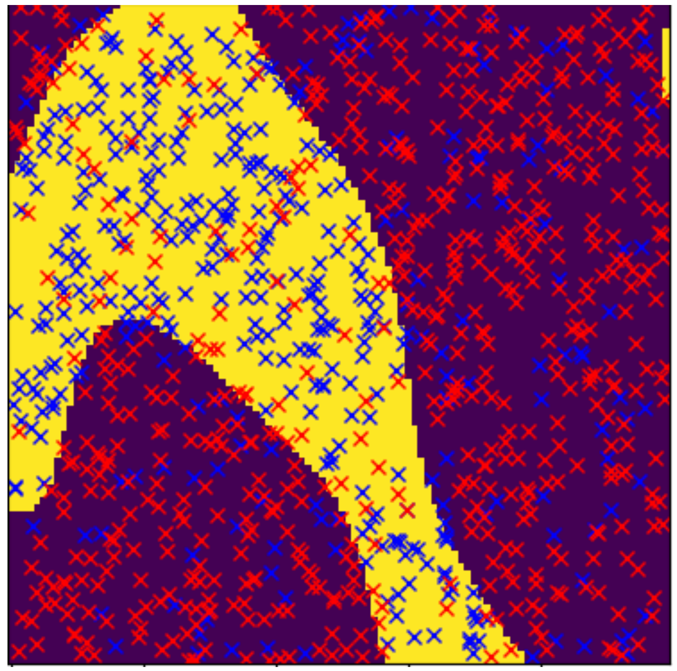
ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE



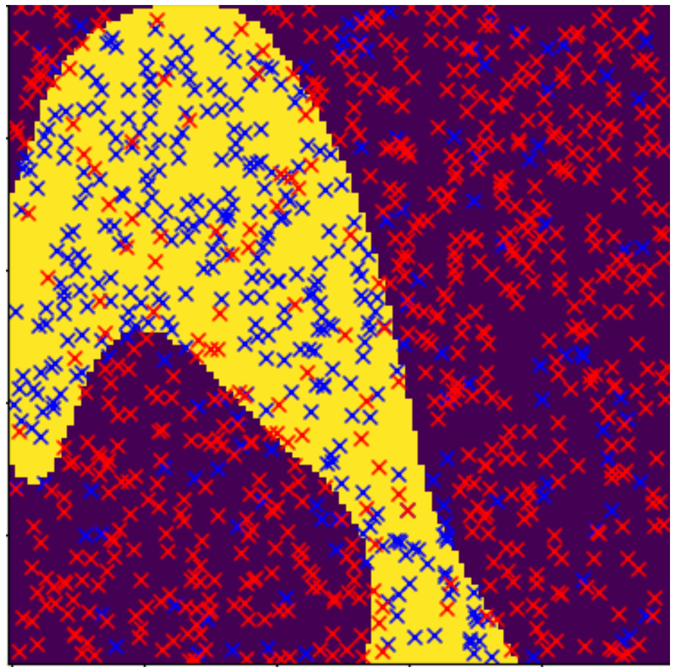
$10\% noise$



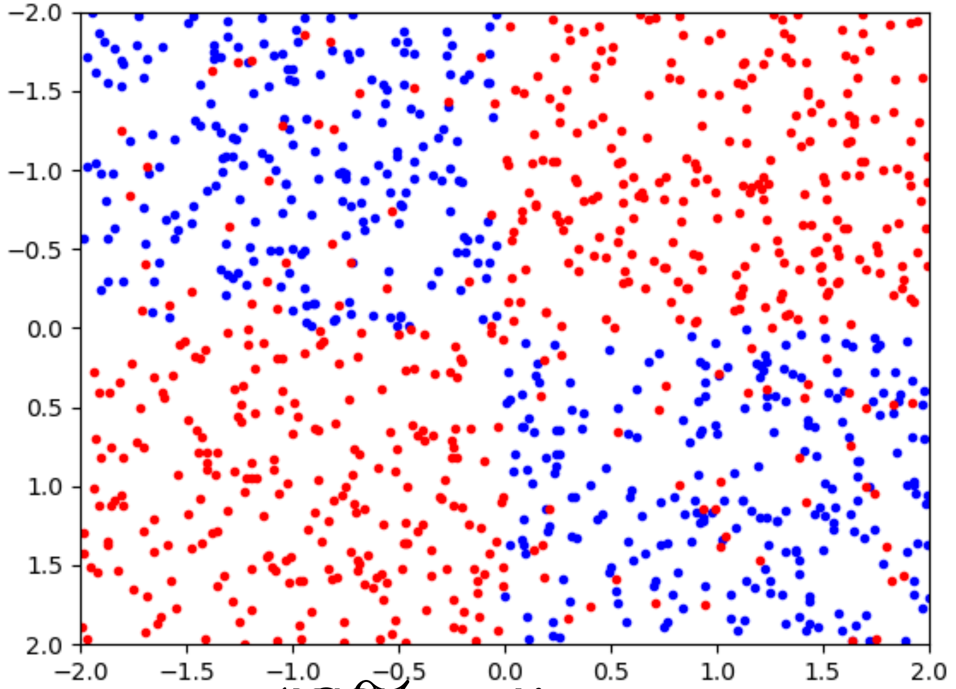
$[x, y, x^2, \ldots, xy^7, y^8]$



$5\% noise$

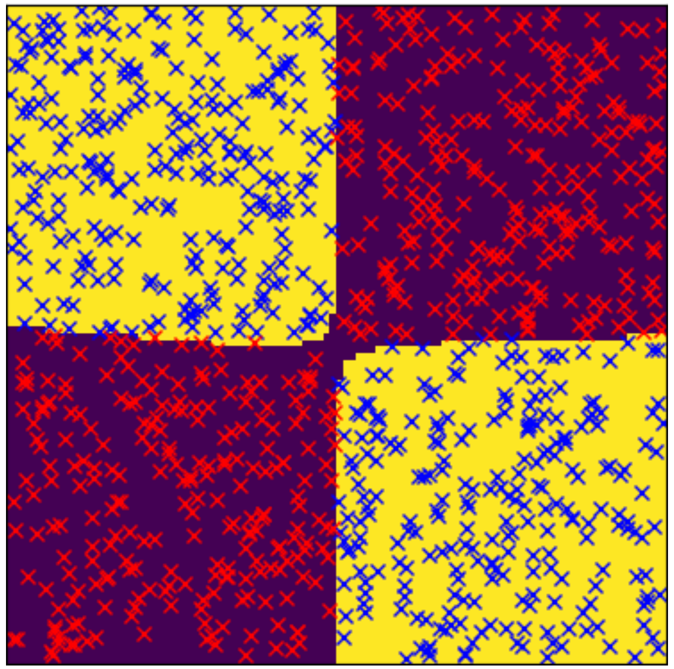


$10\% noise$

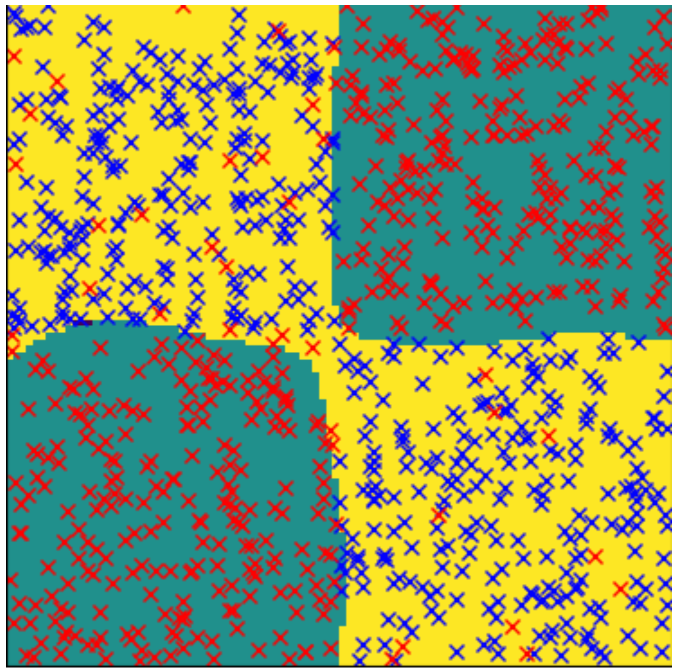


$15\% noise$

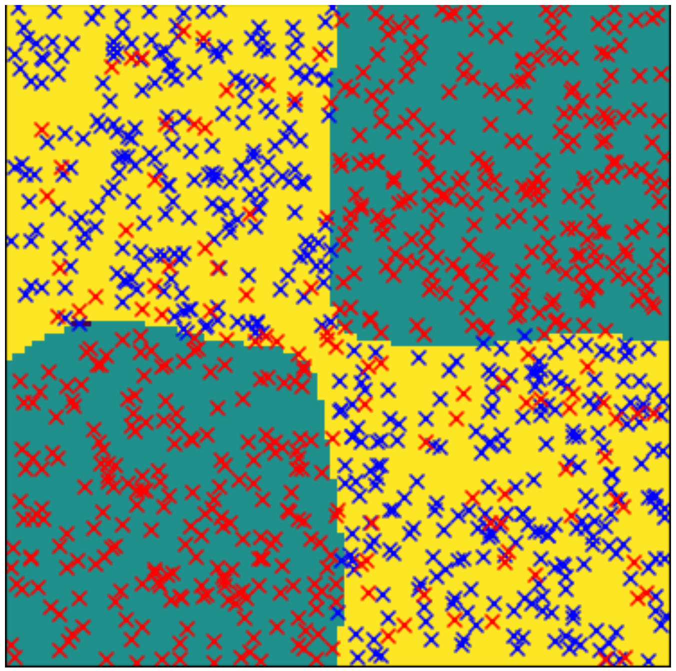
$10\% noise$



$[x, y, x^2, \ldots, xy^7, y^8]$



$5\% noise$



$10\% noise$

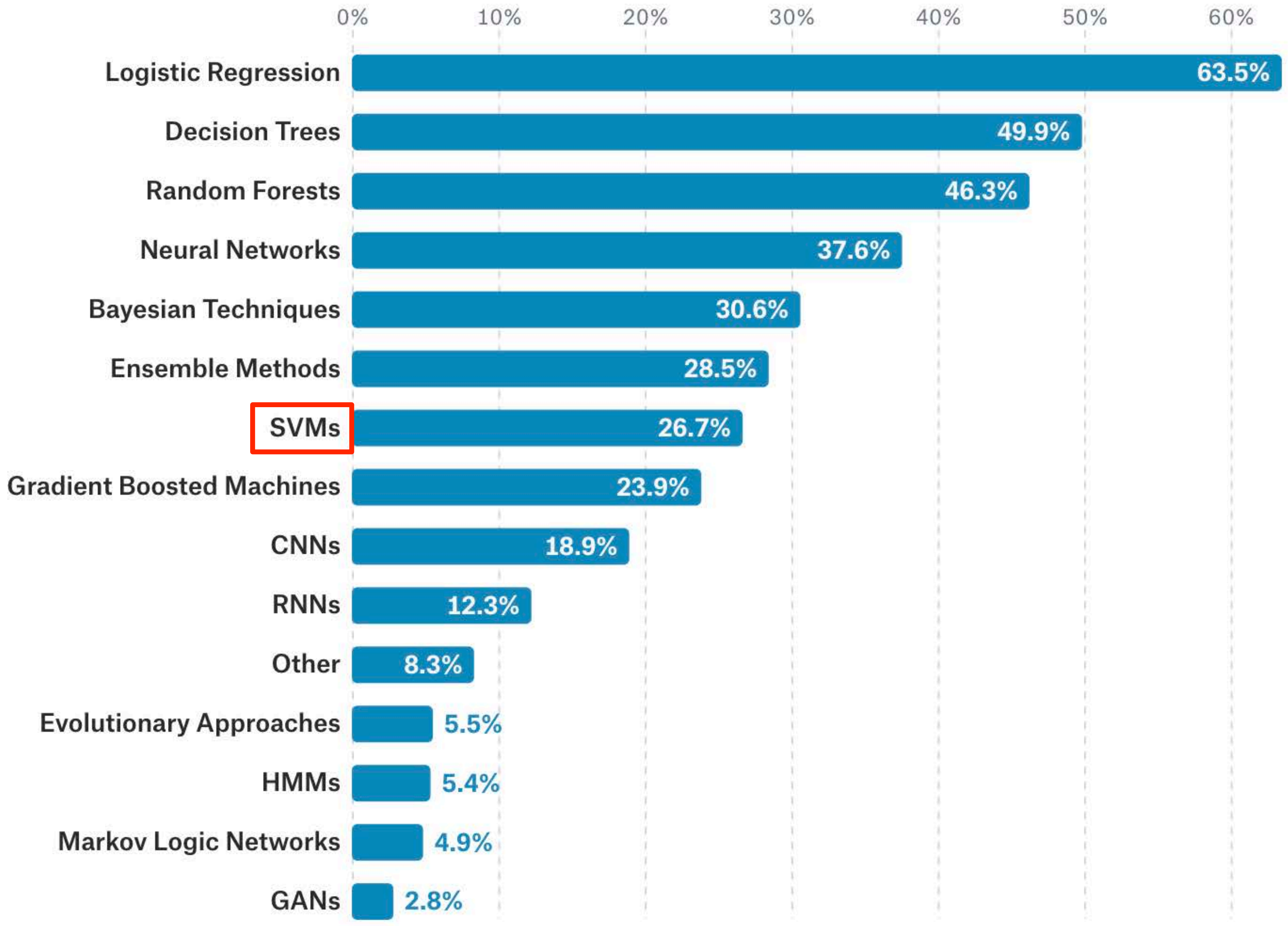
Increasing the dimension further

Can we increase the dimension massively:

- in a principled way,
- while keeping the computational burden down?

