Inteligência Artificial: Uma Abordagem de Aprendizado de Máquina

Máquinas de Vetores de Suporte

Prof. Tiago A. Almeida

Motivação

- Proposto no final da década de 1990
- Ganhou notoriedade a partir de 2000
- Atualmente é considerado o estado da arte na solução de diversos problemas de classificação

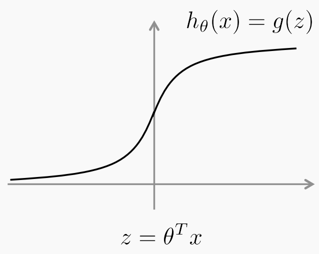
Motivação

- Classificador baseado em otimização
- Capacidade de trabalhar com dados complexos não-lineares
- Baixa sensibilidade à dimensionalidade dos dados
- Após treinamento, classificador resultante é rápido e independente da base de dados
  - Pode ser encapsulado em dispositivos com pouca capacidade computacional
- Possibilidade de escolher *kernels*

Função Objetivo

- **Regressão logística:**

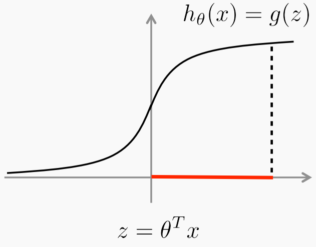
$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$



Função Objetivo

- **Regressão logística:**

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$



- Se $y = 1$, deseja-se que $h_\theta(x) \approx 1,\ \theta^T x \gg 0$

Função Objetivo

- **Regressão logística:**
- Custo por exemplo = $-(y \log h_\theta(x) + (1-y)\log(1 - h_\theta(x)))$

$$= -y \log \frac{1}{1 + e^{-\theta^T x}} - (1-y)\log(1 - \frac{1}{1 + e^{-\theta^T x}})$$

Amostras com y = 1

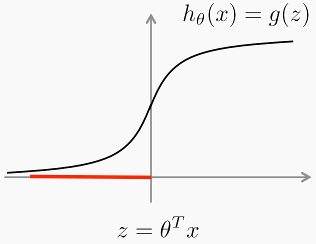
Amostras com y = 0

Contribuição de cada amostra de treinamento (x,y) na Função Custo

Função Objetivo

- **Regressão logística:**

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$



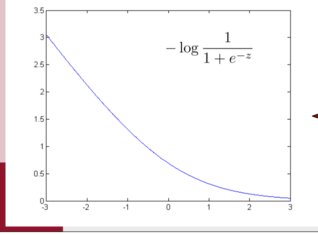
- Se $y = 1$, deseja-se que $h_\theta(x) \approx 1,\ \theta^T x \gg 0$
- Se $y = 0$, deseja-se que $h_\theta(x) \approx 0,\ \theta^T x \ll 0$

Função Objetivo

- **Regressão logística:**
- Custo por exemplo = $-(y \log h_\theta(x) + (1-y)\log(1 - h_\theta(x)))$

$$= -y \log \frac{1}{1 + e^{-\theta^T x}} - (1-y)\log(1 - \frac{1}{1 + e^{-\theta^T x}})$$

- Se $y = 1$ (deseja-se $\theta^T x \gg 0$):



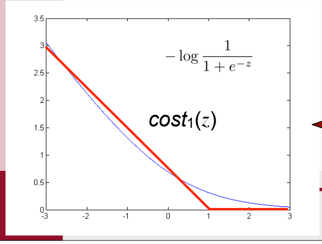
z >> 0 representa pouco aumento na Função Custo (J)

Função Objetivo

- **Regressão logística:**
- Custo por exemplo = $-(y\log h_\theta(x) + (1-y)\log(1-h_\theta(x)))$

$$= -y\log\frac{1}{1+e^{-\theta^T x}} - (1-y)\log(1-\frac{1}{1+e^{-\theta^T x}})$$

- Se $y=1$ (deseja-se $\theta^T x \gg 0$):



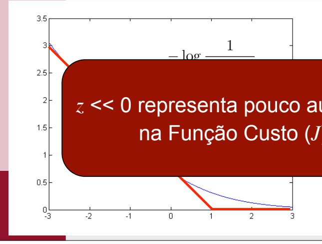
z >> 0 representa pouco aumento na Função Custo (J)

