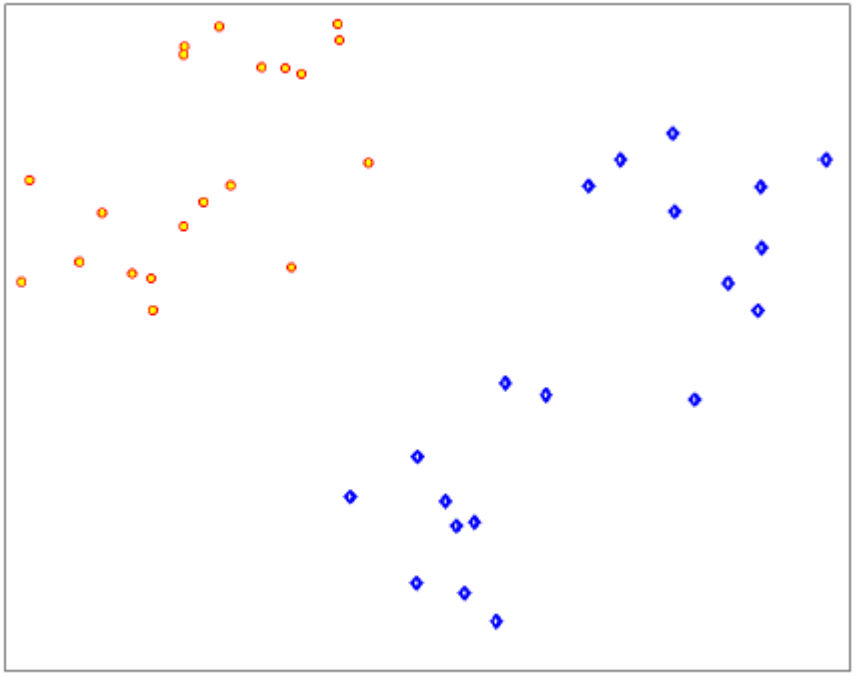
Tema 07

Máquinas de Vetores de Suporte (SVM)

Professora:
Ariane Machado Lima

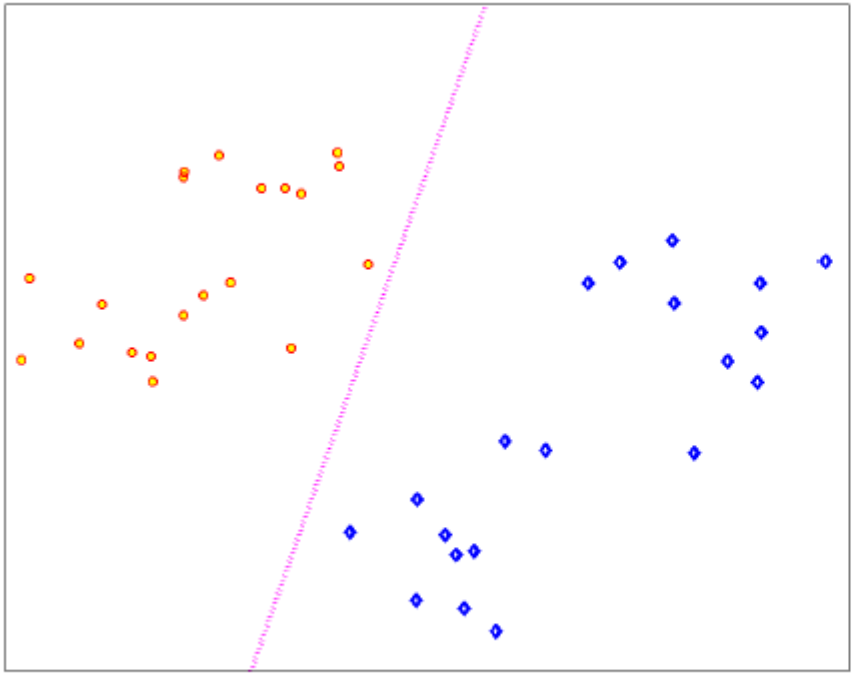
Classificação linear
VAMOS COMEÇAR ASSUMINDO QUE HÁ DUAS CLASSES LINEARMENTE SEPARÁVEIS

- Achar uma reta que separe (classifique) duas categorias



Classificação linear

- Achar uma reta que separe (classifique) duas categorias



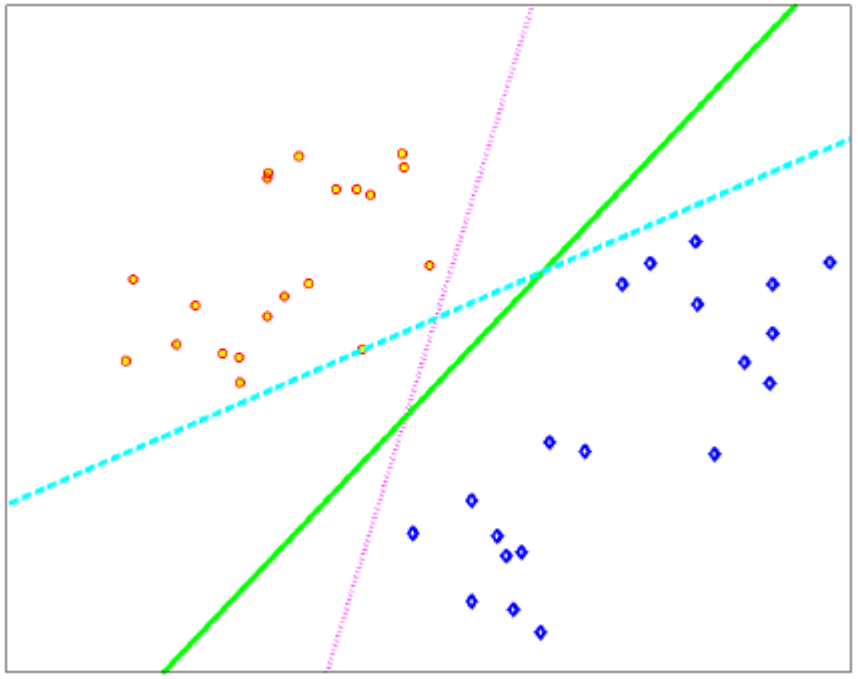
$$D(x) = \text{sign}(\mathbf{v}^\top \mathbf{x} + a)$$

the decision border:

$$\mathbf{v}^\top \mathbf{x} + a = 0$$

Classificação linear

- Mas muitas soluções seriam possíveis...
- É necessário definir o problema melhor



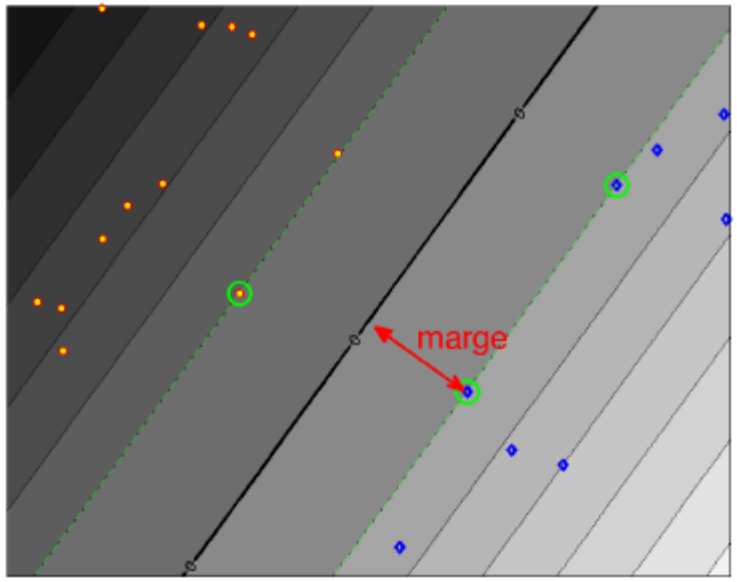
$$D(x) = \text{sign}(\mathbf{v}^\top \mathbf{x} + a)$$

the decision border:

$$\mathbf{v}^\top \mathbf{x} + a = 0$$

Maximização da margem (para maximizar a generalização)

Fronteira de decisão: $\Delta(\mathbf{v}, a) = \{\mathbf{x} \in \mathbb{R}^d \mid \mathbf{v}^\top \mathbf{x} + a = 0\}$



Maximização da margem (*n* objetos de treinamento):

$$\max_{\mathbf{v},a} \underbrace{\min_{i\in[1,n]} \operatorname{dist}(\mathbf{x}_i, \Delta(\mathbf{v}, a))}_{\text{margin: } m}$$

$$\begin{cases} \max\limits_{\mathbf{v},a} & m \\ \text{with} & \min\limits_{i=1,n} |\mathbf{v}^\top \mathbf{x}_i + a| \geq m \end{cases}$$

Se o mínimo é maior que m, então todos os outros são também.

O problema ainda está mal definido:

Se (v, a) é uma solução, (kv, ka) também é para qualquer k real >=0

Maximização da margem

$$\begin{cases} \max\limits_{\mathbf{v},a} & m \\ \text{with} & \min\limits_{i=1,n} |\mathbf{v}^\top \mathbf{x}_i + a| \geq m \\ & \|\mathbf{v}\|^2 = 1 \end{cases}$$

Obs: m >= 0, pois é uma distância

Tirando o módulo: y_i = 1 ou -1:

y_i é de que lado o ponto (vetor) está, ou seja, é a classe predita!

$$\begin{cases} \max\limits_{\mathbf{v},a} & m \\ \text{with} & y_i(\mathbf{v}^\top \mathbf{x}_i + a) \geq m, \quad i = 1:n \\ & \|\mathbf{v}\|^2 = 1 \end{cases}$$

Dividindo por *m* e fazendo uma mudança de variáveis:

$$\mathbf{w} = \frac{\mathbf{v}}{m} \qquad b = \frac{a}{m} \implies \|\mathbf{w}\| = \frac{\|\mathbf{v}\|}{m}$$

$$\begin{cases} \max\limits_{\mathbf{w},b} & m \\ \text{with} & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 \quad ; \; i = 1,n \\ \text{and} & m = \frac{1}{\|\mathbf{w}\|} \end{cases}$$

$$\begin{cases} \min\limits_{\mathbf{w},b} & \|\mathbf{w}\|^2 \\ \text{with} & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 \\ & i = 1,n \end{cases}$$

Maximização da margem

$$\begin{cases} \max\limits_{\mathbf{v},a} & m \\ \text{with} & \min\limits_{i=1,n} |\mathbf{v}^\top \mathbf{x}_i + a| \geq m \\ & \|\mathbf{v}\|^2 = 1 \end{cases}$$

Obs: m >= 0, pois é uma distância

Tirando o módulo: y_i = 1 ou -1:

y_i é de que lado o ponto (vetor) está, ou seja, é a classe predita!

$$\begin{cases} \max\limits_{\mathbf{v},a} & m \\ \text{with} & y_i(\mathbf{v}^\top \mathbf{x}_i + a) \geq m, \quad i = 1:n \\ & \|\mathbf{v}\|^2 = 1 \end{cases}$$

Dividindo por *m* e fazendo uma mudança de variáveis:

$$\mathbf{w} = \frac{\mathbf{v}}{m} \qquad b = \frac{a}{m} \implies \|\mathbf{w}\| = \frac{\|\mathbf{v}\|}{m}$$

$$\begin{cases} \max\limits_{\mathbf{w},b} & m \\ \text{with} & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 \quad ; \; i = 1,n \\ \text{and} & m = \frac{1}{\|\mathbf{w}\|} \end{cases}$$

$$\begin{cases} \min\limits_{\mathbf{w},b} & \|\mathbf{w}\|^2 \\ \text{with} & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 \\ & i = 1,n \end{cases}$$

SVM é a solução do seguinte problema: (formulação primal)

