

4

Aprendizado de Máquina

Aprendizado de Máquina é um ramo da Inteligência Artificial especializado em criar modelos que têm capacidade de aprender com informações. É muito útil, pois os modelos, após treinados, isto é, a partir do momento em que eles conseguem absorver os dados que lhes são passados, podem fazer previsões precisas de dados aos quais nunca tiveram acesso. Um exemplo é o de filtros de spam. Um modelo para distinguir se um e-mail é um spam ou não é treinado com um banco de dados composto de vários exemplos de e-mails e suas possíveis categorias, nesse caso, ser spam ou não. Uma vez que o modelo foi treinado, ele é capaz de fazer previsões para novos e-mails que não estavam nos dados de treinamento.

Além disso, é muito utilizado atualmente para sistemas de recomendação em sites de venda *online*: considerando um cliente e suas características, qual produto teria mais chance de ser comprado por ele? Ou então, considerando que ele comprou um certo produto, quais outros o mesmo tipo de cliente comprou? Sabendo tais respostas, o sistema pode recomendar certos produtos e possivelmente aumentar suas vendas.

Aprendizado de Máquina é usado também em aplicações para a bolsa de valores, bem como análise e extração de informações de textos em linguagem natural.

Os principais algoritmos de Aprendizado de Máquina são: supervisionado, semi-supervisionado e não supervisionado. A diferença básica entre eles está no banco de dados disponível para treinamento. O supervisionado requer exemplos rotulados, isto é, atributos e sua categoria. Para o não supervisionado, os exemplos não estão rotulados e no semi-supervisionado existem tanto exemplos rotulados quanto não rotulados. No decorrer desta dissertação, ao utilizar a nomenclatura atributos, estaremos nos referenciando a esses atributos anteriormente citados, ou seja, os dados que compõem um vetor que será usado tanto para treinamento do SVM quanto para classificação utilizando um modelo já treinado.

O aprendizado pode ser caracterizado como *online* ou *offline*. Quando

offline, o modelo não terá capacidade de agregar conhecimento com o passar do tempo, estando limitado ao que aprendeu durante o treinamento inicial. Já quando *online*, o modelo agrega um exemplo a cada vez. Esse tipo de aprendizado permite que um modelo previamente treinado seja incrementado com a informação de um novo exemplo, sem que para isso seja preciso treiná-lo por completo novamente, qualificando-o como um modelo incremental.

4.1

Support Vector Machines

Support Vector Machines (SVM) é, tradicionalmente, uma técnica de aprendizado supervisionado que estuda e analisa dados com o objetivo de reconhecimento de padrões. Inicialmente proposto por Cortes e Vapnik (1995) (11), este método de aprendizado pode ser facilmente encontrado em literatura mais recente (37) (3). O SVM é treinado a partir de um banco de dados composto de exemplos que consistem num conjunto de atributos e a sua classificação binária, 0 ou 1. Dado um novo conjunto de atributos cuja classificação é desconhecida, o SVM é capaz de realizar uma predição, ou seja, classificar esse conjunto de atributos em 0 ou 1, de acordo com o que aprendeu durante o treinamento.

Em um exemplo, para dados de duas dimensões, o objetivo do SVM é receber um banco de dados de exemplos, que contém os valores dos atributos (x, y) de cada exemplo e também a sua classificação (0 e 1) e encontrar uma reta que traga a maior margem de separação possível entre as duas classes. A Figura 4.1 representa esse exemplo, na qual os círculos azuis representam a classe 0 e os vermelhos representam a classe 1. A reta encontrada separa linearmente os dois grupos a fim de que a margem de separação seja a maior possível. Existem várias maneiras de separar os dois grupos, mas quanto maior a margem, mais confiança o separador possui.

Como o estudo proposto lida com várias emoções, é necessário usar uma extensão do SVM, o SVM multiclasse. Tal extensão torna-o capaz de fazer predições para várias classes, removendo sua limitação de apenas duas classes. Isso pode ser feito de algumas maneiras, sendo a mais simples quebrar a predição em uma cadeia de várias classificações binárias (métodos *one-versus-one* e *one-versus-all*).

Especificamente, o *one-versus-one* cria modelos de comparação entre cada par de classes e realiza a predição para todos os pares. Para cada predição feita, a classe escolhida pelo SVM ganha um voto. A classe final é aquela que obteve mais votos após todas as predições binárias.

