

INSTITUTO DE COMPUTAÇÃO  
ENGENHARIA DE COMPUTAÇÃO

SUPPORT VECTOR MACHINES  
CLASSIFICADOR DE DADOS

MACEIÓ  
JANEIRO DE 2021

RUAN HELENO CORREA DA SILVA

SUPPORT VECTOR MACHINES  
CLASSIFICADOR DE DADOS

MACEIÓ  
JANEIRO DE 2021

# Sumário

<b>Sumário</b>	<b>3</b>
<b>Introdução</b>	<b>4</b>
<b>Resumo</b>	<b>5</b>
<b>Support Vector Machines - SVM</b>	<b>6</b>
Vetores de Suporte	6
Hiperplano	6
Execução	7
Cálculo Matemático	8
Prós e Contras	9
Hiperplano para dados não lineares	10
Cálculo matemático do Kernel Trick	10
Vantagens e Desvantagens do uso do método	11
<b>Referências bibliográficas</b>	<b>12</b>

# Introdução

A tecnologia com o passar dos anos vêm evoluindo e se reconstruindo cada vez mais. A Inteligência Artificial vem tomando cada vez mais espaço em todos os meios possíveis quando o assunto é informática, computação e tecnologia. Nessa busca incessante pelo melhoramento do Software, desenvolvedores se somam e se multiplicam para criar algoritmos que otimizem seu trabalho sem restringir tal evolução. Algoritmos essenciais para determinados programas foram desenvolvidos e hoje são determinantes para qualificar e melhorar tais programas.

Andando lado a lado com a Inteligência Artificial, a Ciência de Dados é uma ciência que visa estudar as informações, seu processo de captura, transformação, geração e, posteriormente, análise de dados. Dentre os algoritmos essenciais criados, alguns tiveram o enfoque na contribuição para tais objetivos que a ciência visa. Algoritmos de aprendizado de máquina, supervisionados, de classificação e regressão de dados, muitos foram criados. Entre eles, está o algoritmo de aprendizado de máquina supervisionado chamado Support Vector Machines(SVM).

# Resumo

É um algoritmo supervisionado que tenta criar uma linha(ou uma fronteira) que melhor separa os dados. Sendo um classificador não probabilístico, o Support Vector Machines tem resultados comparáveis aos obtidos por outros algoritmos de aprendizado como Redes Neurais Artificiais (RNAs) .

# Support Vector Machines - SVM

Idealizado em 1979 por Vladimir Vapnik e alguns colegas, Support Vector Machines são métodos de Machine Learning baseados em aprendizado de máquina supervisionado que podem ser usados para desafios de classificação e regressão a partir da análise de dados e reconhecimento de padrões. Seu foco maior é no treinamento e classificação de um dataset.

## Vetores de Suporte

Os vetores de suporte são simplesmente as coordenadas da observação individual. O classificador Support Vector Machine é uma fronteira que melhor segrega as duas classes (hiperplano/linha). Esses vetores de suporte são chamados assim porque são vetores de pontos usados para formar a margem que cria o limite entre as classes. Um ponto importante sobre os vetores é que eles só são afetados pelos pontos próximos à margem. Em outras palavras, qualquer ponto que não comprometa a margem e os vetores de suporte não afetarão a localização dos vetores de suporte ou essa margem.