Seja $\{(x_i, y_i), i = 1, \ldots, n\}$ um conjunto rotulado de treinamento, $x_i \in R^d$ e $y_i \in \{-1, 1\}$

Uma máquina de vetores de suporte (SVM) é um classificador linear associado com a função de decisão $D(x) = \text{sign}(\mathbf{w}^\top \mathbf{x} + b)$

sendo que $w \in R^d$ e $b \in R$ são a solução de

$$\begin{cases} \min\limits_{\mathbf{w},b} & \frac{1}{2}\|\mathbf{w}\|^2 = \frac{1}{2}\mathbf{w}^\top \mathbf{w} \\ \text{with} & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 \\ & i = 1, n \end{cases}$$

Este é um problema de programação quadrática:

$$\begin{cases} \min\limits_{\mathbf{z}} & \frac{1}{2}\mathbf{z}^\top A\mathbf{z} - \mathbf{d}^\top \mathbf{z} \\ \text{with} & B\mathbf{z} \leq \mathbf{e} \end{cases}$$

$$\mathbf{z} = (\mathbf{w}, b)^\top, \mathbf{d} = (0, \ldots, 0)^\top, A = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}, B = -[\mathbf{y}X, \mathbf{y}] \text{ et } \mathbf{e} = -(1, \ldots, 1)^\top$$

Uso na classificação

Fase de treinamento: **w** e b são calculados para encontrar "a reta" (o hiperplano) que melhor separa as duas classes (que maximiza a margem entre as n instâncias de treinamento)

$$\begin{cases} \min\limits_{\mathbf{w},b} & \frac{1}{2}\|\mathbf{w}\|^2 = \frac{1}{2}\mathbf{w}^\top\mathbf{w} \\ \text{with} & y_i(\mathbf{w}^\top\mathbf{x}_i + b) \geq 1 \\ & i = 1, n \end{cases}$$

Fase de classificação de uma instância **x**: verifica-se de que lado do hiperplano **x** está; se D(**x**) < 0 é uma classe, se D(**x**) > 0 é outra classe (lembrando que d é a dimensão de **x**)

$$D(x) = \sum_{j=1}^{d} w_j x_j + b$$

- d + 1 variáveis desconhecidas (**w** e b)
- n restrições
- Programação quadrática clásssica
- Ok quando d << n

Uso na classificação

Fase de treinamento: **w** e b são calculados para encontrar "a reta" (o hiperplano) que melhor separa as duas classes (que maximiza a margem entre as n instâncias de treinamento)

$$\begin{cases} \min\limits_{\mathbf{w},b} & \frac{1}{2}\|\mathbf{w}\|^2 = \frac{1}{2}\mathbf{w}^\top\mathbf{w} \\ \text{with} & y_i(\mathbf{w}^\top\mathbf{x}_i + b) \geq 1 \\ & i = 1, n \end{cases}$$

Fase de classificação de uma instância **x**: verifica-se de que lado do hiperplano **x** está; se D(**x**) < 0 é uma classe, se D(**x**) > 0 é outra classe (lembrando que d é a dimensão de **x**)

$$D(x) = \sum_{j=1}^{d} w_j x_j + b$$

- d + 1 variáveis desconhecidas (**w** e b)
- n restrições
- Programação quadrática clásssica
- Ok quando d << n

E quando d >> n?

Há outra formulação mais eficiente do problema...

Dualidade

- Normalmente um problema pode ser descrito de duas formas diferentes
- Ideia: transformar a descrição de um problema em uma outra descrição (dual) que permita uma solução mais simples
- Como transformar esse problema de otimização que acabamos de ver (primal) em um correspondente problema dual ?

Um pouco sobre otimização

Problema de otimização: achar o mínimo ou máximo de uma função objetiva J, possivelmente sujeito a determinadas restrições (desigualdades e igualdades)

- Problema primal (genérico):

  min J(**x**), $x \in \Omega$ (subconjunto de R^n)

  sujeito a $g_i(\mathbf{x}) \leq 0$, i = 1, …, k

  $h_i(\mathbf{x}) = 0$, i = 1, …, m

Tratamento lagrangiano

- Manipular restrições de desigualdades diretamente pode ser difícil
- Introdução de multiplicadores de Lagrange (variáveis duais)

Tratamento lagrangiano

x aqui um vetor aleatório genérico, representando as variáveis da função que eu quero minimizar (nada a ver com o **x** dos slides anteriores)

- Problema primal:

  $\min_{\mathbf{x}} J(\mathbf{x})$, $\quad x \in \Omega$ (subconjunto de R^n)

  sujeito a $g_i(\mathbf{x}) <= 0 \quad , i = 1, \ldots, k$

  $h_j(\mathbf{x}) = 0 \quad , j = 1, \ldots, m$

- Função Lagrangiana generalisada:

$$L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = J(\mathbf{x}) + \Sigma_i \mu_i g_i(\mathbf{x}) + \Sigma_j \lambda_j h_j(\mathbf{x})$$

$$= J(\mathbf{x}) + \boldsymbol{\mu}^T \mathbf{g(x)} + \boldsymbol{\lambda}^T \mathbf{h(x)}$$

- Problema dual (Lagrangiano):

  $\max_{\lambda, \mu} \theta (\boldsymbol{\lambda}, \boldsymbol{\mu})$

  sujeito a $\boldsymbol{\mu} >= \mathbf{0}$

  sendo $\theta (\boldsymbol{\lambda}, \boldsymbol{\mu}) = \min_{x \in \Omega} L(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu})$

Condições Karush-Kuhn-Tucker (KKT)

($\mathbf{x}^*$, $\boldsymbol{\lambda}$, $\boldsymbol{\mu}$) satisfazem as condições KKT se:

Definition: Karush, Kuhn and Tucker (KKT) conditions

$$\text{stationarity} \quad \nabla J(x^\star) + \sum_{j=1}^{p} \lambda_j \nabla h_j(x^\star) + \sum_{i=1}^{q} \mu_i \nabla g_i(x^\star) = 0$$

$$\text{primal admissibility} \quad h_j(x^\star) = 0 \qquad j = 1, \ldots, p$$

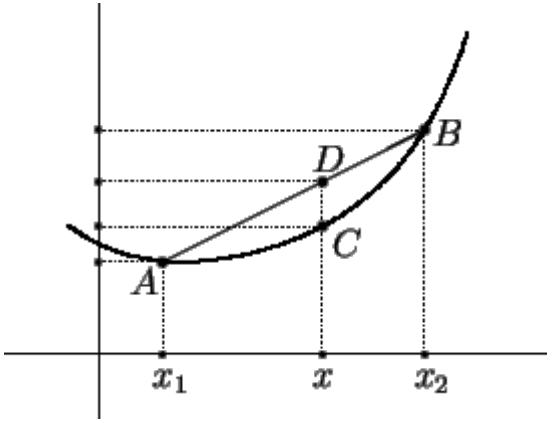
$$g_i(x^\star) \leq 0 \qquad i = 1, \ldots, q$$

$$\text{dual admissibility} \quad \mu_i \geq 0 \qquad i = 1, \ldots, q$$

$$\text{complementarity} \quad \mu_i g_i(x^\star) = 0 \qquad i = 1, \ldots, q$$

Condições Karush-Kuhn-Tucker (KKT)

- Teorema: se **x*** é um ponto estacionário (na qual a derivada da função é zero) do problema primal, então existem multiplicadores de Lagrange **λ, μ** onde (**x***, **λ, μ**) satisfazem as condições KKT
- Se J, g e h forem convexas* (e portanto seus mínimos locais serão também mínimos globais),



* o segmento que une dois pontos está sempre acima da curva

Condições Karush-Kuhn-Tucker (KKT)

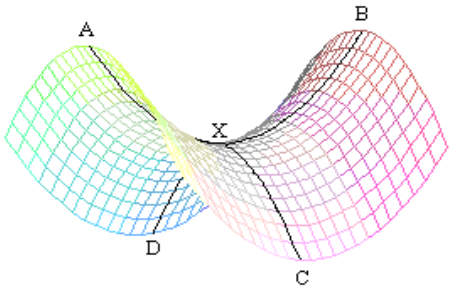
- Teorema: se $\mathbf{x}^*$ é um ponto estacionário (na qual a derivada da função é zero) do problema primal, então existem multiplicadores de Lagrange $\boldsymbol{\lambda}, \boldsymbol{\mu}$ onde ($\mathbf{x}^*$, $\boldsymbol{\lambda}, \boldsymbol{\mu}$) satisfazem as condições KKT
- Se J, g e h forem convexas (e portanto seus mínimos locais serão também mínimos globais), então $\mathbf{x}^*$ que satisfaça as condições KKT é uma solução para o problema.

Condições Karush-Kuhn-Tucker (KKT)

- A condição estacionária (na qual a derivada é zero) pode ser escrita como

$$\nabla \mathcal{L}(\mathbf{x}^\star, \lambda, \mu) = 0$$

- Um ponto de sela é estacionário (onde a derivada parcial em **x** é zero, mesmo não sendo um ponto de mínimo)



- Ponto de sela do lagrangiano: $\max_{\boldsymbol{\lambda}, \boldsymbol{\mu}} \min_{\mathbf{x}} L(\mathbf{x}^*, \boldsymbol{\lambda}, \boldsymbol{\mu})$

Problema dual!

- Variável primal: **x**
- Variáveis duais: **λ, μ**

Voltando para SVMs...

Formulação dual de SVM

- Primal:

$$\begin{cases} \min\limits_{\mathbf{w},b} & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{with} & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 \qquad i = 1, n \end{cases}$$

- Dual:

$$\begin{cases} \max\limits_{\alpha} \min\limits_{\mathbf{w},b} \mathcal{L}(\mathbf{w}, b, \alpha) \\ \alpha_i \geq 0 \end{cases}$$

onde

Porque a desigualdade é "≥ 0"

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

α_i representa a influência da restrição e portanto a influência do exemplo de treinamento (x_i, y_i)

Condições de otimalidade

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

- Computando os gradientes:

$$\begin{cases} \nabla_{\mathbf{w}} \mathcal{L}(\mathbf{w}, b, \alpha) & = \mathbf{w} - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \\ \dfrac{\partial \mathcal{L}(\mathbf{w}, b, \alpha)}{\partial b} & = -\sum_{i=1}^{n} \alpha_i \, y_i \end{cases}$$

- Condições de otimalidade:

$$\begin{cases} \nabla_{\mathbf{w}} \mathcal{L}(\mathbf{w}, b, \alpha) = 0 \;\Rightarrow\; \mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \\ \dfrac{\partial \mathcal{L}(\mathbf{w}, b, \alpha)}{\partial b} = 0 \;\Rightarrow\; \sum_{i=1}^{n} \alpha_i \, y_i = 0 \end{cases}$$

Formulação dual de SVM

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

- Otimalidade: $\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \qquad \sum_{i=1}^{n} \alpha_i \, y_i = 0$

$$\mathcal{L}(\alpha) = \underbrace{\tfrac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_j \alpha_i y_i y_j \mathbf{x}_j^\top \mathbf{x}_i}_{\mathbf{w}^\top \mathbf{w}} - \underbrace{\sum_{i=1}^{n} \alpha_i y_i}_{=0} \underbrace{\sum_{j=1}^{n} \alpha_j y_j \mathbf{x}_j^\top \mathbf{x}_i}_{\mathbf{w}^\top} - b \underbrace{\sum_{i=1}^{n} \alpha_i \, y_i}_{=0} + \sum_{i=1}^{n} \alpha_i$$

