Machine Learning
- Aula 3

1 - Regularização

Função Objetivo

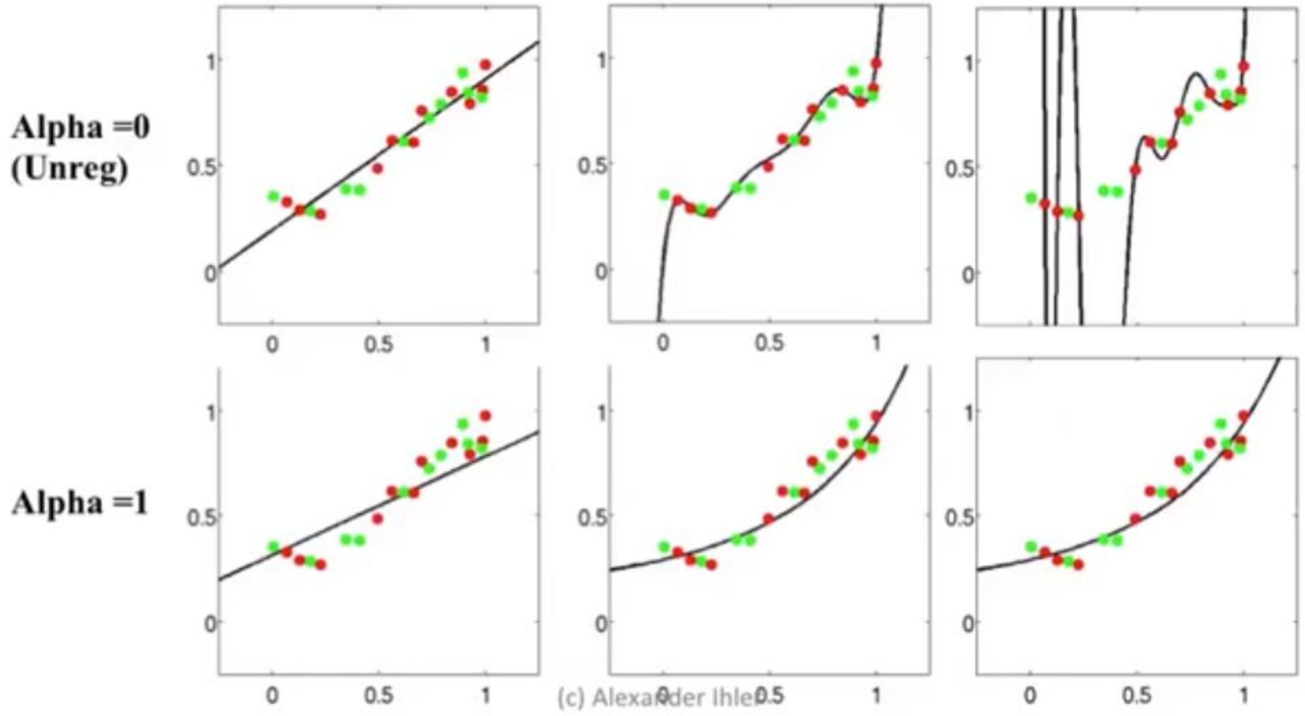
$$\mathrm{Obj}(\Theta) = J(\Theta) + \boxed{R(\Theta)}$$

$$= \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 + \boxed{\lambda \sum_{j=1}^{n} \theta_j^2}\right]$$

(Exemplo)

Regularization

- Compare between unreg. & reg. results

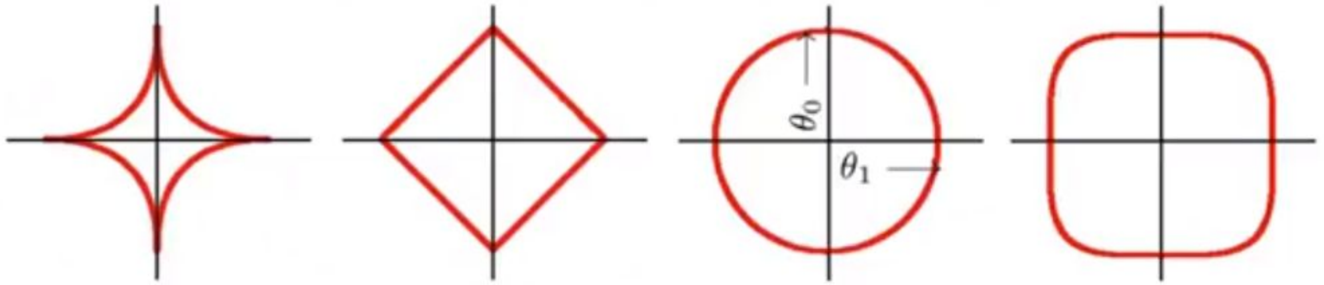


Linear regression (6): Regularization
https://www.youtube.com/watch?v=sO4ZirJh9ds&t=1s

Different regularization functions

- More generally, for the L_p regularizer: $\left(\sum_i |\theta_i|^p \right)$

Isosurfaces: $\|\theta\|_p = \text{constant}$



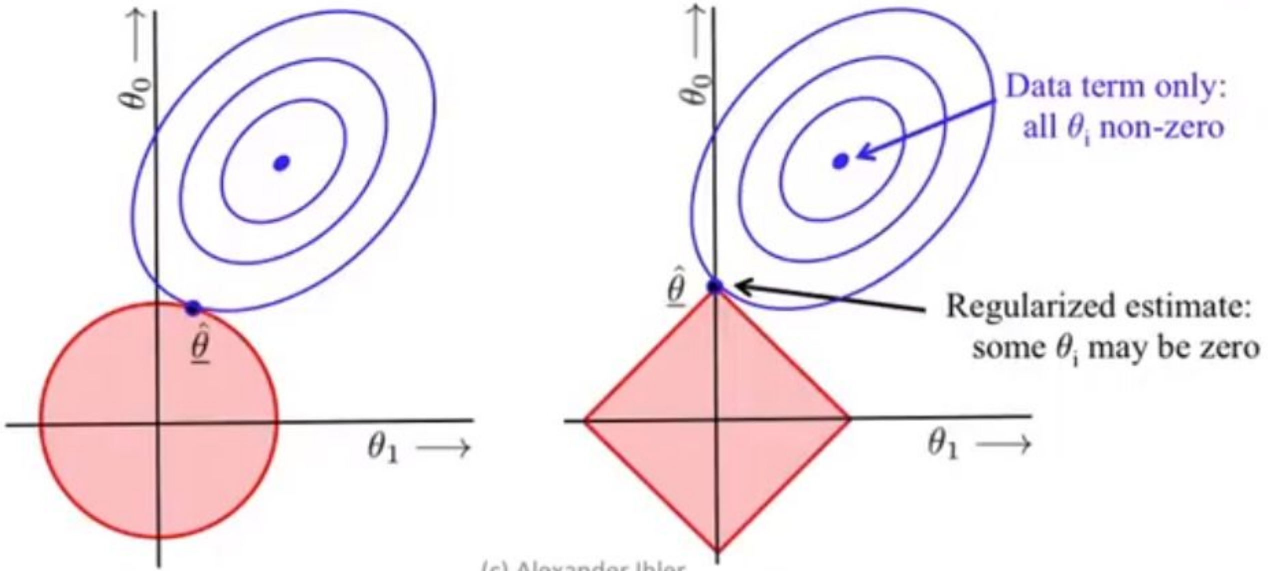
p=0.5 p=1 Lasso p=2 Quadratic p=4

L_0 = limit as $p \to 0$: “number of nonzero weights”, a natural notion of complexity

(c) Alexander Ihler

Regularization: L1 vs L2

- Estimate balances data term & regularization term
- Lasso tends to generate sparser solutions than a quadratic regularizer.

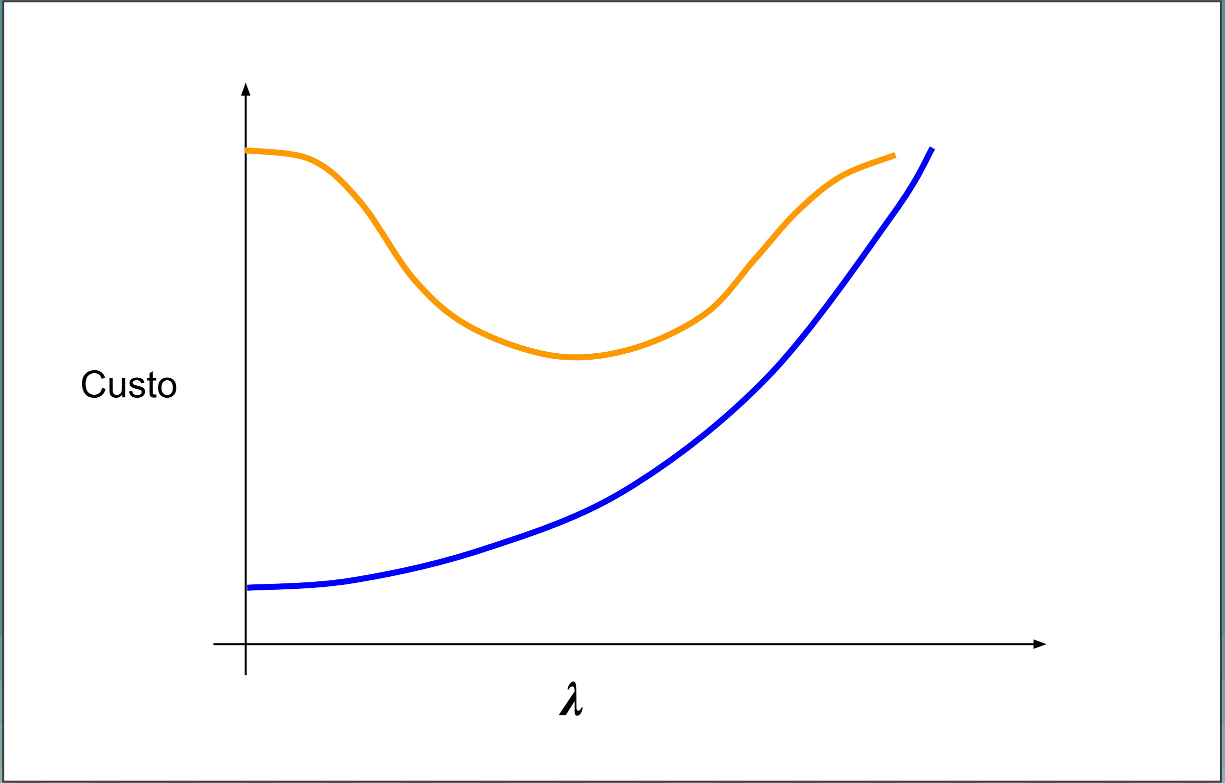


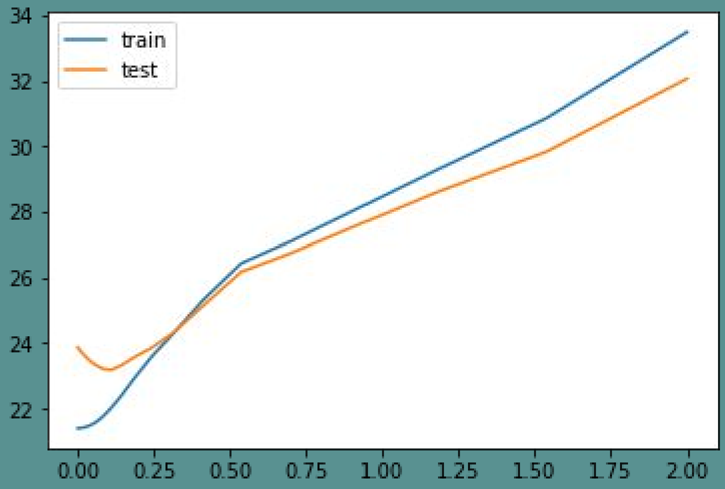
$\theta_0 \longrightarrow$
$\hat{\underline{\theta}}$
$\theta_1 \longrightarrow$
$\theta_0 \longrightarrow$
Data term only: all θ_i non-zero
$\hat{\underline{\theta}}$
Regularized estimate: some θ_i may be zero
$\theta_1 \longrightarrow$

