FUNDAMENTOS DA INTELIGÊNCIA ARTIFICIAL

Aula 08 - Breve introdução às Máquinas de Vetores de Suporte (SVMs)

Prof. Rafael G. Mantovani

Londrina - PR, Brasil
Novembro, 2018

Universidade Estadual de Londrina - UEL
Departamento de Computação - DC

Roteiro

1. Introdução
2. Máquinas de Vetores Suporte (SVMs)
3. SVMs com margens rígidas
4. Exemplo
5. Síntese / Próximas Aulas
6. Referências

Roteiro

Introdução

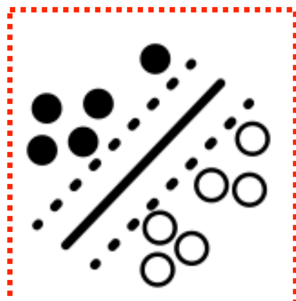
- Máquinas de Vetores de Suporte
  - *Support Vector Machines* (SVMs)
  - Teoria do Aprendizado Estatístico (TAE)

  - Buscam, em resumo, maximizar a margem de separação entre elementos de duas classes {+1, -1}
    - consistente com o princípio de minimização do Risco Empírico (RE)

Introdução

Generalização

Robustez



SVMs

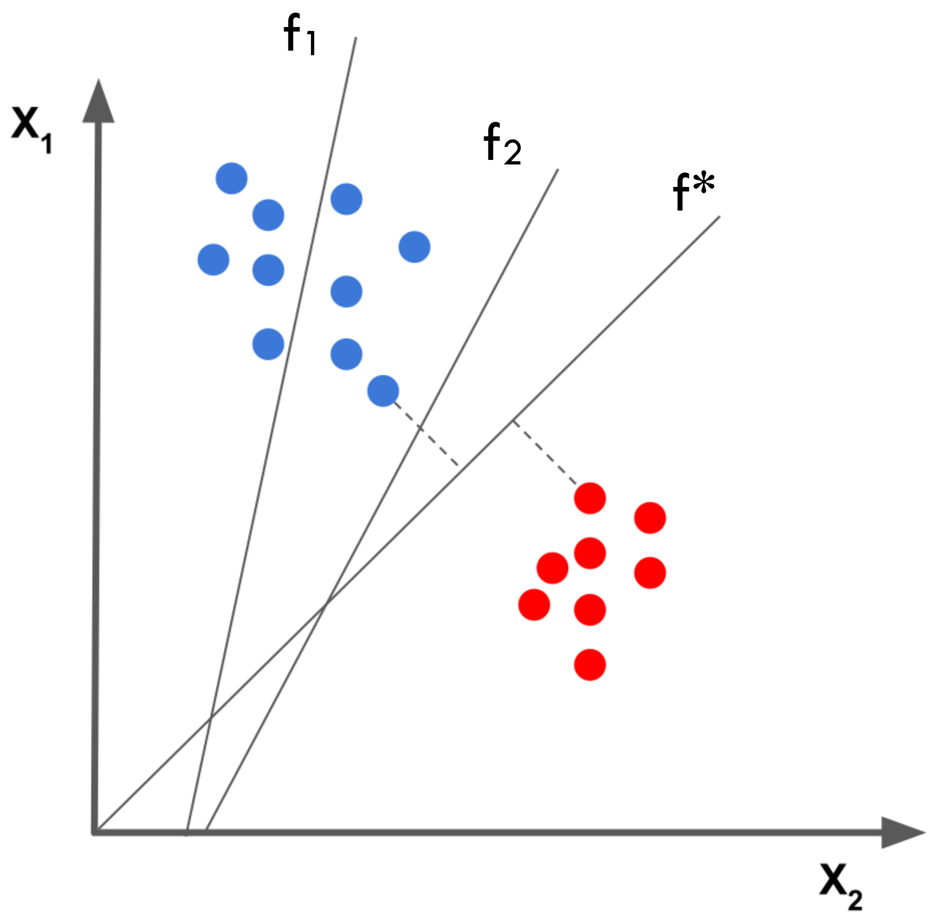
Teoria bem definida

Convexidade

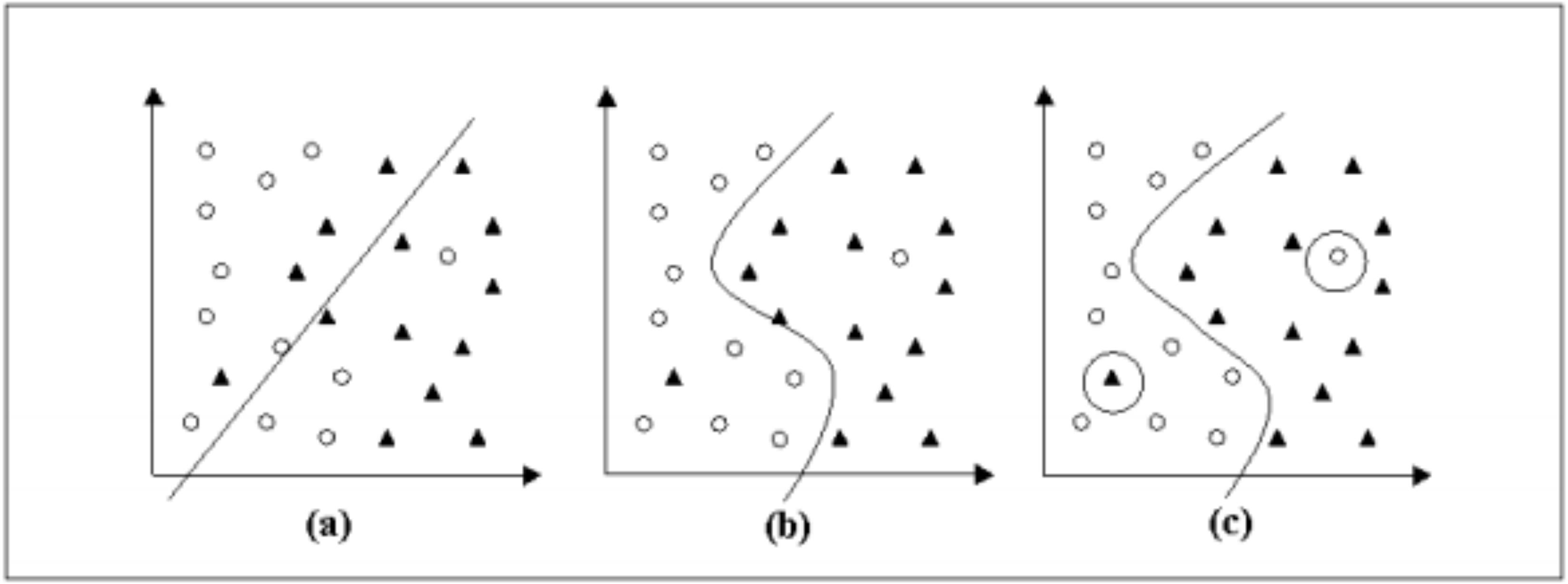
Teoria do Aprendizado Estatístico

- Definições:
  - *f*: é um classificador: dados → espaço de classes