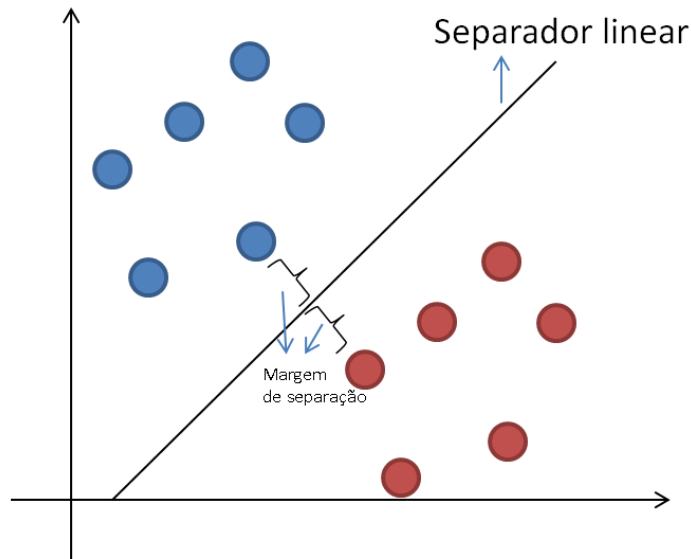


Figura 4.1: Um separador linear que divide dois grupos da melhor maneira possível

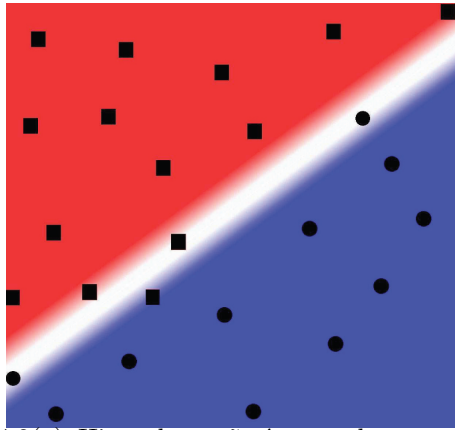
4.2

Kernel Trick

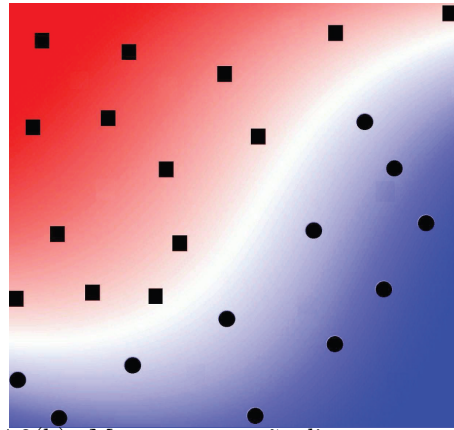
O modelo descrito anteriormente raramente pode ser utilizado na prática. Isso ocorre porque os bancos de dados não são, em geral, linearmente separáveis e são substancialmente mais complexos que o exemplificado. Um dos modos para fazer com que o SVM tenha capacidade de aprender e reconhecer padrões nesse tipo de banco de dados é através do *Kernel Trick*, também chamado de *Kernel Methods*. O *Kernel Trick* pode ser visto como uma extensão a diversos algoritmos de aprendizado de máquina, um deles sendo o SVM, que possibilita um meio de mapear dados de um grupo S em um espaço qualquer para outro espaço V com um número maior de dimensões. A finalidade desse mapeamento é encontrar um espaço V em que exista separação linear dos dados, fazendo com que classificações lineares em V correspondam a classificações genéricas em S . O mapeamento é feito através das chamadas *Kernel Functions* ou apenas *Kernels*. A Figura 4.2 ilustra um grupo de dados que não é linearmente separável, separado através de um mapeamento não linear.

O grande fundamento do *Kernel Trick* é definir um *kernel* K que, para dois pontos quaisquer x e y , $K(x, y)$ seja igual ao valor do produto interno dos vetores $\phi(x)$ e $\phi(y)$, sendo ϕ um mapeamento de S para V .

O uso do *kernel* não é apenas eficiente como também flexível. É geralmente mais eficiente computar K do que o mapeamento ϕ seguido do produto interno. Flexível, porque não é necessário definir ou computar explicitamente



4.2(a): Hiperplano não é capaz de separar os dados



4.2(b): Mapeamento não linear que separa os dados

Figura 4.2: Em 4.2(a) um exemplo de um grupo de dados não linearmente separável por um hiperplano. Em 4.2 um mapeamento não linear que separa os dois grupos de dados. Figuras de (31).

o mapeamento ϕ desde que a sua existência seja garantida. A existência de ϕ é garantida desde que o *kernel* K cumpra a condição de Mercer (28).

O *kernel* utilizado para treinamento do SVM nesta dissertação foi o *Radial Basis Function*, ou RBF que é definido da seguinte maneira:

$$K = \exp(-\gamma * |x - y|^2),$$

sendo γ uma constante positiva. Uma definição mais completa sobre o *Kernel Method* e outros *kernels* podem ser vistos na literatura como Mohri (2012) (31).