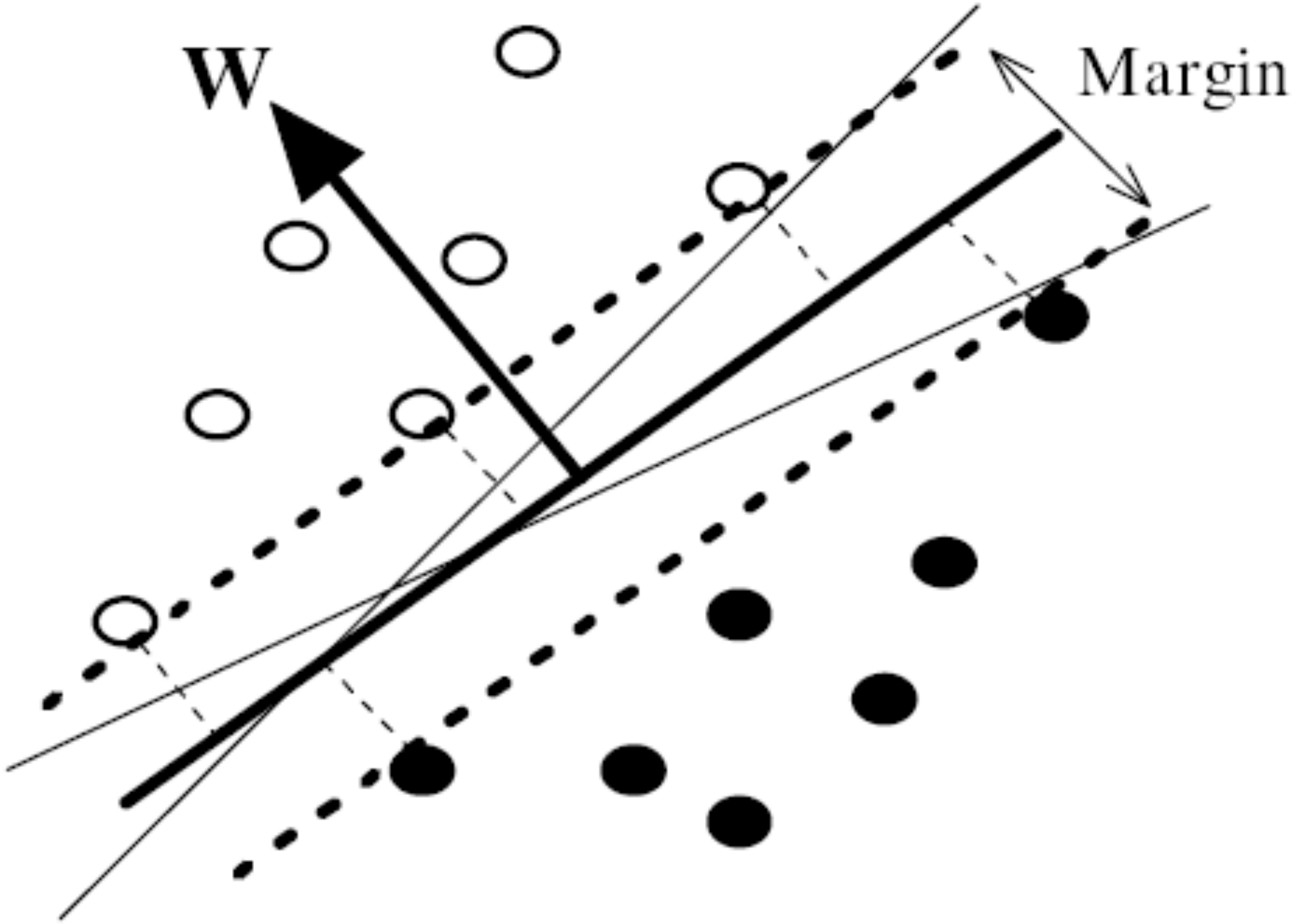
—> Non-linear support vector machines using the kernel trick.

Support Vector Machines



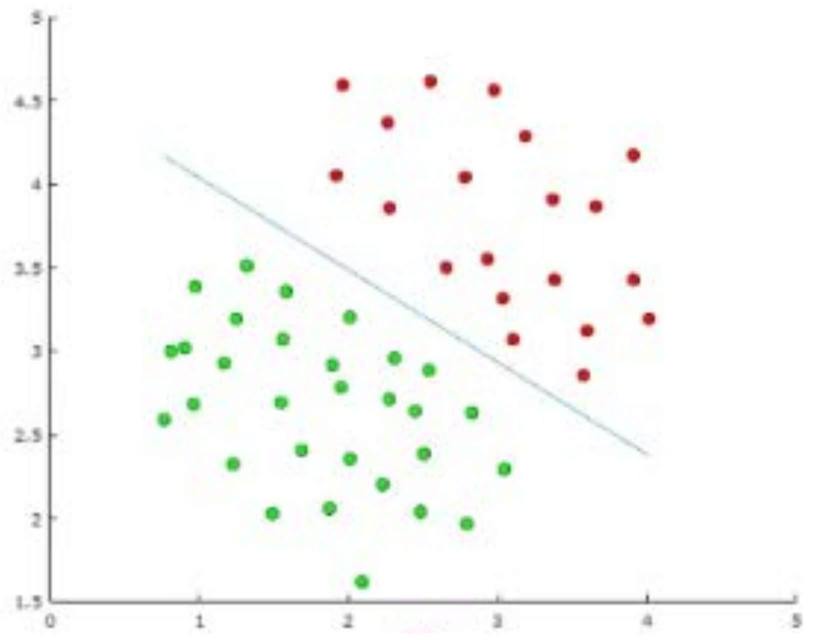
ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Margin

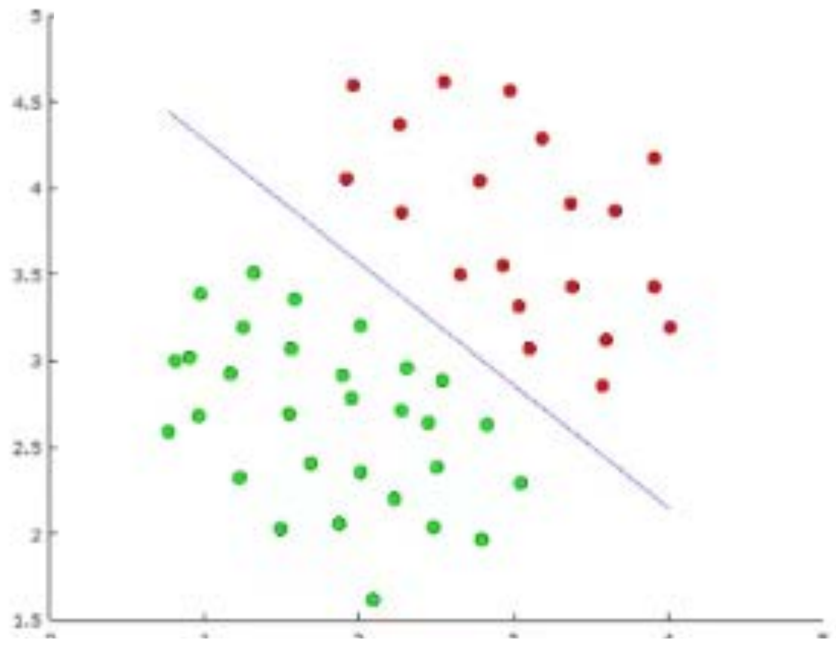


- The larger the margin, the better!
- The logistic regression does not guarantee a large one.

Logistic Regression Revisited



Logistic regression

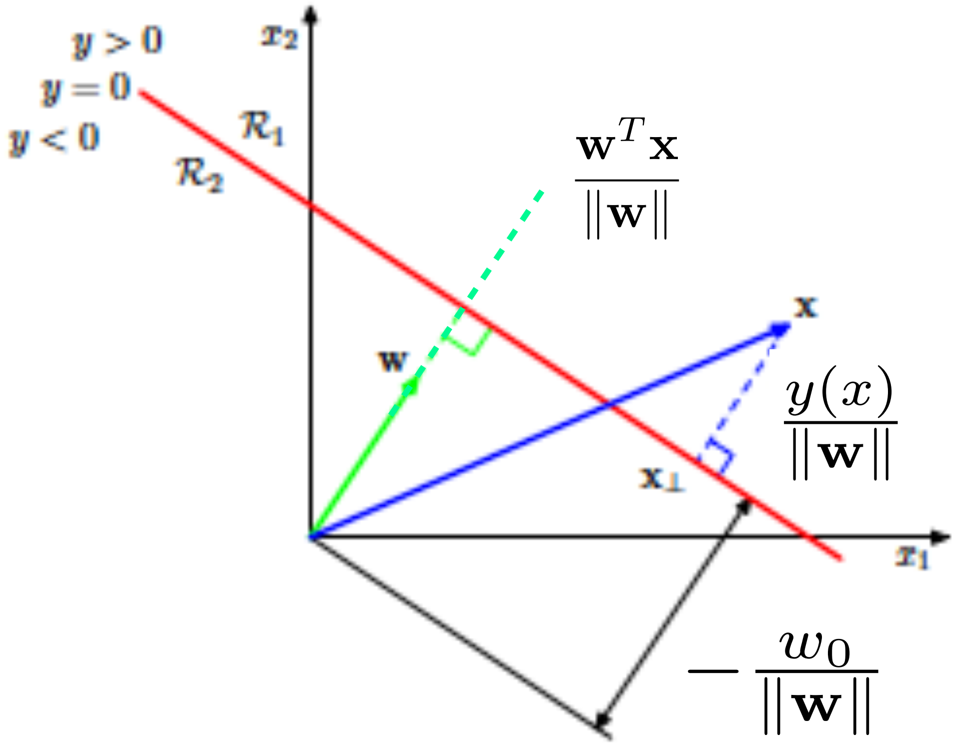


Linear SVM

- The LR decision boundary can come close to some of the training examples.
- This should be prevented if possible.

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Distance to the Decision Boundary



$y > 0$
$y = 0$
$y < 0$
x_2
$\mathcal{R}_1$
$\mathcal{R}_2$
$\frac{\mathbf{w}^T\mathbf{x}}{\|\mathbf{w}\|}$
$\mathbf{x}$
$\mathbf{w}$
$\frac{y(x)}{\|\mathbf{w}\|}$
$\mathbf{x}_\perp$
x_1
$-\frac{w_0}{\|\mathbf{w}\|}$

$$y(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_0$$

$$\frac{y(\mathbf{x})}{\|\mathbf{w}\|} = \frac{\mathbf{w}^T\mathbf{x}}{\|\mathbf{w}\|} + \frac{w_0}{\|\mathbf{w}\|}$$ <— Signed distance to the decision boundary.

Distance to the Decision Surface

When mapping in a high dimension space, we replace **x** by ϕ(**x**).
Therefore we can write:

$$y(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b$$

$$\frac{y(\mathbf{x})}{\|\mathbf{w}\|} = \frac{\mathbf{w}^T \phi(\mathbf{x})}{\|\mathbf{w}\|} + \frac{b}{\|\mathbf{w}\|}$$ <— Signed distance to the decision surface.

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Maximizing the Margin

Assuming that the training data is linearly separable using a linear model of the form

$$y(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b\,,$$

then

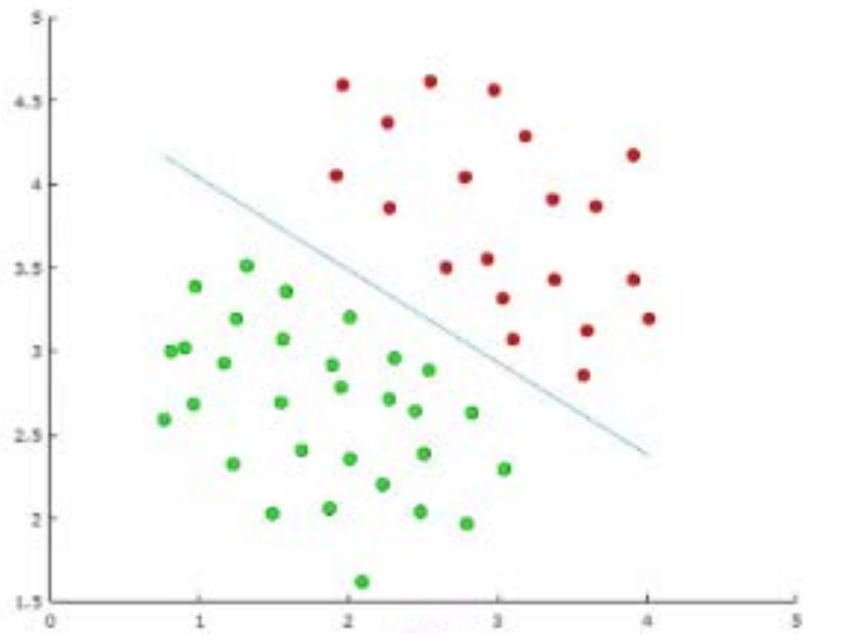
$$\frac{\mathbf{t}_n y(\mathbf{x}_n)}{\|\mathbf{w}\|} = \frac{t_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b)}{\|\mathbf{w}\|}$$ is the distance to the decision surface.