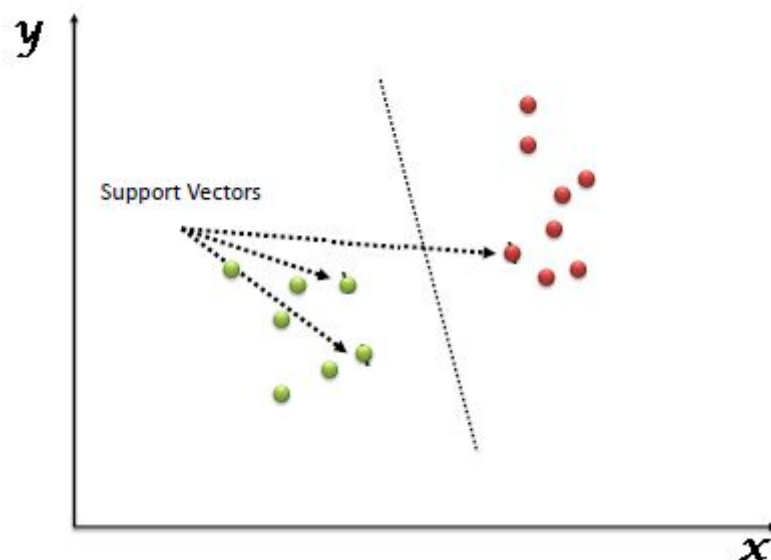


Figura 1: identificando alguns vetores de suporte

## Hiperplano

Hiperplano é a linha/fronteira traçada entre as duas classes no qual é determinado por um subconjunto dos pontos das duas classes.

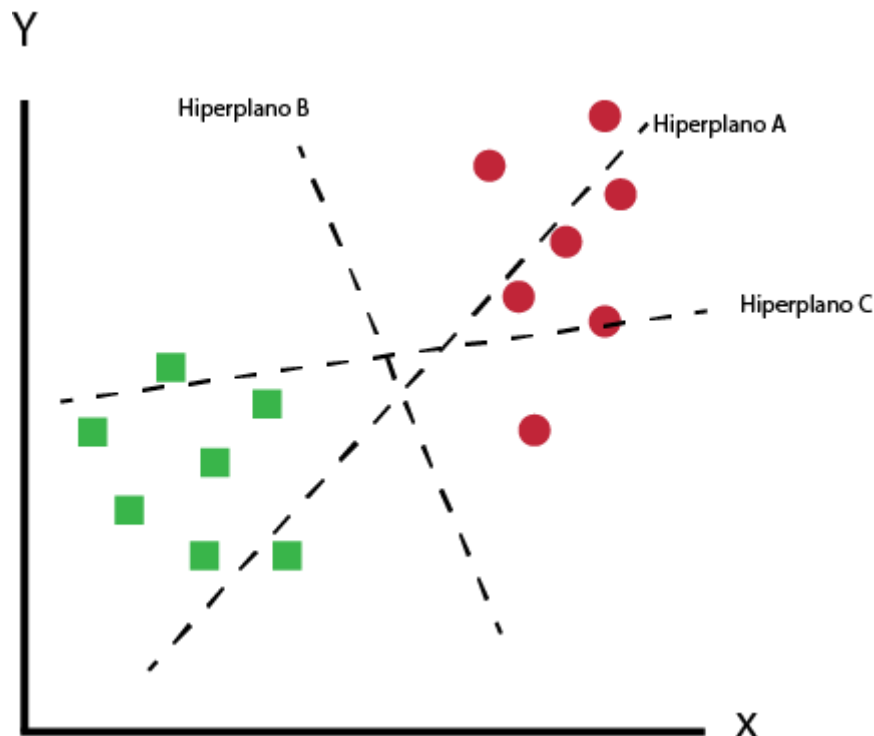


Figura 2: identificando hiperplanos

## Execução

Um dado é visto como um ponto num espaço de  $p$  dimensões e a pergunta é: Podemos separar esses pontos com um hiperplano de  $(p - 1)$  dimensões?

A partir disso, dado um conjunto de dados de treino, cada um marcado com uma classe pertencendo a uma categoria, SVM cria um modelo que assimila novos exemplos em uma das categorias, fazendo uso de um classificador não probabilístico.

Nos métodos presentes, plotamos cada item de dados, como um ponto no espaço  $n$ -dimensional (onde  $n$  é o número de recursos que você tem), com o valor de cada recurso sendo o valor de uma determinada coordenada. Então, nós executamos a classificação encontrando o hiperplano que melhor diferencia as duas classes.

Em alguns conjuntos de dados, temos diversas opções de hiperplanos possíveis. O algoritmo sempre tenta encontrar o hiperplano que vai maximizar a margem entre as classes. Isto é, a fronteira mais distante dos dados de treinamento é a melhor. Logo, um modelo SVM é uma representação dos exemplos como

pontos no espaço, mapeados de forma que as categorias são divididas com a maior distância possível entre elas. Novos exemplos são mapeados de acordo com a zona em que caírem.

