Problemas Clássicos da Computação

SVM

Felipe Augusto Lima Reis

felipe.reis@ifmg.edu.br

Sumário

Introdução

Introdução

- Máquina de Vetores de Suporte (*Support Vector Machine* - SVM) correspondem a um conjuntos de algoritmos de aprendizado de máquinas para reconhecimento de padrões
  - Os algoritmos são muito populares na área de Aprendizado de Máquinas;
  - Fornecem alto desempenho para classificação em conjuntos de dados de tamanhos razoáveis [Marsland, 2014];
- O primeiro algoritmo foi proposto por Vladimir Vapnik em 1992, com revisões e melhorias em anos subsequentes [Burges, 1998].

Introdução

- SVMs são baseados em 2 ideias principais:
  1. Transformação do espaço de dados original e um novo espaço de alta dimensionalidade;
  2. Identificação de uma fronteira de decisão linear nesse novo espaço, de modo que a distância (ou margem) entre elementos das classes seja a maior possível [do Patrocínio Jr., 2018].

Introdução

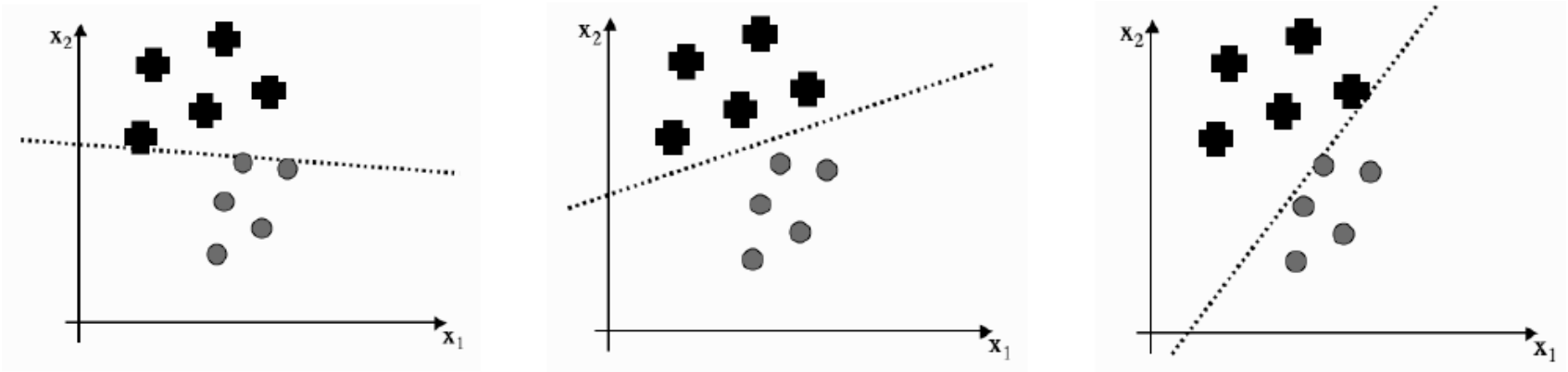
- O classificador SVM é baseado na teoria de Minimização de Risco Estrutural (*Structural Risk Minimization* - SRM)[1]
  - A teoria afirma que maximização de margens do classificador sobre os dados de treinamento deve conduzir a um menor erro de generalização [do Patrocínio Jr., 2018];
  - Para isso, o SVM deve ser treinado em um conjunto de dados, para depois ser aplicado a um conjunto desconhecido (real).

[1] Não confundir com *Empirical Risk Minimization* (ERM).

Conceitos

Intuição

- A partir de um conjunto de dados contendo duas classes linearmente separáveis, pode ser possível traçar um conjunto retas que separam as classes;
  - No entanto, qual a reta mais adequada à separação dos conjuntos?



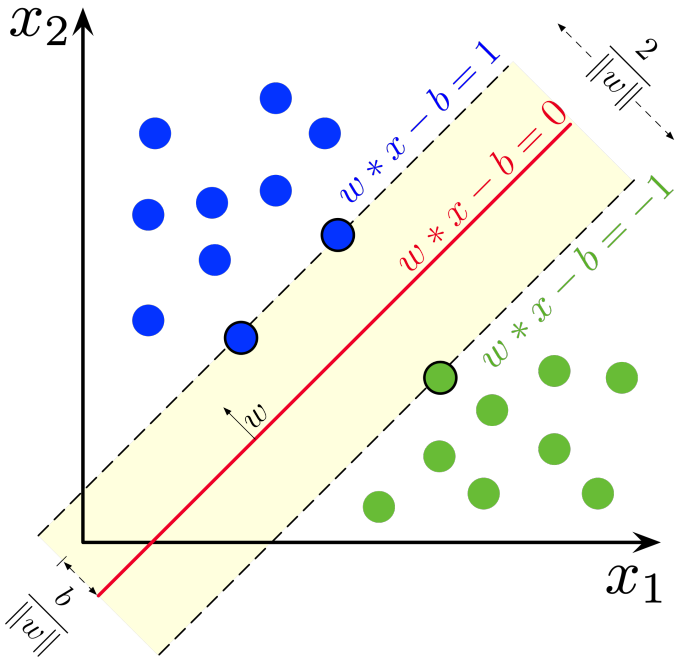
Fonte: [Marsland, 2014]