Therefore the max margin solution is found by solving

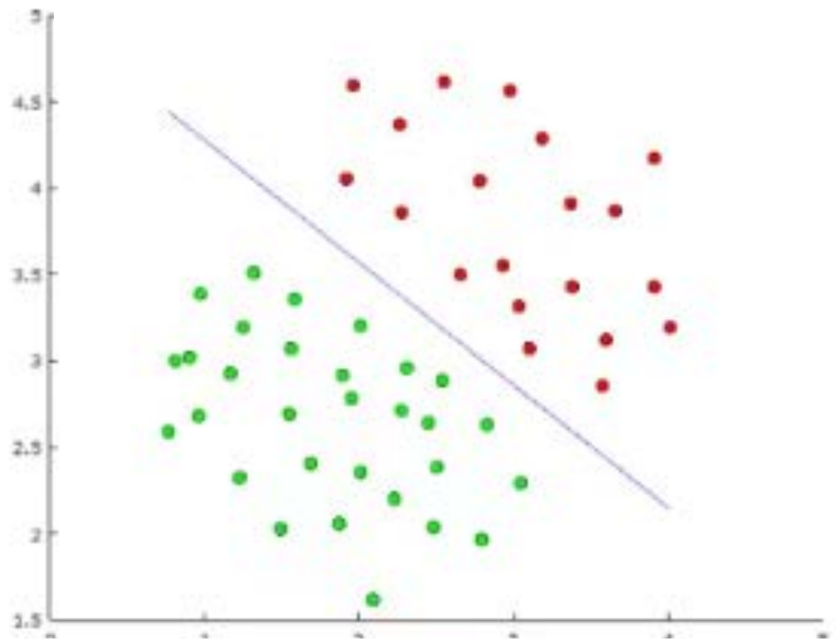
$$\arg\max_{\mathbf{w},b} \left\{ \frac{1}{\|\mathbf{w}\|} \min_n [t_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b)] \right\} \,.$$

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Logistic Regression Revisited



Logistic regression

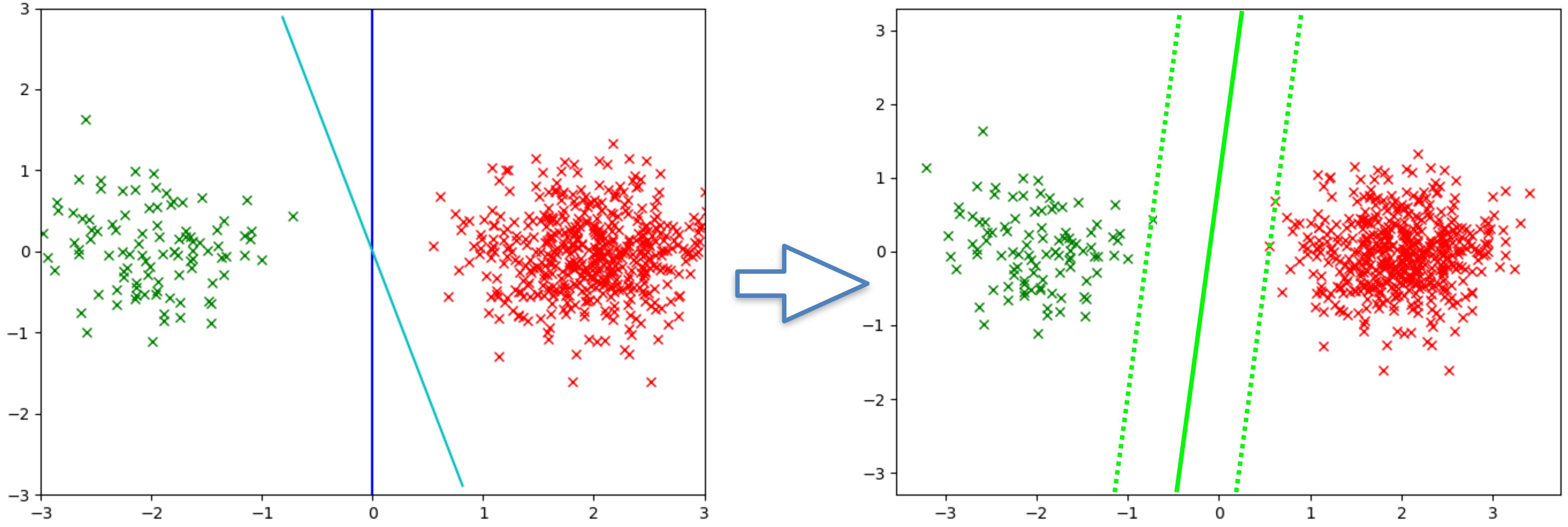


Linear SVM

If ϕ is taken to be the identity, the result is a linear SVM that generally does slightly better than then the logistic regression.

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

From Perceptron to Linear SVM



Perceptron

Linear SVM

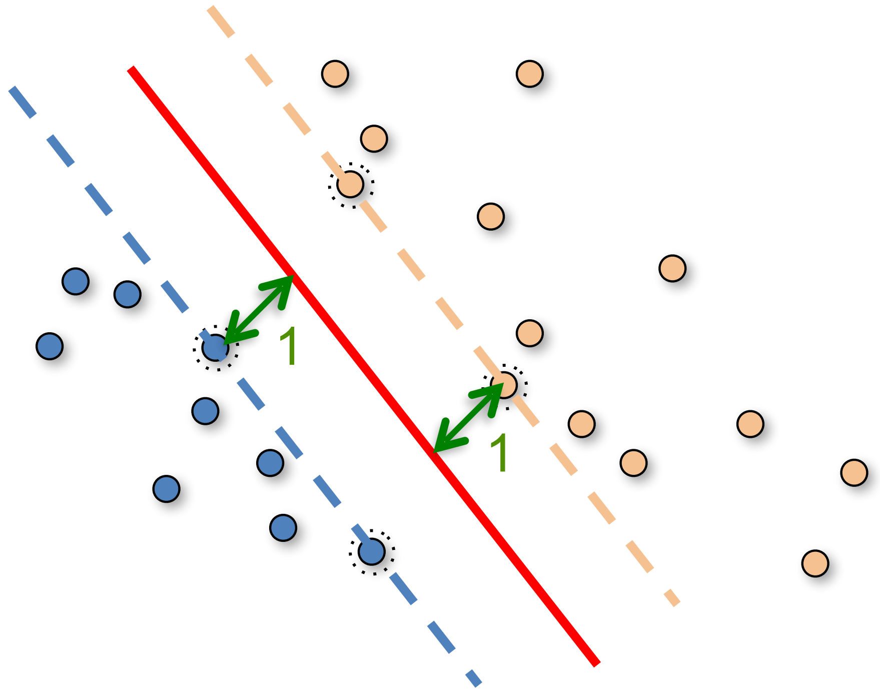
Maximizing the Margin

$\mathbf{w}$ and b can always be rescaled so that for the point closest to the decision surface

$$t_n(\mathbf{w}^T\phi(\mathbf{x}_n)+b) = 1\ ,$$

and for all others

$$t_n(\mathbf{w}^T\phi(\mathbf{x}_n)+b) \geq 1\ .$$



- The points that minimize the distance are known as **support vectors**.
- There are always at least two of them.
- The margin is $t_n y_n / ||\mathbf{w}|| = 1 / ||\mathbf{w}||$
- Maximizing the margin reduces to minimizing

$$\frac{1}{2}||\mathbf{w}||^2$$

under the above constraints.

—> Quadratic programming problem with D variables.

—> Complexity in $O(D^3)$.

Introducing Constraints

Constrained optimization:

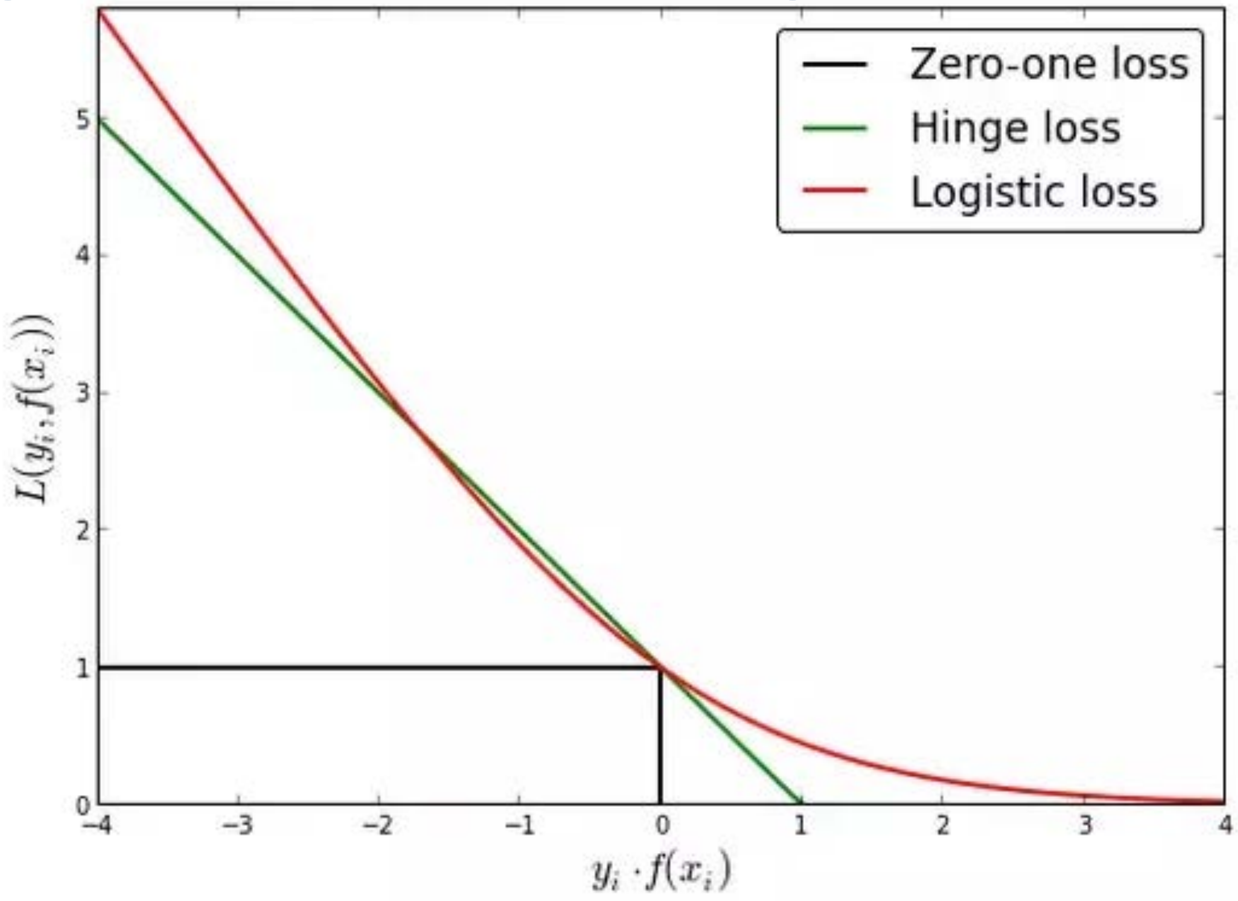
$$\arg\min_{\mathbf{w},b} \tfrac{1}{2}\|\mathbf{w}\|^2 \text{ subject to } \forall n,\ t_n(\mathbf{w}^T\phi(\mathbf{x}_n)+b)\geq 1$$

Unconstrained optimization:

$$\arg\min_{\mathbf{w},b} \tfrac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N}\max(0, 1-t_n(\mathbf{w}^T\phi(\mathbf{x}_n)+b))$$

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Hinge Loss vs Logistic Loss



Hinge loss: $$\arg\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N} \max(0, 1 - t_n(\mathbf{w}^T\phi(\mathbf{x}_n) + b)) \ ,$$

Logistic loss: $$\arg\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N} \log(1 + \exp(1 - t_n(\mathbf{w}^T\phi(\mathbf{x}_n) + b))) \ .$$

https://towardsdatascience.com/support-vector-machine-vs-logistic-regression-94cc2975433f

Back to Constrained Minimization

Definition:

$$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{n=1}^{N} a_n \left\{t_n(\mathbf{w}^T\phi(\mathbf{x}_n) + b) - 1\right\}$$

is known as the **Lagrangian** and the a_n as the **Lagrange multipliers**.

Theorem:

A solution of the constrained minimization problem must be such that L is minimized with respect to the components of vector $\mathbf{w}$ and maximized with respect to the Lagrange multipliers, which must remain greater or equal to zero.

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Back to Constrained Minimization

Definition:

$$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{n=1}^{N} a_n \left\{ t_n(\mathbf{w}^T \phi(\mathbf{x}_n) + b) - 1 \right\}$$

is known as the **Lagrangian** and the a_n as the **Lagrange multipliers**.

Setting the derivatives of $L(\mathbf{w}, b, \mathbf{a})$ to zero with respects to the elements of $\mathbf{w}$ and b yieds

$$\begin{aligned} \mathbf{w} &= \sum_{n=1}^{N} a_n t_n \phi(\mathbf{x}_n) \ , \\ 0 &= \sum_{n=1}^{N} a_n t_n \ . \end{aligned}$$

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Dual Problem

At the optimum, we have

$$\tilde{L}(\mathbf{a}) = L(\mathbf{w}, b, \mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m)$$

subject to

$$a_n \geq 0 \; \forall n \; ,$$

$$\sum_{n=1}^{N} a_n t_n = 0$$

and with

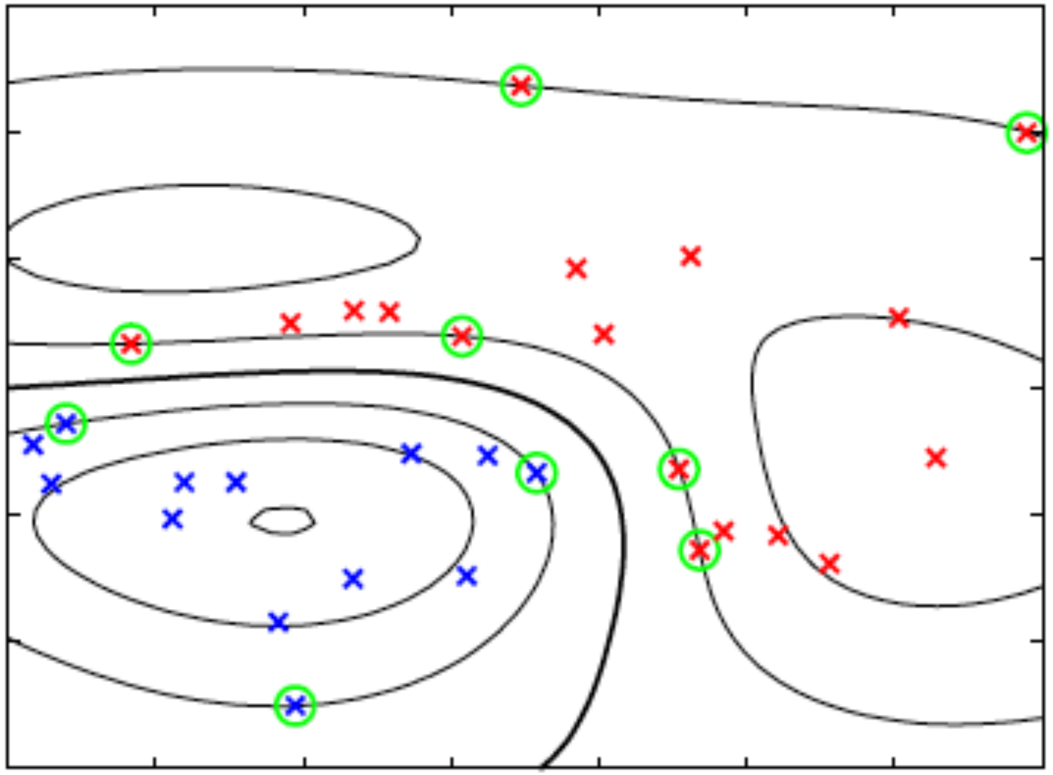
$$k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}') \; .$$

—> Quadratic programming problem with N variables.