Função Objetivo

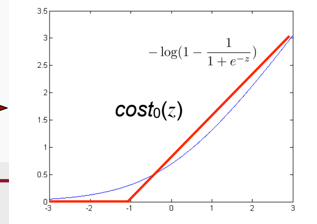
- **Regressão logística:**
- Custo por exemplo = $-(y\log h_\theta(x) + (1-y)\log(1-h_\theta(x)))$

$$= -y\log\frac{1}{1+e^{-\theta^T x}} - (1-y)\log(1-\frac{1}{1+e^{-\theta^T x}})$$

- Se $y=1$ (deseja-se $\theta^T x \gg 0$):



Se $y=0$ (deseja-se $\theta^T x \ll 0$):



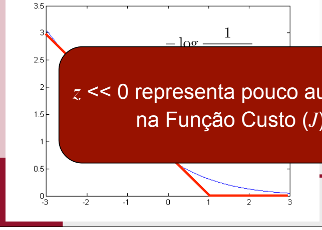
z << 0 representa pouco aumento na Função Custo (J)

Função Objetivo

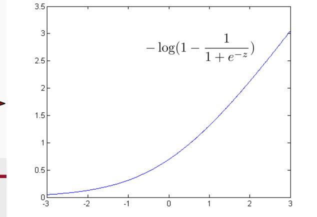
- **Regressão logística:**
- Custo por exemplo = $-(y\log h_\theta(x) + (1-y)\log(1-h_\theta(x)))$

$$= -y\log\frac{1}{1+e^{-\theta^T x}} - (1-y)\log(1-\frac{1}{1+e^{-\theta^T x}})$$

- Se $y=1$ (deseja-se $\theta^T x \gg 0$):



Se $y=0$ (deseja-se $\theta^T x \ll 0$):



z << 0 representa pouco aumento na Função Custo (J)

Função Objetivo

- **Regressão logística:**

$$\min_\theta \frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)}\left(-\log h_\theta(x^{(i)})\right) + (1-y^{(i)})\left((-\log(1-h_\theta(x^{(i)}))\right)\right] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

Função Objetivo

- **Regressão logística:**

$$\min_\theta \frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)}\underbrace{\cancel{\left(-\log h_\theta(x^{(i)})\right)}}_{cost_1(\theta^T x^{(i)})} + (1-y^{(i)})\underbrace{\cancel{\left((-\log(1-h_\theta(x^{(i)}))\right)}}_{cost_0(\theta^T x^{(i)})}\right] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

Função Objetivo

- **Regressão logística:**

$$\min_\theta \frac{1}{\cancel{m}}\left[\sum_{i=1}^{m} y^{(i)}\underbrace{\cancel{\left(-\log h_\theta(x^{(i)})\right)}}_{cost_1(\theta^T x^{(i)})} + (1-y^{(i)})\underbrace{\cancel{\left((-\log(1-h_\theta(x^{(i)}))\right)}}_{cost_0(\theta^T x^{(i)})}\right] + \frac{\lambda}{2\cancel{m}}\sum_{j=1}^{n}\theta_j^2$$

Função Custo

- **Regressão logística:**

$$\min_\theta \frac{1}{\cancel{m}}\left[\sum_{i=1}^{m} y^{(i)}\underbrace{\cancel{\left(-\log h_\theta(x^{(i)})\right)}}_{cost_1(\theta^T x^{(i)})} + (1-y^{(i)})\underbrace{\cancel{\left((-\log(1-h_\theta(x^{(i)}))\right)}}_{cost_0(\theta^T x^{(i)})}\right] + \frac{\overset{C}{\cancel{\lambda}}}{2\cancel{m}}\sum_{j=1}^{n}\theta_j^2$$

Constante de regularização **λ** passa a ser *C* (equivalente a 1/λ) e multiplica a primeira parte de expressão. Ambas as funções conduzem à mesma solução ótima.