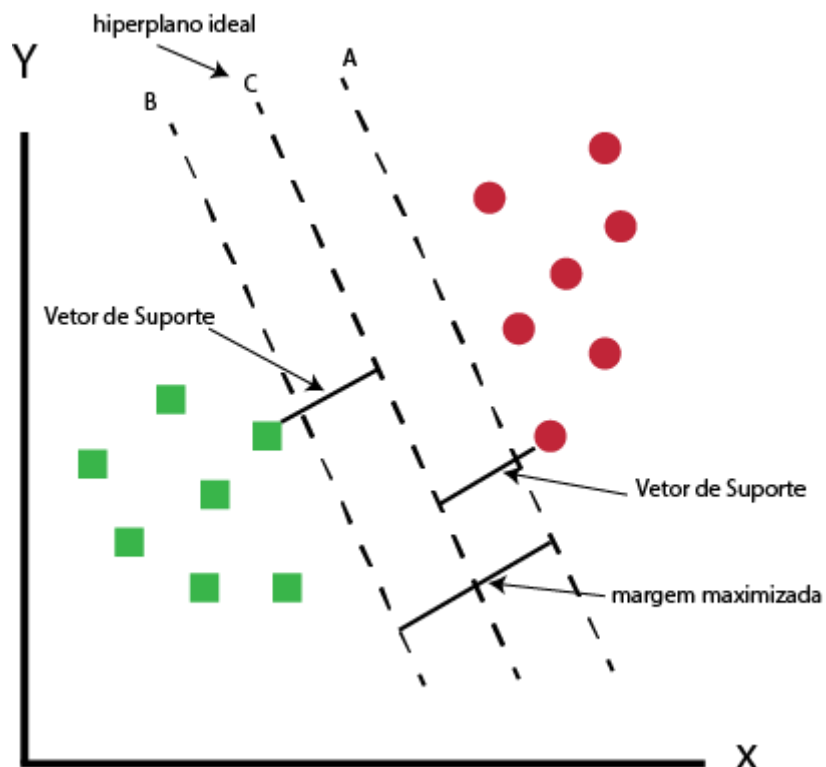


Figura 3: o hiperplano ideal

## Cálculo Matemático

Foi visto até agora com um olhar computacional a forma de como o modelo é executado linearmente, mas matematicamente como o modelo se comporta para traçar os hiperplanos e, posteriormente, escolher o melhor hiperplano?

A equação  $wx + b = 0$ , onde  $w$  é o vetor de pesos (mesma dimensão das amostras) perpendicular ao hiperplano de separação e  $b$  é um escalar, divide o espaço em duas regiões.  $wx + b > 0$  e  $wx + b < 0$ . Com isso apenas o sinal será necessário para fazer a classificação. Sendo então:  $y(x) = \{+1, \text{ se } wx + b > 0; -1, \text{ se } wx + b < 0\}$ .

A partir disso, a escolha do melhor hiperplano depende da seleção de  $w$  e  $b$  de forma que os exemplos mais próximos ao hiperplano satisfaçam  $|wx + b| = 1$ . Logo  $wx + b \geq 1$  se  $y = +1$ ,  $wx + b \leq -1$  se  $y = -1$ .



Sendo dois pontos  $x_1$  e  $x_2$ , onde  $w x_1 + b = +1$  e  $w x_2 + b = -1$ , fazendo a diferença entre as duas equações equivalentes aos hiperplanos traçados temos:  $w(x_1 - x_2) = 2$ . Sendo a margem o vetor diferença projetado na direção de  $w$ , então:  $\text{margem} = \|x_2 - x_1\| = 2 / \|w\|$ . A distância mínima entre o hiperplano separador e os dados é dada por  $w(x_1 - x_2) = 2$ , onde será igual à  $1 / \|w\|$ . Dessa forma, é fácil ver que para maximizar uma margem é necessário minimizar o vetor  $\|w\|$ .