—> Complexity in $O(N^3)$ instead of $O(D^3)$.

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Support Vectors



$$\mathbf{w} = \sum_{n=1}^{N} a_n t_n \phi(\mathbf{x}_n) \; .$$

$$y(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b \; ,$$

$$= \sum_{n=1}^{N} a_n t_n k(\mathbf{x}, \mathbf{x}_n) + b \; ,$$

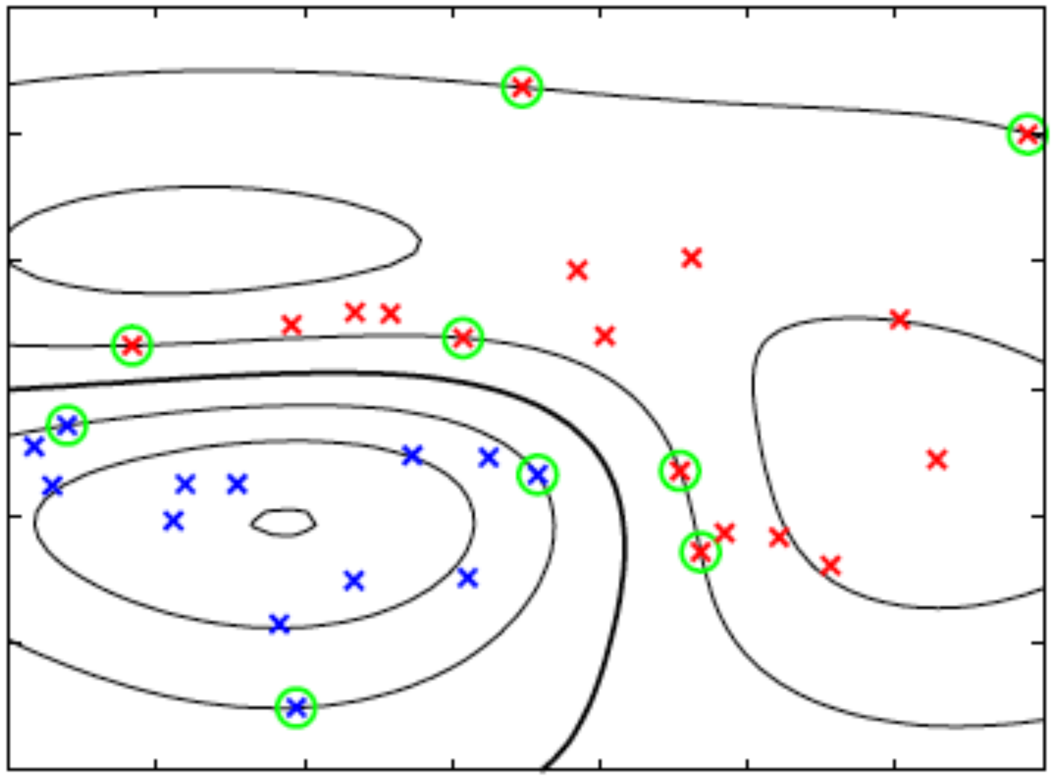
with $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^T \phi(\mathbf{x}') \; .$

- Only for a subset of the data points is a_n is non zero.
- The corresponding x_n are the support vectors and satisfy $t_n y(x_n)=1$.
- They are the only ones that need to be considered as test time.

—> That is what makes SVMs practical!

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

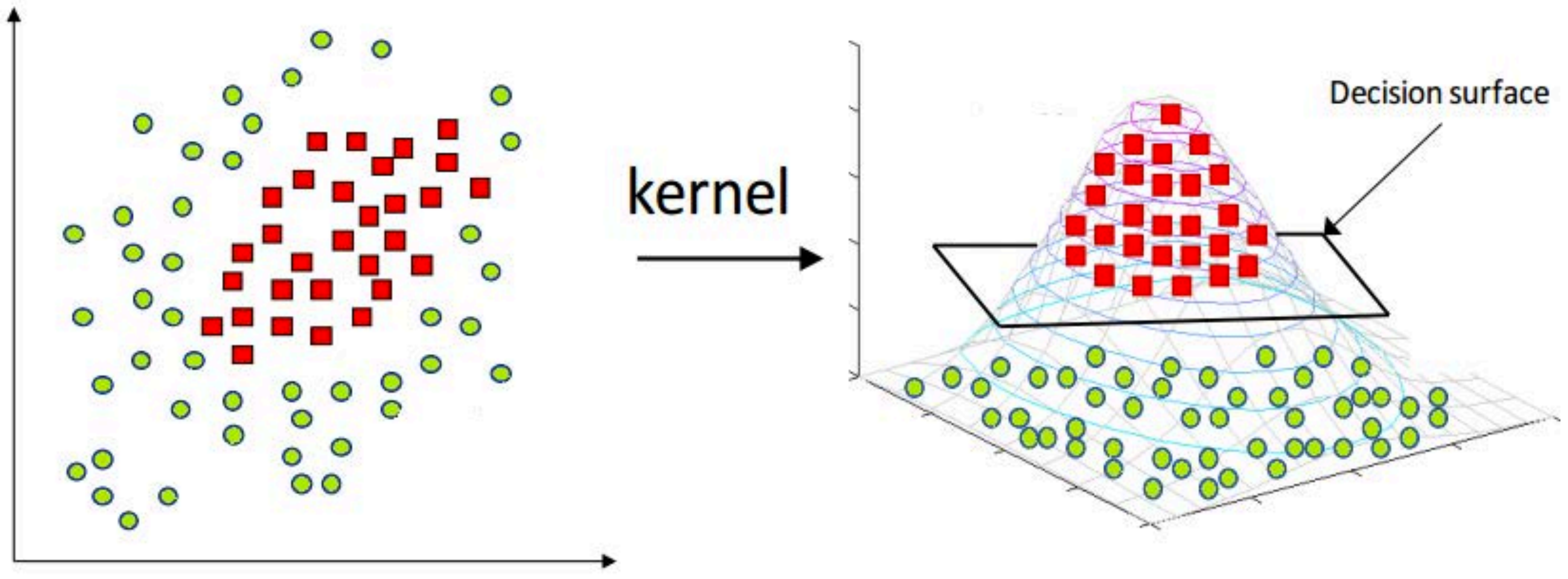
Kernel Functions



$$y(\mathbf{x}) = \sum_{n=1}^{N} a_n t_n k(\mathbf{x}, \mathbf{x}_n) + b ,$$

- The feature vector $\phi(\mathbf{x})$ does **not** appear explicitly anymore.
- The kernel function k(.) can be understood as a similarity measure.

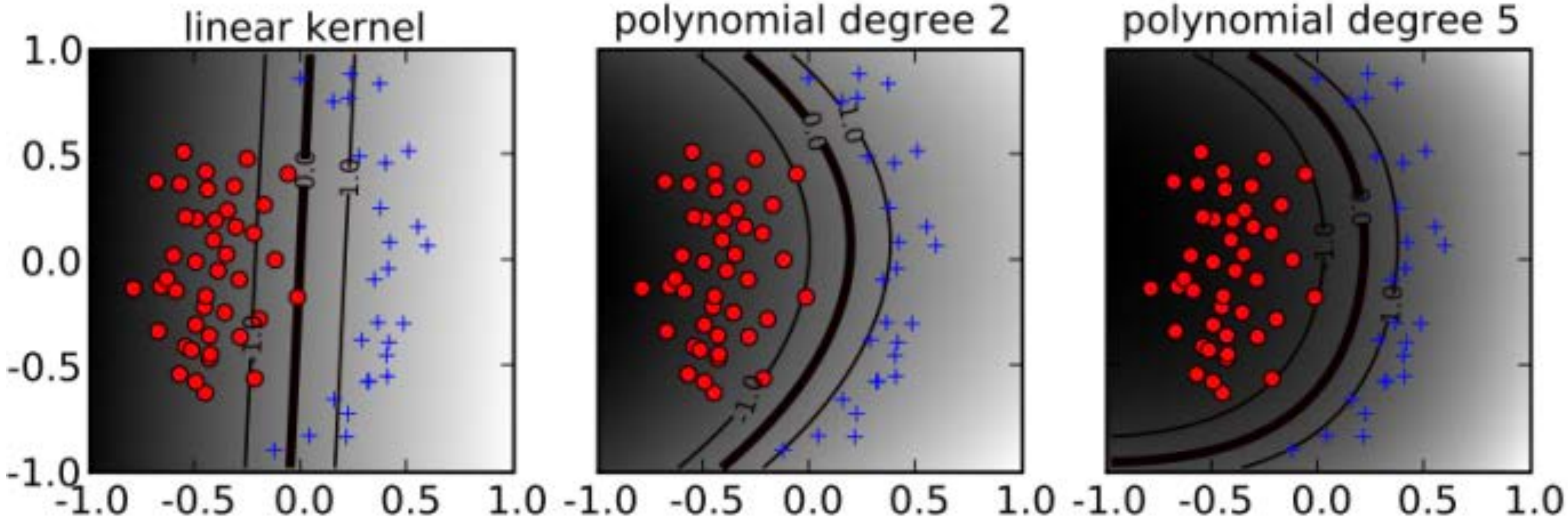
Role of the Kernel



Polynomial kernels: From small to high dimension.
Gaussian   kernels: From small to infinite dimension.

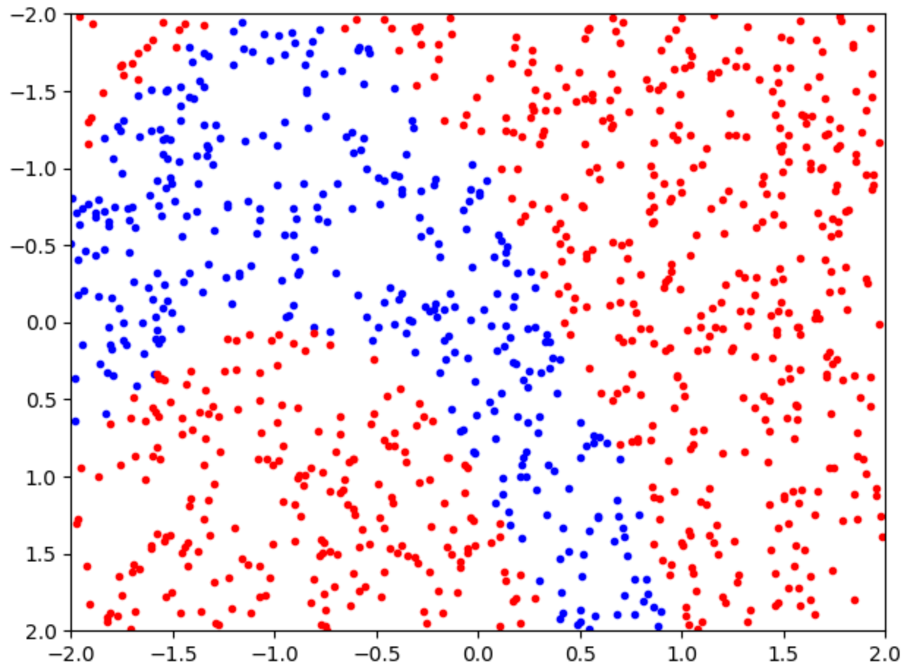
ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

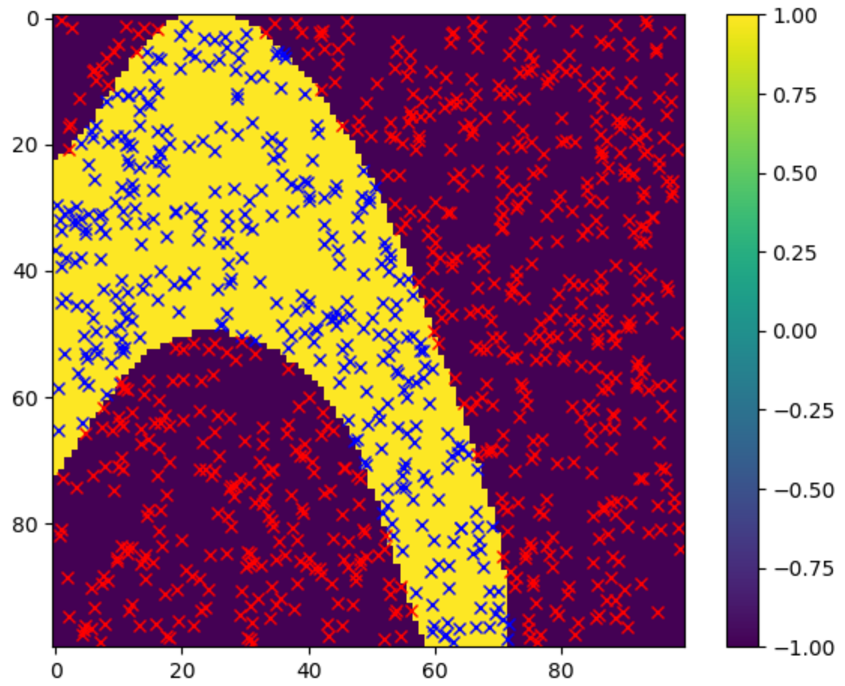
Influence of the Kernel



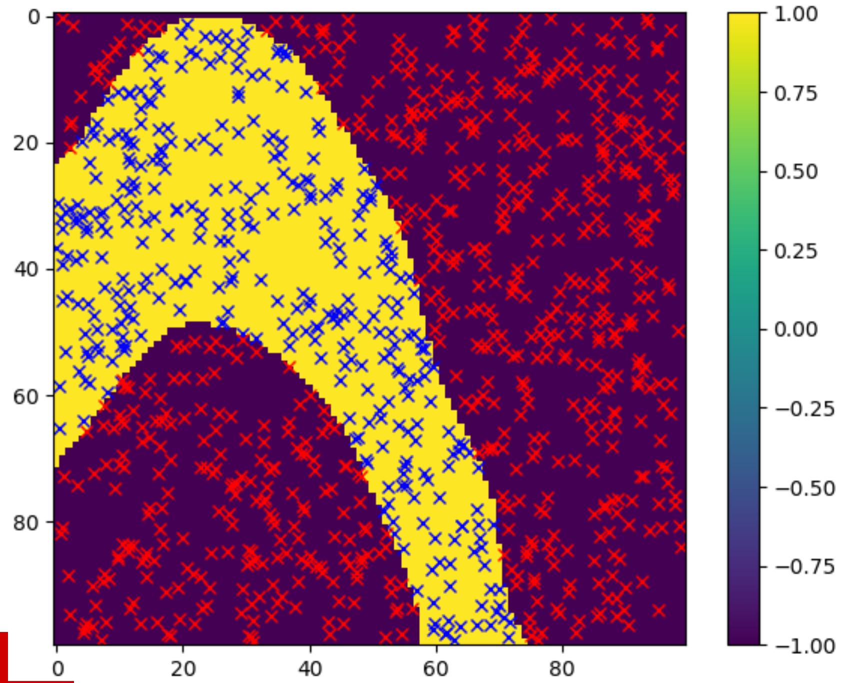
$$
\begin{aligned}
y(\mathbf{x}) &= \sum_{n=1}^{N} a_n t_n k(\mathbf{x}, \mathbf{x}_n) + b\ , \\
k(\mathbf{x}, \mathbf{x}') &= 1 + (\mathbf{x}^T \mathbf{x}')^d \qquad \text{(Polynomial terms up to degree } d\text{).} \\
k(\mathbf{x}, \mathbf{x}') &= \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{\sigma^2}\right) \text{ (Gaussian, feature space of infinite dimension).}
\end{aligned}
$$

