Support Vector Machine - SVM

Anna Eloyr Vilasboas
Arthur Nunes Torres

LAMFO/UnB

6 de julho de 2020

Estrutura da Oficina

1 Introdução

2 Support Vector Classifier

3 Support Vector Machine

4 SVM - Aplicação

5 Referências

Dilema Viés - Variância

Trade-off que surge ao se decidir entre um modelo mais ou menos flexível.

Dilema Viés - Variância

Trade-off que surge ao se decidir entre um modelo mais ou menos flexível.

O uso de um modelo exacerbadamente inflexível gera um alto viés, visto que simplifica uma relação que pode ser complexa.

Dilema Viés - Variância

Trade-off que surge ao se decidir entre um modelo mais ou menos flexível.

O uso de um modelo exacerbadamente inflexível gera um alto viés, visto que simplifica uma relação que pode ser complexa.

Porém, um modelo muito flexível gera *overfit*, ou seja, uma maior divergência entre os dados de treino e de teste.

Dilema Viés - Variância

Trade-off que surge ao se decidir entre um modelo mais ou menos flexível.

O uso de um modelo exacerbadamente inflexível gera um alto viés, visto que simplifica uma relação que pode ser complexa.

Porém, um modelo muito flexível gera *overfit*, ou seja, uma maior divergência entre os dados de treino e de teste.

$$E(y_0 - \hat{f}(x_0))^2 = Var(\hat{f}(x_0)) + [Bias(\hat{f}(x_0))]^2 + Var(\epsilon) \tag{1}$$

Validação Cruzada

Trata-se de uma ferramenta utilizada para avaliar modelos, valendo-se da metodologia de aprendizado supervisionado (treino do modelo com um determinado grupo dos dados e teste com outros).

Validação Cruzada

Trata-se de uma ferramenta utilizada para avaliar modelos, valendo-se da metodologia de aprendizado supervisionado (treino do modelo com um determinado grupo dos dados e teste com outros).

Os dados utilizados para moldar o modelo são divididos em partes iguais, e, um a um, são usados para testar o modelo enquanto os demais são utilizados para treiná-lo.

Validação Cruzada

Trata-se de uma ferramenta utilizada para avaliar modelos, valendo-se da metodologia de aprendizado supervisionado (treino do modelo com um determinado grupo dos dados e teste com outros).

Os dados utilizados para moldar o modelo são divididos em partes iguais, e, um a um, são usados para testar o modelo enquanto os demais são utilizados para treiná-lo.

Os resultados são computados e, então, utilizados para decidir qual modelo utilizar.

Estrutura da Oficina

1 Introdução

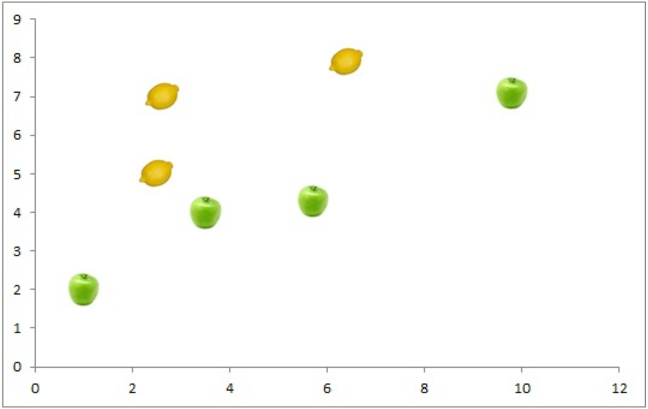
2 Support Vector Classifier

3 Support Vector Machine

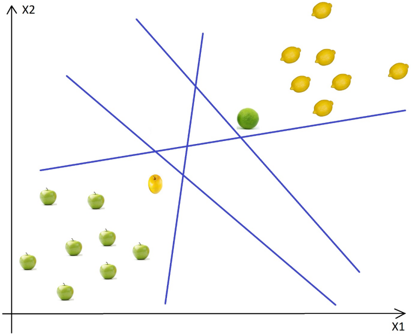
4 SVM - Aplicação

5 Referências

Como separar os elementos baseados no tipo?



Existem várias formas de separar



Maximal Margin Classifier

Utiliza-se um hiperplano de separação ótima, isto é, um hiperplano que divide os grupos ao mesmo tempo que maximiza a margem.

Maximal Margin Classifier

Utiliza-se um hiperplano de separação ótima, isto é, um hiperplano que divide os grupos ao mesmo tempo que maximiza a margem.

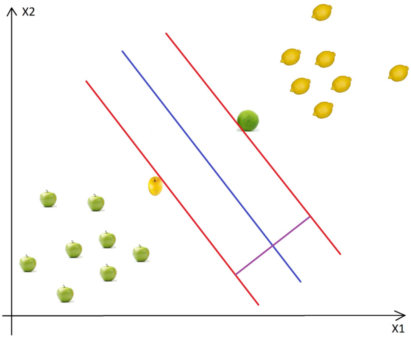
Portanto, o problema de otimização dá por meio de

$$\underset{\beta_0,\beta_1,\ldots,\beta_p,M}{\text{maximize}} \quad M$$

$$\text{subject to} \quad \sum_{j=1}^{p} \beta_j^2 = 1,$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) \geq M \forall i = 1, \ldots, n. \qquad (2)$$

Maximal Margin Classifier



Maximal Margin Classifier

Porém, note que a margem está sendo determinada pelas observações das extremidades dos grupos.