Intuição

- Dentre as figuras anteriores, intuitivamente, a reta mais adequada é aquela na figura central *(b)*;
  - Podemos estabelecer como critério, que a melhor reta é aquela que esteja no centro das duas classes;
- De forma intuitiva, seguindo o critério adotado, buscamos maximizar a distância entre as classes
  - Ao maximizar a distância em um conjunto de treinamento, acredita-se que menor será o erro do algoritmo na classificação em um conjunto de dados real.

Margem de Separação

- A distância entre a reta[2] e o primeiro ponto de cada classe é chamado de margem [Marsland, 2014];



Fonte: [Wikipedia contributors, 2021]

[2] Em espaços multidimensionais, a separação é feita por um hiperplano.

Margem de Separação

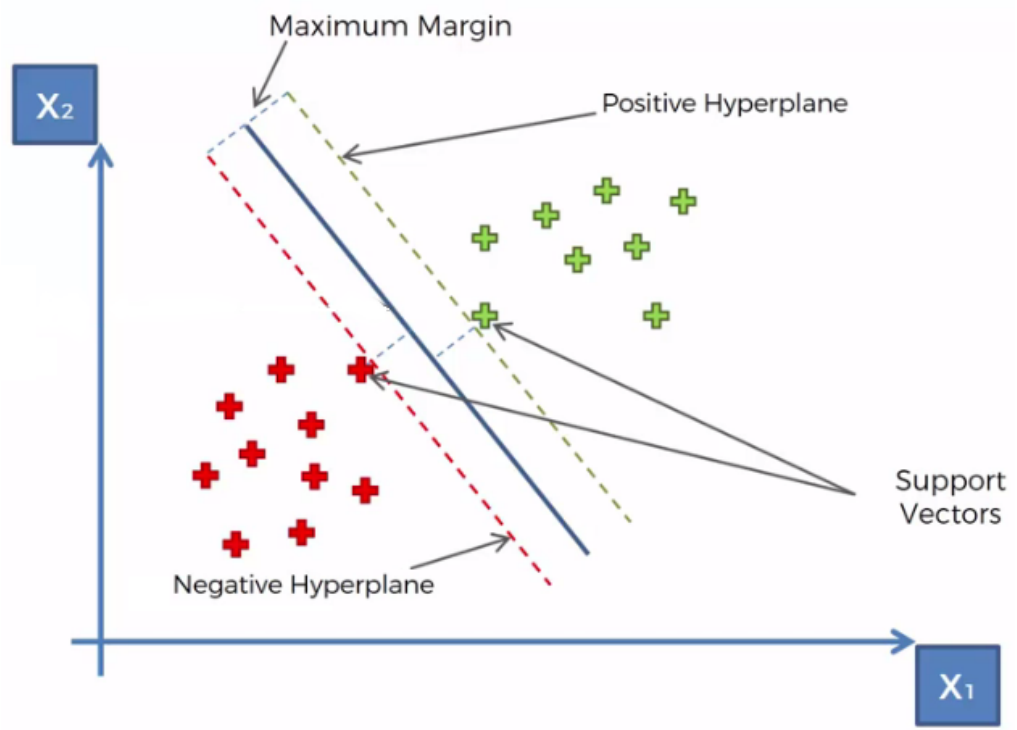
- O hiperplano que divide o conjunto é denominado fronteira de classificação;
- A margem para um ponto x corresponde à distância orientada[3] de x até a fronteira de classificação
  - Se x está corretamente classificado, a margem é positiva; caso contrário, negativa [do Patrocínio Jr., 2018];
- O classificador que é capaz de fornecer uma distância associada à fronteira de decisão é chamado de classificador de margem máxima[4] [Marsland, 2014].

[3] Distância com sinal positivo ou negativo.

[4] Tradução literal de *Maximum Margin Classifier*.

Vetores de Suporte

- Os pontos de dados de cada classe que estão mais próximos da fronteira de classificação recebem um nome especial: vetores de suporte.