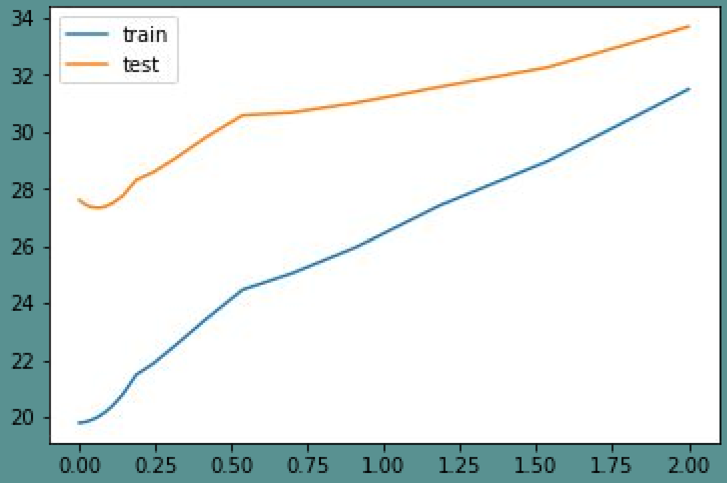
(c) Alexander Ihler

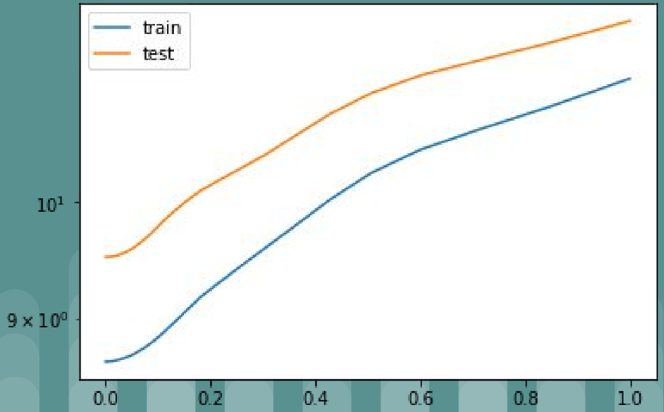
Linear regression (6): Regularization
https://www.youtube.com/watch?v=sO4ZirJh9ds&t=1s

Achando λ



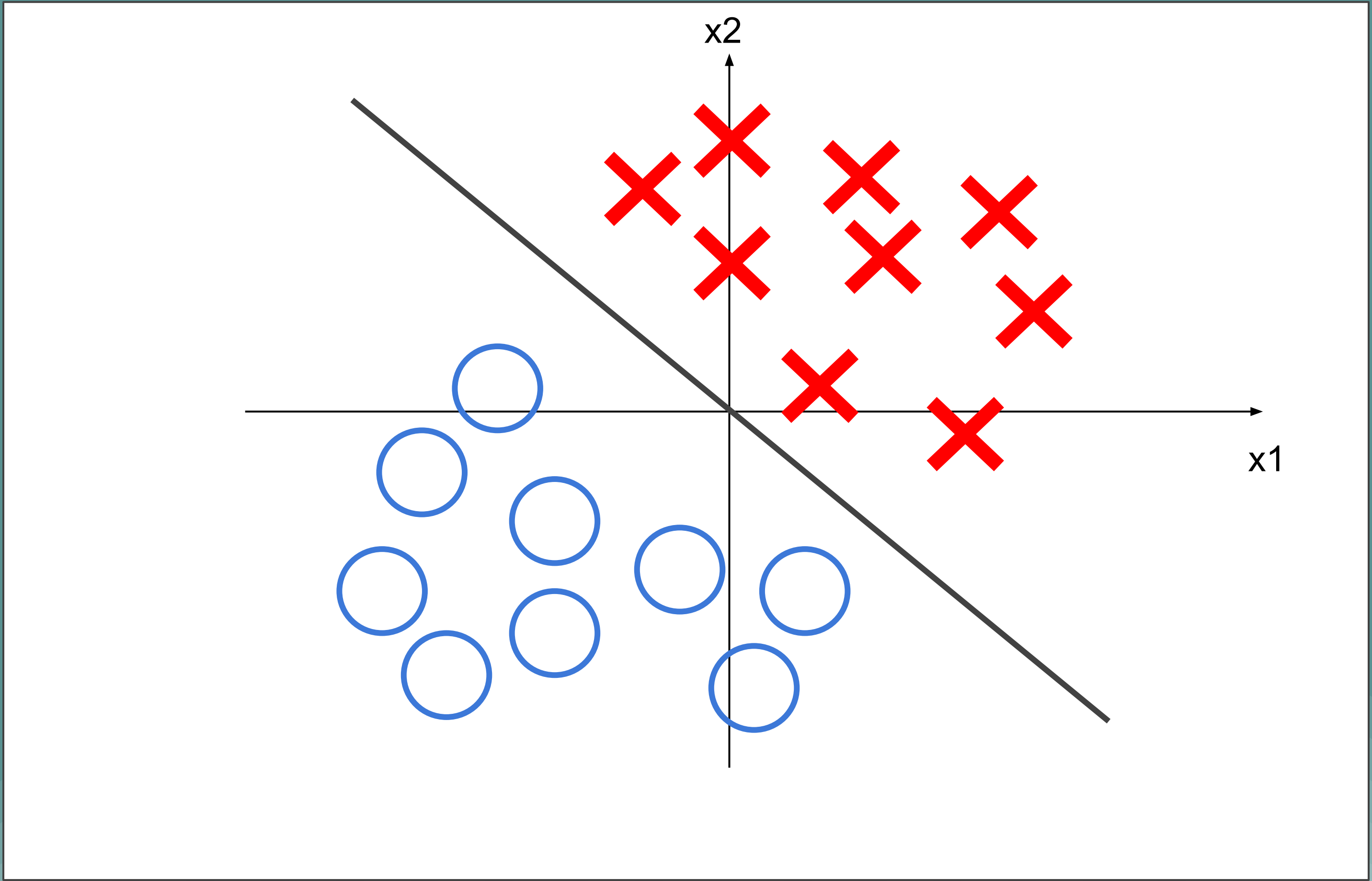

34
32
30
28
26
24
22
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
train
test


34
32
30
28
26
24
22
20
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
train
test