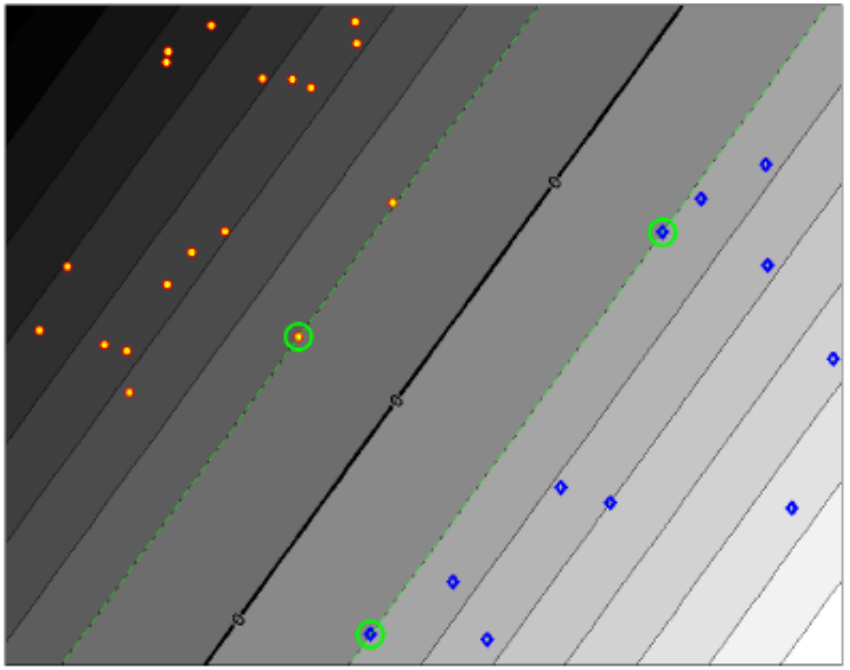
$$= \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_j \alpha_i y_i y_j \mathbf{x}_j^\top \mathbf{x}_i + \sum_{i=1}^{n} \alpha_i$$

- Que também é um problema de programação quadrática:

$$\begin{cases} \min\limits_{\alpha \in \mathbf{R}^n} & \frac{1}{2}\alpha^\top G \alpha - \mathbf{e}^\top \alpha \\ \text{with} & \mathbf{y}^\top \alpha = 0 \\ \text{and} & 0 \leq \alpha_i \qquad\qquad i = 1, n \end{cases}$$

sendo G uma matriz simétrica nxn $\quad G_{ij} = y_i y_j \mathbf{x}_j^\top \mathbf{x}_i$

Resolvendo o dual



- Influência dos pontos:
  - α_i = 0 : ponto inútil
  - α_i != 0 : ponto de suporte

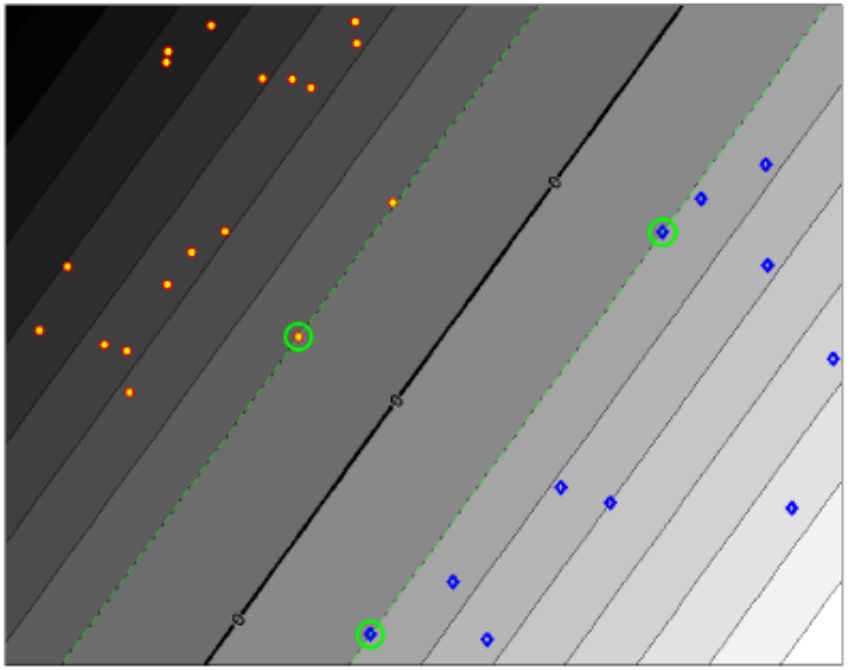
Amostra de treinamento de tamanho n, cada ponto de dimensão d (ex: d = 2)

Fronteira de decisão

$$f(\mathbf{x}) = \sum_{j=1}^{d} w_j x_j + b = \sum_{i=1}^{n} \alpha_i \, y_i (\mathbf{x}^\top \mathbf{x}_i) + b$$

primal dual

Resolvendo o dual



- Influência dos pontos:
  - α_i = 0 : ponto inútil
  - α_i != 0 : ponto de suporte

Amostra de treinamento de tamanho n, cada ponto de dimensão d (ex: d = 2)

Fronteira de decisão

$$f(\mathbf{x}) = \sum_{j=1}^{d} w_j x_j + b = \sum_{i=1}^{3} \alpha_i \, y_i (\mathbf{x}^\top \mathbf{x}_i) + b$$

Fronteira de decisão só depende de 3 pontos (d+1)

Primal x Dual

Primal

$$\begin{cases} \min\limits_{\mathbf{w}\in\mathbf{R}^d, b\in\mathbf{R}} & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{with} & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 \\ & i = 1, n \end{cases}$$

- d + 1 variáveis desconhecidas
- n restrições
- Programação quadrática clásssica
- Ok quando d << n

Dual

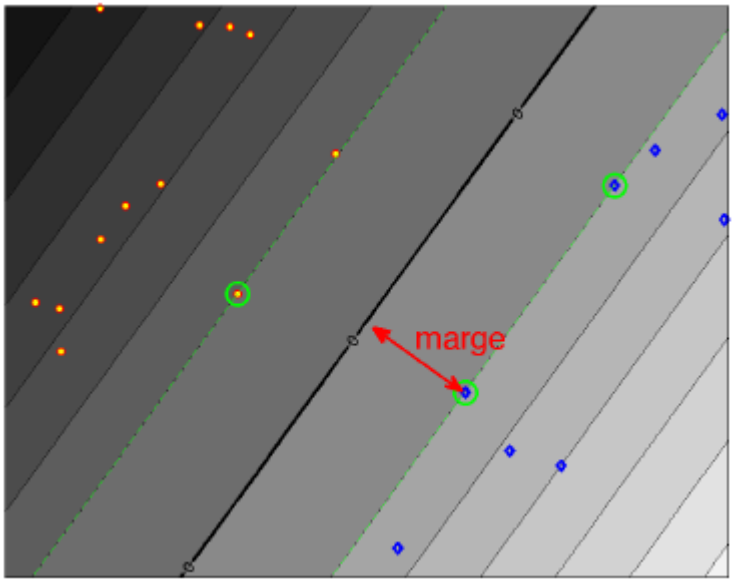
$$\begin{cases} \min\limits_{\alpha\in\mathbf{R}^n} & \frac{1}{2}\alpha^\top G\alpha - \mathbf{e}^\top\alpha \\ \text{with} & \mathbf{y}^\top\alpha = 0 \\ \text{and} & 0 \leq \alpha_i \qquad\qquad i = 1, n \end{cases}$$

- n variáveis desconhecidas
- G matriz pairwise de influência
- n restrições
- Fácil de resolver
- Para ser usado quando d > n

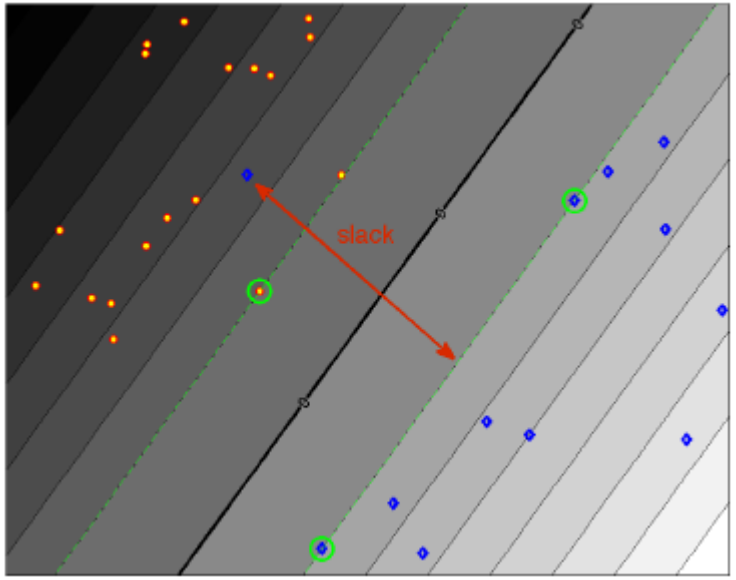
$$f(\mathbf{x}) = \sum_{j=1}^{d} w_j x_j + b = \sum_{i=1}^{n} \alpha_i\, y_i(\mathbf{x}^\top \mathbf{x}_i) + b$$

O que vimos até agora: SVMs *Hard-Margin*

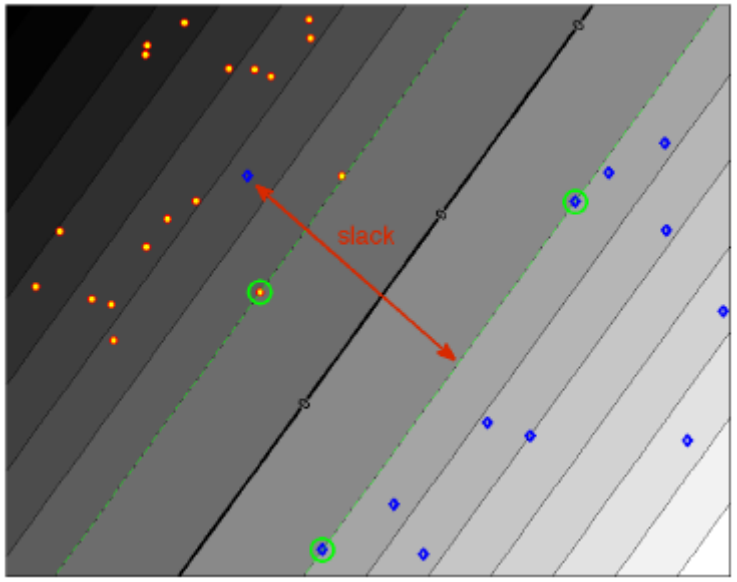
- SVMs lineares
- Casos separáveis



O caso não separável



O caso não separável



- Erro associado à posição do ponto em relação à reta

$$(x_i, y_i) \quad \begin{cases} y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 \Rightarrow & \xi_i = 0 \\ & \xi_i = 1 - y_i(\mathbf{w}^\top \mathbf{x}_i + b) > 0 \end{cases}$$

Sem erro:

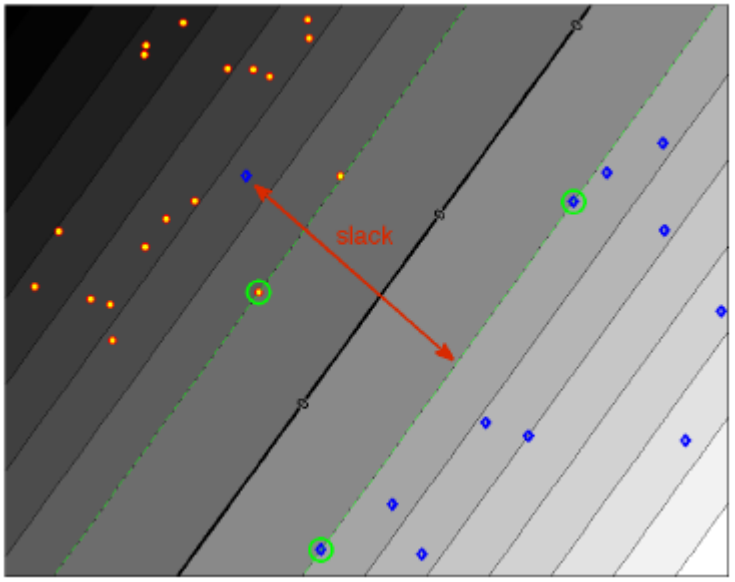
Com erro:

O caso não separável

- Esperamos que o erro seja 0 para a maioria dos pontos
- Esperamos que a soma do erro seja o menor possível

Introdução de variáveis “slack” (folga)

- Esperamos que o erro seja 0 para a maioria dos pontos
- Esperamos que a soma do erro seja o menor possível



$$\begin{cases} \min\limits_{\mathbf{w},b,\xi} & \frac{1}{2}\|\mathbf{w}\|^2 \\ \min\limits_{\mathbf{w},b,\xi} & \frac{C}{p}\sum\limits_{i=1}^{n}\xi_i^p \\ \text{with} & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i \\ & \xi_i \geq 0 \quad i = 1, n \end{cases}$$

Peso dado ao erro

p = 1 ou 2

Problema primal (soft-margin SVM):

$$\begin{cases} \min\limits_{\mathbf{w},b} & \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{p}\sum\limits_{i=1}^{n}\xi_i^p & \\ \text{with} & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i & i = 1, n \\ & \xi_i \geq 0 & i = 1, n \end{cases}$$

Lagrangiano para o problema dual:

$$\mathcal{L}(\mathbf{w}, b, \alpha, \beta) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i^p - \sum_{i=1}^{n}\alpha_i\left(y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1 + \xi_i\right) - \sum_{i=1}^{n}\beta_i\xi_i$$

Condições de otimalidade (p = 1 : L1 SVM)

$$\mathcal{L}(\mathbf{w}, b, \alpha, \beta) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n}\xi_i - \sum_{i=1}^{n}\alpha_i\left(y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1 + \xi_i\right) - \sum_{i=1}^{n}\beta_i\xi_i$$

- Computando os grandientes:

$$\begin{cases} \nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}, b, \alpha) & = \mathbf{w} - \sum_{i=1}^{n}\alpha_i y_i \mathbf{x}_i \\ \dfrac{\partial\mathcal{L}(\mathbf{w}, b, \alpha)}{\partial b} & = -\sum_{i=1}^{n}\alpha_i\, y_i \\ \nabla_{\xi_i}\mathcal{L}(\mathbf{w}, b, \alpha) & = C - \alpha_i - \beta_i \end{cases}$$

- Condições de otimalidade:

  Nada mudou para **w** e b

$$C - \alpha_i - \beta_i = 0 \quad , \text{ como } \quad \beta_i \geq 0 \quad \Rightarrow \quad \alpha_i \leq C$$

Formulação dual (L1 Soft-margin SVM)

$$\begin{cases} \min\limits_{\alpha \in \mathbf{R}^n} & \frac{1}{2}\alpha^\top G \alpha - \mathbf{e}^\top \alpha \\ \text{with} & \mathbf{y}^\top \alpha = 0 \\ \text{and} & 0 \leq \alpha_i \leq C \qquad\qquad i = 1, n \end{cases}$$

Primal x dual (L1 Soft-margin SVM)

Primal

$$\begin{cases} \min\limits_{\mathbf{w},b,\xi\in\mathbf{R}^n} & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum\limits_{i=1}^{n}\xi_i \\ \text{with} & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i \\ & \xi_i \geq 0 \quad i = 1, n \end{cases}$$

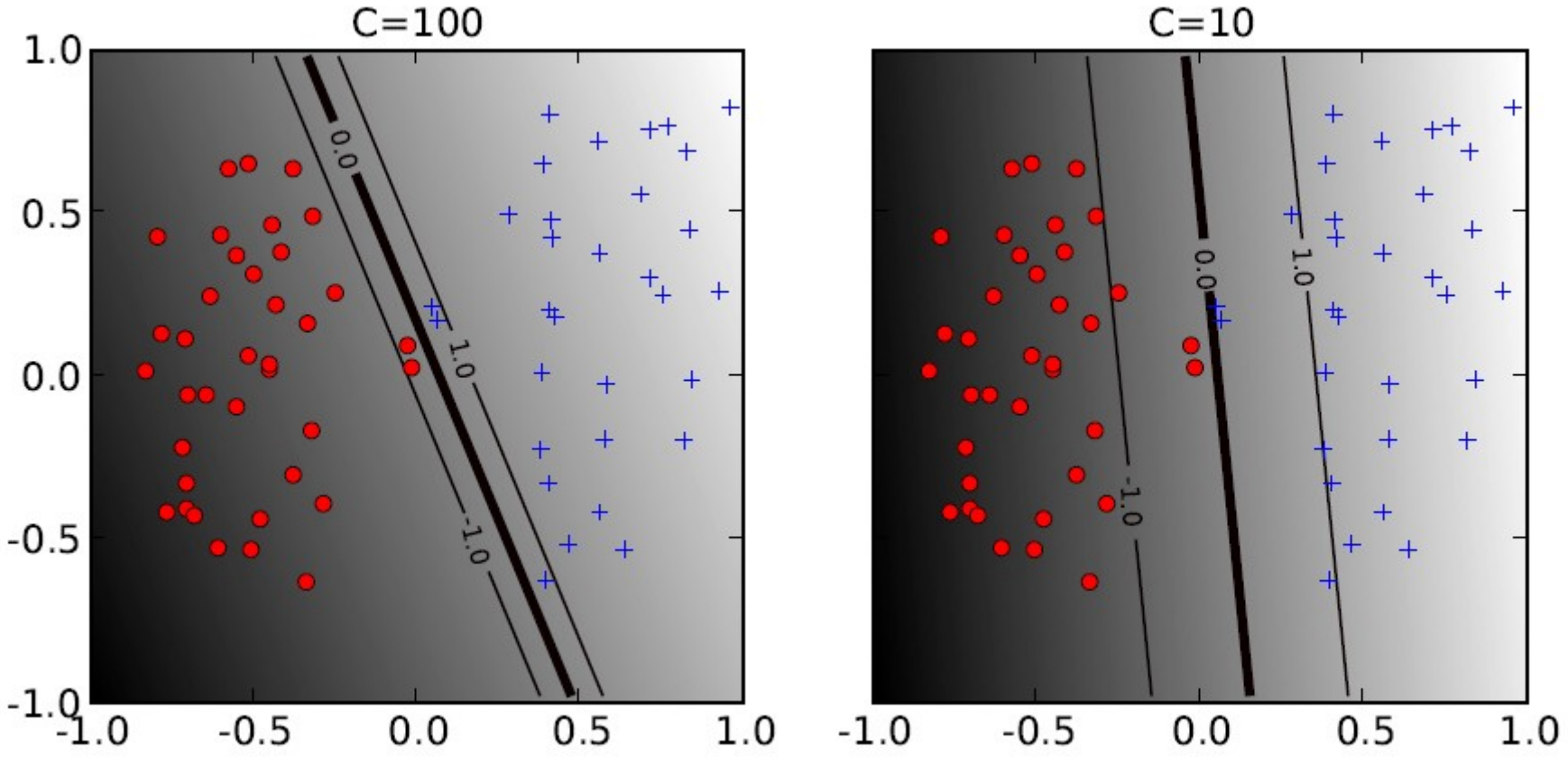
- d + n + 1 variáveis desconhecidas
- 2n restrições
- Programação quadrática clássica
- Para ser usado quando n é muito grande para construir G (d < n)

Dual

$$\begin{cases} \min\limits_{\alpha\in\mathbf{R}^n} & \frac{1}{2}\alpha^\top G\alpha - \mathbf{e}^\top\alpha \\ \text{with} & \mathbf{y}^\top\alpha = 0 \\ \text{and} & 0 \leq \alpha_i \leq C \qquad i = 1, n \end{cases}$$

- n variáveis desconhecidas
- G matriz pairwise de influência nxn
- 2n restrições
- Fácil de resolver
- Para ser usado quando d > n

Efeito da variação do C



http://pyml.sourceforge.net/doc/howto.pdf

Primal x dual (L2 Soft-margin SVM)

Primal

$$\begin{cases} \min\limits_{\mathbf{w},b,\xi\in\mathbf{R}^n} & \frac{1}{2}\|\mathbf{w}\|^2 + \frac{1}{C}\sum\limits_{i=1}^{n}\xi_i^2 \\ \text{with} & y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1 - \xi_i \\ & \xi_i \geq 0 \quad i = 1, n \end{cases}$$

- d + n + 1 variáveis desconhecidas
- 2n restrições
- Programação quadrática clássica
- Para ser usado quando n é muito grande para construir G (d << n)

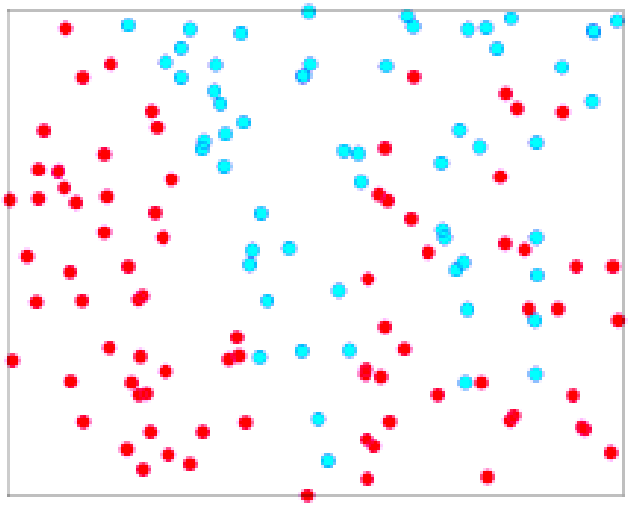
Dual

$$\begin{cases} \min\limits_{\alpha\in\mathbf{R}^n} & \frac{1}{2}\alpha^\top(G + \frac{1}{C}I)\alpha - \mathbf{e}^\top\alpha \\ \text{with} & \mathbf{y}^\top\alpha = 0 \\ \text{and} & 0 \leq \alpha_i \qquad i = 1, n \end{cases}$$

- n variáveis desconhecidas
- G matriz pairwise de influência nxn
- n restrições
- Fácil de resolver
- Para ser usado quando d > n

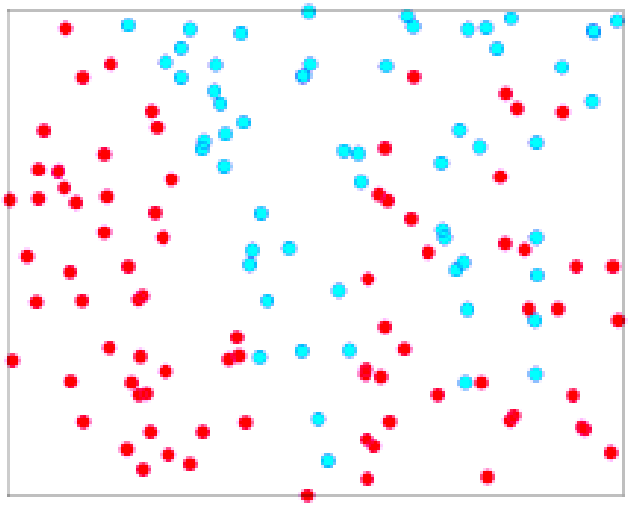
SVMs não lineares

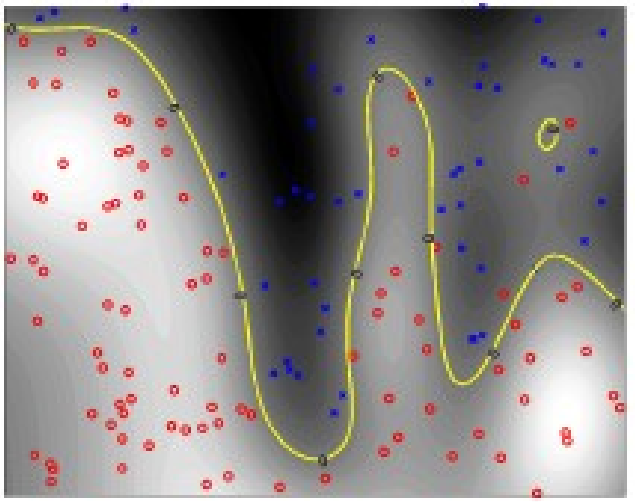
- Até agora vimos SVM lineares (função de decisão é uma reta que separa os dados)
- Mas nem sempre é o caso...