  - F: todos os classificadores que podem ser gerados por um algoritmo A
  - S: conjunto de treinamento (x_i, y_i), N pares
    - usado para gerar um classificador $f' \in F$
    - obter o melhor classificador f^*

Teoria do Aprendizado Estatístico



Teoria do Aprendizado Estatístico



Exemplo de conjunto de dados de treinamento pertencentes a duas classes classificado segundo três diferentes hipóteses.

Teoria do Aprendizado Estatístico

- TAE :
  - condições matemáticas que permitam a escolha de um classificador ***f**** com bom desempenho para os conjuntos de treinamento e teste;
  - ***f**** é capaz de classificar os dados de treinamento da forma mais correta possível
  - ***f**** produz o menor erro durante o treinamento

SVMs

- Máquinas de Vetores de Suporte
  - originalmente para problemas binários {-1, +1}
  - adaptáveis para problemas mutilasse

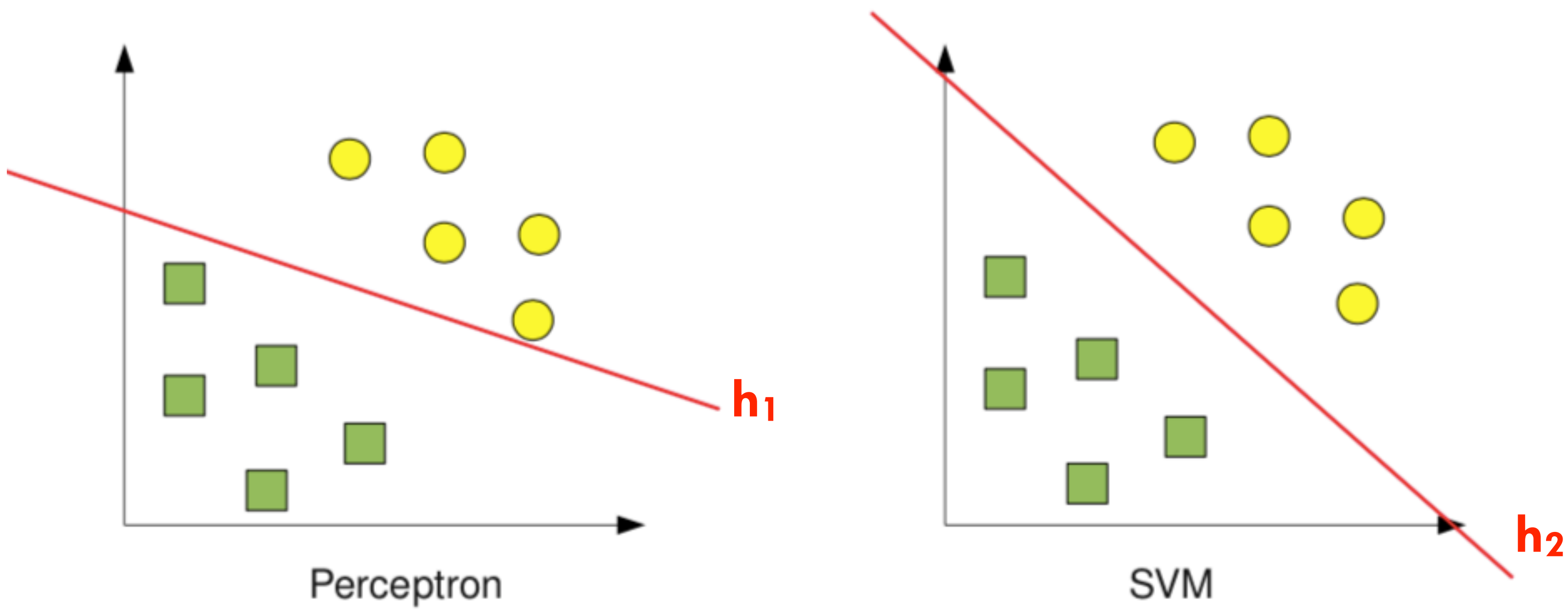
  - SVMs lineares → problemas linearmente separáveis
  - SVMs não lineares → problemas mais complexos

