Aula13

DATA SCIENCE IPT

TURMA 02

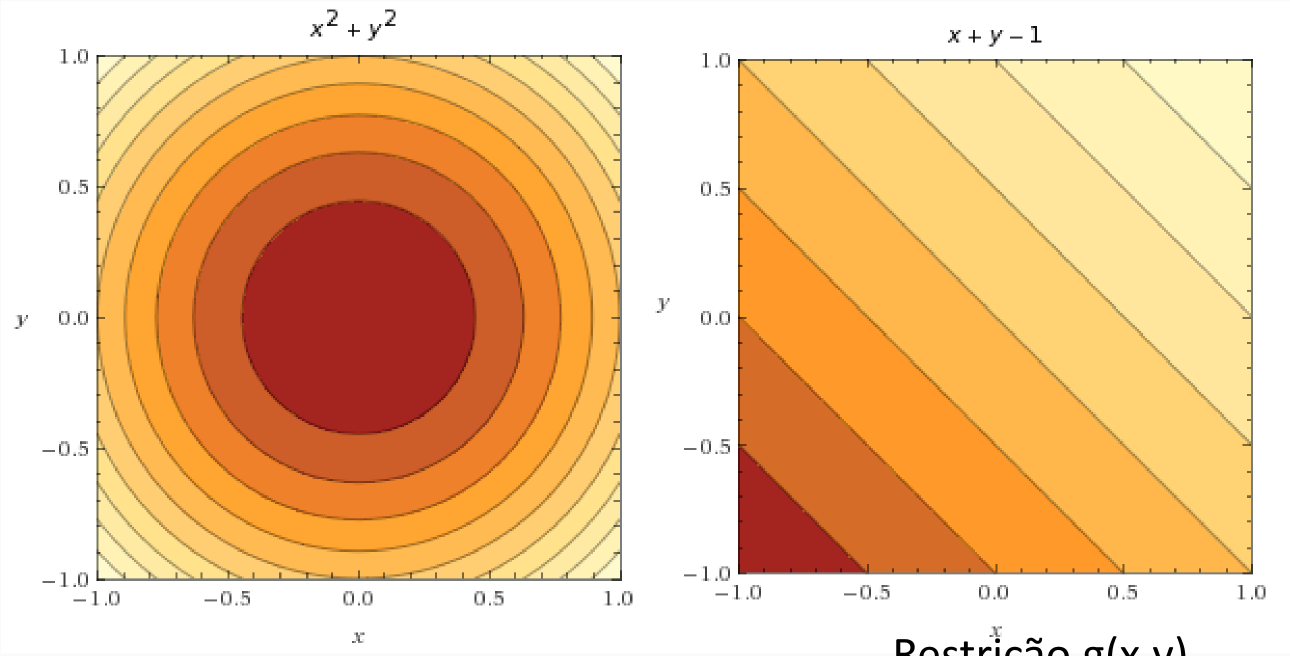
LINKED EDUCATION

V2019

SVM Parte 2

Conteúdo

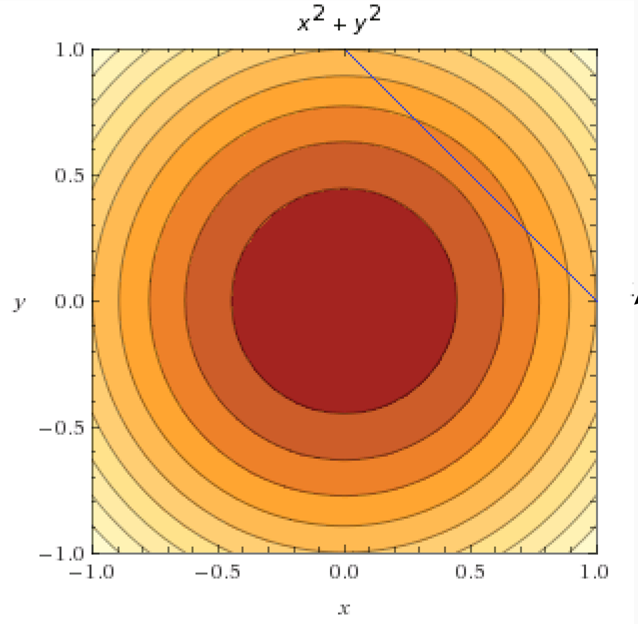
Um pouco de **Lagrange**..... Minimizar função sujeita a restrições...exemplo:



Função f(x,y)