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

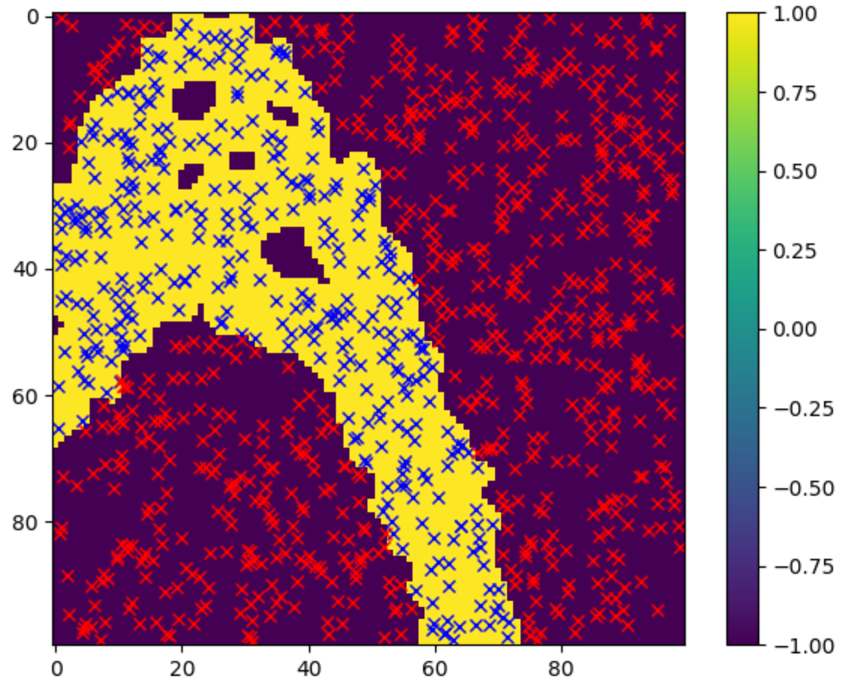




Rbf, σ = 10.0

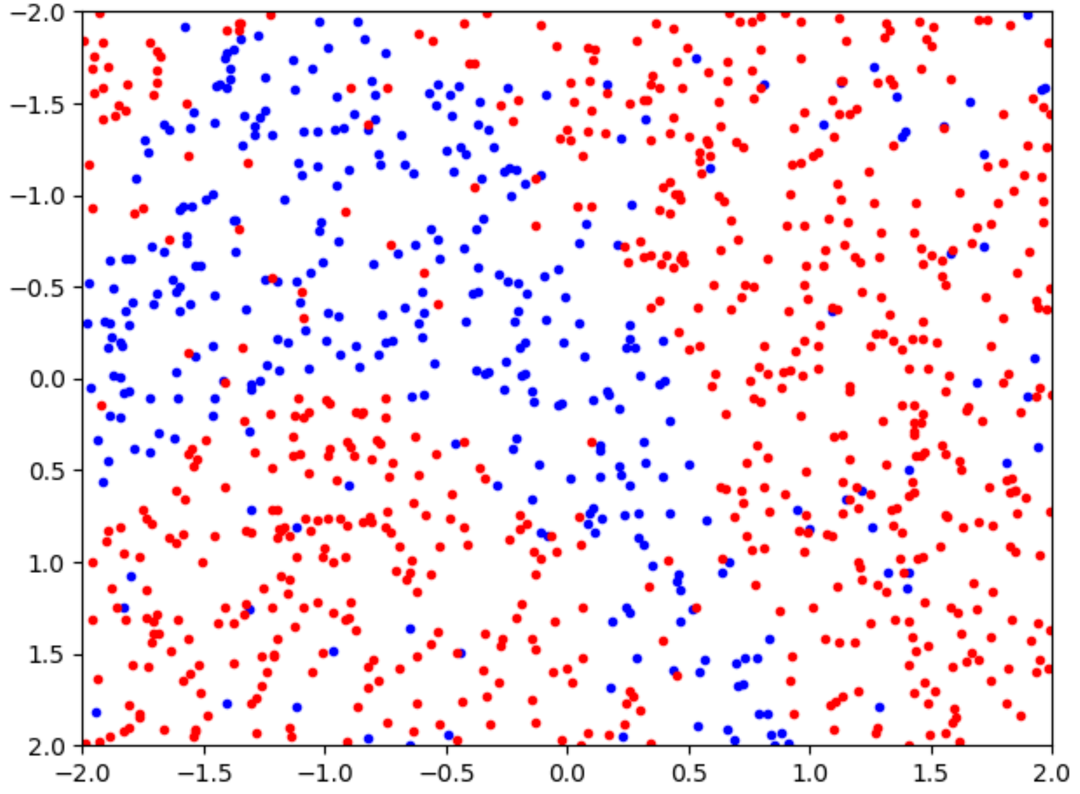


Rbf, σ = 1.0



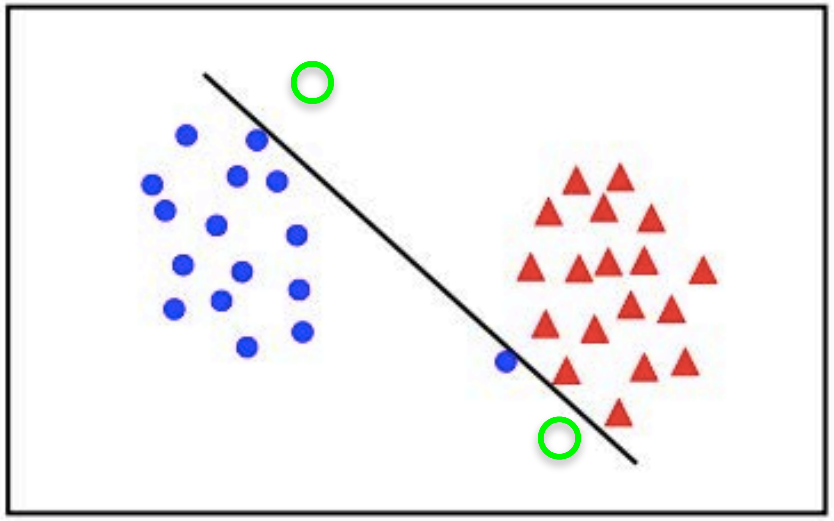
Rbf, σ = 0.01

Non-Separable Distributions

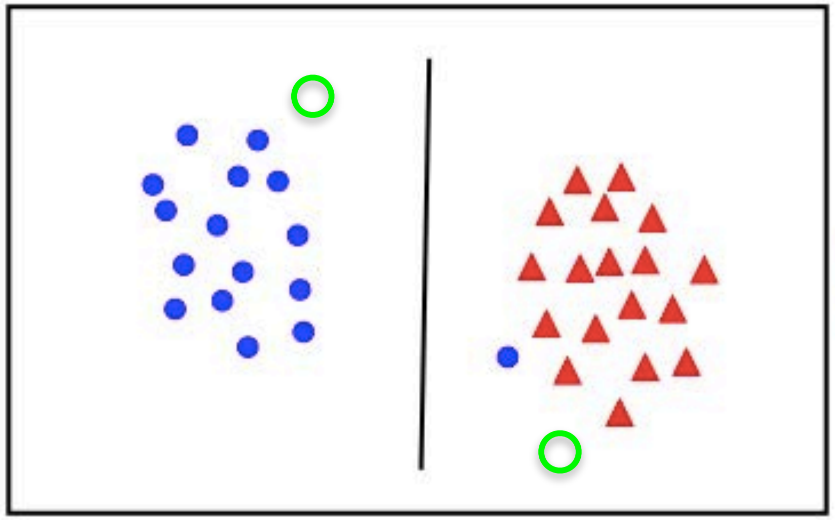


Cannot easily satisfy the hard constraints!

Optimal vs Best



- The points can be linearly separated but the margin is still very small.
- At test time the two green circles will be misclassified.

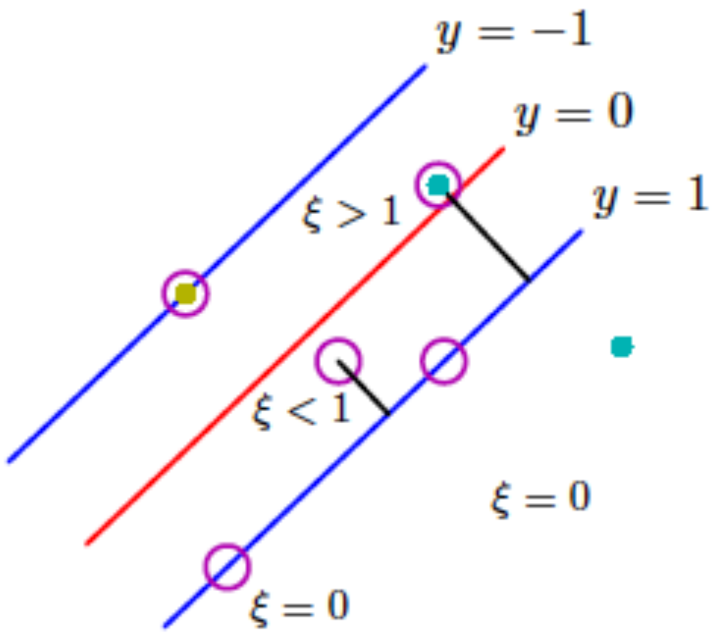


- The margin is much larger but one training example is misclassified.
- At test time the two green circles will be classified correctly.

—> Tradeoff between the number of mistakes on the training data and the margin.

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Slack Variables



$$t_n y(x_n) \geq 1 - \xi_n$$

Controls the balance between large margin and training misclassification rate.

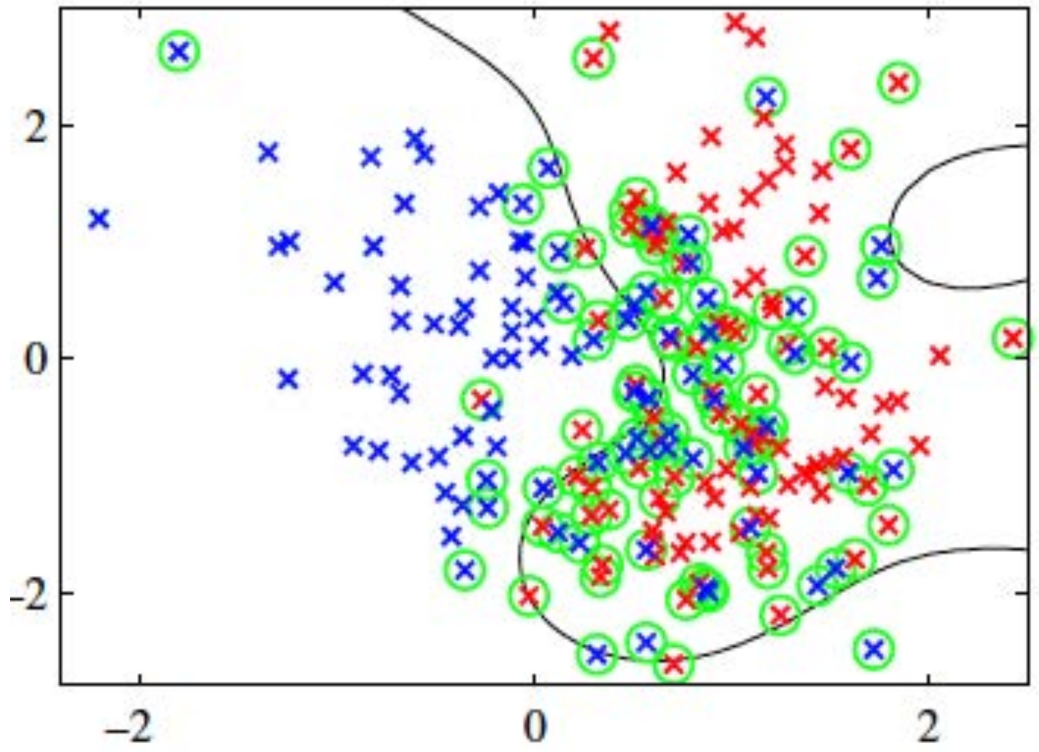
$$\arg\min_{\mathbf{w},b,\xi} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N} \xi_n$$

subject to

$$\begin{aligned} \forall n,\ \xi_n &\geq 0 \\ \forall n,\ t_n(\mathbf{w}^T\phi(\mathbf{x}_n) + b) &\geq 1 - \xi_n \end{aligned}$$

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Lagrangian



$$y(\mathbf{x}) = \sum_{m\in\mathcal{S}} a_m t_m k(\mathbf{x}, \mathbf{x}_m) + b$$