Maximal Margin Classifier

Porém, note que a margem está sendo determinada pelas observações das extremidades dos grupos.

Esse modelo se aplica a todos os casos?

E o que ocorre caso tenhamos outliers? Ou caso os grupos não sejam inteiramente separáveis?

Support Vector Classifier

Este método utiliza-se de validação cruzada para determinar o hiperplano que gerará melhores previsões no futuro, mesmo que para isso, ele permita que algumas observações violem a margem.

Support Vector Classifier

Este método utiliza-se de validação cruzada para determinar o hiperplano que gerará melhores previsões no futuro, mesmo que para isso, ele permita que algumas observações violem a margem.

Para isso, ao invés de utilizar as observações das extremidades dos grupos para determinar a posição do hiperplano e, consequentemente, da margem, esse classificador utiliza as observações que vão gerar essa melhores previsões no longo prazo. A essas observações, dá-se o nome de *support vectors*.

Support Vector Classifier

Portanto, a otimização vista no *maximal margin classifier* ganha a forma

$$\underset{\beta_0,\beta_1,\ldots,\beta_p,M}{\text{maximize}} \quad M$$

$$\text{subject to} \quad \sum_{j=1}^{p} \beta_j^2 = 1, \tag{3}$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip}) \geq M(1-\epsilon_i), \tag{4}$$

$$\epsilon_i \geq 0, \sum_{i=1}^{n} \epsilon_i \leq C \tag{5}$$

Support Vector Classifier

Desta forma, o C (e consequentemente ϵ_i) está diretamente ligado ao trade-off entre viés e variância, e sua definição sujeita ao método de validação cruzada.

Support Vector Classifier

Desta forma, o C (e consequentemente ϵ_i) está diretamente ligado ao trade-off entre viés e variância, e sua definição sujeita ao método de validação cruzada.

Note que, como o próprio nome indica, apenas as observações que ficam na margem ou a ultrapassam, influenciam na posição desta.

Estrutura da Oficina

1 Introdução

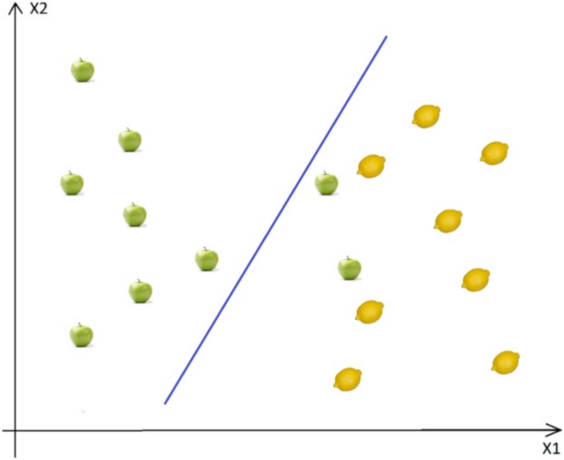
2 Support Vector Classifier

3 Support Vector Machine

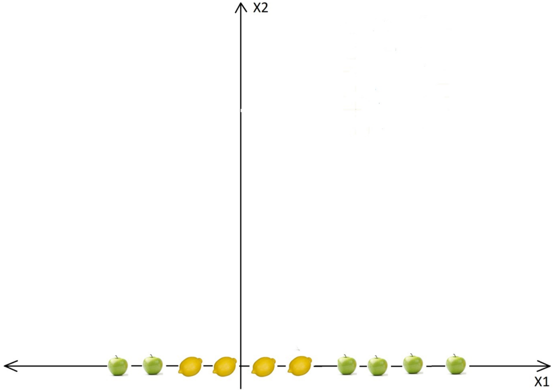
4 SVM - Aplicação

5 Referências

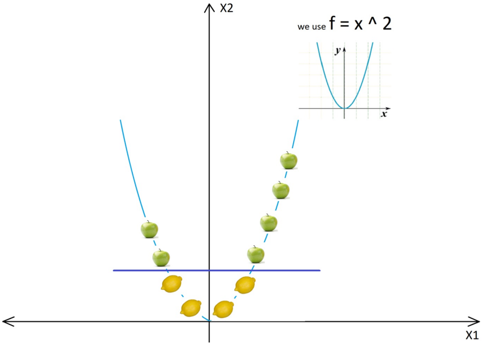
Classificação linear



Se o problema de classificação não for linear?