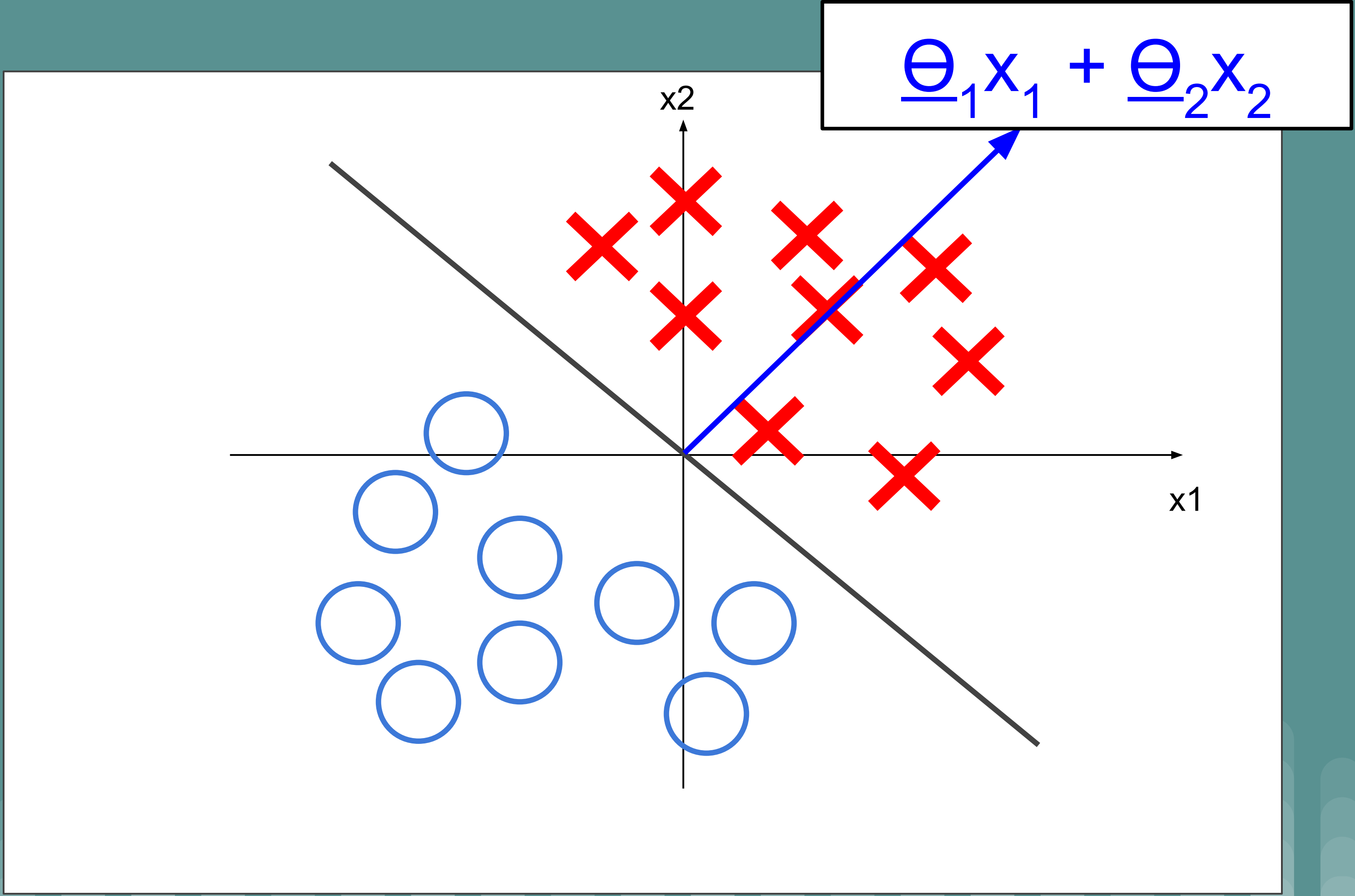

10^1
9×10^0
0.0
0.2
0.4
0.6
0.8
1.0
train
test

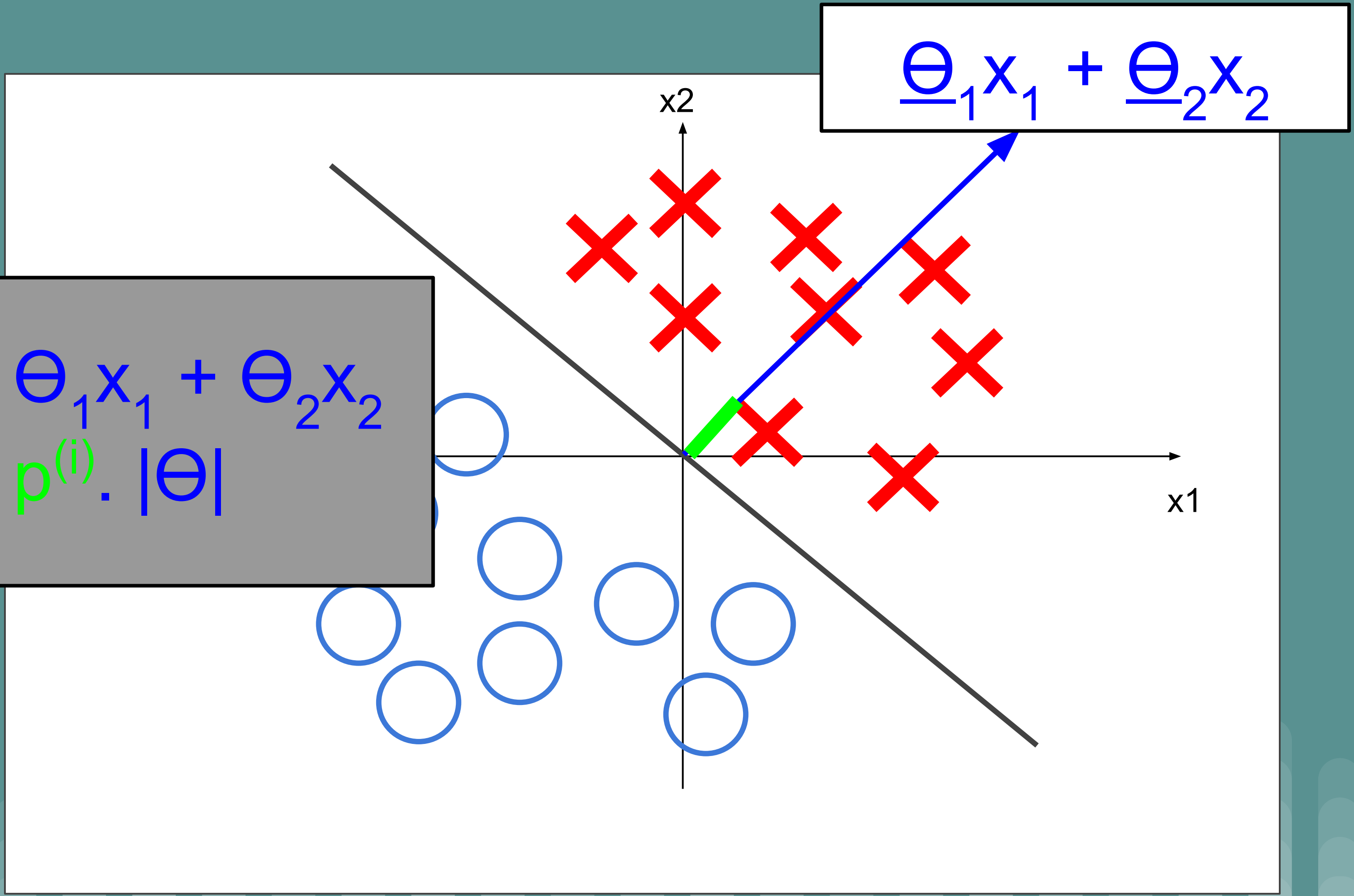
2 - SVM

2.1 - Classificação

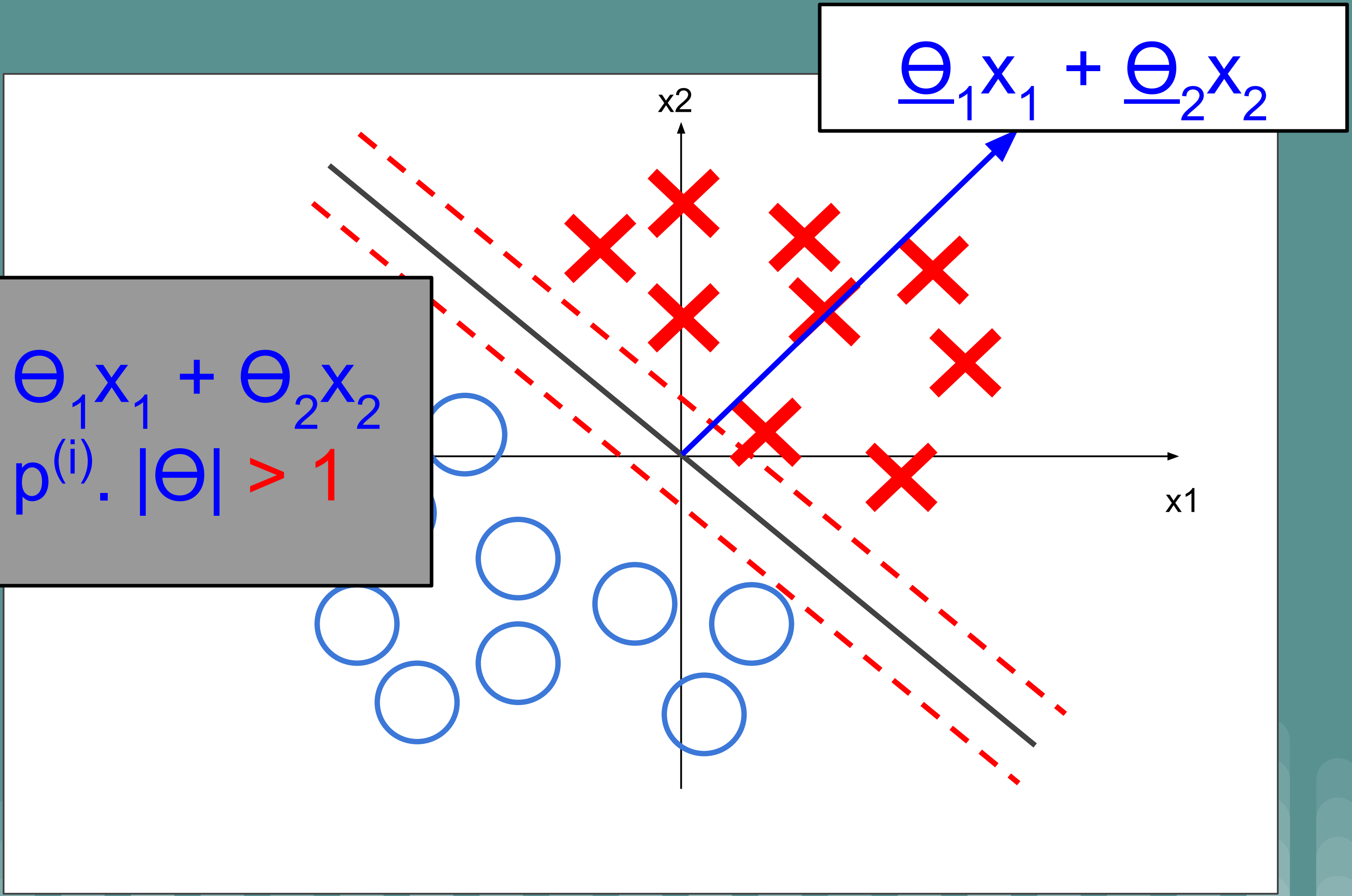

x2
x1

$$\underline{\theta}_1 x_1 + \underline{\theta}_2 x_2$$





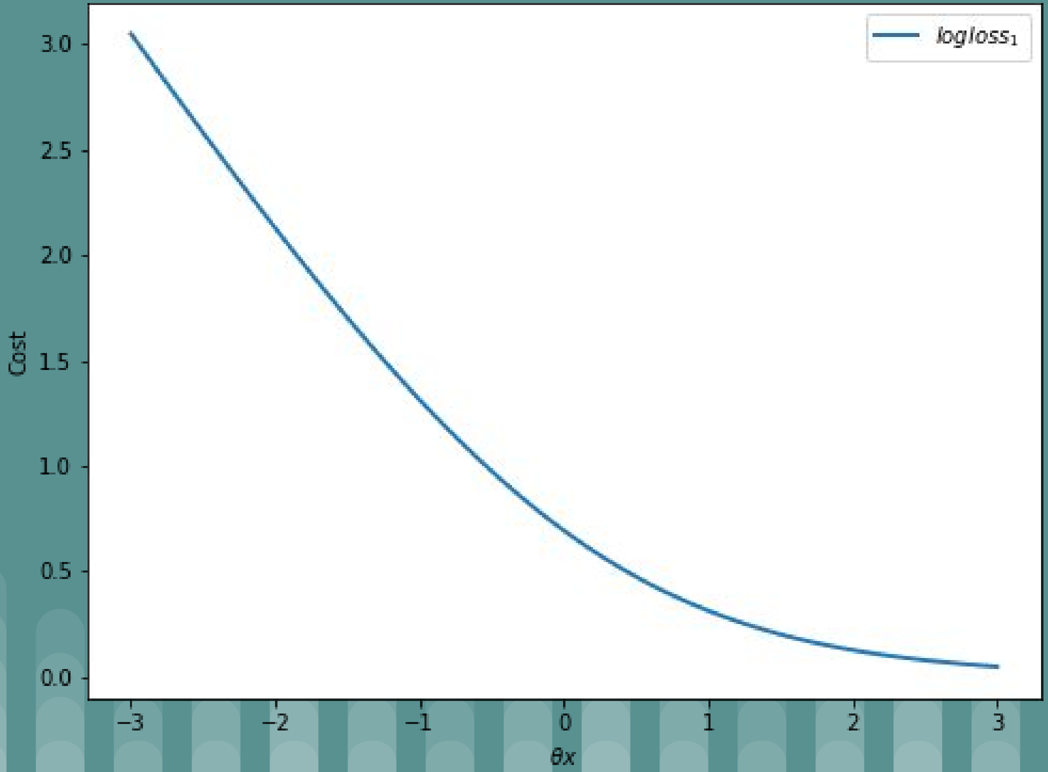
$$\mathbf{x}.\boldsymbol{\theta} = \theta_1 x_1 + \theta_2 x_2$$
$$\mathbf{x}.\boldsymbol{\theta} = p^{(i)}.\ |\theta|$$



$\mathbf{x.\Theta} = \Theta_1 x_1 + \Theta_2 x_2$

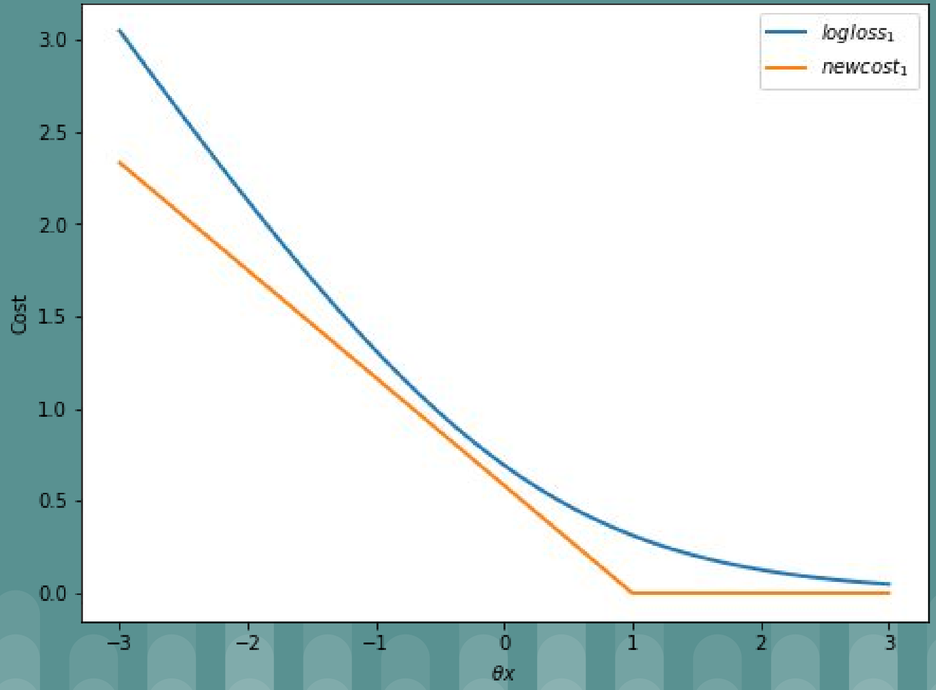
$\mathbf{x.\Theta} = p^{(i)}.\ |\Theta| > 1$

$$logloss := -((y)\log(\frac{1}{1+e^{-\theta x}}) + (1-y)\log(1-\frac{1}{1+e^{-\theta x}}))$$