SVMs não lineares

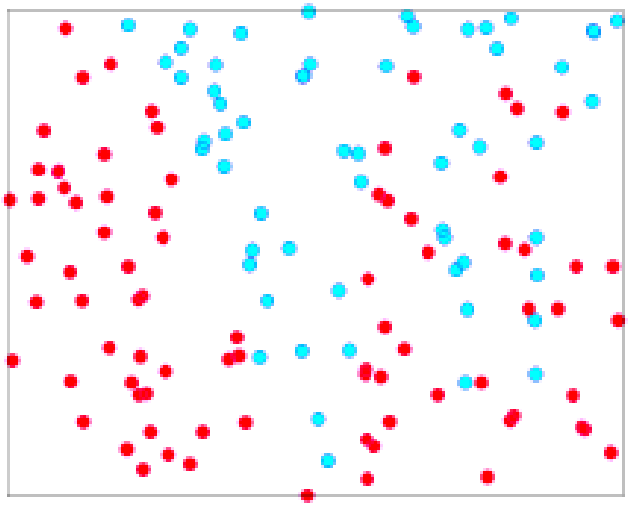
- Até agora vimos SVM lineares (função de decisão é uma reta que separa os dados)
- Mas nem sempre é o caso...

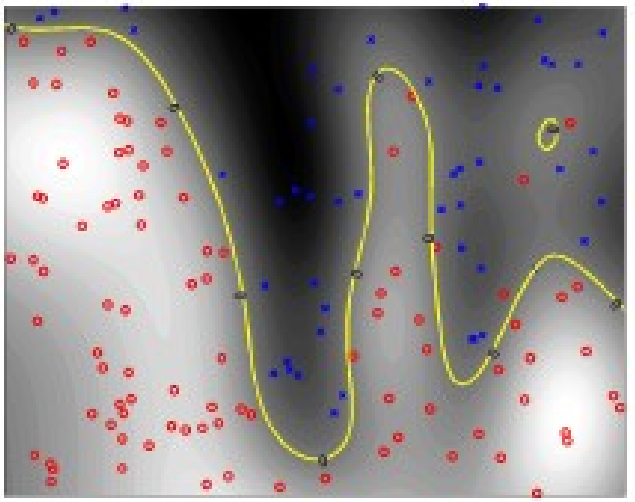




SVMs não lineares

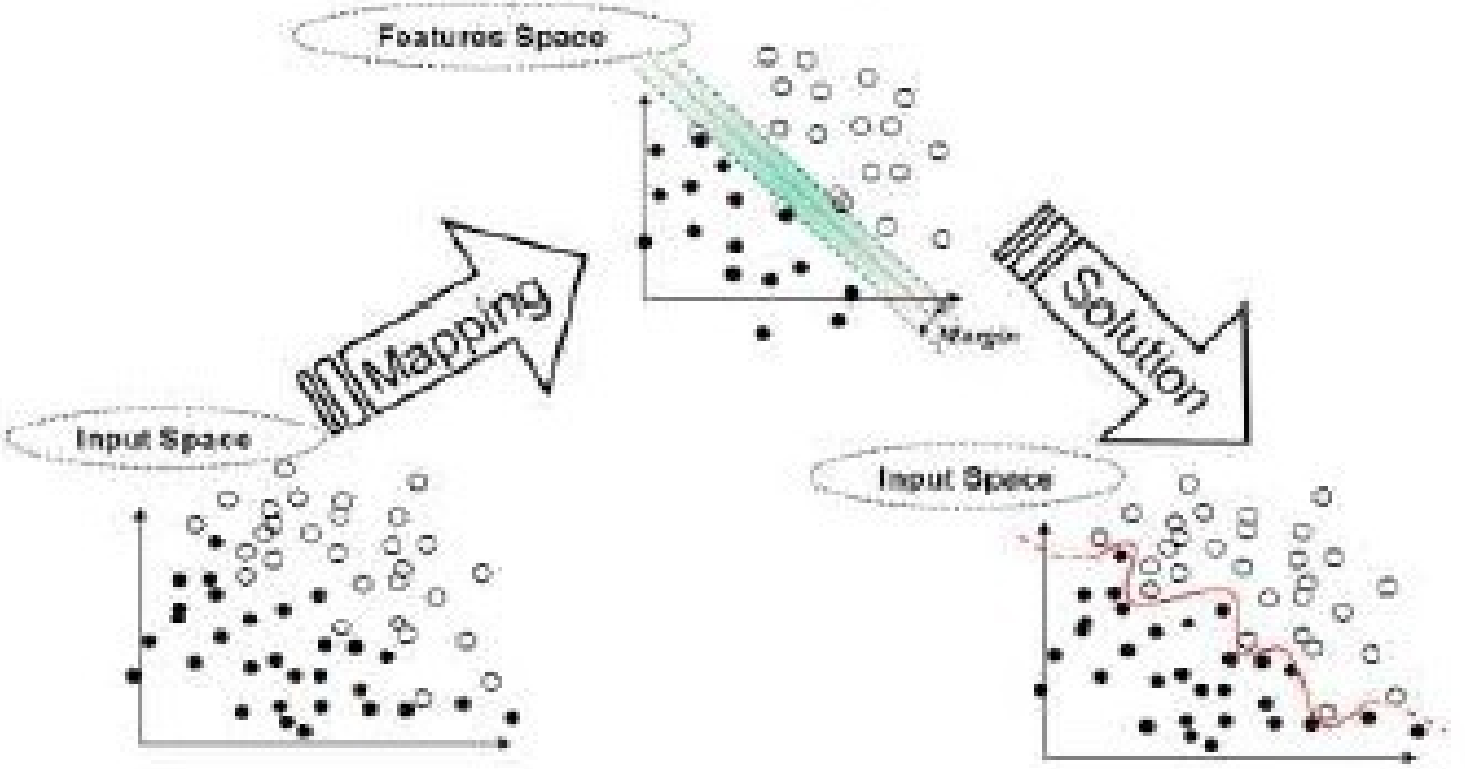
- Até agora vimos SVM lineares (função de decisão é uma reta que separa os dados)
- Mas nem sempre é o caso...



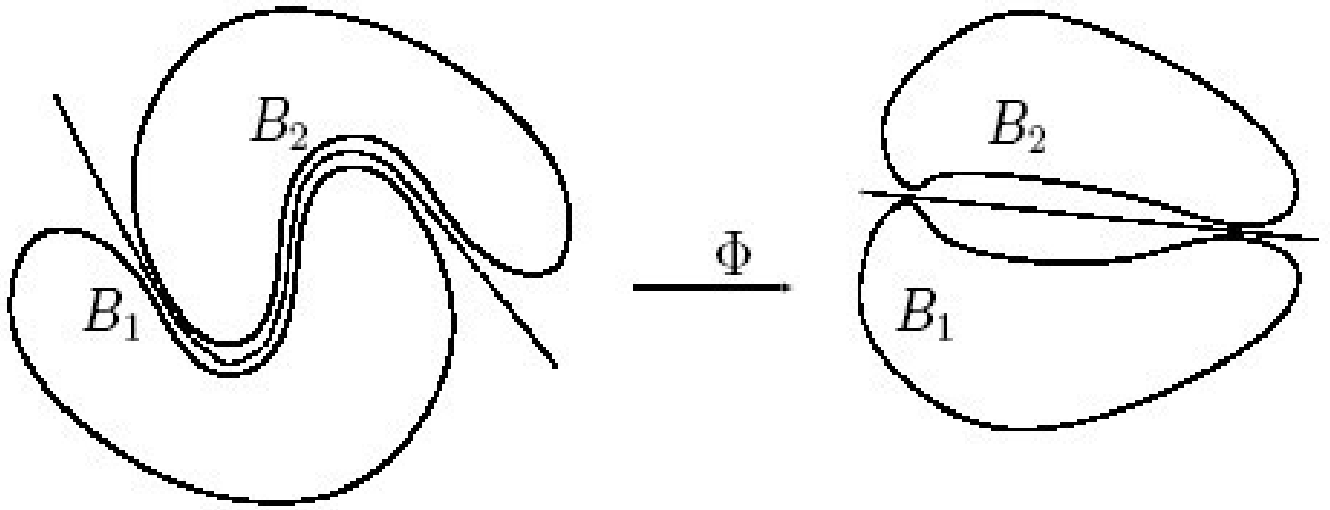


- Segredo: mapear os dados para outro espaço (geralmente de dimensão maior) e criar um classificador linear nesse novo espaço (com cara de não linear no espaço atual) – para isso usaremos a "kernel trick"

The SVM algorithm

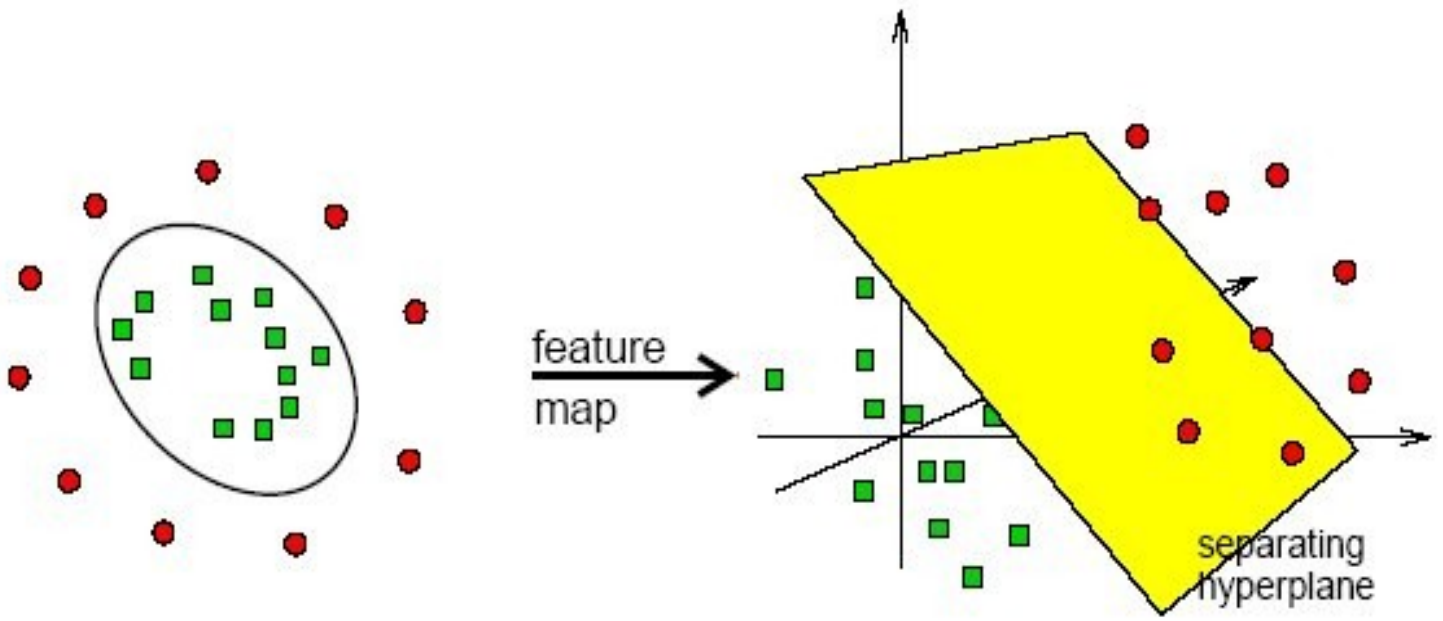


http://www.dtreg.com/svm.htm



http://www.dtreg.com/svm.htm

Separation may be easier in higher dimensions



complex in low dimensions

simple in higher dimensions

http://www.dtreg.com/svm.htm

Kernel polinomial

https://www.youtube.com/watch?time_continue=42&v=3liCbRZPrZA

SVMs não lineares

- Forma dual de SVMs lineares: produto interno de pares de exemplos de treinamento (x_i . x_j)
  - Vai facilitar a formulação não-linear