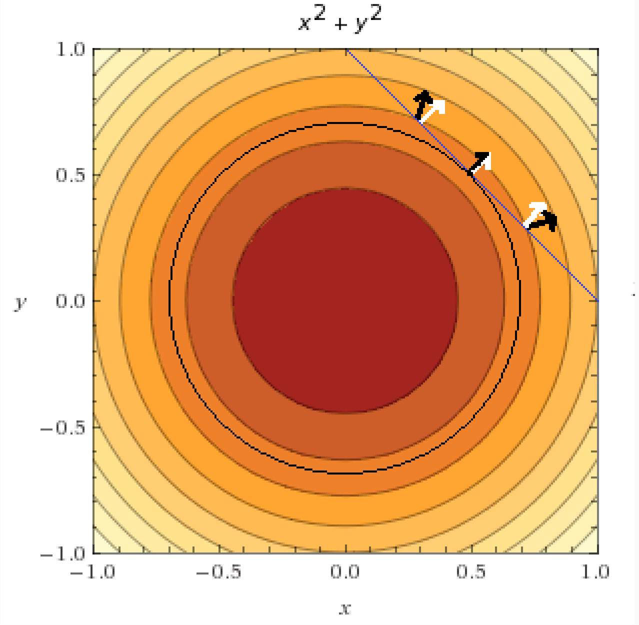
Restrição g(x,y)

SVM Parte 2



Função

Restrição
(y=1-x)



Gradientes de f (pretos) e de g (brancos)

Observe que o ponto onde os gradientes são paralelos, acontece o mínimo

Fonte: https://www.svm-tutorial.com/2016/09/duality-lagrange-multipliers/

SVM Parte 2

Se os gradientes de f e g são paralelos:

$$\nabla f = \lambda \nabla g$$

ou...

$$\nabla f - \lambda \nabla g = 0$$

Podemos definir a função Lagrangeana como:

$$L(x,y,\lambda) = f - \lambda g$$

E impor

$$\nabla L(x,y,\lambda) = 0$$

(gradientes de f e g paralelos)...

SVM Parte 2

No exemplo:

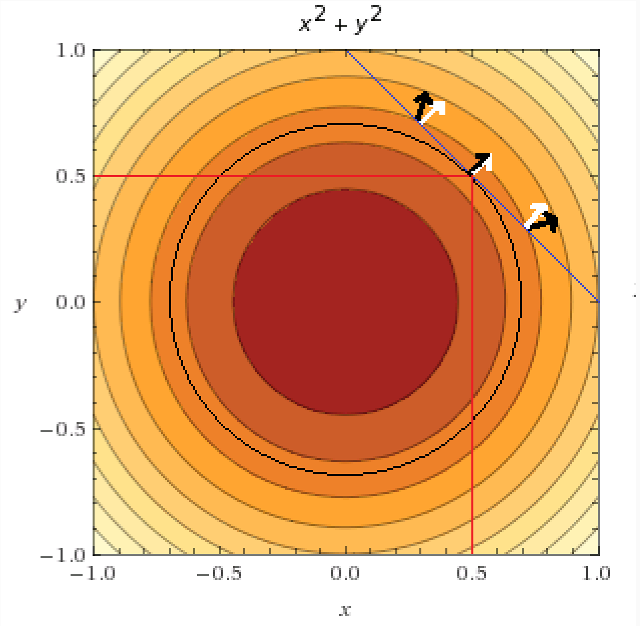
$L(x,y,\lambda)= x^2+y^2 - \lambda(x+y-1)$

$\nabla L=0 \Rightarrow 2x-\lambda=0$ (deriv. Parc. x)