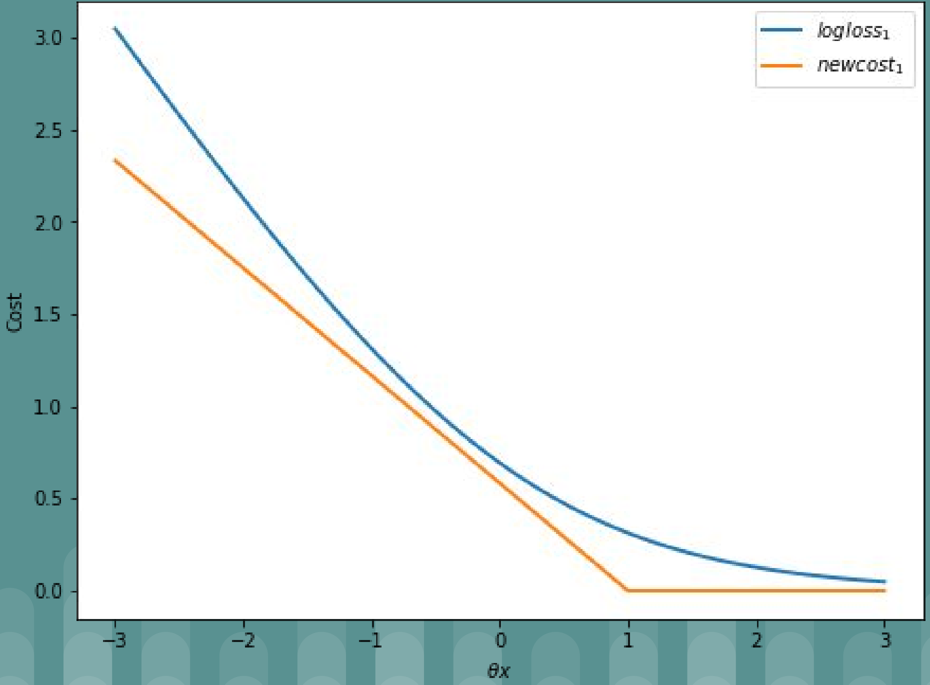
$$logloss := -((y)\log(\frac{1}{1+e^{-\theta x}}) + (1-y)\log(1-\frac{1}{1+e^{-\theta x}})$$

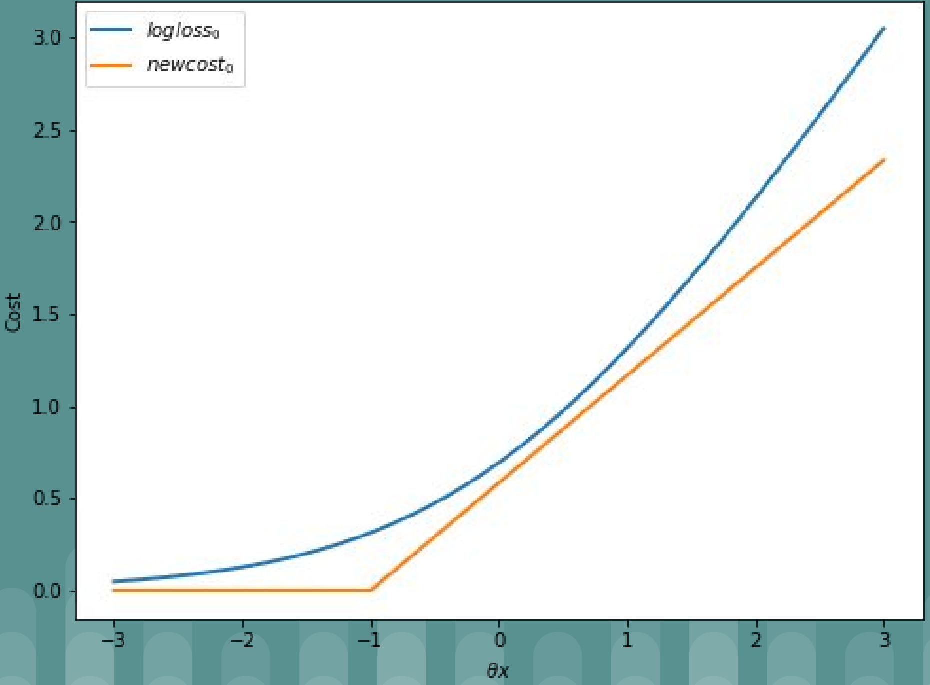
$$newcost := -((y)cost_1(\theta x) + (1-y)cost_0(\theta x))$$

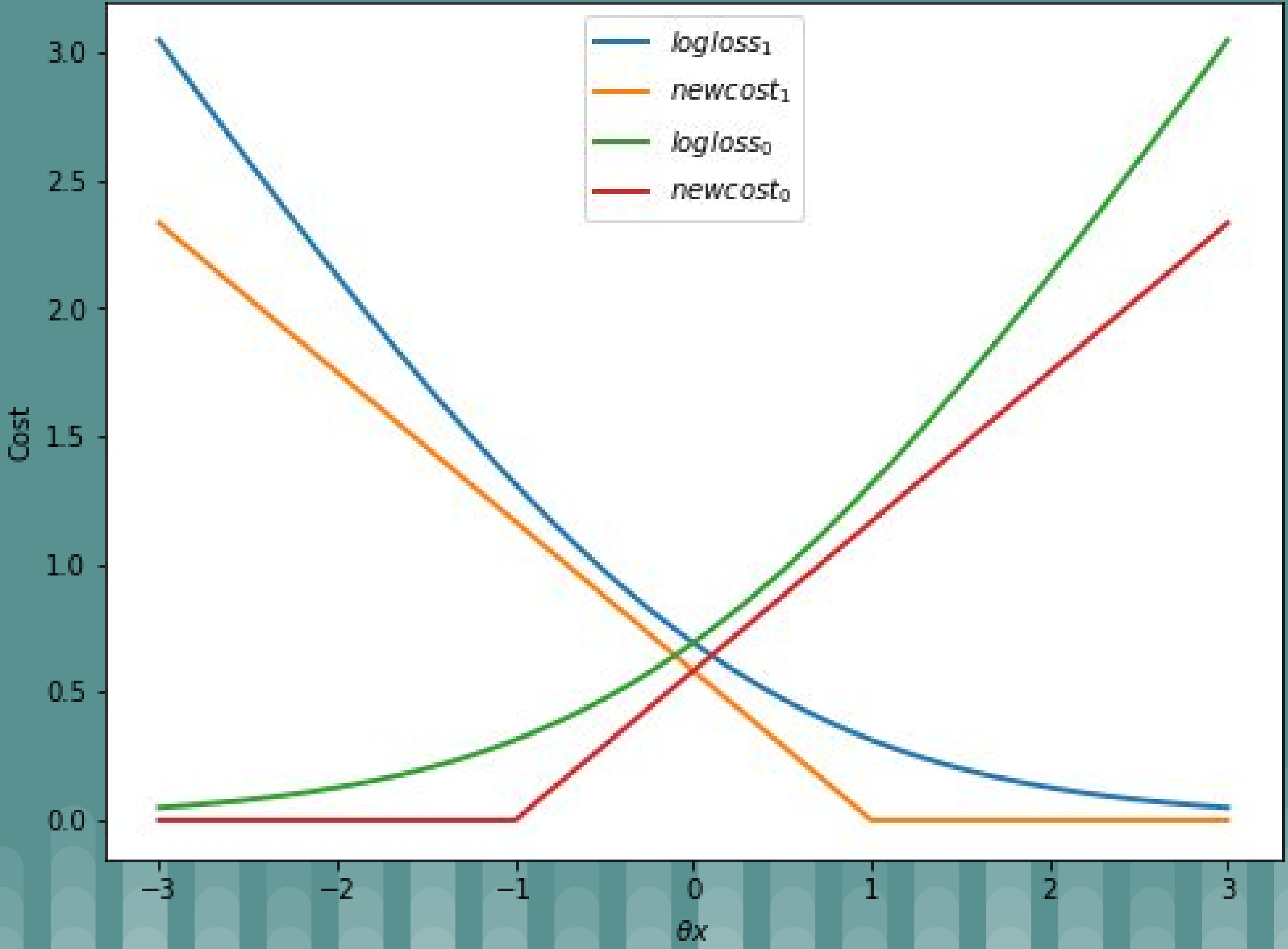


$$logloss := -((y)\log(\frac{1}{1+e^{-\theta x}}) + (1-y)\log(1-\frac{1}{1+e^{-\theta x}})$$

$$newcost := -((y)cost_1(\theta x) + (1-y)cost_0(\theta x))$$







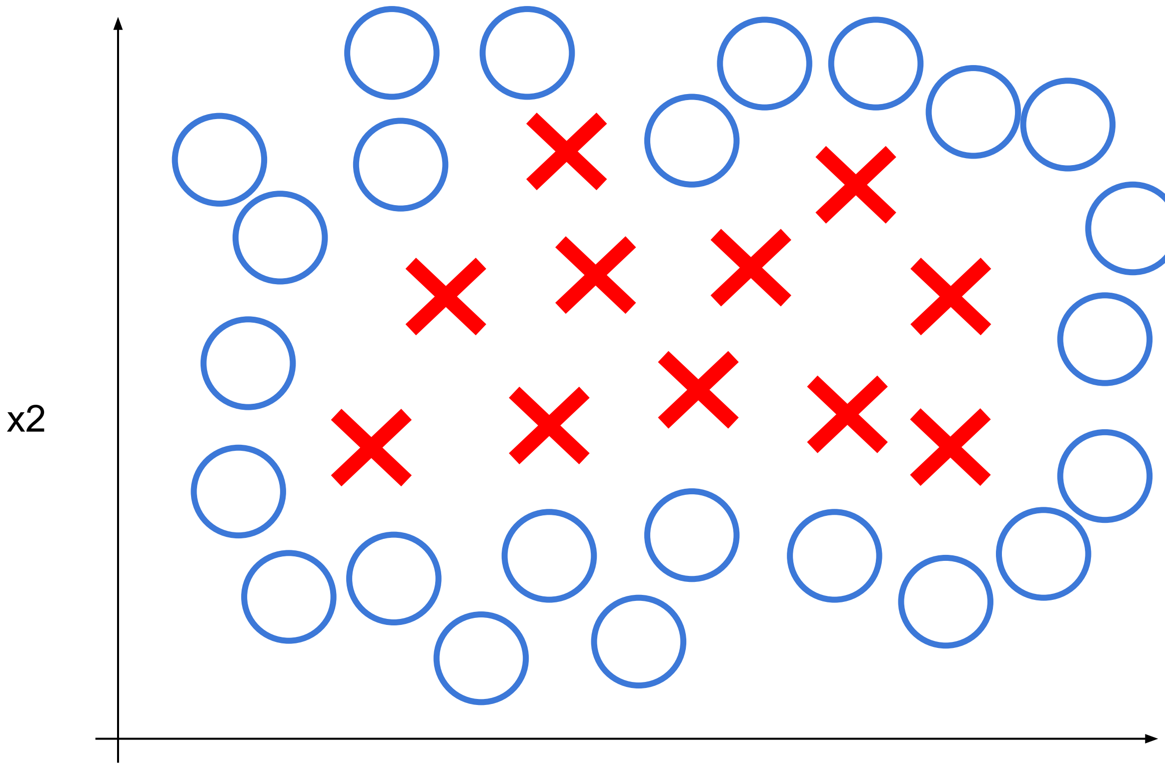
$logloss_1$
$newcost_1$
$logloss_0$
$newcost_0$
Cost
3.0
2.5
2.0
1.5
1.0
0.5
0.0
−3
−2
−1
0
1
2
3
θx