Minimize

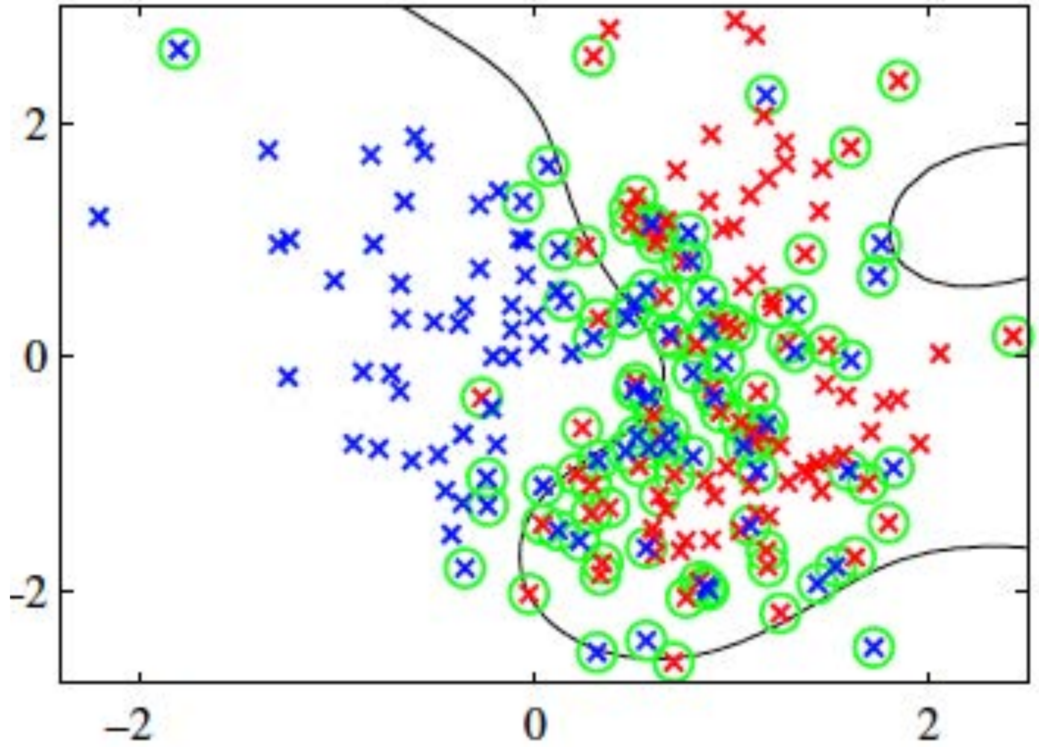
$$\tilde{L}(\mathbf{a}) = L(\mathbf{w}, b, \mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2}\sum_{n=1}^{N}\sum_{m=1}^{N} a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m) \,,$$

subject to

$$0 \le a_n \le C \,,\ \forall n \,,$$

$$\sum_{n=1}^{N} a_n t_n = 0 \,.$$

Interpretation



$a_n > 0$: x_n is a support vector.

- $a_n < C$: x_n lies on the margin.
- $a_n = C$: x_n lies inside the margin.
- $a_n > C$: x_n is correctly classified.

Minimize

$$\tilde{L}(\mathbf{a}) = L(\mathbf{w}, b, \mathbf{a}) = \sum_{n=1}^{N} a_n - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} a_n a_m t_n t_m k(\mathbf{x}_n, \mathbf{x}_m) \ ,$$

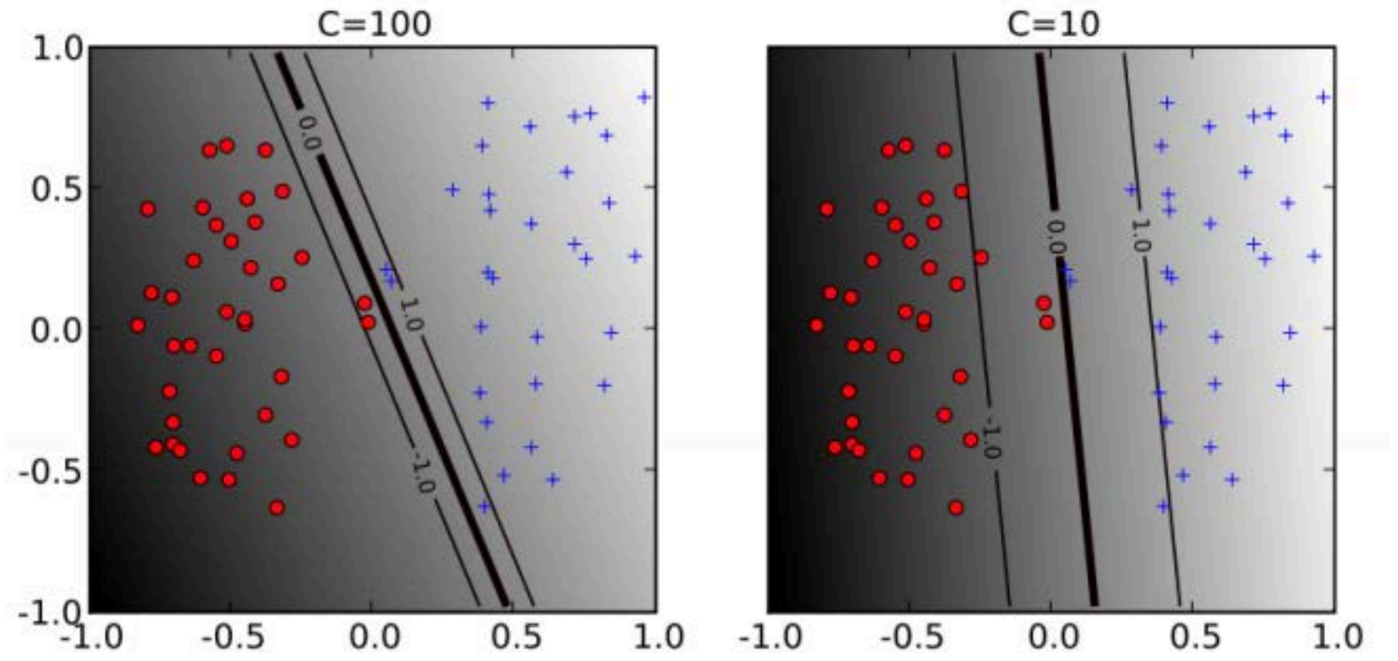
subject to

$$0 \leq a_n \leq C \ , \ \forall n \ ,$$

$$\sum_{n=1}^{N} a_n t_n = 0 \ .$$

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

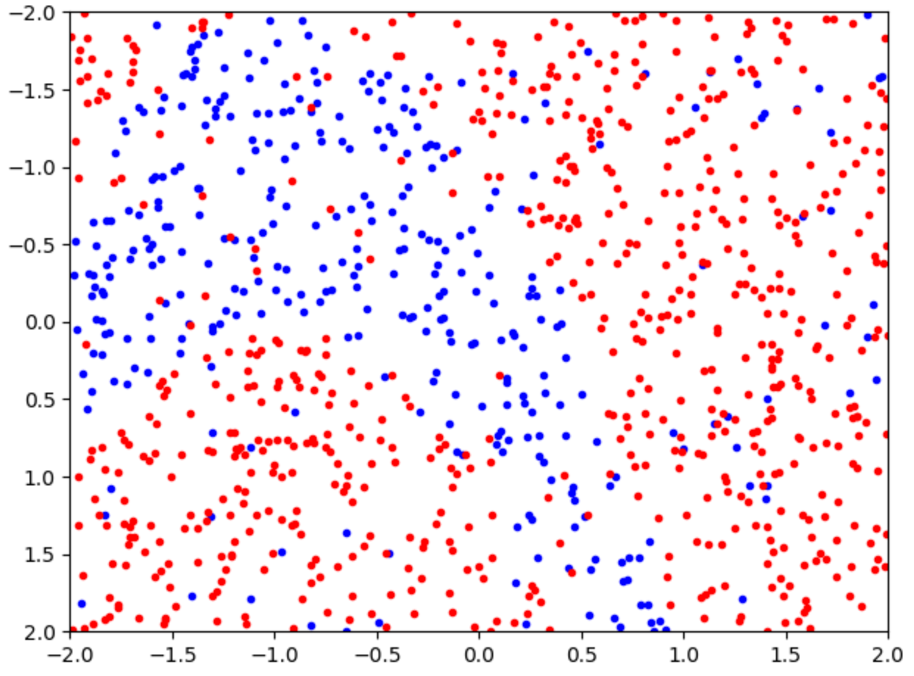
Impact of the C constant

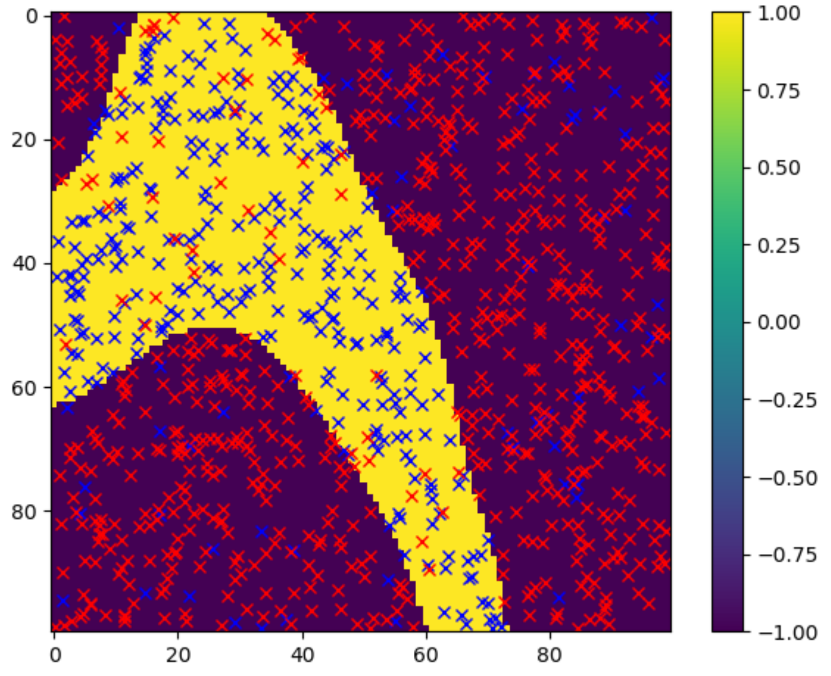


$$\arg\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{n=1}^{N}\xi_n$$

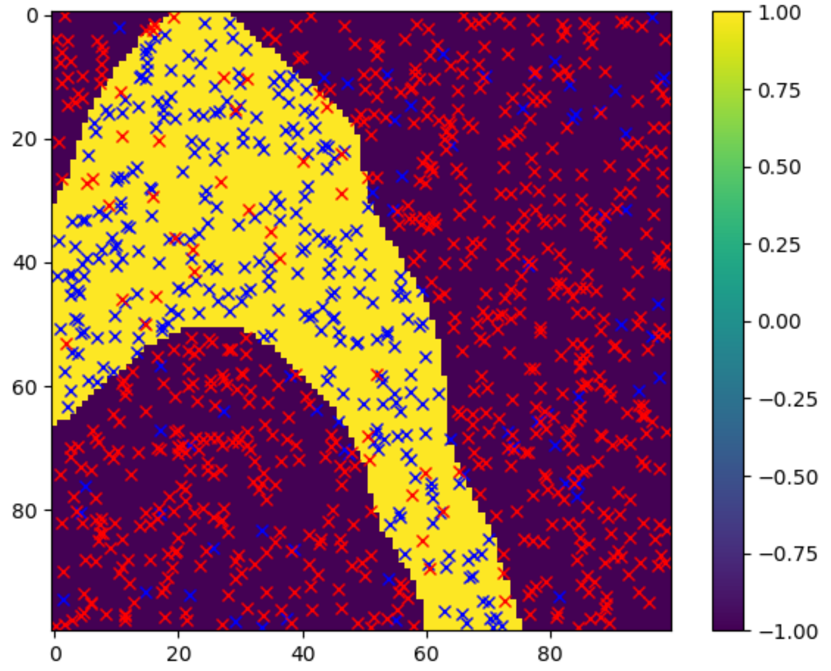
subject to

$$\begin{aligned} \forall n,\ \xi_n &\geq 0 \\ \forall n,\ t_n(\mathbf{w}^T\phi(\mathbf{x}_n)+b) &\geq 1-\xi_n \end{aligned}$$