Fonte: [Manglick, 2017]

Vetores de Suporte

- Supondo que o melhor classificador passa pelo meio da margem, podemos estabelecer as seguintes considerações:
  1. A margem deve ser a maior possível;
  2. Vetores de suporte são os pontos de dados mais úteis, pois influenciam no tamanho da margem e na posição da fronteira de decisão [Marsland, 2014].

SVM

Definições Matemáticas

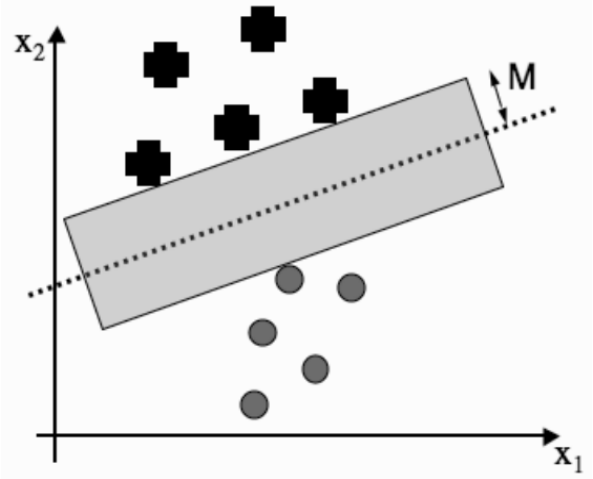
- Supondo um vetor de pontos x e um vetor de pesos w, o produto escalar é dado por:

$$w \cdot x = \sum_i w_i \cdot x_i$$

- A fim de evitar a multiplicação escalar, uma matriz de pesos degenerada w^T, é frequentemente utilizada;
  - Nesse cenário utilizam-se regras de multiplicação convencional de matrizes;
  - A mesma equação é frequentemente escrita como $\boldsymbol{w^T x}$.

SVM

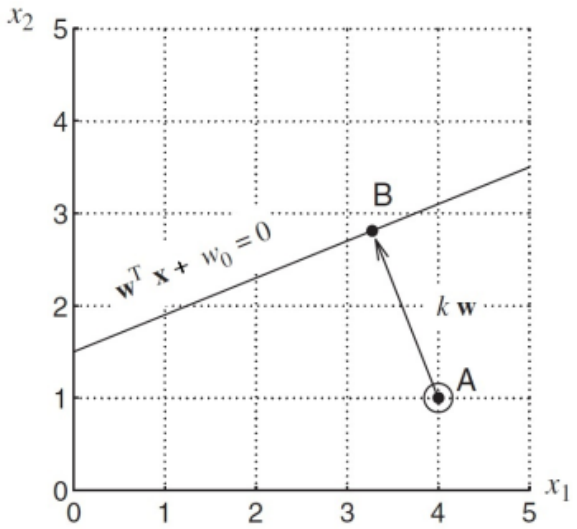
- Considerando a figura abaixo, um dado valor de margem M e um peso *bias* w_0, temos as seguinte possibilidades:
  - $w^T + w_0 > M$: corresponde a um elemento da classe "+";
  - $w^T + w_0 < -M$: corresponde a um elemento da classe "o";
  - $w^T + w_0 = 0$: corresponde a um ponto no hiperplano de separação de classes.



Fonte: [Marsland, 2014]

SVM

- Podemos supor um ponto de interesse $A \in \mathbb{R}^n$;
- Podemos também supor um ponto B no hiperplano, tal que $\vec{AB}$ seja ortogonal ao hiperplano;



Fonte: [do Patrocínio Jr., 2018]

SVM

- Após uma série de cálculos, teremos que a distância de A ao hiperplano será a magnitude do vetor $\vec{AB}$:

$$\vec{AB} = k||\boldsymbol{w}|| = \frac{\boldsymbol{w}^T \vec{A} + w_0}{||\boldsymbol{w}||}$$

- No entanto, esse hiperplano do qual calculamos as distâncias, pode não ser o melhor hiperplano possível
  - O algoritmo deve avaliar outros hiperplanos e escolher aquele que maximiza a margem (ou minimiza o vetor de pesos $\boldsymbol{w}^T$);
  - Com isso, o algoritmo pode ser associado a um problema de Otimização;
  - Constantes e restrições podem ser adicionadas para penalizar pontos que estejam longe do objetivo [Marsland, 2014].

