

IFT6390

Fondements de l'apprentissage machine

Beyond Linear Classifiers

First solution:

Explicit non-linear mappings

Professor: Ioannis Mitliagkas

Slides: Pascal Vincent

How can we build a *non-linear* classifier using a linear classifier?

An old trick...

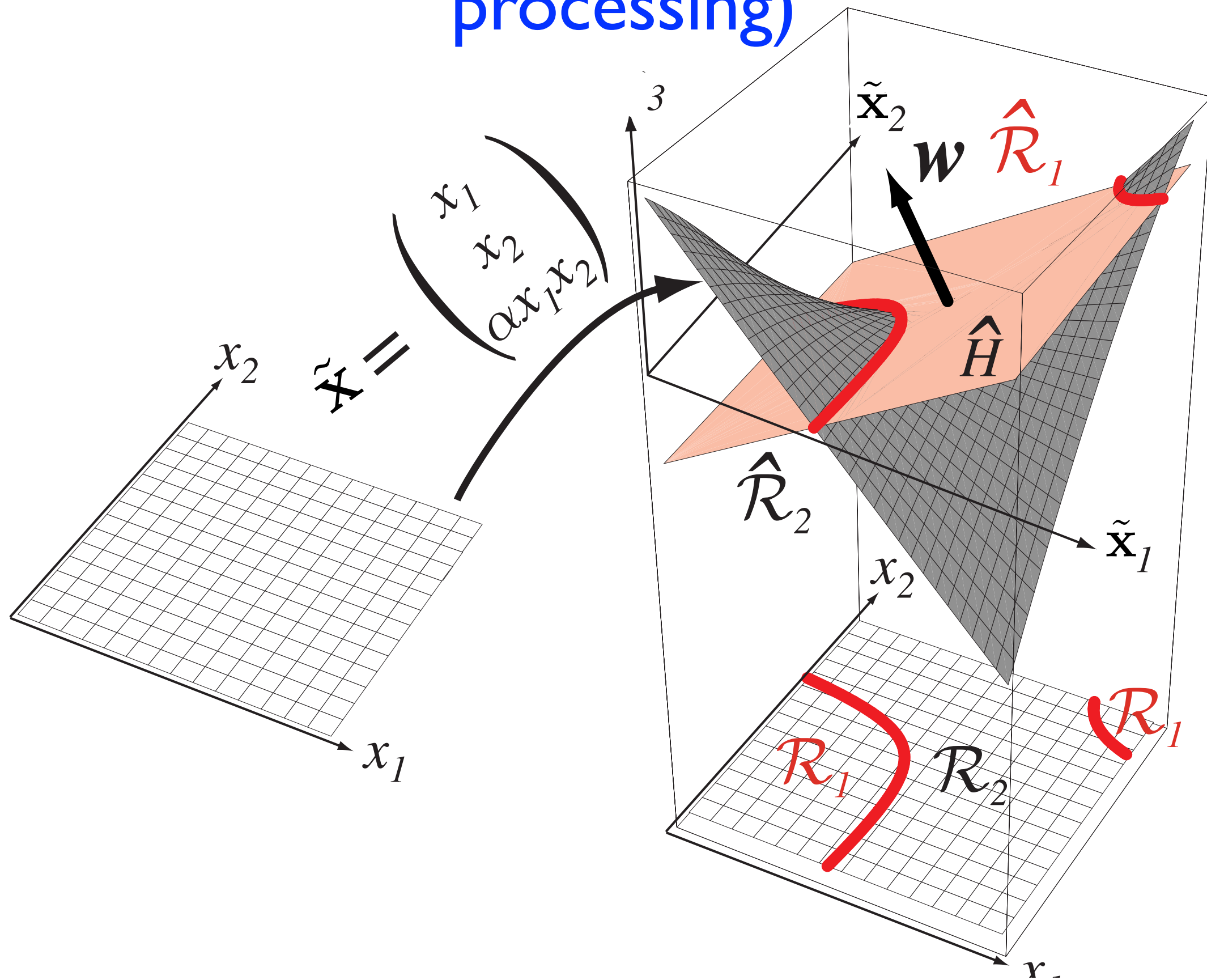
- First, **apply a non-linear transform to x** (projection into a new feature space)

$$\tilde{\mathbf{x}} = \phi(\mathbf{x})$$

 non-linear feature map

- Learn a linear discriminant function in this new space (using one of the many algorithms we saw for learning linear classifiers)
- The separating hyperplane in this new space corresponds to a **non-linear decision boundary in the original space!**
- And voilà! \Rightarrow **A non-linear classifier (with increased capacity).**

Ex. *a priori* linear transform (pre-processing)



Ex. *a priori* linear transform (pre-processing)

How to build a *polynomial* classifier (or regressor) using a linear classifier (or regressor)?

What is a *linear (affine)* function of x (e.g. in 2D)?

$$y = f_{\theta}([x_1, x_2]) = w_1x_1 + w_2x_2 + b$$

What is a *degree 2 polynomial* of x (e.g. in 2D)?

$$\text{poly}_2([x_1, x_2]) = a_{11}x_1^2 + a_{22}x_2^2 + a_{12}x_1x_2 + a_1x_1 + a_2x_2 + b$$

What is a *linear (affine)* function of x (e.g. in 2D)?

...

Ex. *a priori* linear transform (pre-processing)

How to build a *polynomial* classifier/regressor using a linear classifier/regressor?

- First **apply some non-linear feature map to x** (projection into a new feature space)

$$\tilde{x} = \varphi(x) = \varphi([x_1, x_2]) = [x_1^2, x_2^2, x_1 x_2, x_1, x_2]$$

- Then, **learn a linear discriminant function in this new space** (using one of the many algorithms we saw for learning linear classifiers)

$$f(x) = w^T \tilde{x} + b = w_1 \underbrace{\tilde{x}_1}_{x_1^2} + w_2 \underbrace{\tilde{x}_2}_{x_2^2} + w_3 \underbrace{\tilde{x}_3}_{x_1 x_2} + w_4 \underbrace{\tilde{x}_4}_{x_1} + w_5 \underbrace{\tilde{x}_5}_{x_2} + b$$

- The separating hyperplane in this new space corresponds to a **non-linear decision boundary in the original space!**
- And voilà! \Rightarrow **A non-linear classifier (with increased capacity).**

But which transformation?

Three ways to obtain a non-linear feature map

- **Explicitly** choose a **fixed** transformation *a priori*
 - ➡ Previous example
- **Implicitly** choose a **fixed** transformation *a priori*
 - ➡ Kernel method (kernelized SVM, kernelized logistic regression, ...)
- **Learn** the parameters of a parametrized feature map:
 - ➡ Neural networks
 - Multilayer perceptron (MLP)