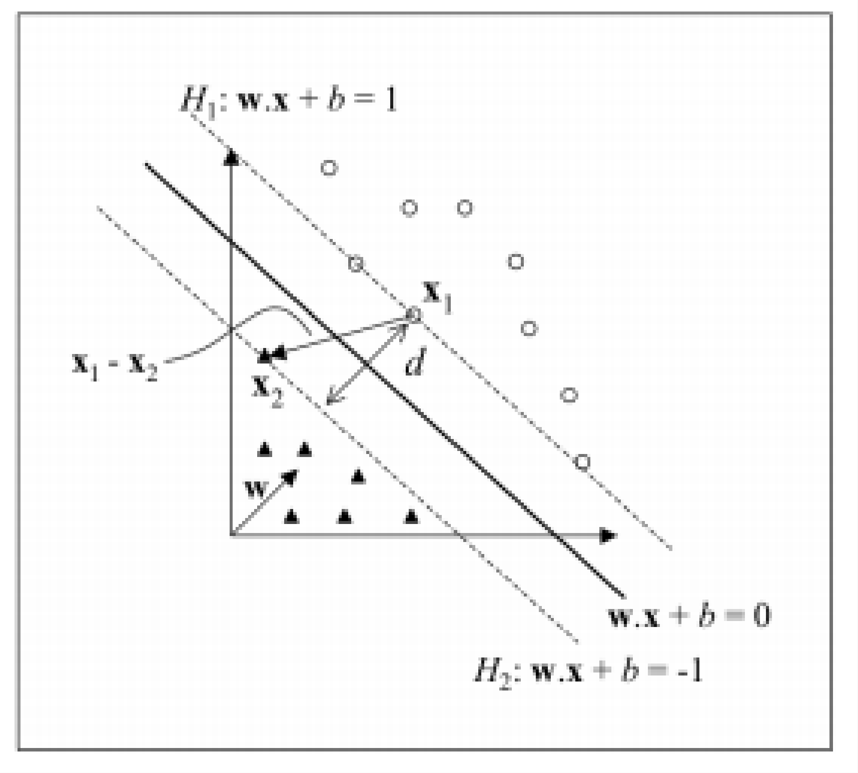
$2y-\lambda=0$ (deriv. Parc. y)

$x+y-1=0$ (restrição)

3 equações, e 3 variáveis... x=y=1/2 é o Ponto que minimiza f sob a restrição g

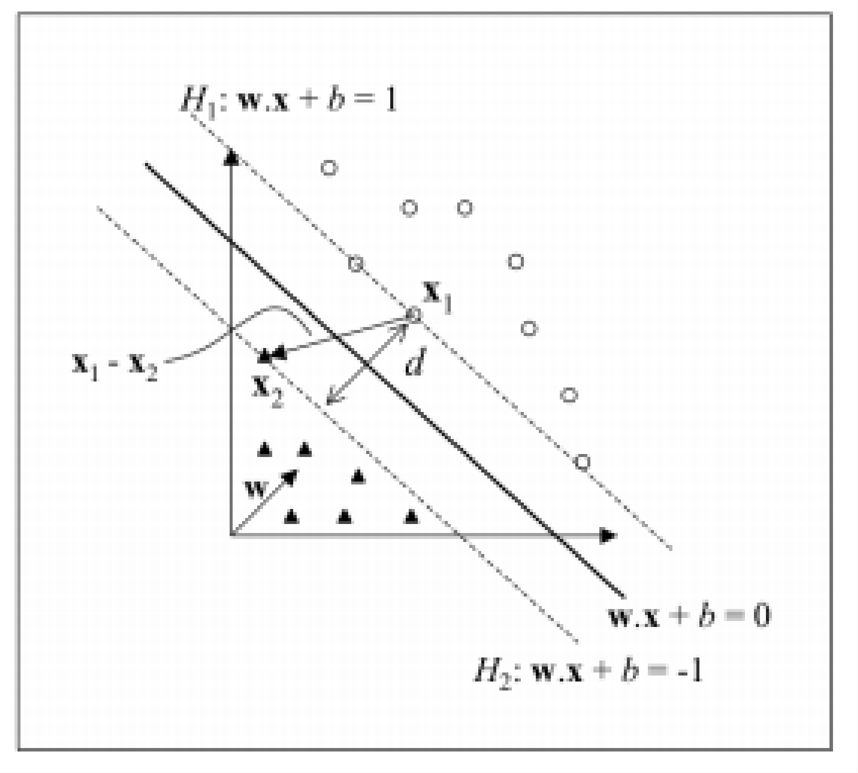


No SVM queremos maximizar a margem que separa as duas classes. Na aula passada, vimos que isso acontece quando MINIMIZAMOS o módulo do vetor normal ao hiperplano (w), já que a margem d é 2/||w||.



SVM Parte 2

Além disso, para pontos x positivos: w.x+b>=1 e para pontos negativos: w.x+b<=-1 (a igualdade acontece na borda da margem (1 e -1)).