SVM

- Devido às características do problema, o modelo produz um Problema de Programação Não Linear[5]
  - Problemas de programação não linear devem respeitar as condições de Karush-Kuhn-Tucker (KKT).
- O problema a ser resolvido possui a seguinte forma[6]:

$$\text{minimizar } \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} \quad \text{sujeito a} \quad t_i(\boldsymbol{w}^T\boldsymbol{x_i} + b) \geq 1 \quad \text{para todo} \quad i \geq 1, ..., n$$

[5] Esse tipo de programa não será abordado nesse curso.

[6] Cálculos algébricos necessários à definição da forma não serão apresentados nesta disciplina.

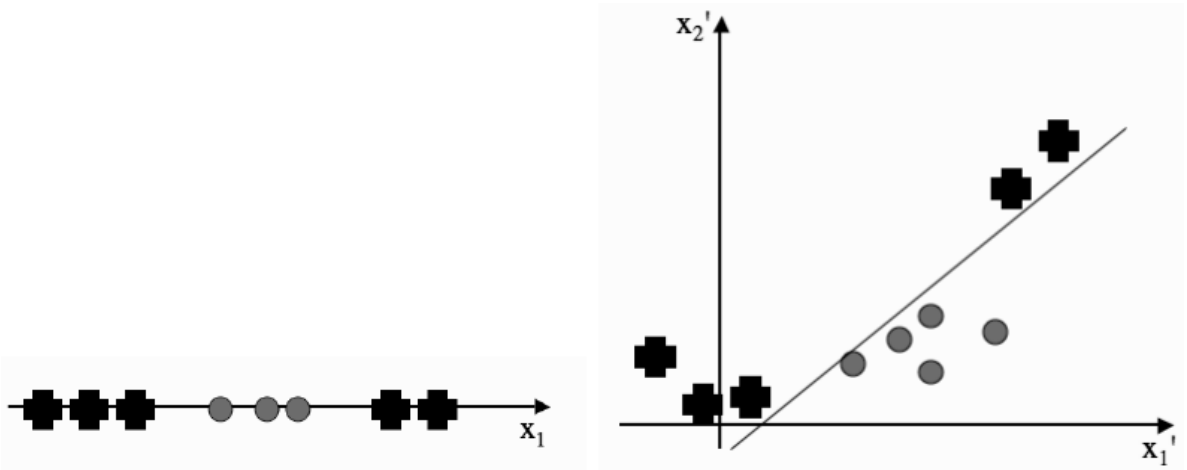
Kernels

Problemas Não Linearmente Separáveis

- Até o momento, todos os problemas estudados eram linearmente separáveis
  - No entanto, uma grande parte dos conjuntos não apresenta essa característica;
- Para conjuntos de dados não linearmente separáveis pode-se adicionar dimensões extras
  - Para isso, devem-se utilizar funções $\phi(x)$ nos dados de entrada;
  - Com o uso de funções, novos dados não são criados - apenas os dados anteriores são transformados [Marsland, 2014].

Problemas Não Linearmente Separáveis

- Para escolha mais assertiva das funções a serem utilizadas, é necessário conhecimento prévio do domínio do problema
  - Isso pode ser feito a partir da identificação de características mais importantes e/ou visualização de dados[7];
- A transformação permite a separação linear de conjuntos até então não linearmente separáveis.



Fonte: [Marsland, 2014]

[7] Ver aula de Redução de Dimensionalidade.

Kernels

- Uma solução para evitar o cálculo de produtos internos em espaços de altas dimensões é utilizar funções especiais, chamadas de funções de kernel
  - Essas funções permitem operar em um espaço de recurso implícito de alta dimensão, sem computar as coordenadas de dados nesse espaço;
  - As operações costumam ser mais baratas que o cálculo das coordenadas;
  - De forma mais informal, pode-se dizer que não é necessário transformar explicitamente os dados do espaço original para o novo espaço;
  - Essa solução é denominada kernel trick [Marsland, 2014].

Kernels

- Para um dado conjunto de dados $Z = \{\boldsymbol{z}_1, ..., \boldsymbol{z}_n\}$, o processo de otimização do SVM busca encontrar coeficientes α_j e w_0 para uma função $f(x)$, onde $K(\cdot, \cdot)$ corresponde ao kernel:

$$f(\boldsymbol{x}) = \sum_{j=1}^{N} \alpha_j K(\boldsymbol{x}, \boldsymbol{z}_j) + w_0$$

- Após o treinamento do SVM, todos os parâmetros α_j são zerados, exceto aqueles associados aos vetores de suporte [do Patrocínio Jr., 2018].