Roteiro

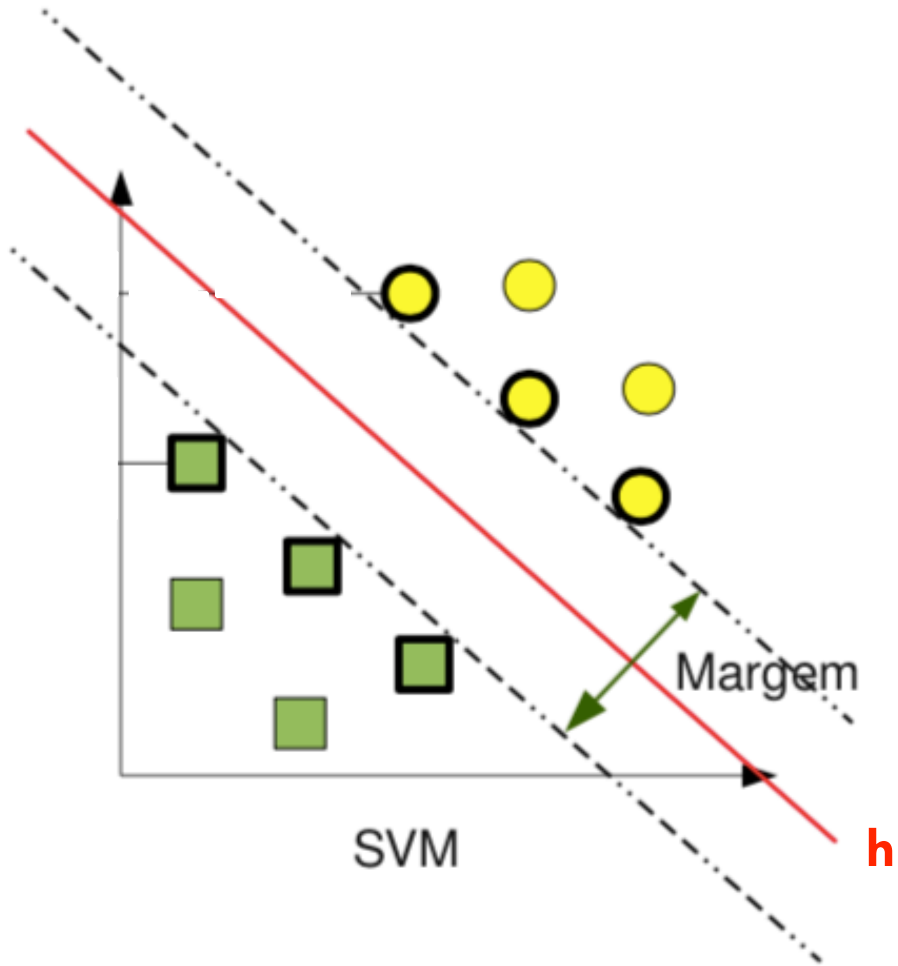
1. Introdução
2. **Máquinas de Vetores Suporte (SVMs)**
3. SVMs com margens rígidas
4. Exemplo
5. Síntese / Próximas Aulas
6. Referências

SVMs lineares

- Definem fronteiras lineares (hiperplanos) para separar dados
  - buscam maximizar a margem de separação entre classes