Solução

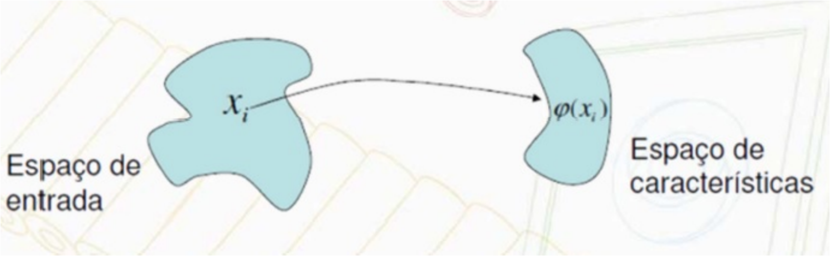


Mapear o conjunto de treinamento de seu espaço original (não linear) para um novo espaço de maior dimensão, denominado espaço de características (feature space), que é linear.

Tranformação não linear

- Para isso, precisamos encontrar uma transformação não linear, $\boldsymbol{\varphi}(x) = [\varphi_1(x), \ldots, \varphi_m(x)]$
  - Essa transformação mapeia o espaço original das observações para um novo espaço de atributos m - dimensional;
  - Nesse novo espaço, as observações passam a ser linearmente separáveis;
  - m pode ser muito maior que a dimensão do espaço original.
- Com a função de transformação, nosso problema de otimização recai pra uma SVM linear.

Transformação $x \to \phi(x)$



Produto Escalar

Ao analisar a estimação dos coeficientes no problema de otimização e a representação do classificador linear $f(x)$, que é dada por:

$$f(x) = \beta_0 + \sum_{i \in \mathcal{S}} \alpha_i \langle x, x_i \rangle$$

onde $\mathcal{S}$ é a coleção de índices desses pontos de suporte, concluímos que apenas precisamos dos **produtos escalares**.

Função Kernel

O algoritmo linear depende somente de $\langle x, x_i \rangle$, portanto o algoritmo transformado também dependerá somente de $\langle \boldsymbol{\varphi}(x), \boldsymbol{\varphi}(x_i) \rangle$.

Esse produto escalar entre os vetores transformados é chamado de função Kernel:

$$K(x, x_i) = \langle \boldsymbol{\varphi}(x), \boldsymbol{\varphi}(x_i) \rangle$$

Truque de Kernel

A função Kernel nos permite operar no espaço original, sem precisar computar as coordenadas dos dados em um espaço dimensional superior.

Por exemplo, vamos supor que x e y são observações em 3 dimensões:

$$\mathbf{x} = (x_1, x_2, x_3)^T$$
$$\mathbf{y} = (y_1, y_2, y_3)^T$$

Vamos assumir que precisamos mapear x e y para um espaço 9-dimensional.

Truque de Kernel

Para isso, precisamos realizar as seguintes operações para chegar ao resultado final, que é apenas um escalar:

$$\phi(\mathbf{x}) = \left(x_1^2, x_1x_2, x_1x_3, x_2x_1, x_2^2, x_2x_3, x_3x_1, x_3x_2, x_3^2\right)^T$$

$$\phi(\mathbf{y}) = \left(y_1^2, y_1y_2, y_1y_3, y_2y_1, y_2^2, y_2y_3, y_3y_1, y_3y_2, y_3^2\right)^T$$

$$\phi(\mathbf{x})^T\phi(\mathbf{y}) = \sum_{i,j=1}^{3} x_ix_jy_iy_j$$

Truque de Kernel

Porém, se usarmos a função Kernel, ao invés de fazer operações complexas em um espaço 9-dimensional, obtemos o mesmo resultado com um espaço 3-dimensional ao calcular o produto escalar do transposto de x e y:

$$
\begin{aligned}
K(\mathbf{x}, \mathbf{y}) &= \left(\mathbf{x}^T \mathbf{y}\right)^2 \\
&= (x_1 y_1 + x_2 y_2 + x_3 y_3)^2 \\
&= \sum_{i,j=1}^{3} x_i x_j y_i y_j
\end{aligned}
$$

Kernel Polinomial

Um exemplo de Kernel é:

$$K\left(x_i, x_{i'}\right) = \left(1 + \sum_{j=1}^{p} x_{ij} x_{i'j}\right)^d$$

que é conhecido como Kernel polinomial de grau d, onde d é um inteiro positivo.

Kernel Radial

Outra opção bastante popular é o Kernel radial, que possui a seguinte forma:

$$K\left(x_i, x_{i'}\right) = \exp\left(-\gamma \sum_{j=1}^{p} \left(x_{ij} - x_{i'j}\right)^2\right) \tag{6}$$

onde γ é uma constante positiva.

Support Vector Machine

Quando o *support vector classifier* é combinado com uma função Kernel, como o polinomial, o classificador resultante é conhecido como *support vector machine*.

Note que, de forma geral, a função do hiperplano terá a seguinte forma:

$$f(x) = \beta_0 + \sum_{i \in \mathcal{S}} \alpha_i K(x, x_i)$$

Estrutura da Oficina

1 Introdução

2 Support Vector Classifier

3 Support Vector Machine

4 SVM - Aplicação

5 Referências

Estrutura da Oficina

1 Introdução

2 Support Vector Classifier

3 Support Vector Machine

4 SVM - Aplicação

5 Referências

Bibliography I

Gareth James et al. *An introduction to statistical learning*. Vol. 112. Springer, 2013.

Alexandre Kowalczyk. “Support vector machines succinctly”. Em: *Syncfusion Inc* (2017).