Kernels

- As funções de kernel mais adequadas ao SVM devem seguir o Teorema de Mercer [Mercer and Forsyth, 1909]
  - O teorema corresponde à representação de uma função simétrica positiva-definida[8] em um quadrado como a soma de uma sequência convergente de funções produto;
  - O teorema ainda indica que é possível convolver[9] kernels juntos e o resultado será outro kernel [Marsland, 2014];
  - Essas funções reforçam a positividade na integral de funções arbitrárias [Marsland, 2014];
- De forma geral, qualquer função simétrica positiva-definida pode ser usada como kernel.

[8] Funções com forma quadrática definida (*positive-definite function* ou *definite quadratic form*).

[9] Ver conceito de convoluções, na aula de Redes Neurais Convolucionais.

Kernels

- Frequentemente são utilizadas as seguintes funções de kernel: [Marsland, 2014] [do Patrocínio Jr., 2018]
  - Família de funções polinomiais com algum grau s

$$\boxed{K(x, y) = (1 - x^T y)^s}$$

  - Funções sigmoides[10] com parâmetros κ_1 e κ_2

$$\boxed{K(x, y) = tanh(\kappa_1 x^T y - \kappa_2)}$$

  - Família de funções de base radial (RBF)

$$\boxed{K(x, y) = exp\left(-\frac{(x - y)^2}{2\sigma^2}\right)}$$

[10] Conjunto de funções em formato de "s", não necessariamente a função logística.

Algoritmo

Algoritmo

INSTITUTO FEDERAL
Minas Gerais

- O algoritmo do SVM pode ser sumarizado em:

The Support Vector Machine Algorithm

- **Initialisation**
  - for the specified kernel, and kernel parameters, compute the kernel of distances between the datapoints
    - the main work here is the computation $\mathbf{K} = \mathbf{X}\mathbf{X}^T$
    - for the linear kernel, return $\mathbf{K}$, for the polynomial of degree d return $\frac{1}{\sigma}\mathbf{K}^d$
    - for the RBF kernel, compute $\mathbf{K} = \exp(-(\mathbf{x} - \mathbf{x}')^2/2\sigma^2)$
- **Training**
  - assemble the constraint set as matrices to solve:

$$\min_{\mathbf{x}} \frac{1}{2}\mathbf{x}^T t_i t_j \mathbf{K}\mathbf{x} + \mathbf{q}^T\mathbf{x} \text{ subject to } \mathbf{G}\mathbf{x} \leq \mathbf{h}, \mathbf{A}\mathbf{x} = b$$

  - pass these matrices to the solver
  - identify the support vectors as those that are within some specified distance of the closest point and dispose of the rest of the training data
  - compute b^* using equation (8.10)
- **Classification**
  - for the given test data $\mathbf{z}$, use the support vectors to classify the data for the relevant kernel using:
    - compute the inner product of the test data and the support vectors
    - perform the classification as $\sum_{i=1}^{n} \lambda_i t_i \mathbf{K}(\mathbf{x}_i, \mathbf{z}) + b^*$, returning either the label (hard classification) or the value (soft classification)

Fonte: [Marsland, 2014]

Referências I

Burges, C. J. C. (1998).
A tutorial on support vector machines for pattern recognition.
Data Mining and Knowledge Discovery, 2:121–167.

do Patrocínio Jr., Z. K. G. (2018).
Aprendizado de máquina e reconhecimento de padrões - análise de discriminantes lineares.
Slides de Aula.

Kopec, D. (2019).
Classic Computer Science Problems in Python.
Manning Publications Co, 1 edition.

Manglick, A. (2017).
Support vector machine (svm).
[Online]; acessado em 24 de Fevereiro de 2021. Disponível em:
http://arun-aiml.blogspot.com/2017/07/support-vector-machine-svm.html.

Marsland, S. (2014).
Machine Learning: An Algorithm Perspective.
CRC Press, 2 edition.
Disponível em: https://homepages.ecs.vuw.ac.nz/ marslast/MLbook.html.

Mercer, J. and Forsyth, A. R. (1909).
Xvi. functions of positive and negative type, and their connection the theory of integral equations.
Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character, 209(441-458):415–446.

Referências II

Richert, W. and Coelho, L. P. (2013).
Building Machine Learning Systems with Python.
Packt Publishing Ltd., 1 edition.

Shalev-Shwartz, S. and Ben-David, S. (2014).
Understanding Machine Learning: From Theory to Algorithms.
Cambridge University Press, 1 edition.
Disponível em: http://www.cs.huji.ac.il/ shais/UnderstandingMachineLearning.

Wikipedia contributors (2021).
Support-vector machine.
[Online]; acessado em 24 de Fevereiro de 2021. Disponível em:
https://en.wikipedia.org/wiki/Support-vector_machine.