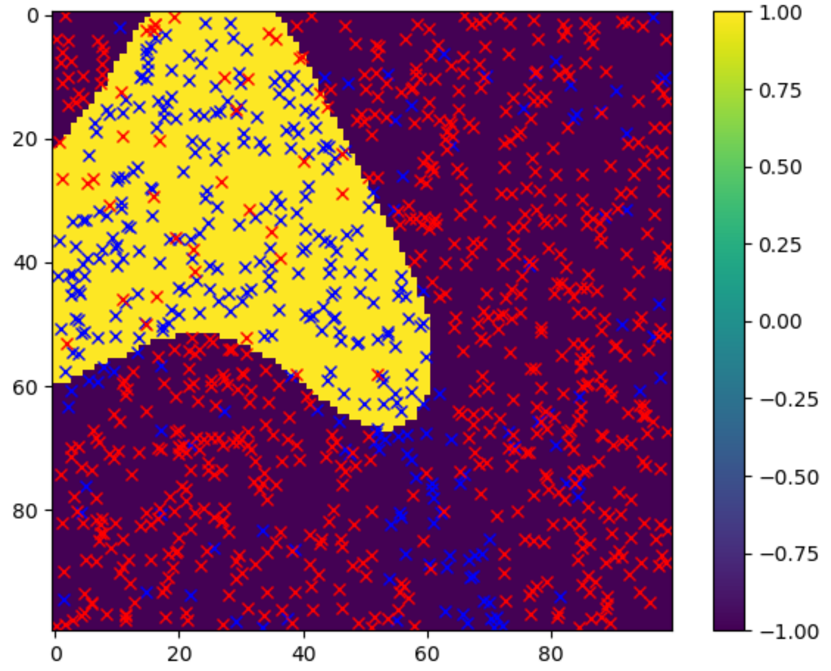




Rbf, σ = 1.0, C=1.0

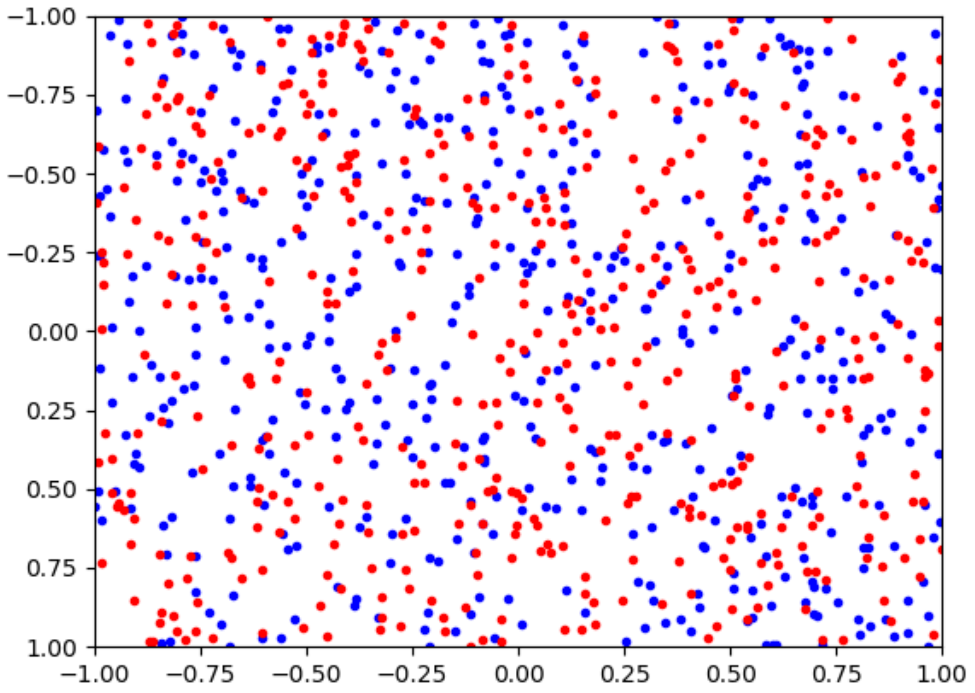


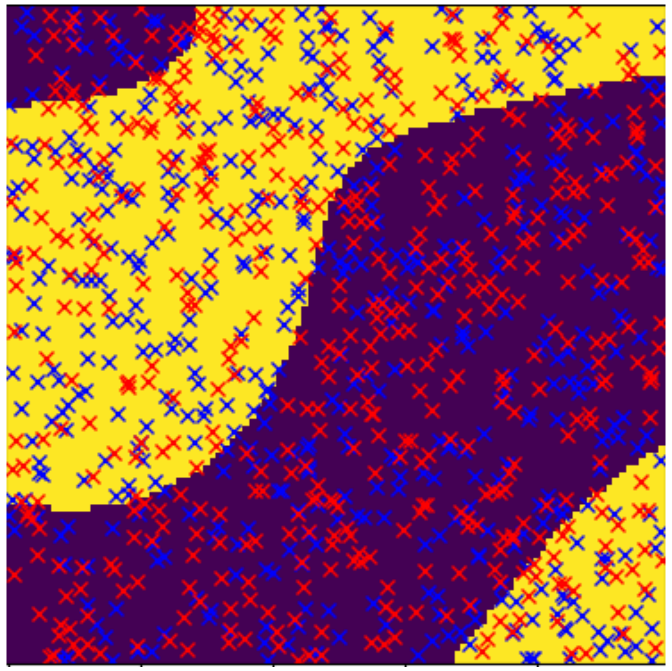
Rbf, σ = 1.0, C= 100.0

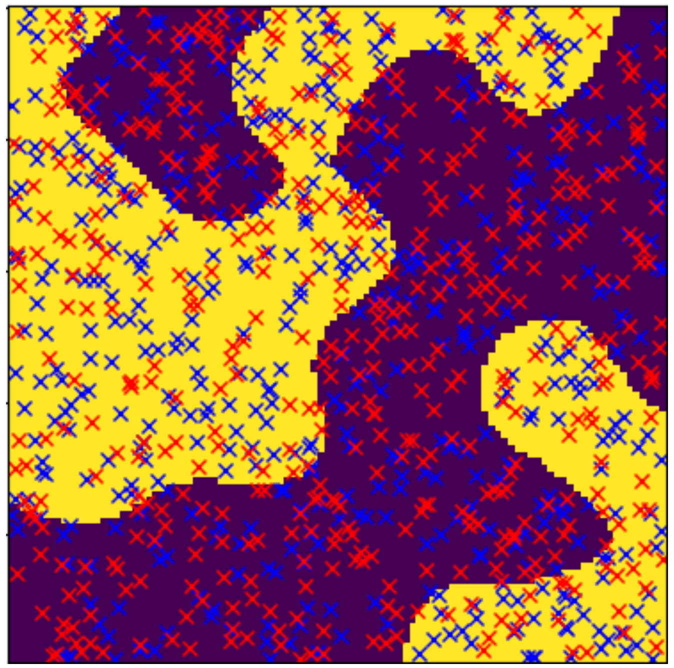


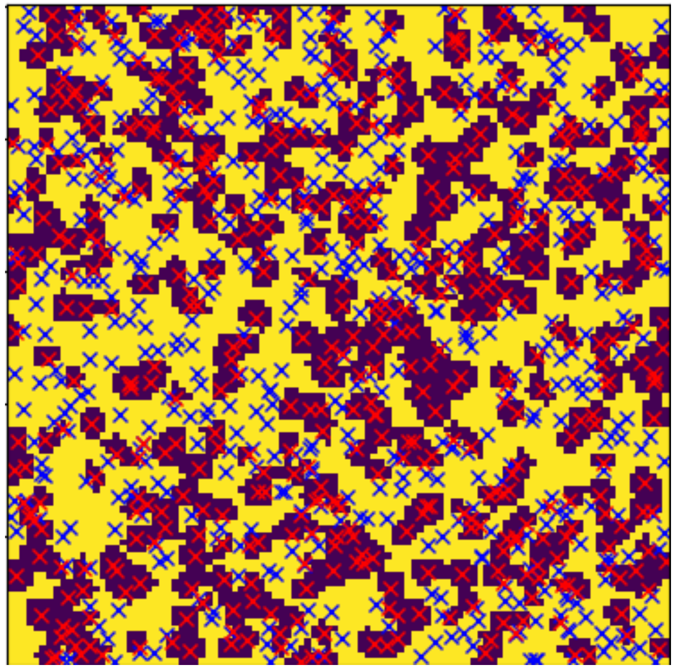
Rbf, σ = 1.0, C= 0.1

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE



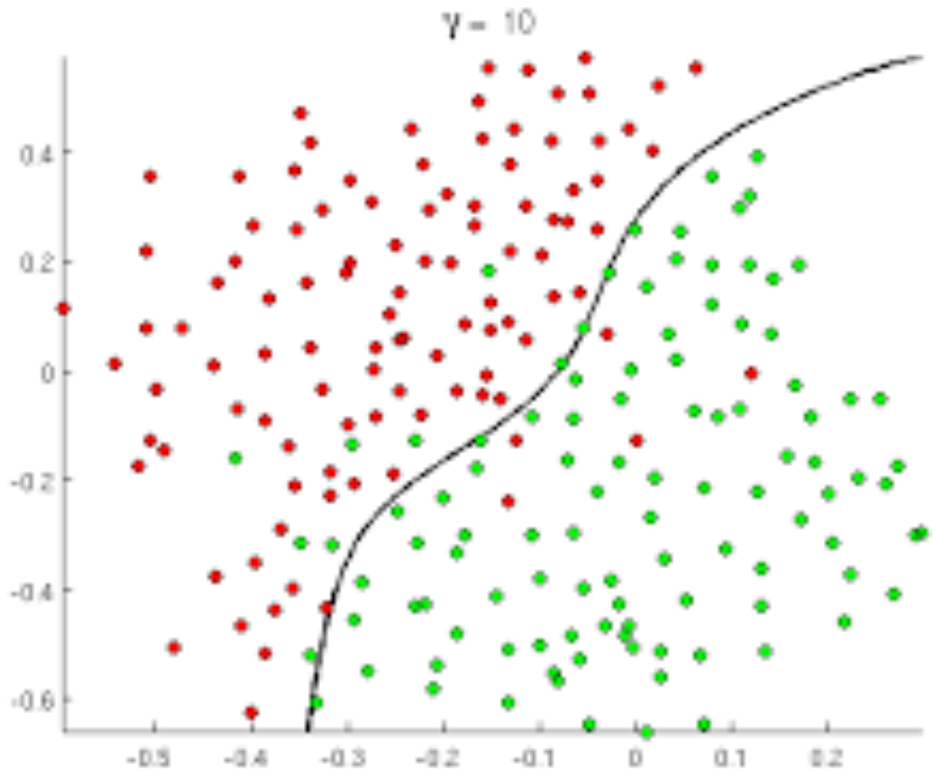

Rbf, σ = 1.0, C = 1.0


Rbf, σ = 0.1, C = 1.0


Rbf, σ = 0.05, C = 1.0

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Non-Separable Distributions

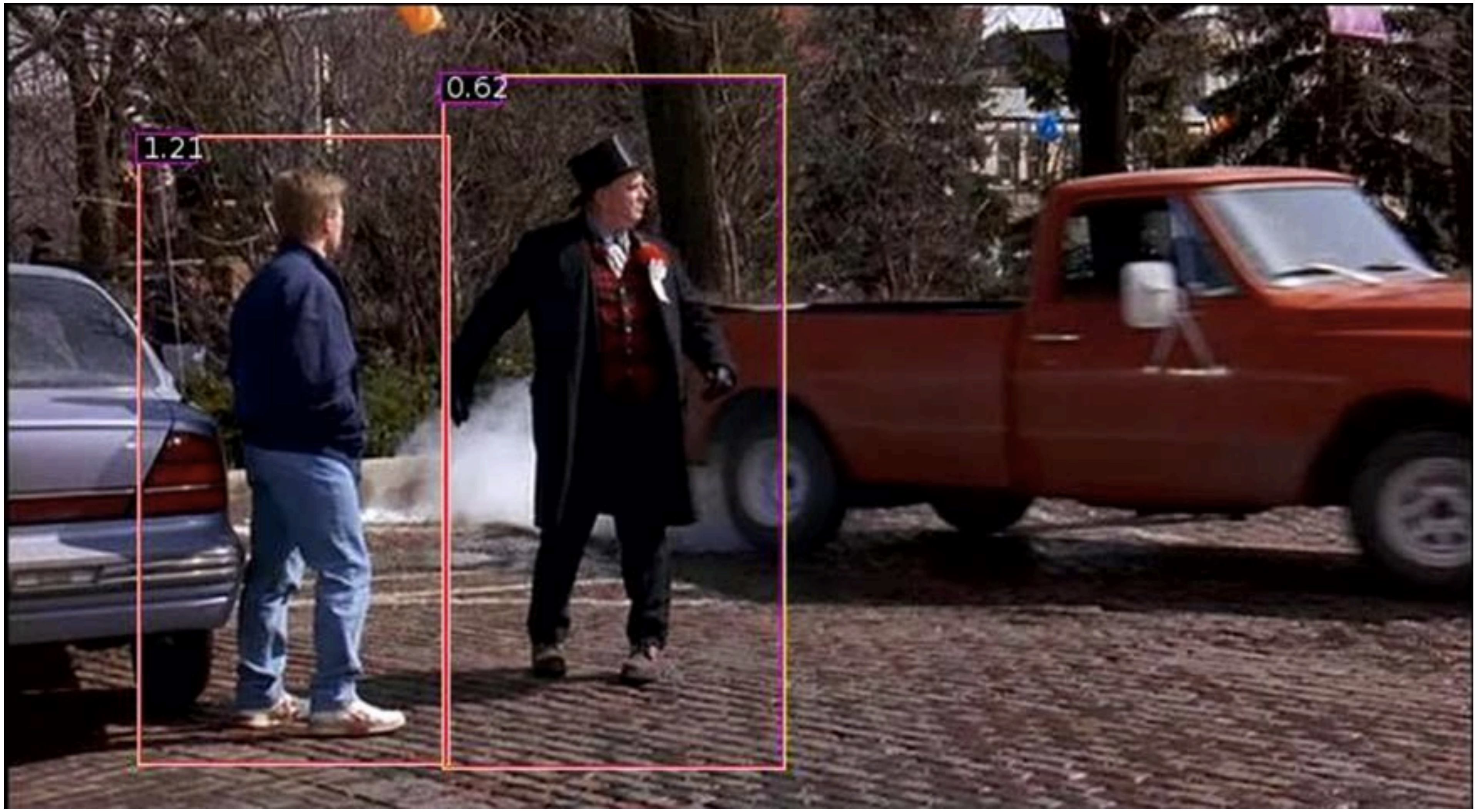


The slack variables allow some training points to be misclassified.

- A large σ sigma tends to smooth the decision boundary.
- A large C tends to minimize the number of misclassified training points.

—> Validation data is required to choose them properly.