SVM Parte 2

Chegamos a um problema de minimização com restrições....

$$\underset{\mathbf{w},b}{\text{Minimizar}} \quad \frac{1}{2}\|\mathbf{w}\|^2$$

$$\text{Com as restrições:} \quad y_i\,(\mathbf{w}\cdot\mathbf{x}_i + b) - 1 \geqslant 0, \quad \forall i = 1, \ldots, n$$

yi são os rótulos das amostras (1 ou -1). Quando yi =1,w.x+b>=1. Quando yi=-1, w.x+b <=-1

SVM (Support Vector Machines)

Conteúdo

Minimizar

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i \left(y_i \left(\mathbf{w} \cdot \mathbf{x}_i + b\right) - 1\right)$$

Equação 1

Impondo $\nabla L = 0$

$$\frac{\partial L}{\partial b} = 0 \quad \text{e} \quad \frac{\partial L}{\partial \mathbf{w}} = 0$$

Equação 2

Chegamos a:

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$$

Equação 3

Substituindo 3 em 1,
Chegamos ao problema dual:

$$\underset{\boldsymbol{\alpha}}{\text{Maximizar}} \quad \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \left(\mathbf{x}_i \cdot \mathbf{x}_j\right)$$

$$\text{Com as restrições:} \begin{cases} \alpha_i \geqslant 0, \quad \forall i = 1, \ldots, n \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \end{cases}$$

SVM (Support Vector Machines)

Conteúdo

Sendo α* vindo do problema dual e os correspondentes w* e b*....há as condições de Kühn-Tucker para problemas de otimização (temos inequações e Lagrange prevê equações nas restrições):

$$\alpha_i^* \left(y_i \left(\mathbf{w}^* \cdot \mathbf{x}_i + b^* \right) - 1 \right) = 0, \ \forall i = 1, \ldots, n$$

Para que as condições sejam satisfeitas com α*> 0, xi deverá estar nas bordas....será um Support Vector!

A função decisora será o sinal da fórmula abaixo...

$$g\left(\mathbf{x}\right) = \mathrm{sgn}\left(f\left(\mathbf{x}\right)\right) = \mathrm{sgn}\left(\sum_{\mathbf{x}_i \in \mathrm{SV}} y_i \alpha_i^* \mathbf{x}_i \cdot \mathbf{x} + b^* \right)$$

SVM (Support Vector Machines)

O vídeo calcula pela forma dual um SVM com apenas dois pontos...simples e didático.

https://www.youtube.com/watch?v=5zRmhOUjjGY

SVM (Support Vector Machines)

Partindo de lousa-svm.ipynb

Atividade1 : analisar código

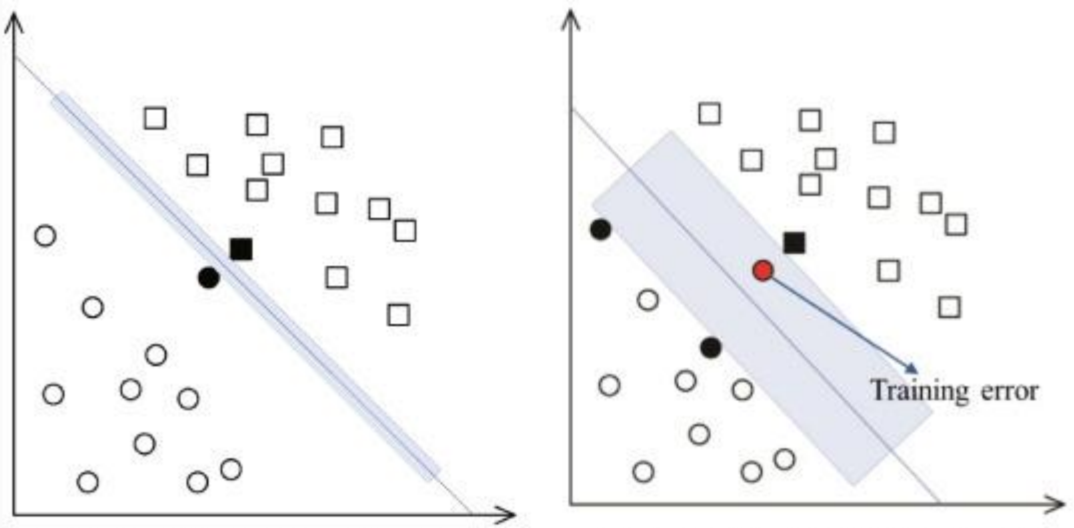
Atividade 2 : mostrar que a reta é x1=1.5

Atividade 3 : apresentar vetores de suporte

Atividade 4: criar função decisora com base em support vectors e coeficientes da solução dual...

SVM (Support Vector Machines)

Quando as classes não são 100% separáveis linearmente, utiliza-se a “soft margin”, ou seja, um relaxamento da condição de não haver pontos entre as margens e pontos classificados errados no treinamento.