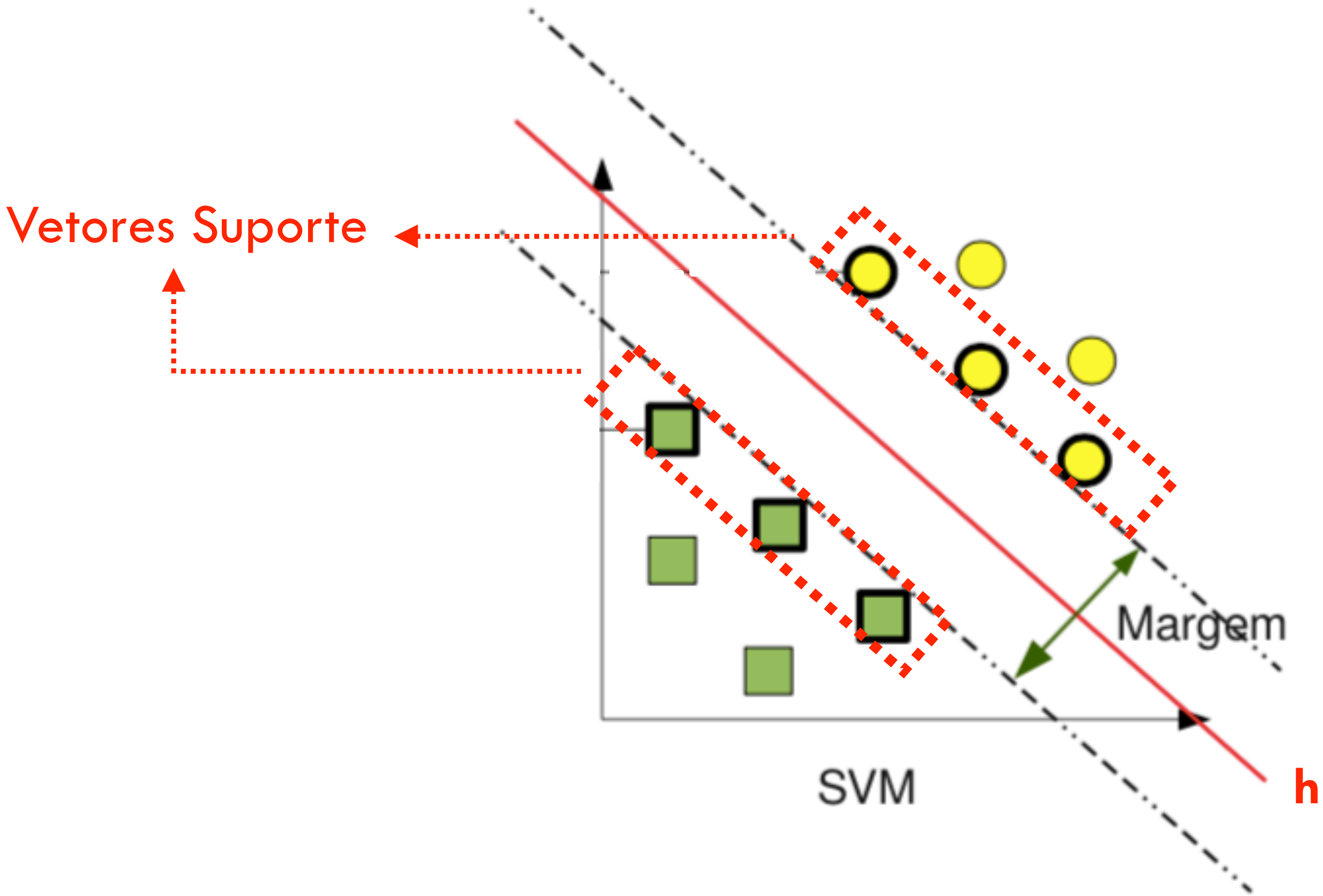
SVMs lineares



SVMs lineares

- **Margem**
  - menor distância entre os exemplos do conjunto de treinamento e o hiperplano separador
- **Vetores suporte**
  - pontos/exemplos (do dataset original) que ajudam a definir o hiperplano separador de margem mínima