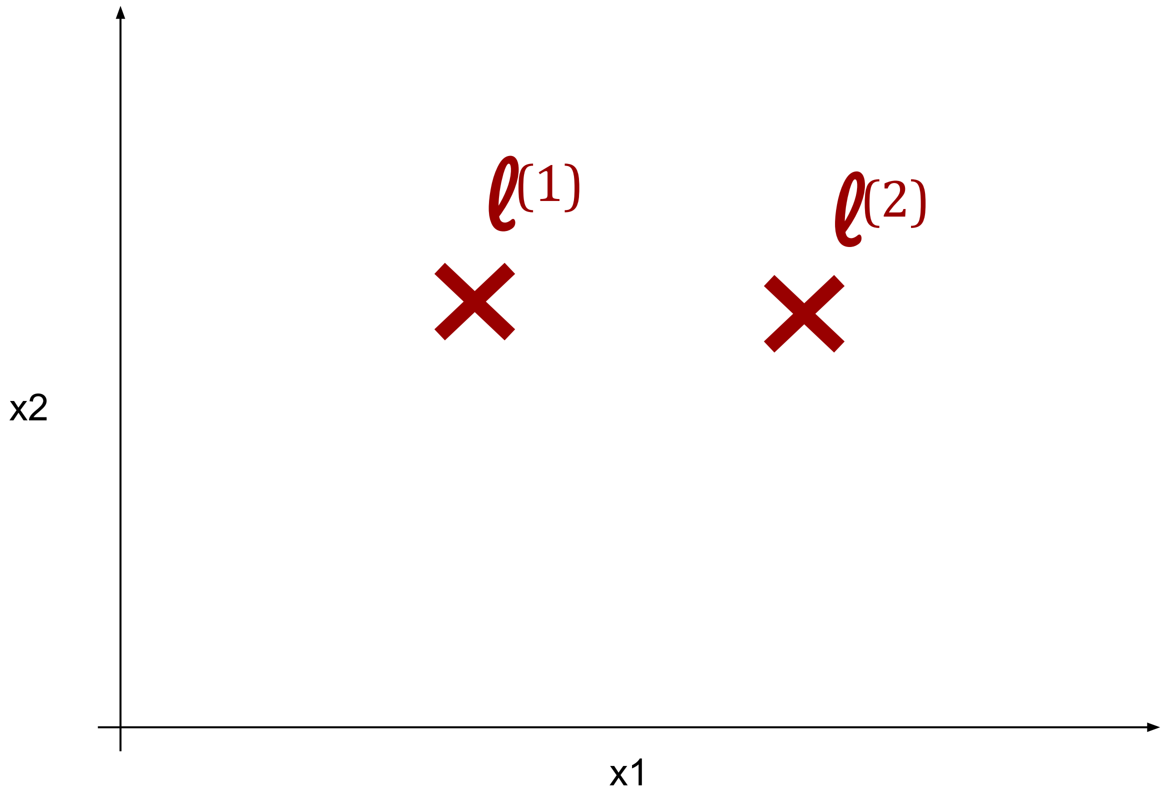
$$\min_{\theta} \frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)}\Big(-\log(g(h_\theta(x^{(i)})))\Big) + (1-y^{(i)})\Big(-\log(1-g(h_\theta(x^{(i)})))\Big)\right] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

$$\downarrow \qquad\qquad\qquad \downarrow$$

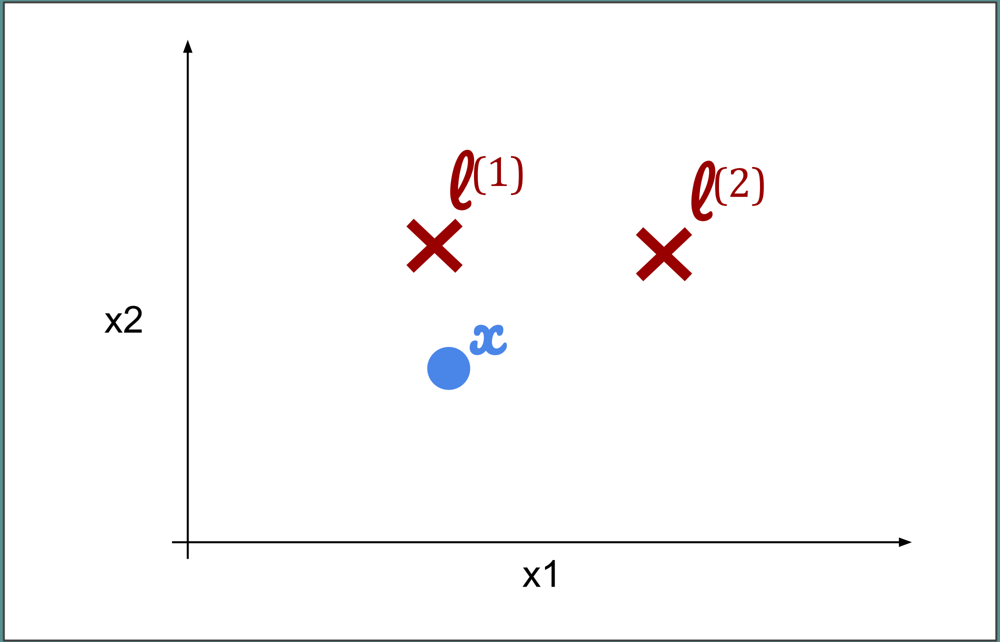
$$\min_{\theta} C\left[\sum_{i=1}^{m} y^{(i)} cost_1(\theta x^{(i)}) + (1-y^{(i)}) cost_0(\theta x^{(i)})\right] + \frac{1}{2}\sum_{j=1}^{n}\theta_j^2$$

2.2 - Kernel

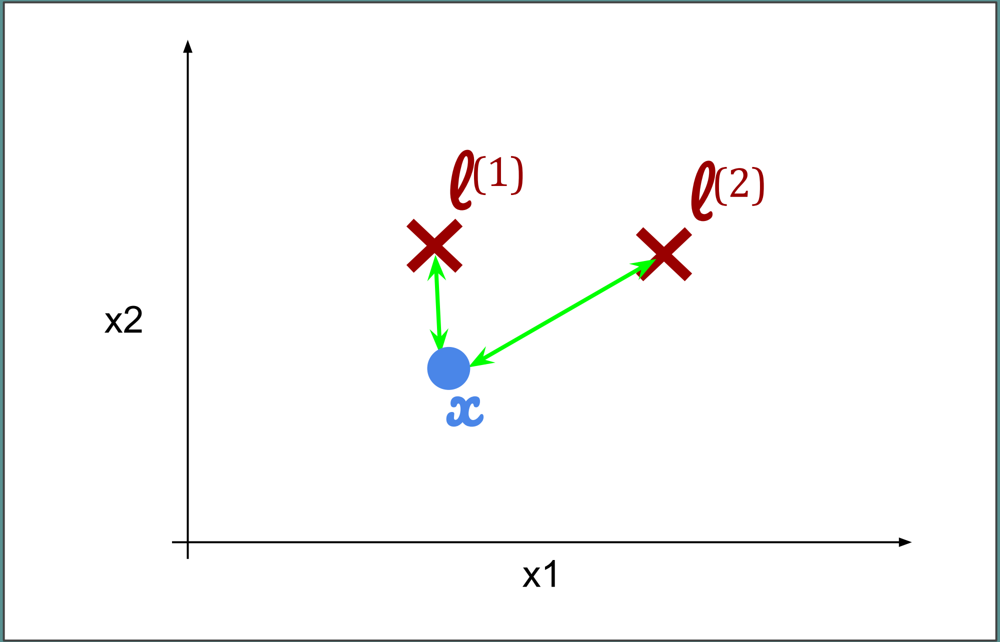

x2
x1



$\ell^{(1)}$
$\ell^{(2)}$
x2
x1



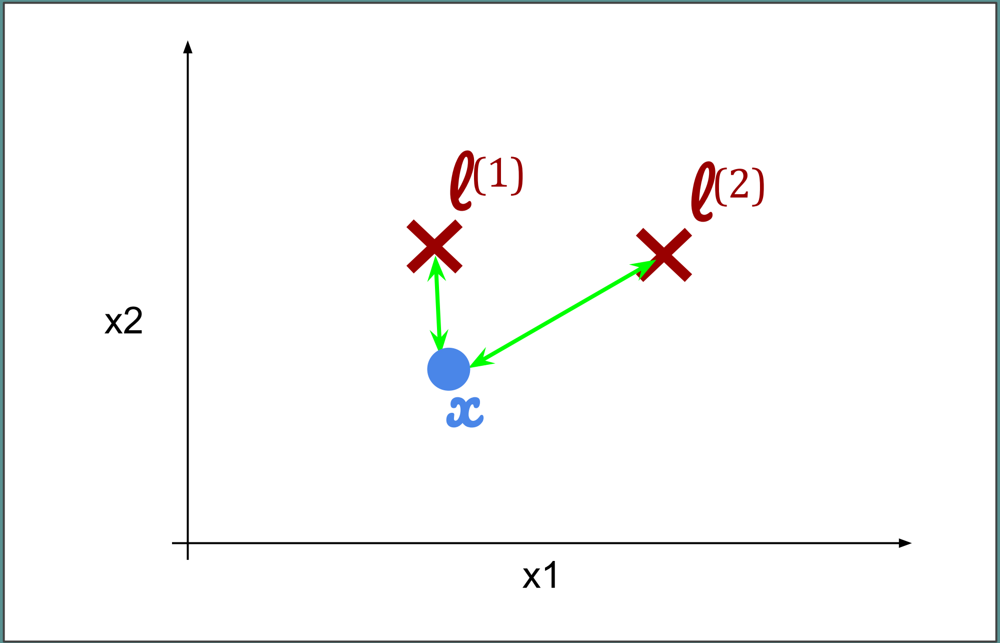
$\ell^{(1)}$
$\ell^{(2)}$
x
x2
x1



CRIM	ZN	AGE
0.004	0.0	50
0.001	1.8	65
0.012	0.0	80
...	...	...

$\rightarrow$

f_1	f_2
1.2	3.4
1.8	5.0
0.5	1.33
…	….



CRIM	ZN	AGE
0.004	0.0	50
0.001	1.8	65
0.012	0.0	80
...	...	...

f_1	f_2
0.0	50
1.8	65
0.0	80
...	....

$$f_1 = similarity(x, l^{(1)}) = \exp\left(-\frac{\|x - l^{(1)}\|^2}{2\sigma^2}\right)$$

$$f_2 = similarity(x, l^{(2)}) = \exp\left(-\frac{\|x - l^{(2)}\|^2}{2\sigma^2}\right)$$

$$\min_{\theta} C\left[\sum_{i=1}^{m} y^{(i)} cost_1(\theta x^{(i)}) + (1 - y^{(i)}) cost_0(\theta x^{(i)})\right] + \frac{1}{2}\sum_{j=1}^{n}\theta_j^2$$

$$\downarrow \qquad\qquad\qquad \downarrow$$

$$\min_{\theta} C\left[\sum_{i=1}^{m} y^{(i)} \underbrace{cost_1(\theta f^{(i)}) + (1 -}_{\mathbf{f.\Theta} = \Theta_0 + \Theta_1 f_1 + \Theta_2 f_2} y^{(i)}) cost_0(\theta f^{(i)})\right] + \frac{1}{2}\sum_{j=1}^{n}\theta_j^2$$

Onde colocar os centroides?