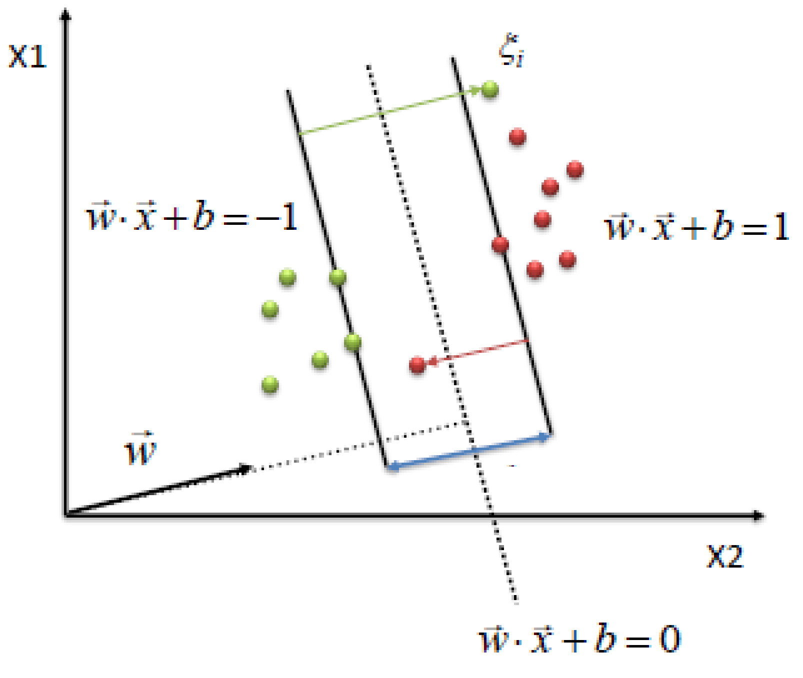
(a) LSVM with hard margin (b) LSVM with soft margin

Fonte imagem : https://www.sciencedirect.com/science/article/pii/S0926580516301297

SVM (Support Vector Machines)

Conteúdo

Soft Margin



Constraint becomes :

$$y_i(w\cdot x_i+b)\geq 1-\xi_i,\ \forall x_i$$
$$\xi_i \geq 0$$

Objective function penalizes for misclassified instances and those within the margin

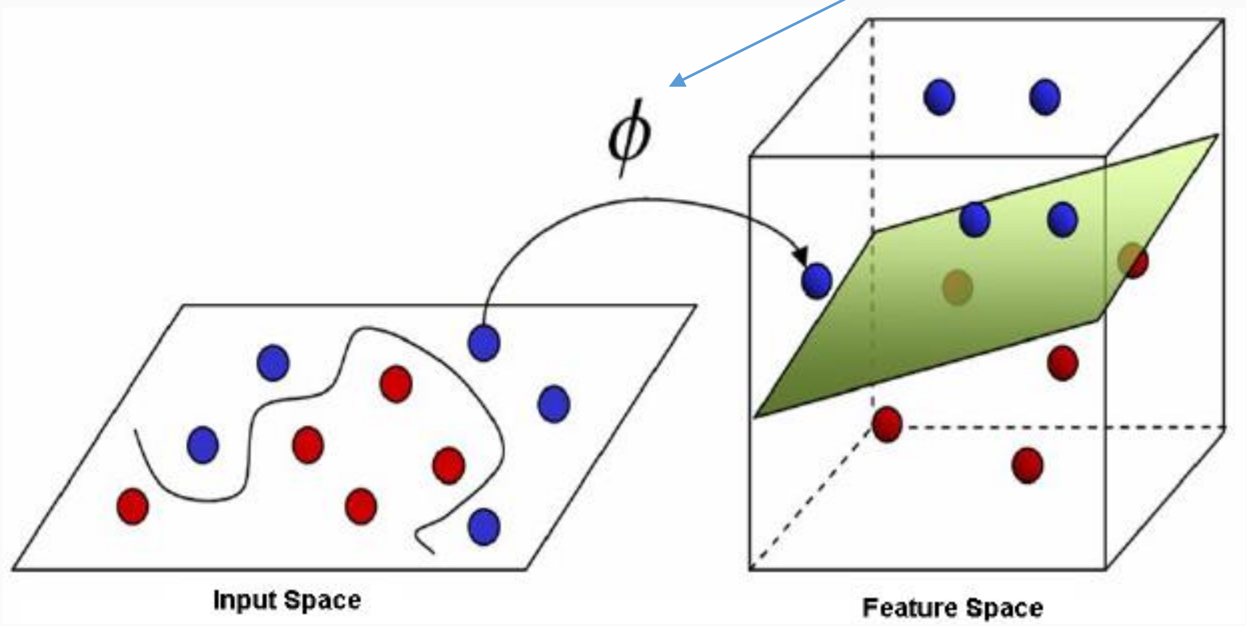
$$\min \frac{1}{2}\|w\|^2 + C\sum_i \xi_i$$