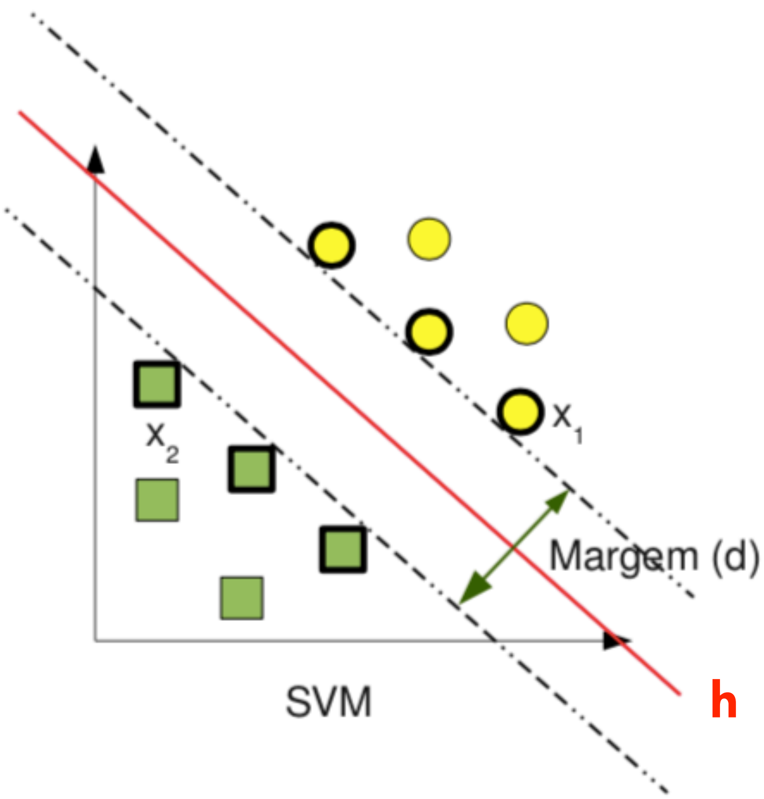
SVMs lineares



Roteiro

1. Introdução
2. Máquinas de Vetores Suporte (SVMs)
3. **SVMs com margens rígidas**
4. Exemplo
5. Síntese / Próximas Aulas
6. Referências

SVMs com margens rígidas



Como buscamos pela máxima margem ou distância projetada entre x_1 e x_2, buscamos maximizar:

$$d = \mathbf{x_1} - \mathbf{x_2} = \frac{\mathbf{2}}{||\ \mathbf{w}\ ||}$$

Sendo a ***distância mínima*** entre o hiperplano separador e os dados de treinamento dada por:

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

Logo podemos maximizar esse termo acima ou minimizar o termo abaixo:

$$||\ \mathbf{w}\ ||$$

SVMs com margens rígidas

- Assim, propõe-se o problema de otimização (***primal***):

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

$$\text{Com as restrições:} \quad \mathbf{w} \cdot \mathbf{x_i} + \mathbf{b} \geq +\mathbf{1} \quad \forall \mathbf{i}$$
$$\text{e} \quad \mathbf{w} \cdot \mathbf{x_j} + \mathbf{b} \leq -\mathbf{1} \quad \forall \mathbf{j}$$

- As restrições são propostas para que não haja dado de treinamento entre as margens de separação das classes
  - Devido a isso dá-se o nome de SVM Linear com Margens Rígidas

- O problema a ser minimizado é quadrático:
  - Função objetivo é convexa, logo há somente um mínimo, que é o global

SVMs com margens rígidas

- Assim, propõe-se o problema de otimização (***primal***):

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

$$\text{Com as restrições: } \mathbf{w} \cdot \mathbf{x_i} + \mathbf{b} \geq +\mathbf{1} \quad \forall \mathbf{i}$$
$$\text{e } \mathbf{w} \cdot \mathbf{x_j} + \mathbf{b} \leq -\mathbf{1} \quad \forall \mathbf{j}$$

- As restrições são propostas para que não haja dado de treinamento entre as margens de separação das classes
  - Devido a isso dá-se o nome de SVM Linear com Margens Rígidas

- O problema a ser minimizado é quadrático:
  - Função objetivo é convexa, logo há somente um mínimo, que é o global

SVMs com margens rígidas

- Uma vez formulado o problema:

$$\textbf{Maximizar: } \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i\alpha_j y_i y_j \mathbf{x_i}\cdot\mathbf{x_j}$$

$$\textbf{Sujeito a: } \begin{cases} \alpha_i \geq 0,\ i = 1,\ \ldots,\ n \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \end{cases}$$

- Pode-se resolvê-lo conforme visto anteriormente
  - Calculando a derivada parcial de L na direção dos alphas
  - Substituindo os alfas e aplicando a função objetivo original, i.e., aquela que considera a minimização de $||w||^2$

SVMs com margens rígidas

Algoritmo 3.1 Determinação do hiperplano ótimo para conjuntos linearmente separáveis (Vert, 2001).

1: Para cada conjunto de treinamento linearmente separável $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$

2: Seja $\alpha^* = (\alpha_1^*, \ldots, \alpha_n^*)$ a solução do seguinte problema de otimização com restrições:

3: Maximizar: $\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j \mathbf{x_i} \cdot \mathbf{x_j}$

4: Sob as restrições: $\begin{cases} \sum_{i=1}^{n} y_i \alpha_i = 0 \\ \alpha_i \geq 0,\ i = 1, \ \ldots, n \end{cases}$

5: O par $(\mathbf{w}^*, b^*)$ apresentado a seguir define o hiperplano ótimo.