Primal

$$\begin{cases} \min\limits_{\mathbf{w},b,\xi\in\mathbf{R}^n} & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum\limits_{i=1}^{n}\xi_i \\ \text{with} & y_i(\mathbf{w}^\top\mathbf{x}_i + b) \geq 1 - \xi_i \\ & \xi_i \geq 0 \quad i = 1, n \end{cases}$$

Dual

$$\begin{cases} \min\limits_{\alpha\in\mathbf{R}^n} & \frac{1}{2}\alpha^\top G\alpha - \mathbf{e}^\top\alpha \\ \text{with} & \mathbf{y}^\top\alpha = 0 \\ \text{and} & 0 \leq \alpha_i \leq C \qquad i = 1, n \end{cases}$$

$$G_{ij} = y_i y_j \mathbf{x}_j^\top \mathbf{x}_i$$

$$f(\mathbf{x}) = \sum_{j=1}^{d} w_j x_j + b = \sum_{i=1}^{n} \alpha_i\, y_i(\mathbf{x}^\top\mathbf{x}_i) + b$$

SVMs não lineares

$\Phi: \mathcal{L} \rightarrow \mathcal{H}$,

sendo que $\mathcal{L}$ é R^d (*"low" dimensional space*),

$\mathcal{H}$ é um novo espaço (*"high" dimensional space*, potencialmente de dimensão infinita)

- O aprendizado do classificador linear depende de $x_i \cdot x_j$
  - No novo espaço, dependerá de $\Phi(x_i) \cdot \Phi(x_j)$

- Se definirmos uma função (kernel)
  $K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$
  não precisamos conhecer Φ ("kernel trick")

SVMs não lineares

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i \, y_i (\mathbf{x}^\top \mathbf{x}_i) + b$$

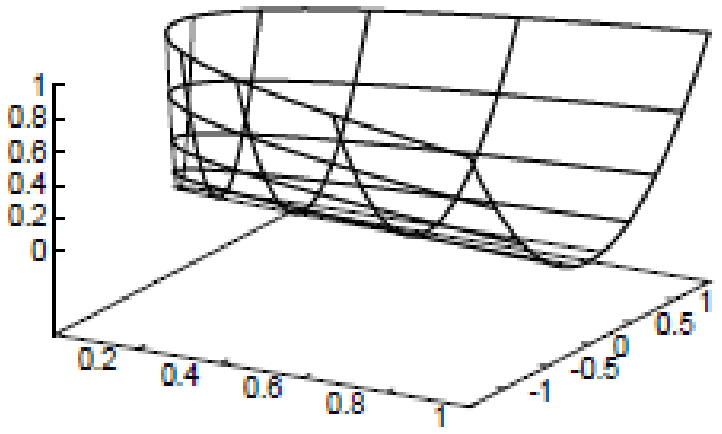
$$f(\mathbf{x}) = \sum_{i=1}^{N_S} \alpha_i y_i \Phi(\mathbf{s}_i) \cdot \Phi(\mathbf{x}) + b = \sum_{i=1}^{N_S} \alpha_i y_i K(\mathbf{s}_i, \mathbf{x}) + b$$

Onde cada s_i é um dos N_s vetores de suporte

Exemplo

(Φ mostrada apenas para exemplificar, pois não precisamos saber Φ !!!)

- $K(x, y) = (x \cdot y)^2$
- $\Phi : R^2 \rightarrow R^3$
- $\Phi(x) = (x_1^2, \sqrt{2}\, x_1\, x_2, x_2^2)$
- Quadrado $[-1,1] \times [-1,1]$ (R^2), temos em R^3 :



[BURGES, 1998]

“Reutilização” de certos kernels

Um mesmo kernel pode servir para diferentes $\mathcal{H}$ e Φ.

Exemplo: (o mesmo kernel do exemplo anterior)

$K(x,y) = (x \cdot y)^2$

$\mathcal{H} = R^4 \quad \Phi = (x_1^2, x_1x_2, x_1x_2, x_2^2)$

Toda função funciona como kernel?

- Pergunta: para toda função K existe um par {$\mathcal{H}$, Φ} no qual:
  - x, y $\in \mathcal{L}$ (espaço atual - “low dimensional”)
  - Φ : $\mathcal{L} \rightarrow \mathcal{H}$
  - K(x, y) = Φ (x) . Φ(y)
  - ?
-
-

Toda função funciona como kernel?

- Pergunta: para toda função K existe um par {$\mathcal{H}$, Φ} no qual:
  - x, y $\in \mathcal{L}$ (espaço atual - “low dimensional”)
  - Φ : $\mathcal{L} \rightarrow \mathcal{H}$
  - K(x, y) = Φ (x) . Φ(y)
  - ?
- Não!
- Somente as que satisfazem a condição de Mercer

Condição de Mercer

There exists a mapping Φ and an expansion

$$K(\mathbf{x}, \mathbf{y}) = \sum_i \Phi(\mathbf{x})_i \Phi(\mathbf{y})_i$$

if and only if, for any $g(\mathbf{x})$ such that

$$\int g(\mathbf{x})^2 d\mathbf{x} \text{ is finite}$$

then

$$\int K(\mathbf{x}, \mathbf{y}) g(\mathbf{x}) g(\mathbf{y}) d\mathbf{x} d\mathbf{y} \geq 0.$$

[BURGES, 1998]

Condição de Mercer

Pode ser difícil provar que certas funções podem ser usadas como kernel (ou seja, que satisfazem a condição)

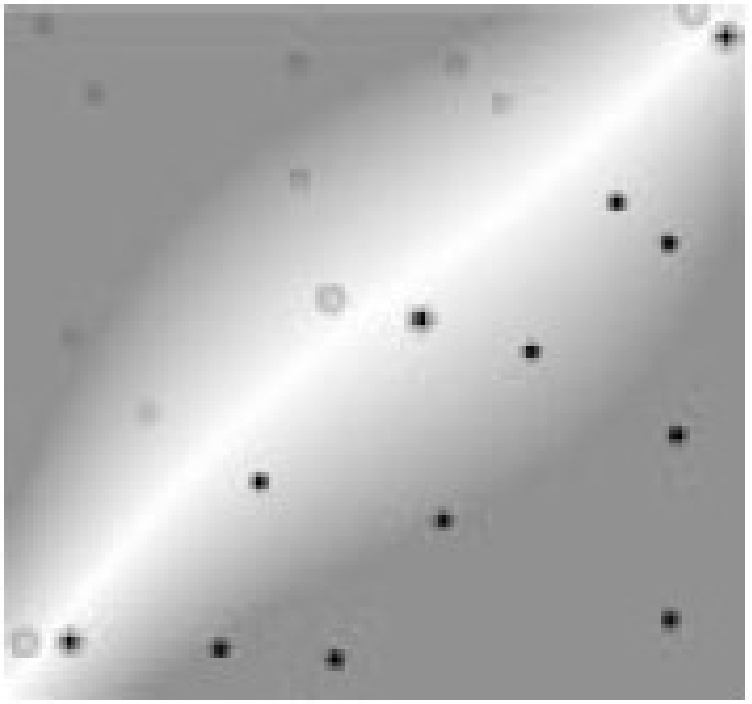
Para algumas existe a prova

Em particular vale para...

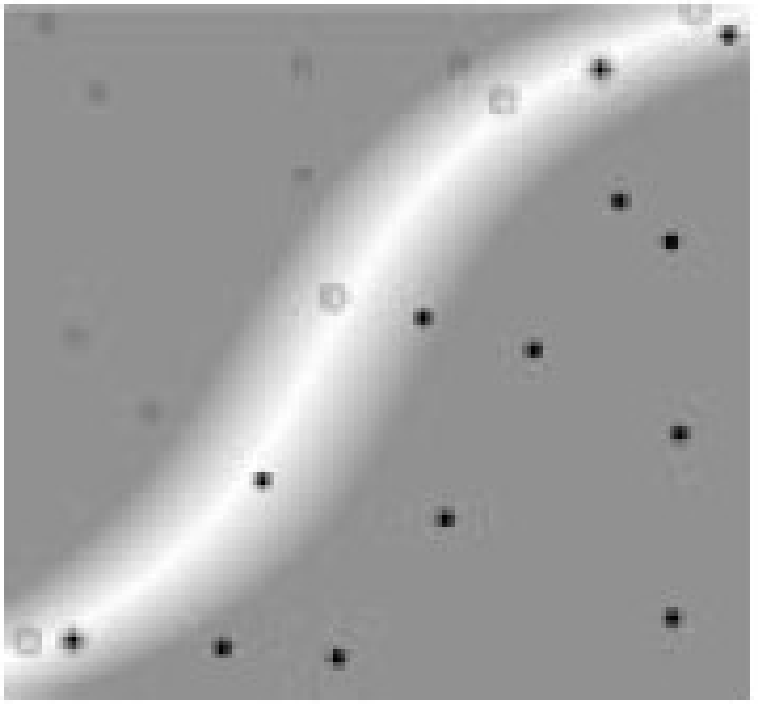
- Kernel polinomial homogêneo
  $K(x,y) = (x \cdot y)^p$, p inteiro positivo

- Kernel polinomial não-homogêneo
  $K(x,y) = (x \cdot y + 1)^p$, p inteiro positivo

Exemplo Polinomial não-homogêneo (p = 3)



Linearmente separáveis



Linearmente não-separáveis
(mas separáveis na SVM não-linear)

[BURGES, 1998]

Vários possíveis kernels

- Boa notícia: há vários possíveis kernels (cada um com parâmetros a serem escolhidos) que podem ser usados

Vários possíveis kernels

- Boa notícia: há vários possíveis kernels (cada um com parâmetros a serem escolhidos) que podem ser usados

- Má notícia: como escolher o kernel e os valores de seus parâmetros ainda é uma questão em aberto

Alguns kernels bastante usados

Kernel Linear:

$K(x,y) = x.y$

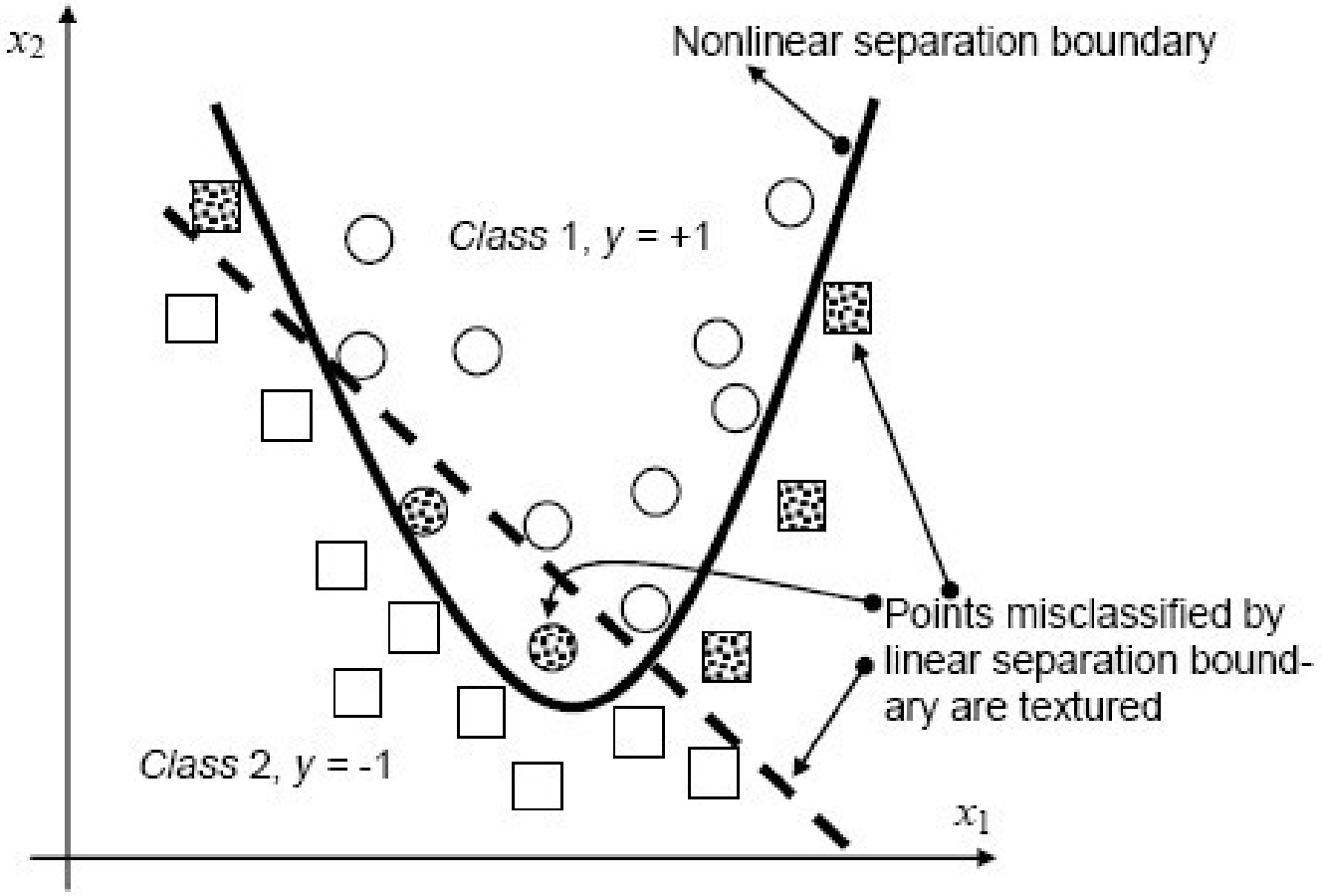
- Recomendado quando:
  - os dados são linearmente separáveis (SVM linear)
  - os dados são esparsos (ex: categorização de textos)
  - Os dados possuem um grande número de características

Alguns kernels bastante usados

Kernel Polinomial:

$K(x,y) = (x \cdot y)^p$, p inteiro positivo

- P = 1 equivale ao linear
- Popular em processamento de imagens
- Espaço $\mathcal{H}$ de dimensão $\binom{d+p-1}{p}$
  - Ex: p = 4 e imagem 16x16 => $\dim(\mathcal{H}) = 183\,181\,376$



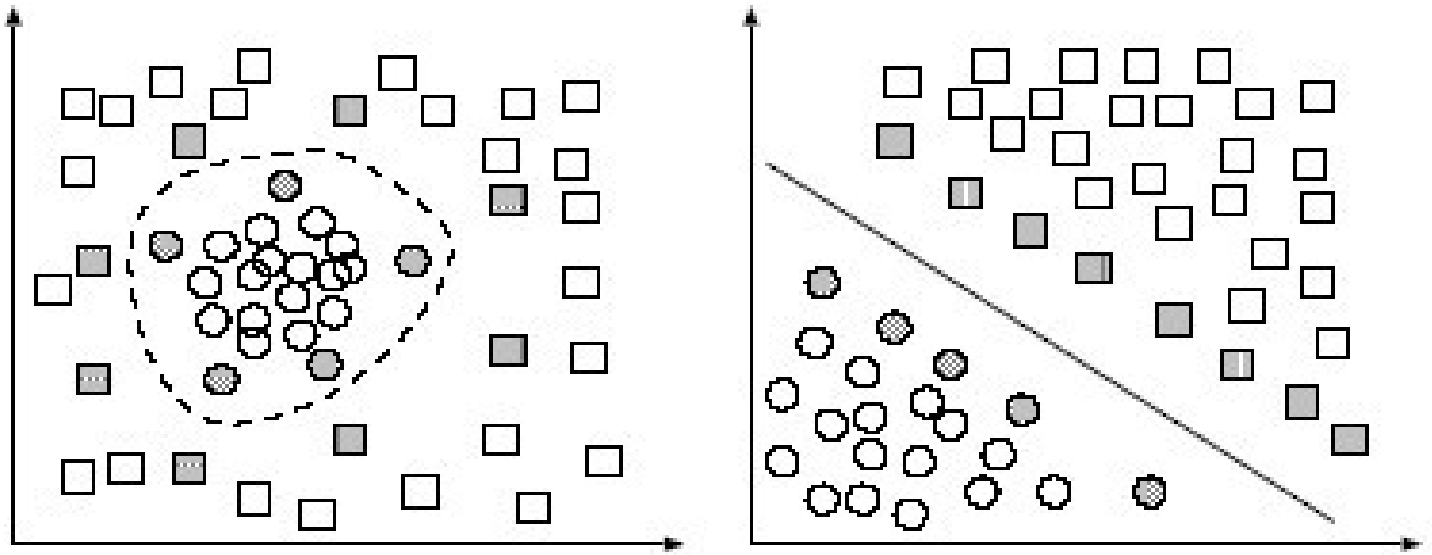
http://www.dtreg.com/svm.htm

Alguns kernels bastante usados

Kernel (Gaussian) Radial Basis Function (RBF):

$K(x,y) = \exp(-\gamma \|x.y\|^2)$, γ um parâmetro positivo

- Considerado como uma boa primeira tentativa quando não se sabe muito sobre os dados



(a) Radial Basis Function (b) RBF mapping

Separable classification with Radial Basis kernel functions in different space. Left: original space. Right: feature space.

http://www.dtreg.com/svm.htm

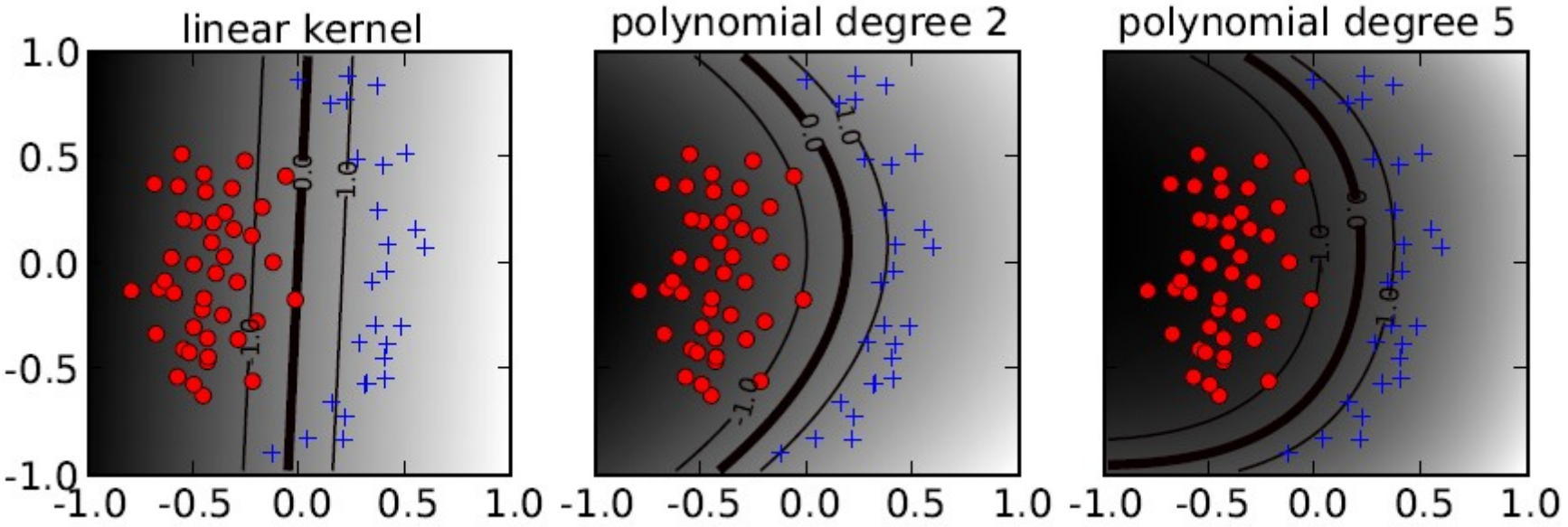
Alguns kernels bastante usados

Kernel Sigmóide:

$K(x,y) = \tanh(\kappa\, x.y - \delta)$

- Considerado equivalente a uma rede neural de duas camadas (rede perceptron) usando função de ativação sigmóide