C trades-off margin width and misclassifications

SVM (Support Vector Machines)

SVM não linear

Um interessante recurso do SVM é a possibilidade de utilizar um “mapeamento” que leva um problema não separável linearmente na dimensão “n” para uma dimensão “n+1” onde os pontos são linearmente separáveis por um hiperplano.



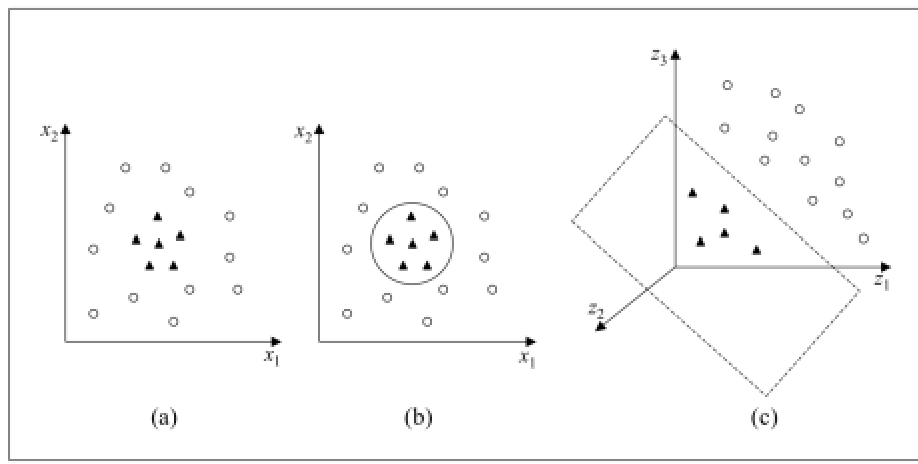
SVM (Support Vector Machines)

A função Kernel permite calcular o produto interno de vetores em outras dimensões, onde a separação linear por hiperplanos pode ser mais fácil...e podemos verificar a similaridade desses vetores em outras dimensões

Exemplo de mapeamento que leva de R^2 para R^3

$$\Phi(\mathbf{x}) = \Phi(x_1, x_2) = \left(x_1^2, \sqrt{2}x_1x_2, x_2^2\right)$$

Com esse mapeamento do exemplo,
Os dados bidimensionais a são linearmente separáveis no espaço!



Para esse mapeamento, o Kernel é :

$K(x,y)=\langle x,y\rangle^2$

Fonte imagens :http://www.seer.ufrgs.br/rita/article/download/rita_v14_n2_p43-67/3543

Kernels do scikit

- linear: $\langle x, x' \rangle$.
- polynomial: $(\gamma\langle x, x' \rangle + r)^d$. d is specified by keyword `degree`, r by `coef0`.
- rbf: $\exp(-\gamma\|x - x'\|^2)$. γ is specified by keyword `gamma`, must be greater than 0.
- sigmoid $(\tanh(\gamma\langle x, x' \rangle + r))$, where r is specified by `coef0`.

SVM (Support Vector Machines)

Quais seriam os parâmetros do Kernel Polinomial

$K(x,y)=(x.y)^2$ no scikit?

SVM (Support Vector Machines)

O fator Gamma no Kernel rbf

Valores baixos de γ fazem com que vetores de suporte mesmo longe de amostras tenham influência na classificação...é um modelo menos complexo

Com Kernels que dependem de γ e C é necessário testar a performance do algoritmo em várias combinações ...esse processo é denominado grid-search

SVM (Support Vector Machines)

Partindo de kernel-not-linear.ipynb

Atividade1 : Analisar código

Atividade 2: Fazer grid-search com c e gamma :[0.01,0.1,1,10,100]
Apresentar melhor acurácia no teste e correspondente par c-gamma

Ensemble Methods

Os Ensemble Methods combinam as predições de vários algoritmos visando diminuir bias e variância...o objetivo é melhorar a generalização das predições.
Há dois tipos clássicos :