2.3 - Regressão

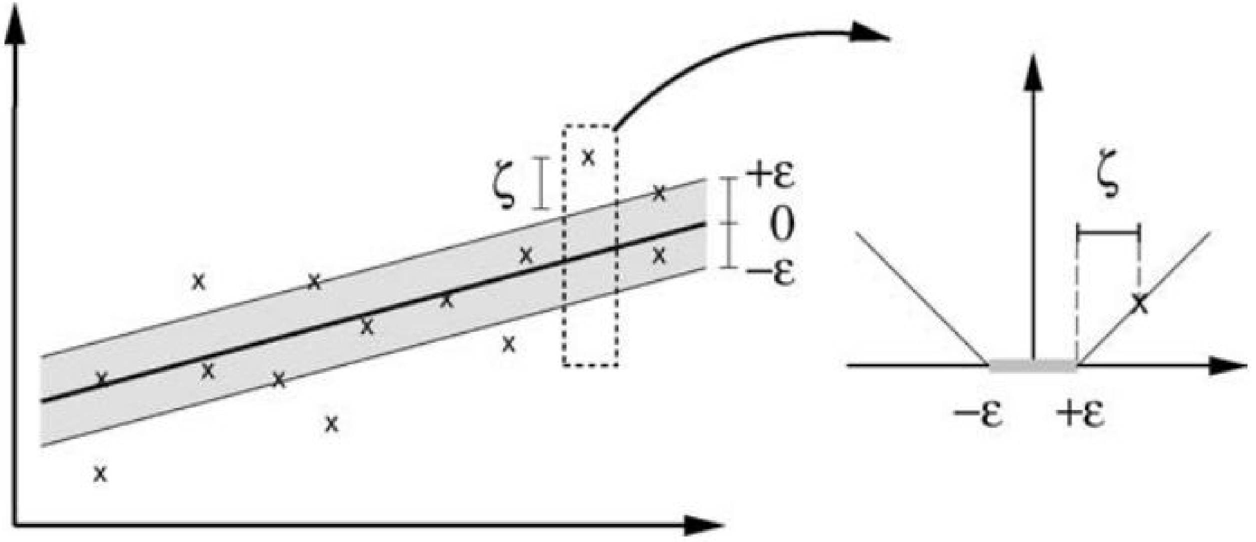
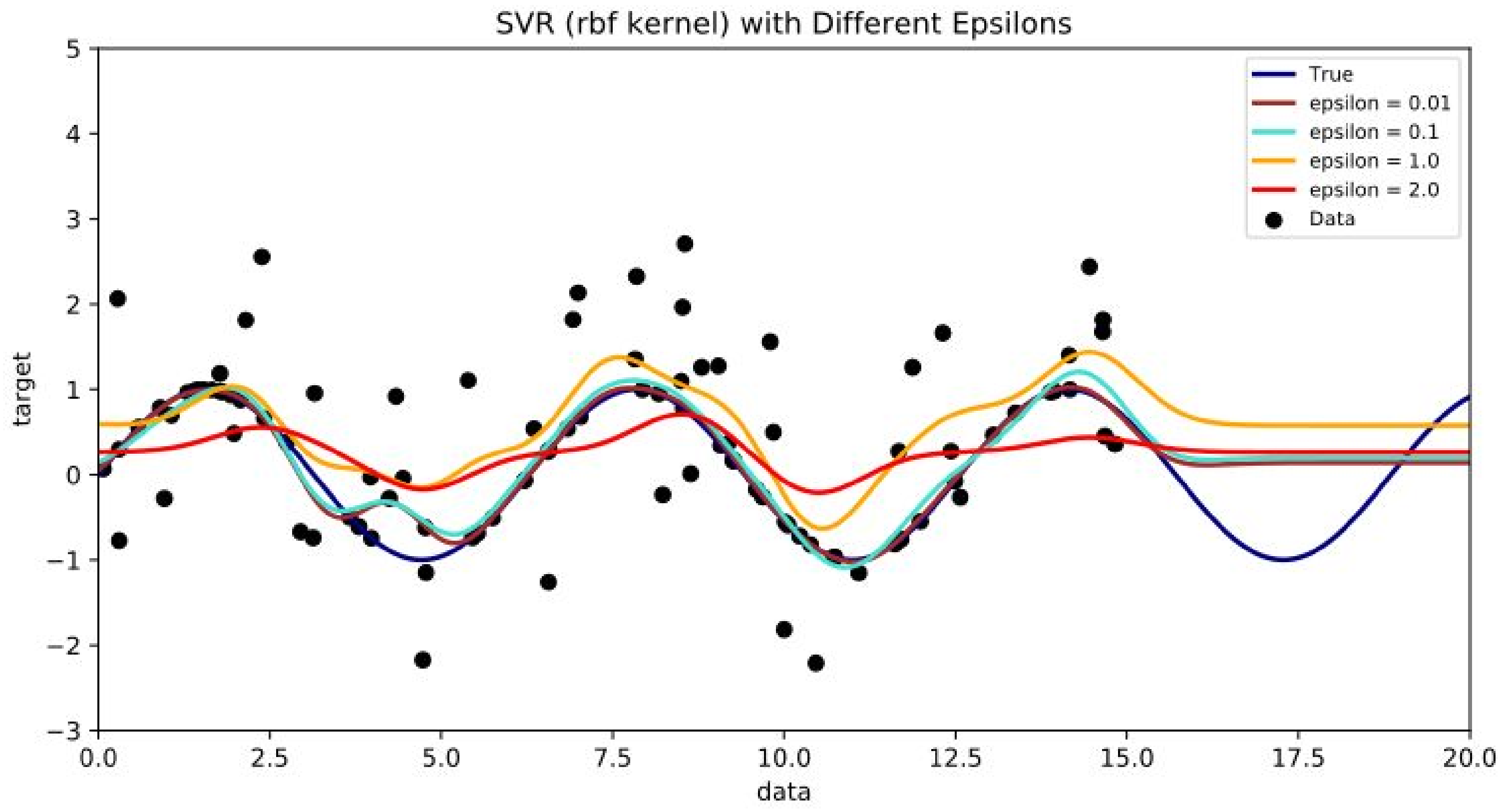
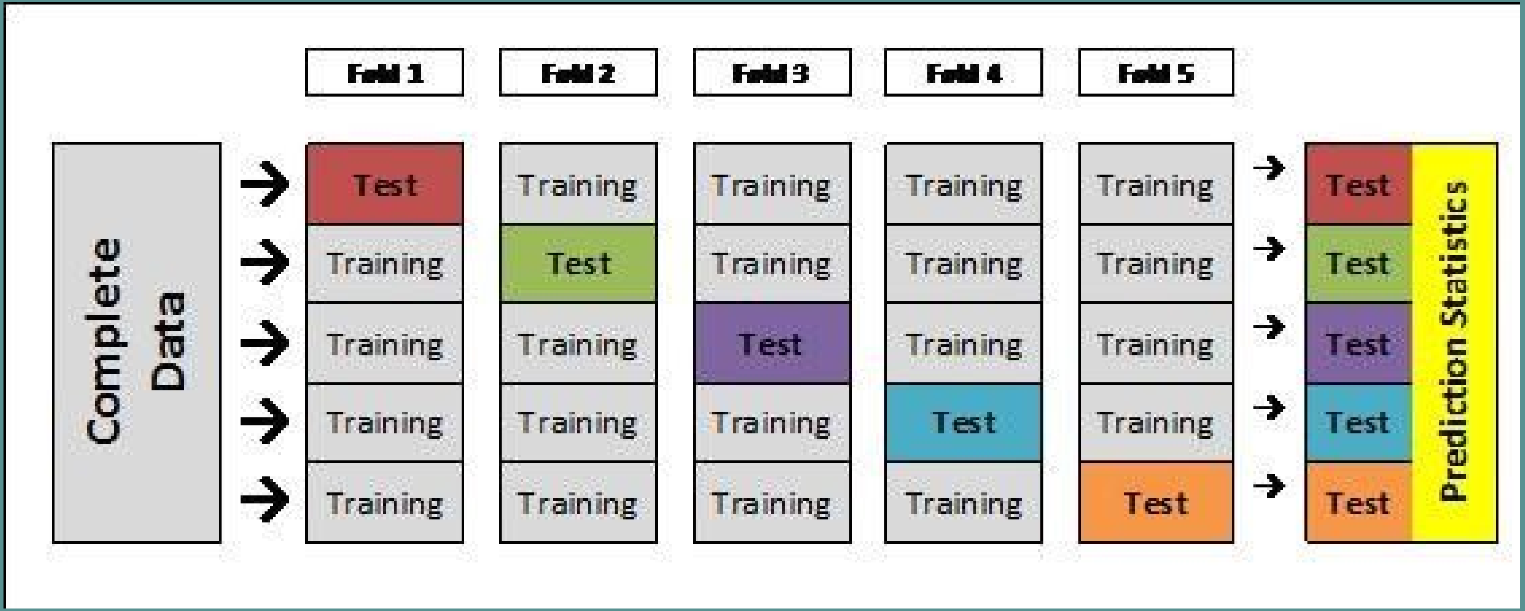


Fig. 1. *The soft margin loss setting for a linear SVM (from Schölkopf and Smola, 2002)*



[By Shiyu Ji - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=60632948]

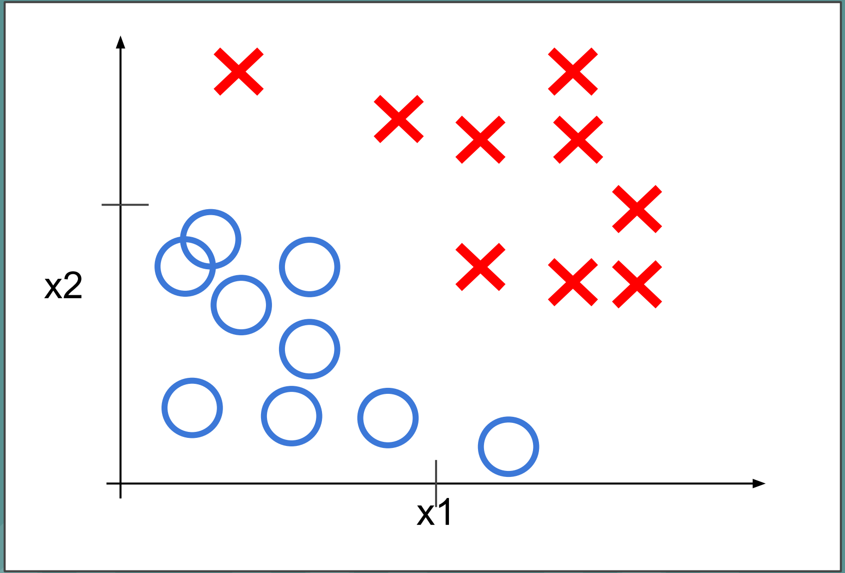
3 - Cross Validation

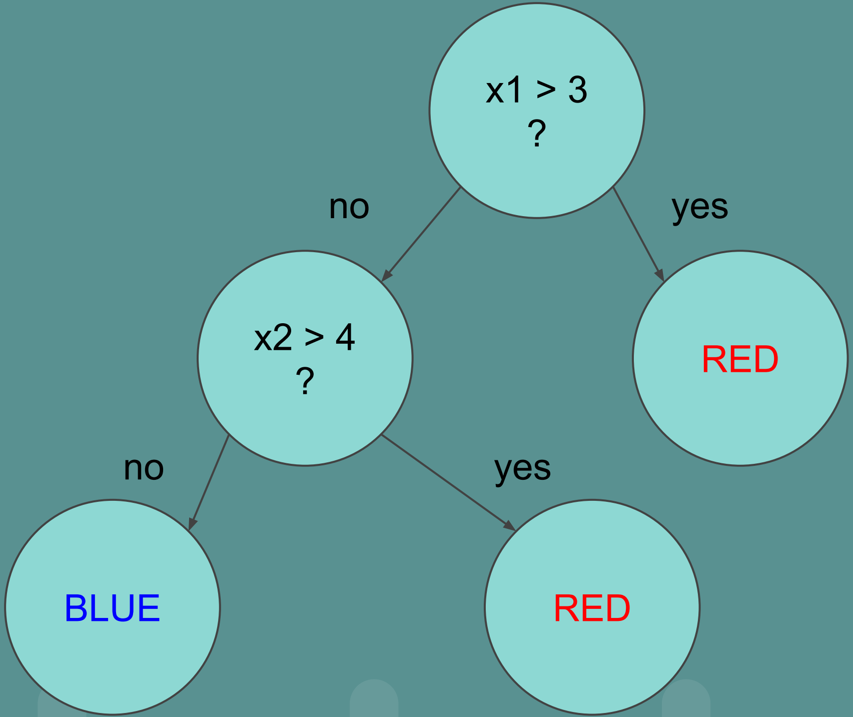

Fold 1
Fold 2
Fold 3
Fold 4
Fold 5
Complete Data
Test
Training
Prediction Statistics