6: $\mathbf{w}^* = \sum_{i=1}^{n} \alpha_i{}^* y_i \mathbf{x_i}$

7: $b^* = -{}^1\!/_2 \left[\max_{\{i|y_i=-1\}} (\mathbf{w}^* \cdot \mathbf{x_i}) + \min_{\{i|y_i=+1\}} (\mathbf{w}^* \cdot \mathbf{x_i}) \right]$

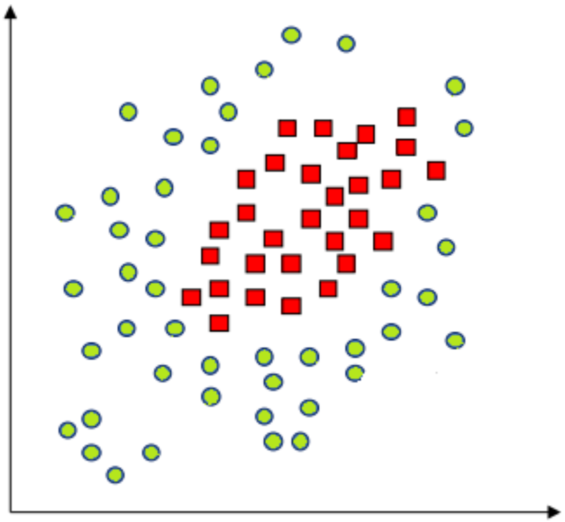
Roteiro

SVMs não-lineares

- Uso de SVMs lineares:
  - conjuntos de dados linearmente separáveis
  - ou possuem uma distribuição aproximadamente linear

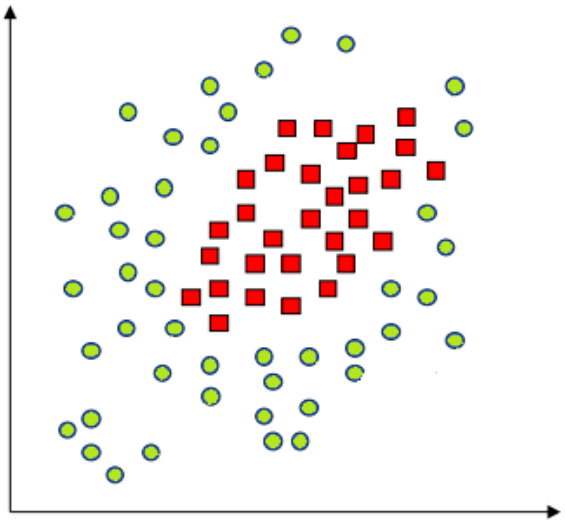
- Porém há casos em que não é possível dividir satisfatoriamente os dados por um hiperplano