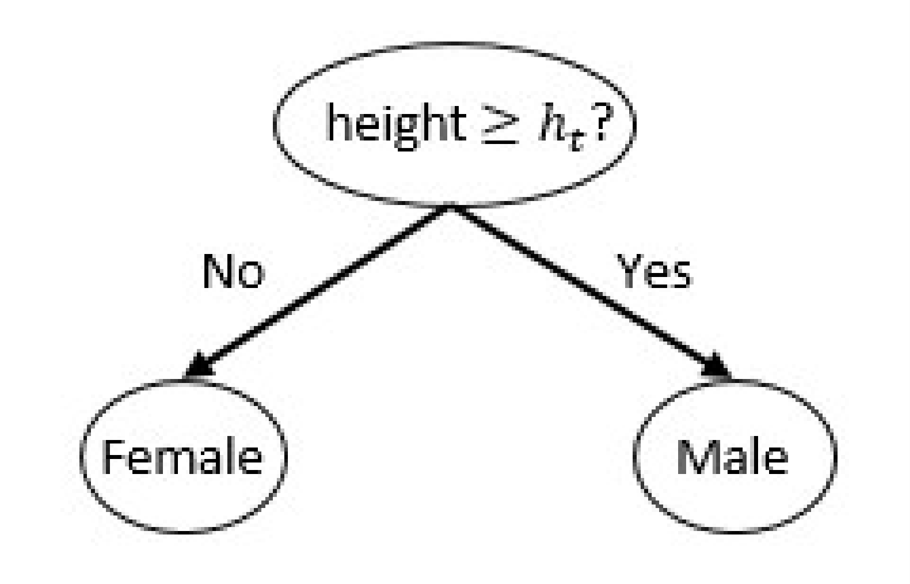
Averaging methods : geram a média de várias predições independentes.
Exemplo : Bagging

Boosting methods : usa várias predições e, sequencialmente, há a tentativa de reduzir o bias da predição combinada. **A ideia é produzir uma estimativa melhor combinando muitos estimadores "fracos (?)".**
Exemplo : Adaboost

Weak Learners

Weak Learners são hipóteses (modelos) com performance levemente superior ao aleatório (50% de acurácia em uma classificação com 2 classes balanceadas, por exemplo)

Árvores de decisão de um nível apenas (decision stumps) são weak learners muito utilizados...mas por que utilizar weak learners?



Boosting

A ideia do boosting é combinar vários weak learners e produzir um Strong learner...os weak learners combinados são robustos quanto a overfitting...

AdaBoost (adaptive boosting)

Adaboost é o mais popular algoritmo de boosting.
Basicamente sua ideia é :