Efeitos da variação do kernel e parâmetros



http://pyml.sourceforge.net/doc/howto.pdf

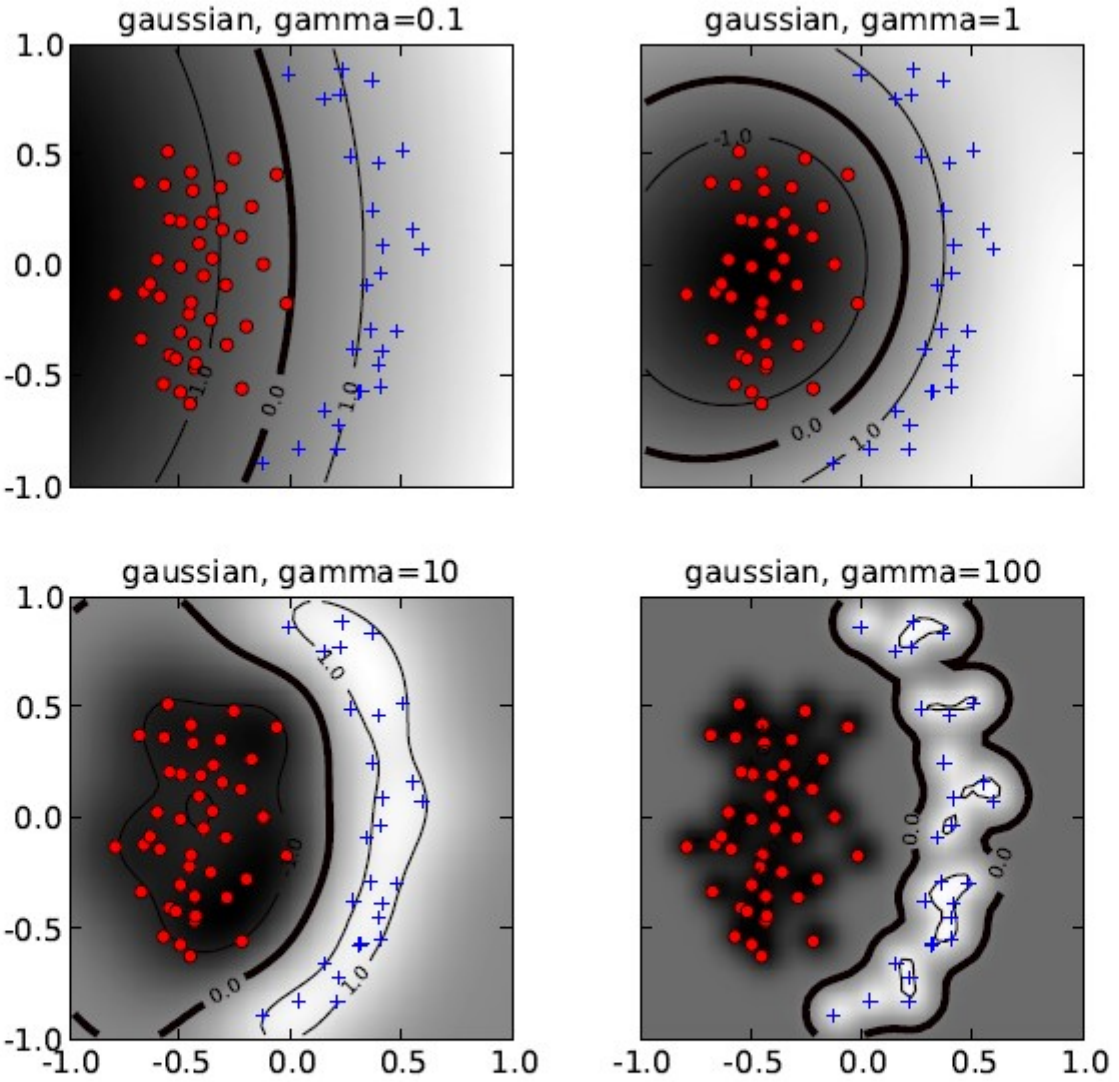
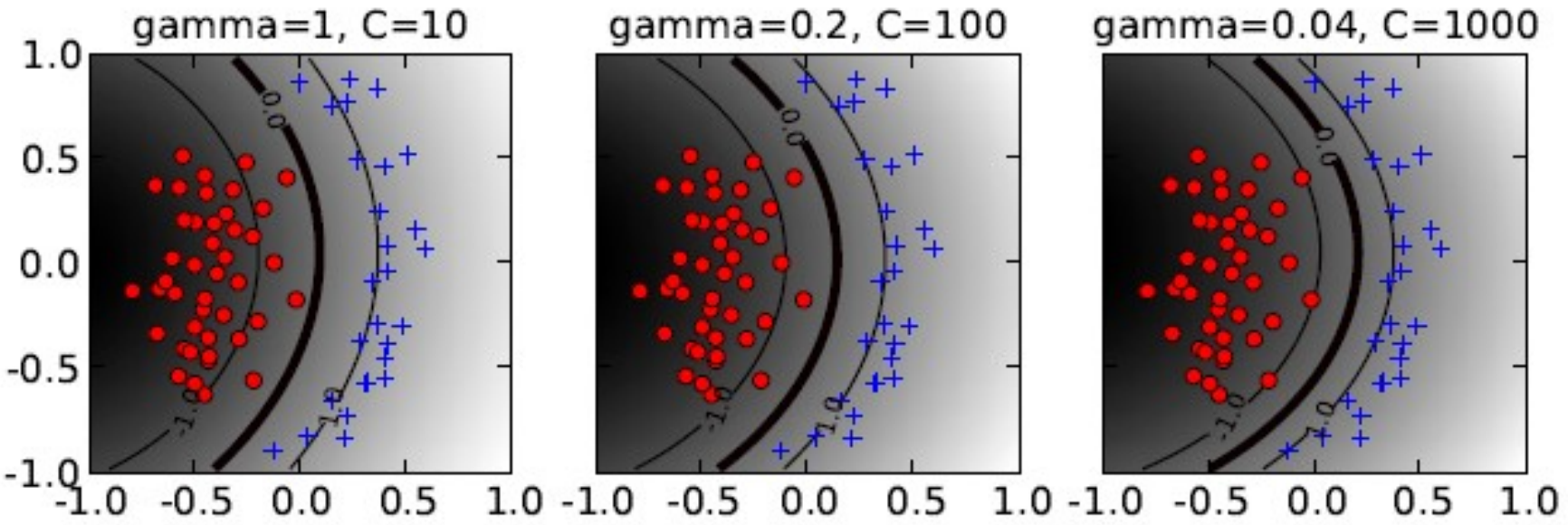


Figure 5: The effect of the inverse-width parameter of the Gaussian kernel (γ) for a fixed value of the soft-margin constant. For small values of γ (upper left) the decision boundary is nearly linear. As γ increases the flexibility of the decision boundary increases. Large values of γ lead to overfitting (bottom). The figure style follows that of Figure 3.



Escolha de kernel e parâmetros

Não há uma regra direta

Metodologia utilizada: seleção por validação cruzada

Dá para usar com variáveis categóricas?

Dá para usar com variáveis categóricas?

Não adequado

Se for usar, será preciso:

- transformar as variáveis categóricas em variáveis dummies
- normalizar todas as variáveis numéricas (para valores de 0 a 1)

Normalização (scaling)

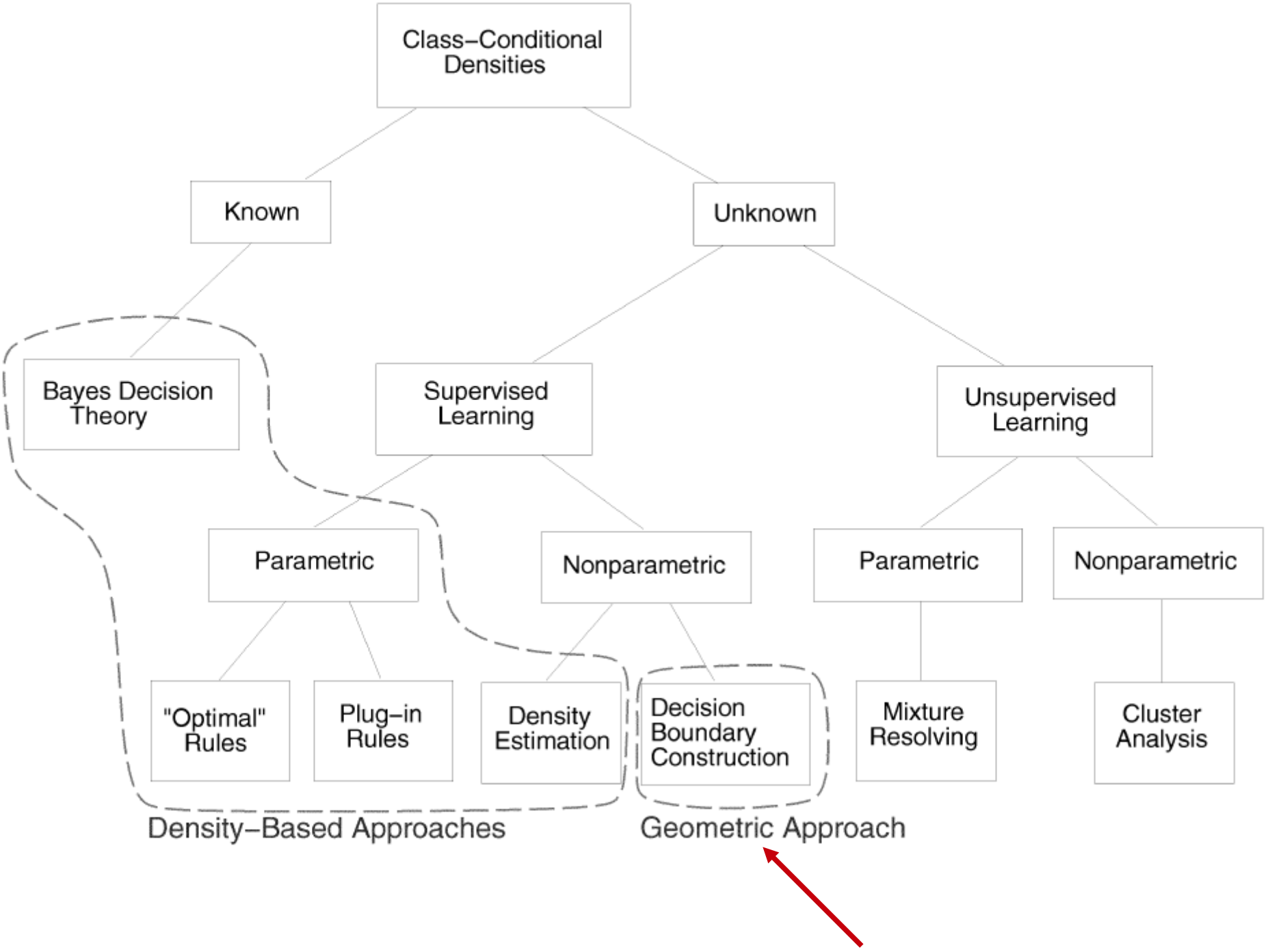
Importante para qualquer tipo de variável.

Lembrar de utilizar nos dados de teste a mesma normalização utilizada nos dados de treinamento

Ex: se dados de treinamento estão no intervalo [-10;+10] e são normalizados para [-1,1]:

se os dados de teste estiverem no intervalo [-11;+8] devem ir para [-1.1;0.8]

Guia prático bem interessante:
https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf


Class–Conditional Densities
Known
Unknown
Bayes Decision Theory
Supervised Learning
Unsupervised Learning
Parametric
Nonparametric
Parametric
Nonparametric
"Optimal" Rules
Plug–in Rules
Density Estimation
Decision Boundary Construction
Mixture Resolving
Cluster Analysis
Density–Based Approaches
Geometric Approach

SVM no R

Pacote e1071 (não é o único)

Função svm()

Parâmetro cost (“C” - custo do erro)

Kernels: (parâmetros gamma, coef0 e degree)

- Linear
- Polinomial: $(\text{gamma} * x.y + \text{coef0})^{\text{degree}}$
- Radial basis: $\exp(-\text{gamma}*|u-v|^2)$
- Sigmoide: $\tanh(\text{gamma} * x.y + \text{coef0})$

Trabalho para dia 24/10

SVM:

- Realizar validação cruzada para testar SVMs utilizando:

- todas a características
- apenas com os componentes principais
- apenas com as características selecionadas pelo selecionador 1
- apenas com as características selecionadas pelo selecionador 2 (opcional)

Em cada um, calcular os valores médios de precisão, revocação e acurácia

Teste diferentes valores de C e funções kernels e seus parâmetros

Apresentar:

- os resultados em uma tabela (métricas nas colunas, com/sem seleção de características nas linhas)
- destacar o que apresentou menor erro
- discutir os resultados

Trabalho - Exemplo

RADIAL									
TODAS AS CARACTERÍSTICAS									
Custo	**0,1**			**0,5**			**10**		
Gama	**0,1**								
	Recall	Precisão	Erro	Recall	Precisão	Erro	Recall	Precisão	Erro
	0,794	1,000	0,204	0,794	1,000	0,204	0,839	0,924	0,200
RELIEF									
Gama	**0,1**								
Custo	**0,1**			**0,5**			**10**		
	Recall	Precisão	Erro	Recall	Precisão	Erro	Recall	Precisão	Erro
	0,794	1,000	0,204	0,793	0,991	0,211	0,874	0,877	0,200
PCA									
Gama	**0,1**								
Custo	**0,1**			**0,5**			**10**		
	Recall	Precisão	Erro	Recall	Precisão	Erro	Recall	Precisão	Erro
	0,794	1,000	0,204	0,794	1,000	0,204	0,883	0,920	0,159

Referências

- BURGES, C. J. C. A tutorial on support vector machines for pattern recognition. **Data Mining and Knowledge Discovery**, v.2, pags. 121-167, 1998
- CANU, S. SVM and Kernel Machines. Tutorial no CIARP, 2010