4 - Decision Trees

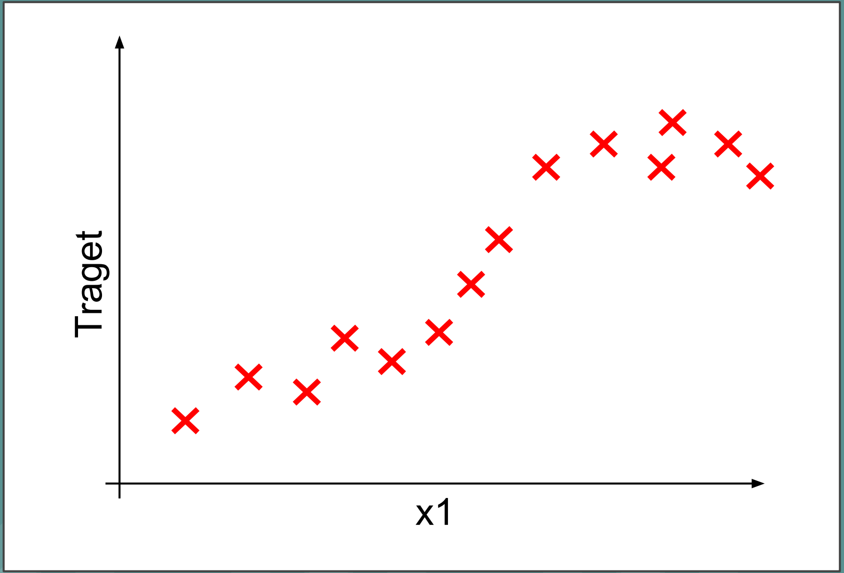
4.1 - Básico

Classificação

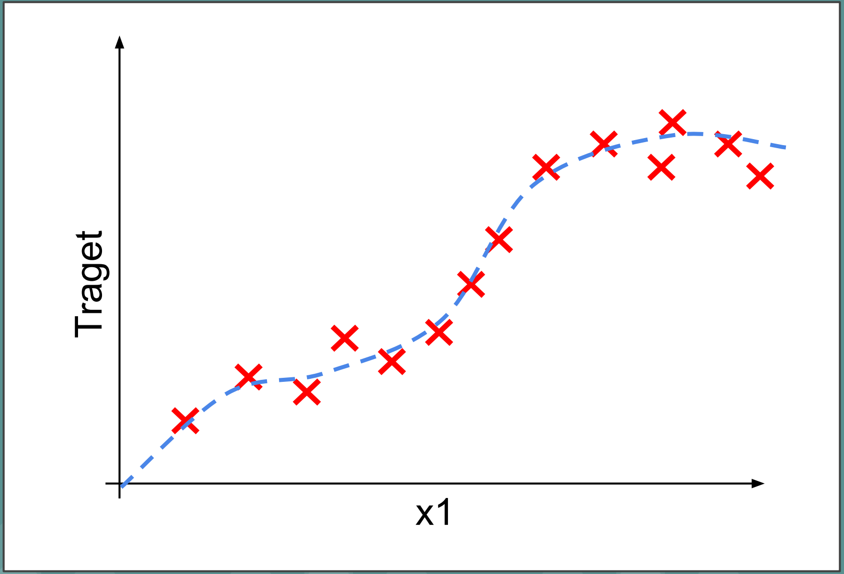




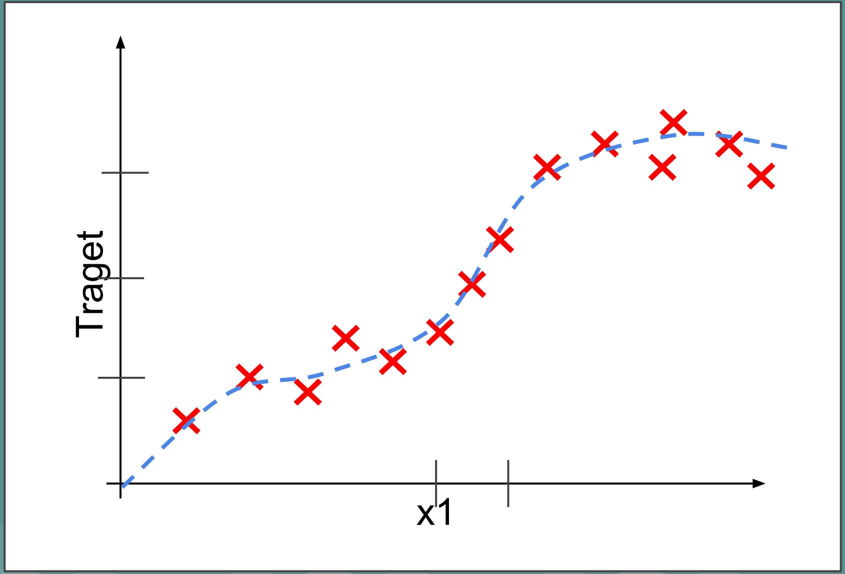
Regressão

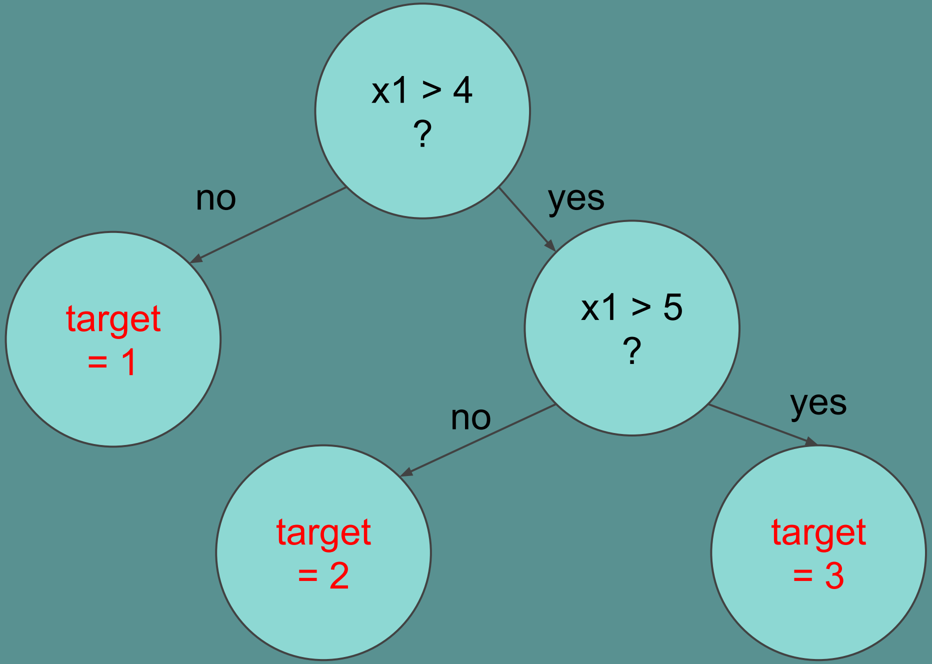


Regressão

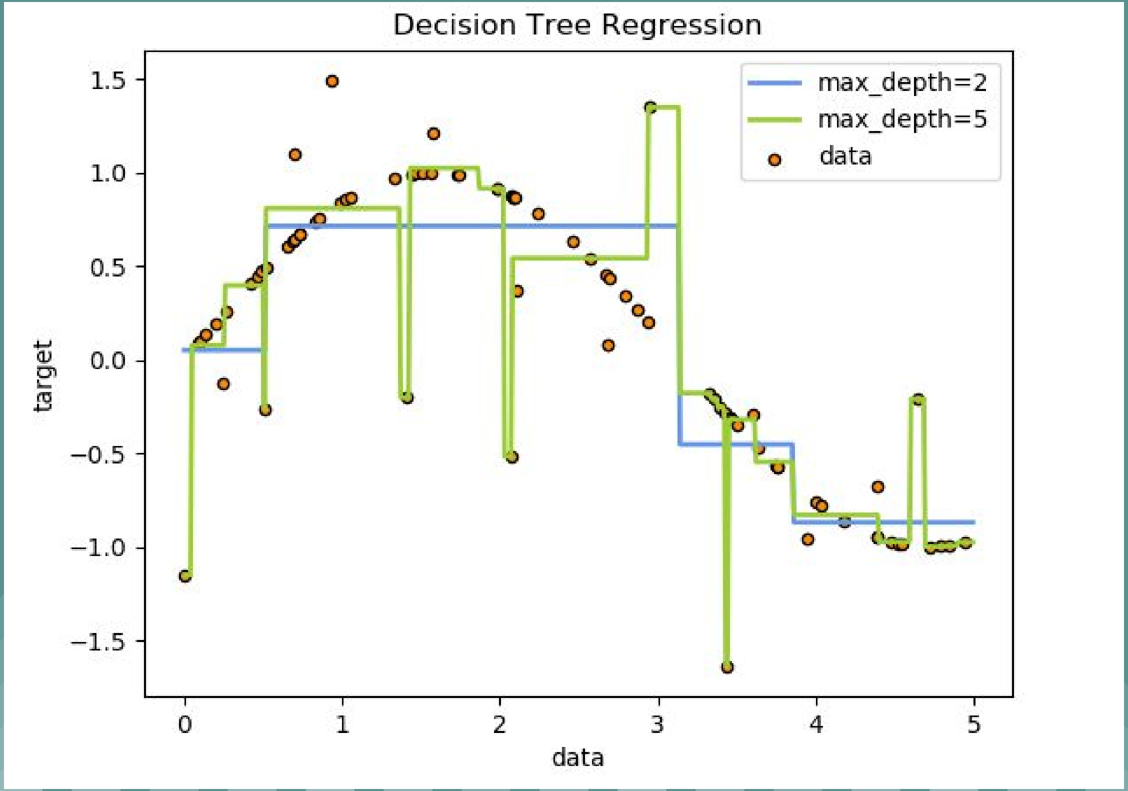


Regressão





Overfitting



4.2 Métricas

Gini Impurity (classificação)

A measure of how often a randomly chosen element from the set would be incorrectly labeled if it was randomly labeled according to the distribution of labels in the subset.

$$I_G(p) = \sum_{i=1}^{J} p_i \sum_{k \neq i} p_k = 1 - \sum_{i=1}^{J} {p_i}^2$$

Information Gain (classificação)

At each step we should choose the split that results in the purest daughter nodes. For each node of the tree, the information value "represents the expected amount of information that would be needed to specify whether a new instance should be classified yes or no, given that the example reached that node"

$$H(T) = -\sum_{i=1}^{J} p_i \log_2 p_i$$

$$\overbrace{IG(T,a)}^{\text{Information Gain}} = \overbrace{H(T)}^{\text{Entropy(parent)}} - \overbrace{H(T|a)}^{\text{Weighted Sum of Entropy(Children)}}$$

Variance reduction (Regressão)

Defined as the total reduction of the variance of the target variable x due to the split at this node.

$$I_V(N) = \frac{1}{|S|^2}\sum_{i\in S}\sum_{j\in S}\frac{1}{2}(x_i - x_j)^2 - \left(\frac{1}{|S_t|^2}\sum_{i\in S_t}\sum_{j\in S_t}\frac{1}{2}(x_i - x_j)^2 + \frac{1}{|S_f|^2}\sum_{i\in S_f}\sum_{j\in S_f}\frac{1}{2}(x_i - x_j)^2\right)$$

4.2 Métodos de Ensemble

4.2.1 Bagging

Bagging

A Bagging classifier is an ensemble meta-estimator that fits base classifiers each on random subsets of the original dataset and then aggregate their individual predictions (either by voting or by averaging) to form a final prediction.