Mas a minimização do  $\|w\|$  não é nada simples. Ela depende da norma de  $w$ , a qual envolve uma raiz. Para solucionar isso é necessário utilizar métodos de substituição, utilização de constantes no qual resultará em problema de otimização de função quadrática com restrições lineares. Quando surgem problemas de otimização, faz-se necessário o uso de uma formulação via multiplicadores de Lagrange. Esses multiplicadores permitem encontrar extremos (máximos e mínimos) de uma função de uma ou mais variáveis suscetíveis a uma ou mais restrições e é uma ferramenta importante em restrições de igualdade. A resolução de tal problema de otimização pode ser dada minimizando a função de Lagrange que engloba as restrições à função objetivo, associadas a parâmetros aos multiplicadores de Lagrange e encontrar seu gradiente.

## Prós e Contras

Prós:

- Funciona muito bem com margem de separação clara.
- É eficaz nos casos em que o número de dimensões é maior que o número de amostras.
- Ele usa um subconjunto de pontos de treinamento na função de decisão (chamamos de vetores de suporte), portanto, também é eficiente em termos de memória.

Contras:

- Não tem um bom desempenho quando temos um grande conjunto de dados porque o tempo de treinamento necessário é grande.
- Ele também não funciona muito bem quando o conjunto de dados tem mais ruído, ou seja, as classes de destino estão sobrepostas.

## Hiperplano para dados não lineares

Há muitos casos em que não é possível dividir satisfatoriamente os dados de treinamento por um hiperplano. O método “kernel trick” foi criado para tal finalidade, onde podemos expandir o conceito de Support Vector Machines para dados não linearmente separáveis. O kernel trick se baseia em aplicar uma transformação nos dados e levá-los para um espaço dimensional superior onde lá ele possa ser separado linearmente e depois, os trazemos de volta aplicando a inversa.

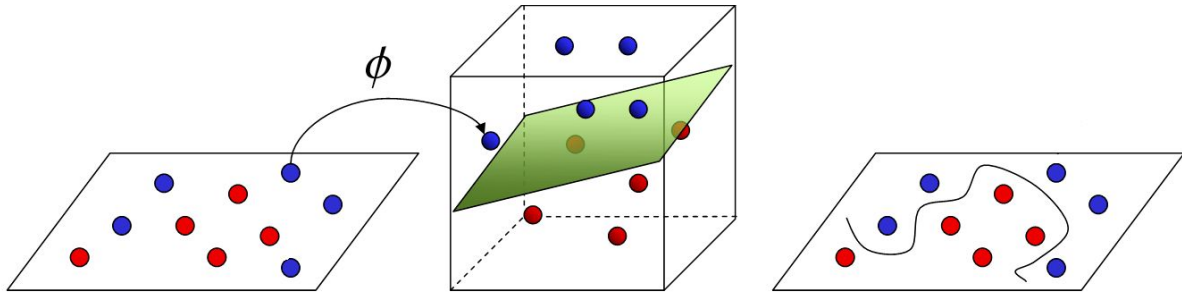


Figura 4: exemplo das transformações feitas

## Cálculo matemático do Kernel Trick

Inicialmente há o mapeamento do conjunto de treinamento de seu espaço original (não linear) para um novo espaço de maior dimensão, denominado espaço de características (*feature space*), que é linear. Para isso, precisa-se encontrar uma transformação não linear, no qual mapeia o espaço original dos padrões para um novo espaço de atributos m-dimensional e nesse novo espaço, os padrões  $x$  passam a ser linearmente separáveis. Com a função de transformação, nosso problema de otimização retoma para uma SVM linear.

Assim como já foi feito na minimização da margem de um SVM linear, faz-se necessário o uso de uma formulação via multiplicadores de Lagrange.

Na formulação via multiplicadores de Lagrange, a solução depende apenas do produto  $\varphi(x_i) \cdot \varphi(x_j)$  para cada par de padrões  $x_i$  e  $x_j$ , e não dos termos individuais. Isso é obtido com o uso de funções denominadas Kernels  $K(x_i, x_j) = \varphi(x_i) \cdot \varphi(x_j)$ . O Kernel realiza transformações de espaço. A sua utilidade está, portanto, na simplicidade de seu cálculo e em sua capacidade de representar espaços abstratos.