Amostras inicialmente têm o mesmo peso

Para os “m” weak learners

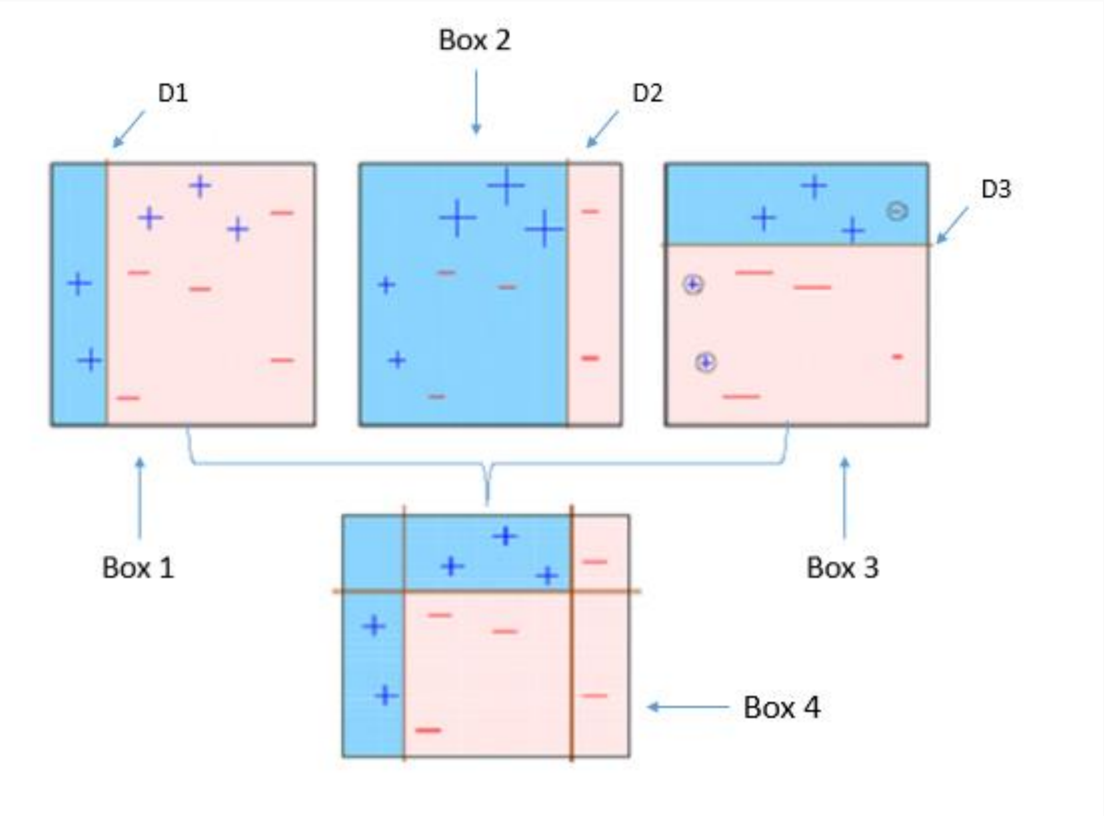
treinar amostras e obter modelo mod i

mudar os pesos das amostras, reforçando as que tiveram predição errada

Fazer um preditor com a ponderação dos modelos individuais

AdaBoost

Exemplo : Box1,2 e 3 são weak learners...box4 os combina
Note que após o box1, o box2 “reforçou” as amostras erradas em box 1 (os 3 +)



AdaBoost - algoritmo

Given: $(x_1, y_1), \ldots, (x_m, y_m)$ where $x_i \in \mathscr{X}$, $y_i \in \{-1, +1\}$.
Initialize: $D_1(i) = 1/m$ for $i = 1, \ldots, m$.
For $t = 1, \ldots, T$:

- Train weak learner using distribution D_t.
- Get weak hypothesis $h_t : \mathscr{X} \to \{-1, +1\}$.
- Aim: select h_t with low weighted error:

$$\varepsilon_t = \Pr_{i \sim D_t}\left[h_t(x_i) \neq y_i\right].$$

- Choose $\alpha_t = \frac{1}{2} \ln\left(\frac{1 - \varepsilon_t}{\varepsilon_t}\right)$.
- Update, for $i = 1, \ldots, m$:

$$D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

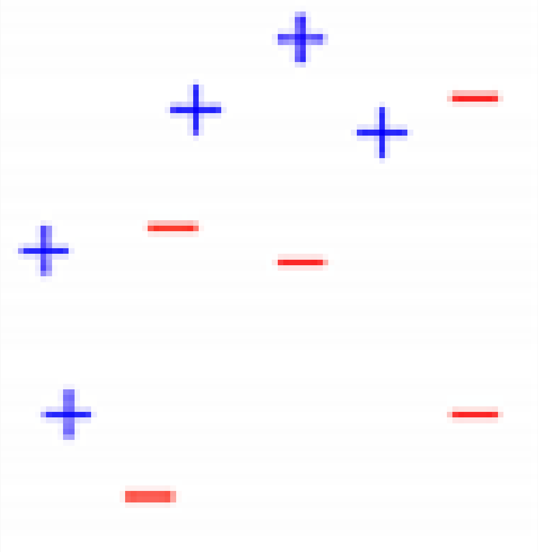
where Z_t is a normalization factor (chosen so that D_{t+1} will be a distribution).

Output the final hypothesis:

$$H(x) = \operatorname{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right).$$

Fonte : Robert E. Schapire

AdaBoost Data Set teste



x,y,classe

29,11,-1
56,51,-1
86,25,-1
87,81,-1
33,58,-1
14,25,1
11,44,1
37,79,1
56,92,1
70,75,1

ada.txt

AdaBoost

Com base em : ada-raiz.ipynb

Atividade 1 :

Obter erro médio

Atividade 2 :

Obter alpha

Atividade 3 : atualizar pesos

Discussão: Como criar um preditor genérico com o modelo criado?

AdaBoost

Com base em ada_scikit.ipynb

Atividade :

Obter acurácia na amostra toda até atingir 100%
variando o número de weak learners..de 1 para cima

Revisão

K-means no Excel
Analisando o código da classe MKMeans (Prof. Leston)

LINKED EDUCATION

Cursos com Alta Performance de Aprendizado