SVMs não-lineares

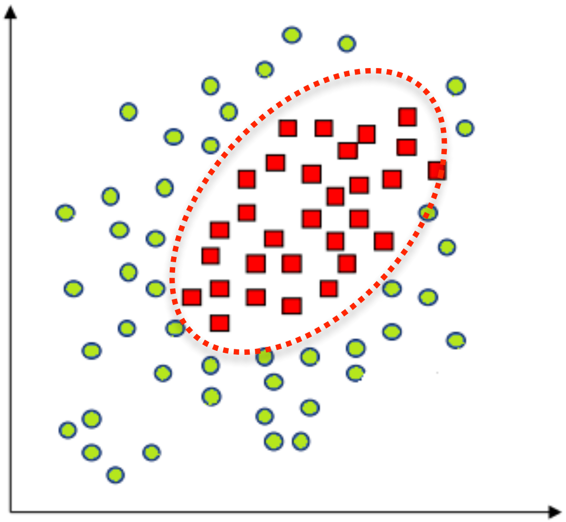


Espaço original

SVMs não-lineares

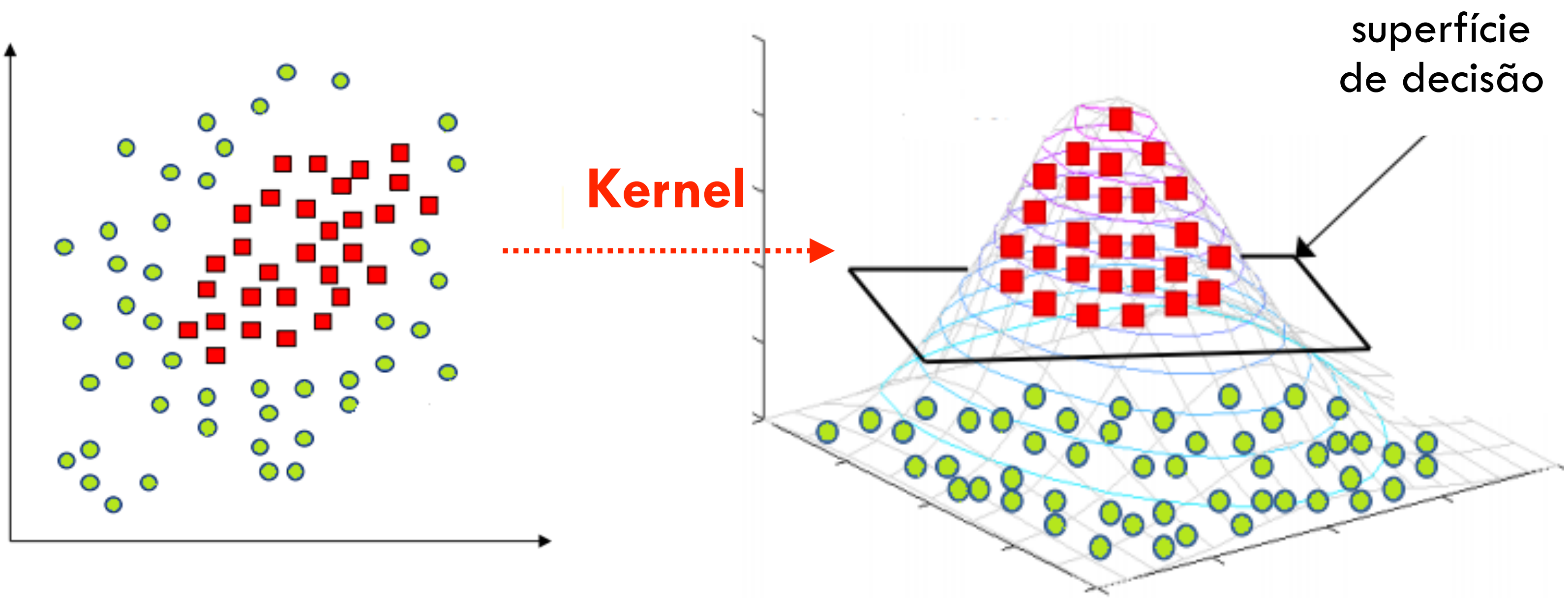


Espaço original



Superfície de decisão
não linear

SVMs não-lineares



Espaço original

Espaço mapeado
(kernel)

SVMs não-lineares

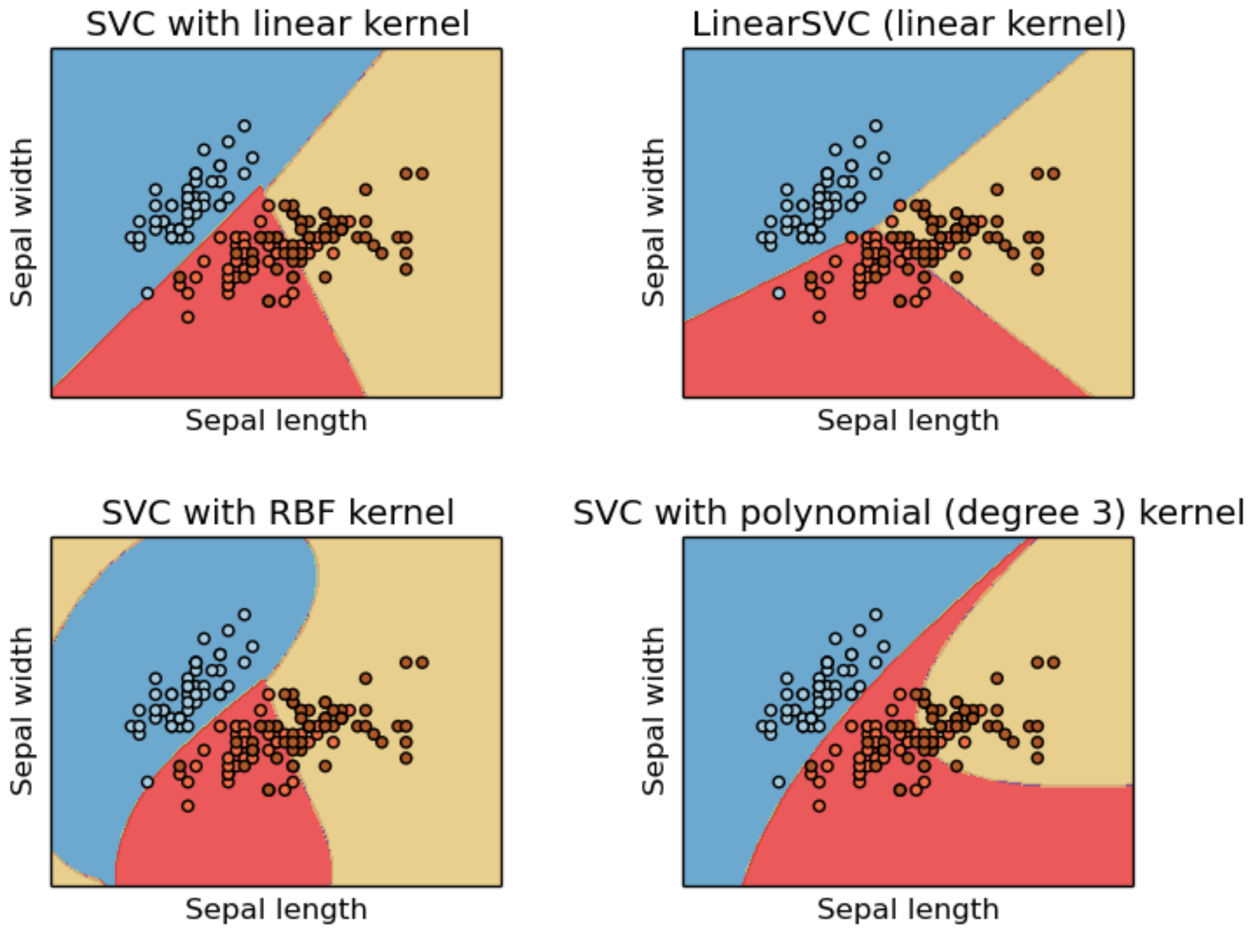
- Kernéis:
  - realizam transformações de espaço
  - comum considerar um kernel sem mesmo saber exatamente o mapeamento que esse faz para os espaço de características
  - **Tipos**: Polinomial, Gaussiano, Sigmoidal, Linear