4.2.2 Random Forest

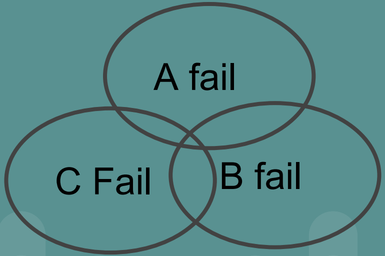
Random Forest

A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples (and sub-features) of the dataset and use averaging to improve the predictive accuracy and control over-fitting.

4.2.3 AdaBoost (Boosting)

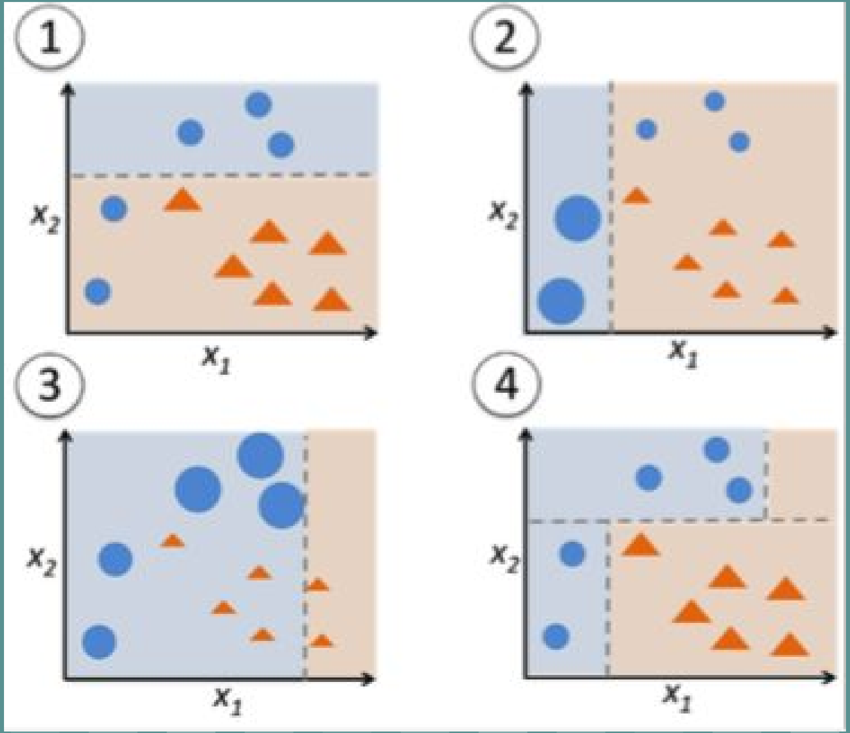
Boosting

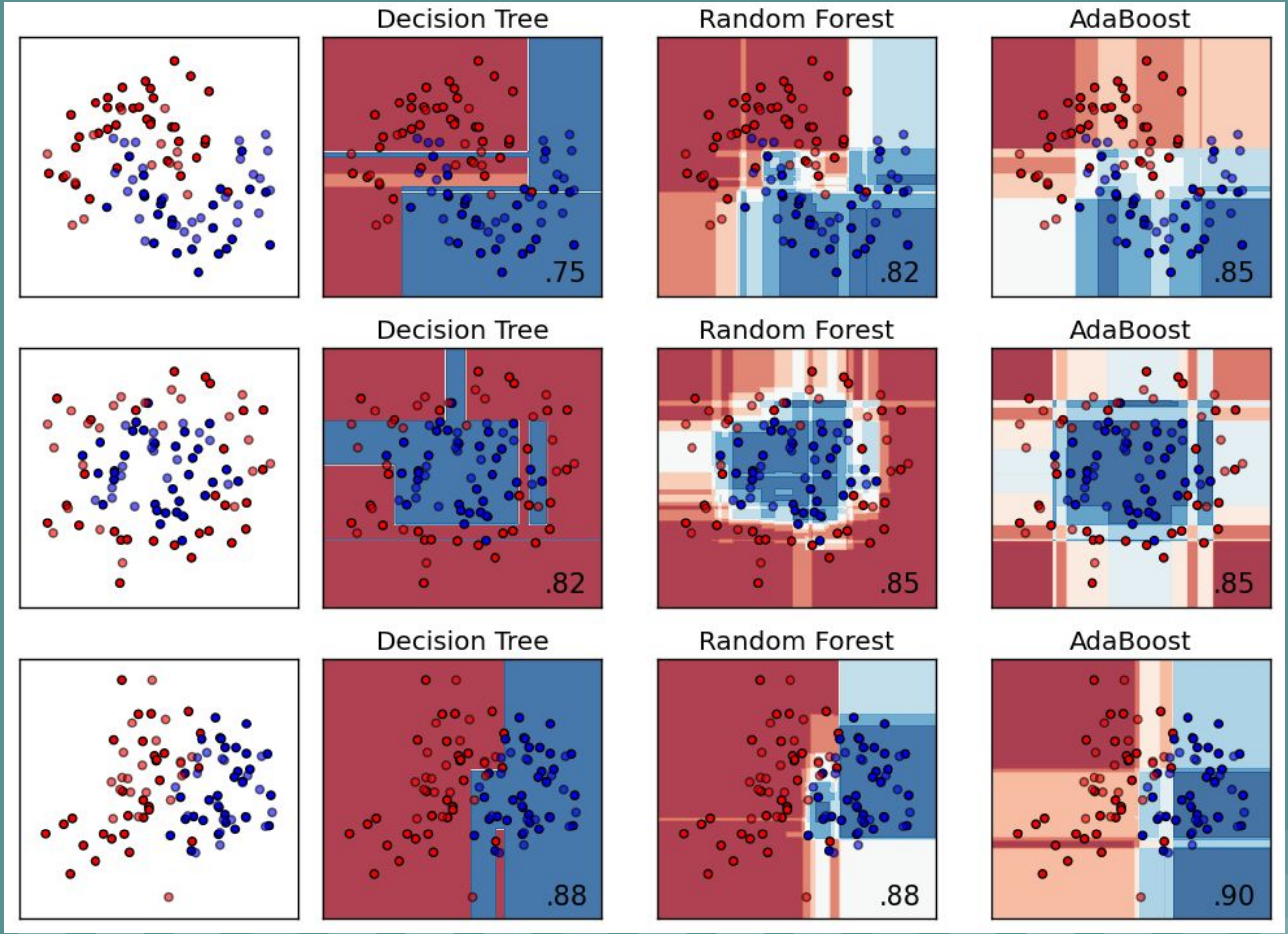
An AdaBoost classifier is a meta-estimator that begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset but where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases.



17. Learning: Boosting
https://youtu.be/UHBmv7qCey4

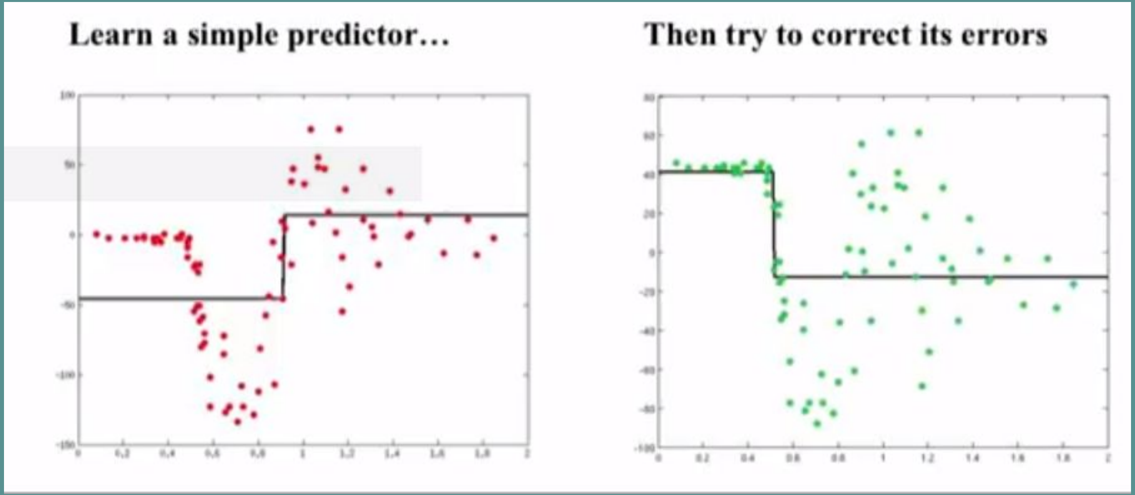
Boosting

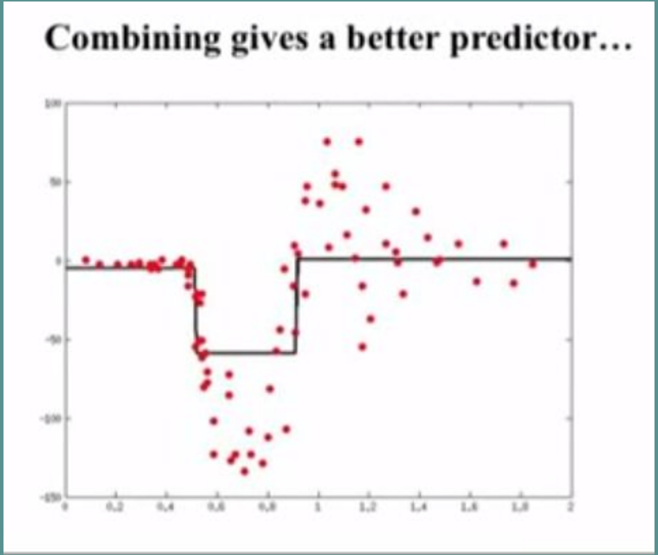



Decision Tree
Random Forest
AdaBoost
.75
.82
.85
Decision Tree
Random Forest
AdaBoost
.82
.85
.85
Decision Tree
Random Forest
AdaBoost
.88
.88
.90

4.2.4 XGBoost (Gradient Boosting)

Gradient Boosting





Ensembles (3): Gradient Boosting
https://www.youtube.com/watch?v=sRktKszFmSk

XGBoost

Utiliza a derivada de 1ª e 2ª ordem do custo em respeito a predição de cada instância, para treinar os próximos estimadores.
Altamente otimizado: calcula em tempo constante a contribuição de inserir ou remover um valor em um *split* (com regularização!).
Desvantagem: Precisa gerar uma cópia da matriz original.
http://xgboost.readthedocs.io/en/latest/model.html
http://datascience.la/xgboost-workshop-and-meetup-talk-with-tianqi-chen/

$$g_i = \partial_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)})$$

$$h_i = \partial^2_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)})$$

$$G_j = \sum_{i \in I_j} g_i \text{ and } H_j = \sum_{i \in I_j} h_i$$

$$Gain = \frac{1}{2}\left[\frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{(G_L+G_R)^2}{H_L+H_R+\lambda}\right] - \gamma$$