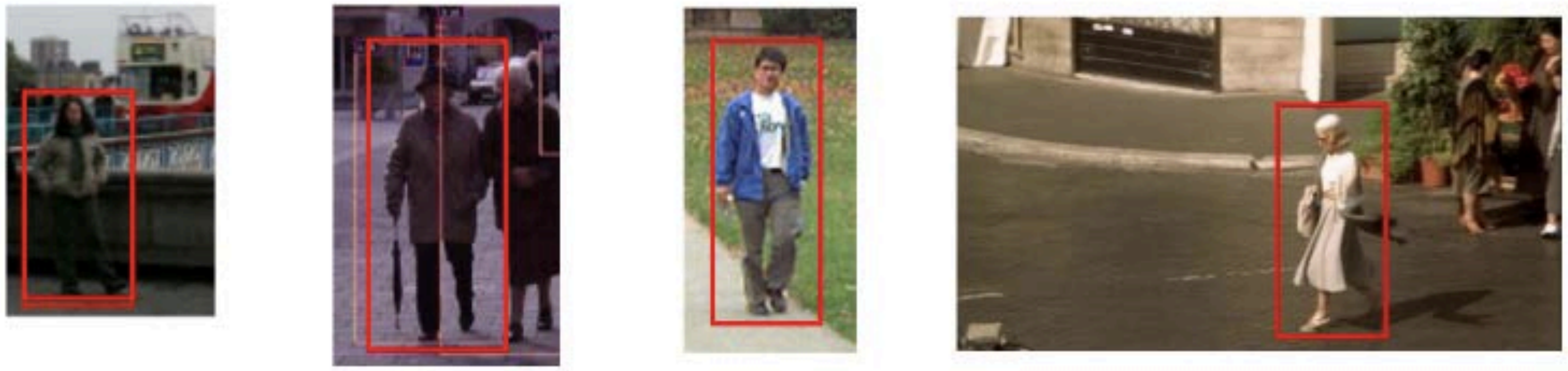
People Detection



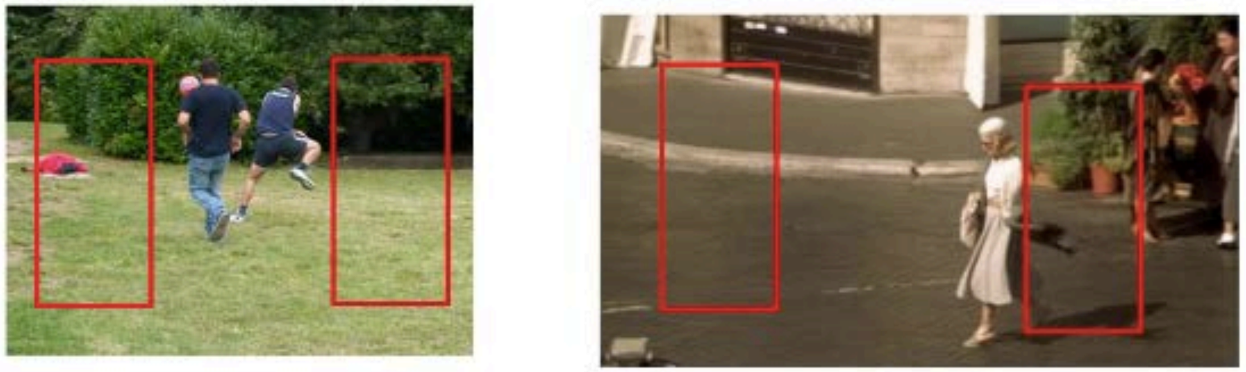
ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Training Data

- Positive data – 1208 positive window examples

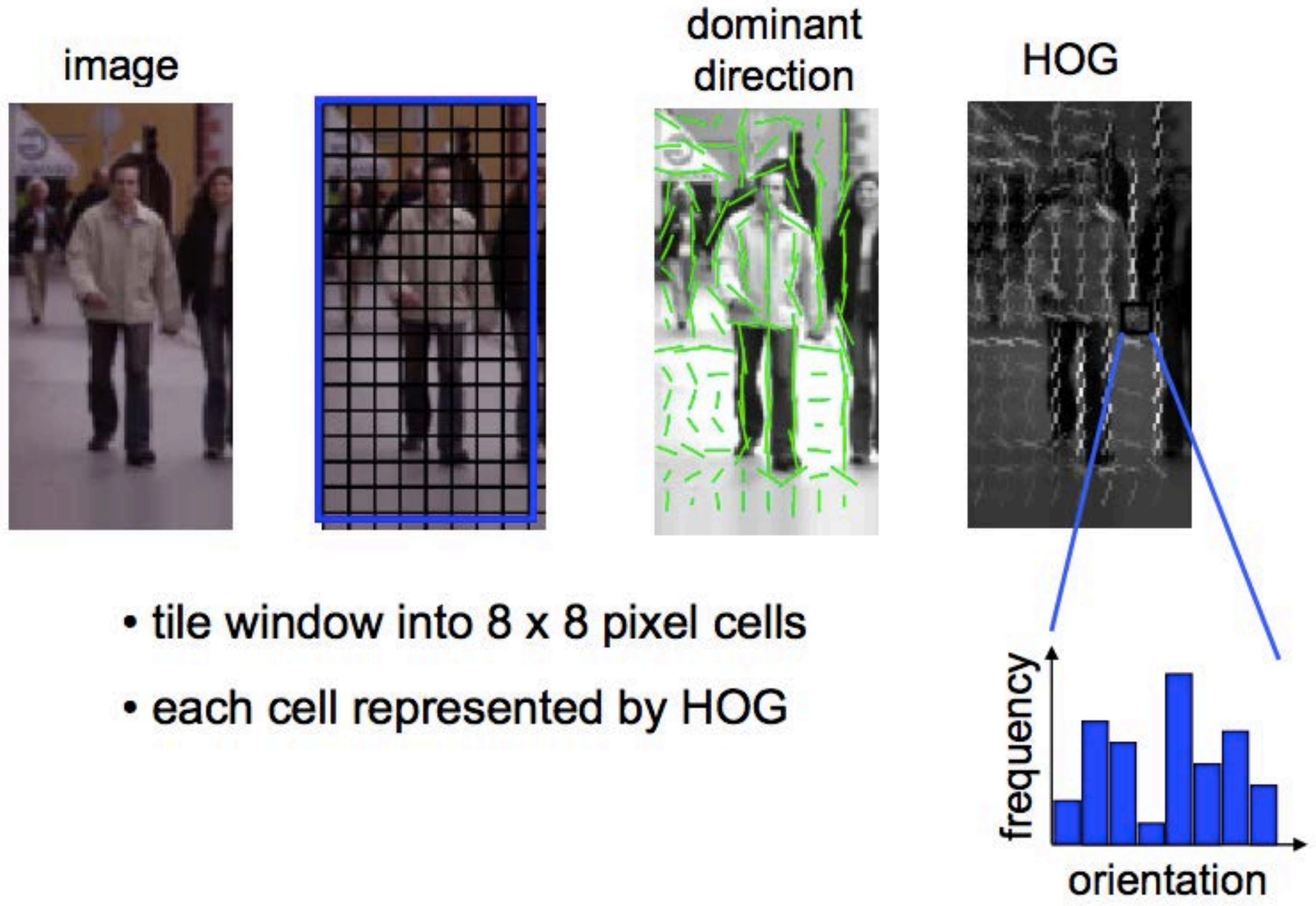


- Negative data – 1218 negative window examples (initially)



ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Histogram of Oriented Gradients



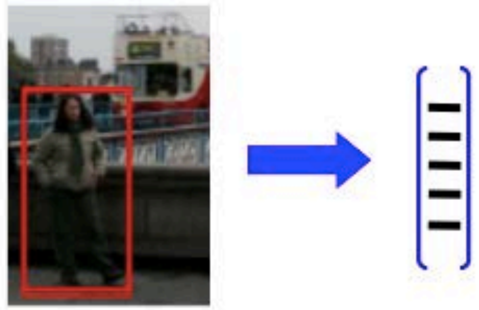
- tile window into 8 x 8 pixel cells
- each cell represented by HOG

Feature vector dimension = 16 x 8 (for tiling) x 8 (orientations) = 1024

Algorithm

Training (Learning)

- Represent each example window by a HOG feature vector



$$\mathbf{x}_i \in \mathbb{R}^d, \text{ with } d = 1024$$

- Train a SVM classifier

Testing (Detection)

- Sliding window classifier

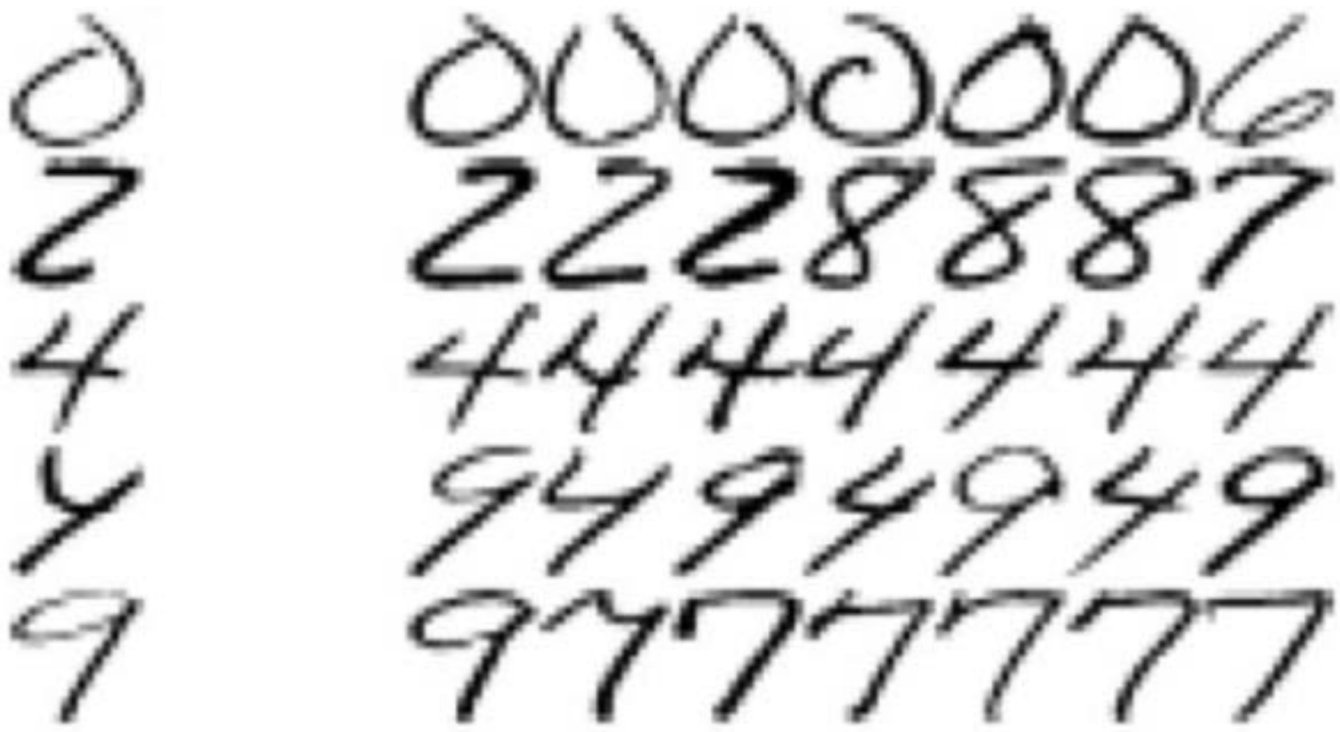
$$f(x) = \mathbf{w}^\top \mathbf{x} + b$$

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Recognizing Hand-Written Digits

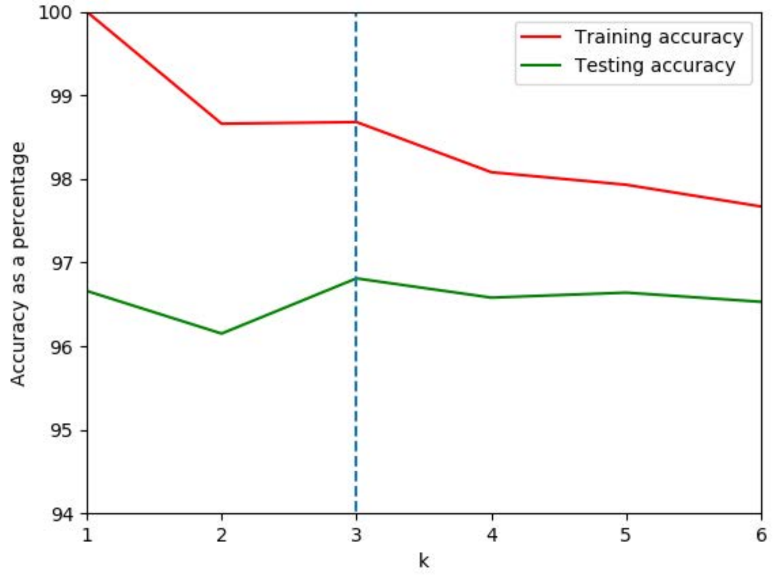
Test sample

Nearest neighbors

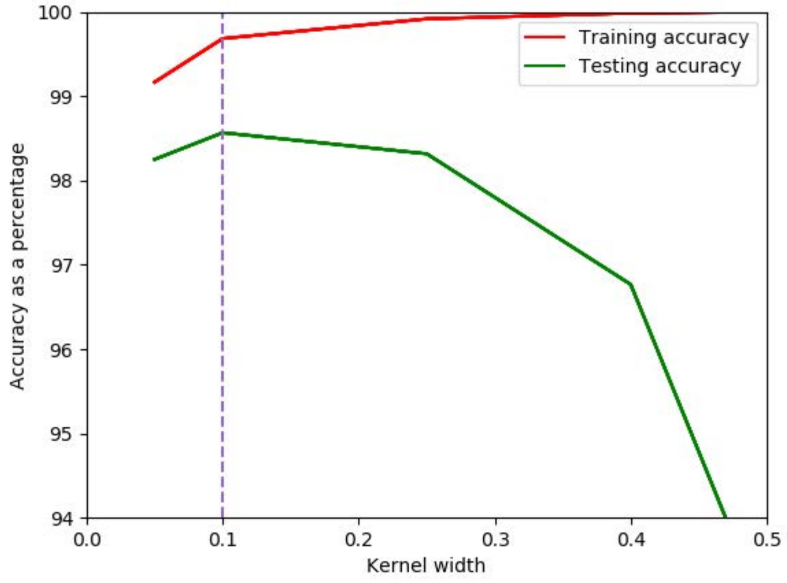


ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

k-Nearest Neighbors vs SVM on MNIST



Knn: 96.8%



Rbf-SVM: 98.6%

- Better accuracy.
- But the kernel and its parameters must be well chosen.

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

SVMs in Short

- The data can be separable in a high-dimensional feature space without being separable in the input space.
- Classifiers can be learned in the feature space without having to actually perform the mapping.
- However the $O(N^3)$ complexity at training time makes it hard to exploit large training sets.

Active Learning

- Train using a relatively small of the training set.
- Progressively add new samples and retrain.
- The simplest strategy is to add training samples that are close to the decision boundary to progressively refine its shape.

—> One approach among others to tackling the computational complexity issue.