Kernels

Tipo de Kernel	Função $K(\mathbf{x_i}, \mathbf{x_j})$ correspondente	Comentários
Polinomial	$(\mathbf{x_i}^T \cdot \mathbf{x_j} + 1)^p$	A potência p deve ser especificada pelo usuário
Gaussiano	$\exp\left(-\frac{1}{2\sigma^2} \lVert\mathbf{x_i} - \mathbf{x_j}\rVert^2\right)$	A amplitude σ^2 é especificada pelo usuário
Sigmoidal	$\tanh(\beta_0 \mathbf{x_i} \cdot \mathbf{x_j} + \beta_1)$	Utilizado somente para alguns valores de β_0 e β_1

Tabela 5.1: Sumário dos principais Kernels utilizados nas SVMs (Haykin, 1999).

Kernels


SVC with linear kernel
LinearSVC (linear kernel)
SVC with RBF kernel
SVC with polynomial (degree 3) kernel
Sepal width
Sepal length

SVMs multiclasse

- Estratégias:
  - 1-contra-todos
  - todos-contra-todos

Exemplo1 [R]

- Executando SVMs em um dataset artificial

```r
# seeing our data
plot(my.data[,-3], col=(ys+3)/2, pch=19, xlim=c(-1,6), ylim=c(-1,6))

# training a svm
svm.model = e1071::svm(type ~ ., data = my.data, type='C-classification',
  kernel='linear', scale = FALSE)
svm.model
```

Exercício 01 (em sala)

- Adaptar o script "testeSVM.R" para plotar o hiperplano separador executando no iris
  - transformar o problema original em um problema de 2 classes

- 30 min

Exemplo2 [R]

- Executando SVMs no dataset *iris (mlr)*

```r
library("mlr")

# iris (Species)
task  = mlr::makeClassifTask(id = "iris", data = iris, target = "Species")
lrn   = mlr::makeLearner(cl = "classif.svm", predict.type = "prob")
rdesc = mlr::makeResampleDesc(method = "CV", iters = 10, stratify = TRUE)
meas  = list(ber, kappa, logloss, multiclass.aunu, timetrain, timepredict, timeboth)

res   =  mlr::resample(learner = lrn, task = task, resampling = rdesc, measures = meas,
  models = FALSE, show.info = TRUE)
# print(res)
```

Exercício 02 (em sala)

- Adaptar o script “testeSvmMlr” para comparar SVMs e MLPs executando no dataset "vehicle" muticlasse
  - testar diferentes kernels
- 45 min

Roteiro

Síntese

- **SVMs**
  - teoria do aprendizado estatístico
  - SVMs lineares
    - aprendizado redução de margem
  - SVMs não-lineares
    - Kernéis
  - Exemplo [R]

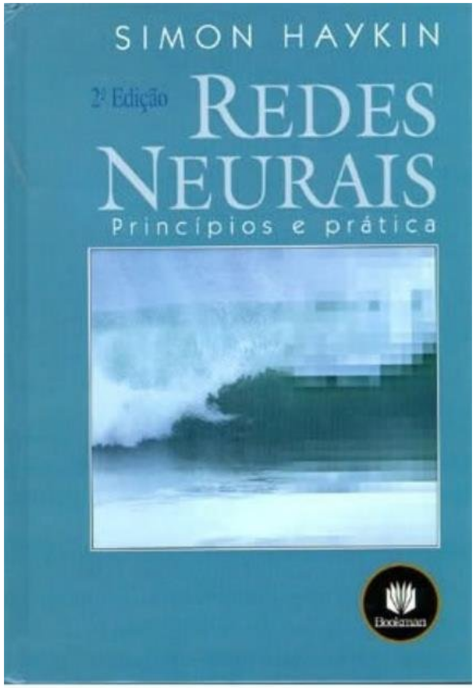
Próxima Aula

- Paradigma evolucionista
  - Algoritmos Evolutivos
    - Programação Evolutiva
    - Estratégia Evolutiva
    - Algoritmos Genéticos

Roteiro

1. Introdução
2. Máquinas de Vetores Suporte (SVMs)
3. SVMs com margens rígidas
4. SVMs não-lineares
5. Síntese / Próximas Aulas
6. **Referências**

Literatura Sugerida



(Haykin, 1999)

[Faceli et al, 2011]

Perguntas?

Prof. Rafael G. Mantovani

rgmantovani@uel.br