O Teorema de Mercer garante que, para algumas classes de Kernels  $K(x_i, x_j)$ , sempre existe uma transformação  $\varphi$ . O teorema não garante nada sobre a dimensão  $m$  do espaço transformado  $\varphi$  (pode até ser infinita!).

Alguns Kernels muito utilizados, são:

- **Polinomial** -  $K(x_i, x_j) = (\delta(x_i \cdot x_j) + k)^d$ ;
- **Gaussianos ou RBF (Radial-Basis Function)** -  $K(x_i, x_j) = \exp(-\sigma \cdot ||x_i - x_j||^2)$ ;
- **Sigmoidal** -  $K(x_i, x_j) = \tanh(\delta(x_i \cdot x_j) + k)$ .

Quando usamos um kernel RBF em uma SVM, temos que o problema recai exatamente em uma rede neural do tipo RBF. Nesse caso, os centros e o número de neurônios da rede são dados automaticamente pelos vetores de suporte.

Maximizar a margem no espaço transformado pelo SVM não-linear não garante a inexistência de *overfitting* no classificador. Sempre existe um número de dimensões suficientemente grande que separa os dados de treinamento. É possível controlar o *overfitting* através da técnica de relaxamento.

## Vantagens e Desvantagens do uso do método

Vantagens:

- Sempre encontram a melhor solução possível para o problema de otimização em questão;
- Um dos mais eficientes classificadores para problemas de elevada dimensionalidade (muitos atributos);
- Sua técnica de relaxamento minimiza o risco de overfitting;
- Podem ser adaptados e/ou estendidos para problemas de regressão.

Desvantagens:

- São classificadores do tipo “caixa-preta”, ou seja, não permitem interpretação da estratégia de decisão como as árvores;
- Voltados apenas para atributos numéricos;
- Possuem complexidade mínima  $O(N^2)$ , usualmente  $O(N^3)$ , onde  $N$  é o número de padrões de treinamento.

# Referências bibliográficas

COELHO, Lucas. Ciência de Dados: O que é, conceito e definição, 2020. Disponível em: <[Ciência de Dados: O que é, conceito e definição | Blog Cetax](#)>. Acesso em 14 de Janeiro de 2021.

SANTANA, Rodrigo. Café com Código #18: O Famoso SVM, 2017. Disponível em: <[Minerando Dados › Café com Código #18: O Famoso SVM](#)>. Acesso em 14 de Janeiro de 2021.

ADDAN, Daniel. Support Vector Machine, 2019. Disponível em: <[IA07.pdf \(ufpr.br\)](#)>. Acesso em 15 de Janeiro de 2021.

J. G. B. CAMPELLO, Ricardo; FERNANDES DE MELLO, Rodrigo. Support Vector Machine - SVM, 2018. Disponível em: <[Reconhecimento de Padrões \(ufu.br\)](#)>. Acesso em 15 de Janeiro de 2021.

LEARN, Scikit. 1.4. Support Vector Machines. Disponível em <[1.4. Support Vector Machines — scikit-learn 0.24.1 documentation \(scikit-learn.org\)](#)>. Acesso em 16 de Janeiro de 2021.

LEARN, Scikit. sklearn.datasets.make\_blobs. Disponível em <[sklearn.datasets.make\\_blobs — scikit-learn 0.24.1 documentation \(scikit-learn.org\)](#)>. Acesso em 17 de Janeiro de 2021.

LEARN, Scikit. sklearn.datasets.load\_breast\_cancer. Disponível em <[sklearn.datasets.load\\_breast\\_cancer — scikit-learn 0.24.1 documentation \(scikit-learn.org\)](#)>. Acesso em 17 de Janeiro de 2021.

MAYERS, Gabriel. Classifying Malignant or Benignant Breast Cancer using SVM, 2020. Disponível em: <[Classifying Malignant or Benignant Breast Cancer using SVM | by Gabriel Mayers | Analytics Vidhya | Medium](#)>. Acesso em 19 de Janeiro de 2021.