Função Custo

- **Regressão logística:**

$$\min_\theta \frac{1}{\cancel{m}}\left[\sum_{i=1}^{m} y^{(i)}\underbrace{\cancel{\left(-\log h_\theta(x^{(i)})\right)}}_{cost_1(\theta^T x^{(i)})} + (1-y^{(i)})\underbrace{\cancel{\left((-\log(1-h_\theta(x^{(i)}))\right)}}_{cost_0(\theta^T x^{(i)})}\right] + \frac{\overset{C}{\cancel{\lambda}}}{2\cancel{m}}\sum_{j=1}^{n}\theta_j^2$$

- **Máquina de vetores de suporte:**

$$min_\theta \quad C\sum_{i=1}^{m}[y^{(i)}cost_1(\theta^T x^{(i)}) + (1-y^{(i)})cost_0(\theta^T x^{(i)})] + \frac{1}{2}\sum_{j=1}^{n}\theta_j^2$$

Função Custo

- **Máquina de vetores de suporte:**

$$min_\theta \;\; C\sum_{i=1}^{m}[y^{(i)}cost_1(\theta^T x^{(i)}) + (1-y^{(i)})cost_0(\theta^T x^{(i)})] + \frac{1}{2}\sum_{j=1}^{n}\theta_j^2$$

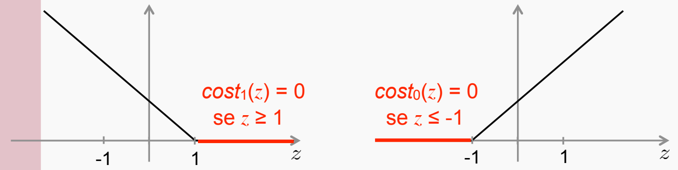
- **Uma vez otimizados os parâmetros, a hipótese:**

$h_\theta(x) = 1$ se $\theta^T x \geq 0$

0 caso contrário

Margem

- SVM: $min_\theta \;\; C\sum_{i=1}^{m}[y^{(i)}cost_1(\theta^T x^{(i)}) + (1-y^{(i)})cost_0(\theta^T x^{(i)})] + \frac{1}{2}\sum_{j=1}^{n}\theta_j^2$



- Se $y = 1$, deseja-se que $\theta^T x \geq 1$ (não apenas ≥ 0)
- Se $y = 0$, deseja-se que $\theta^T x \leq -1$ (não apenas < 0)

Aumenta a margem

Margem

- Durante o treinamento o SVM seleciona vetores de suporte que **maximizam a margem** de separação entre as classes

Limite de decisão

$$min_\theta \;\; C\sum_{i=1}^{m}[y^{(i)}cost_1(\theta^T x^{(i)}) + (1-y^{(i)})cost_0(\theta^T x^{(i)})] + \frac{1}{2}\sum_{j=1}^{n}\theta_j^2$$

Para C relativamente grande

$$\min_\theta \frac{1}{2}\sum_{j=1}^{n}\theta_j^2$$

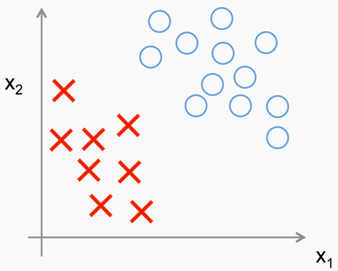
s.a. $\theta^T x^{(i)} \geq 1$ se $y^{(i)} = 1$

$\theta^T x^{(i)} \leq -1$ se $y^{(i)} = 0$

Formulação equivalente

Limite de decisão

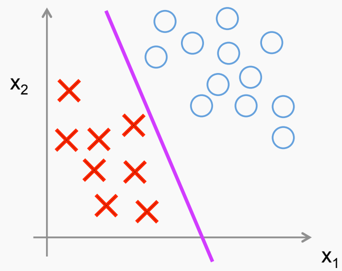
$$min_\theta \;\; C\sum_{i=1}^{m}[y^{(i)}cost_1(\theta^T x^{(i)}) + (1-y^{(i)})cost_0(\theta^T x^{(i)})] + \frac{1}{2}\sum_{j=1}^{n}\theta_j^2$$



Dados linearmente separáveis

Limite de decisão

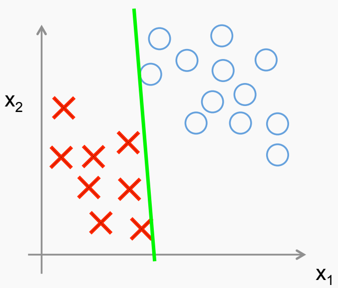
$$min_\theta \;\; C\sum_{i=1}^{m}[y^{(i)}cost_1(\theta^T x^{(i)}) + (1-y^{(i)})cost_0(\theta^T x^{(i)})] + \frac{1}{2}\sum_{j=1}^{n}\theta_j^2$$



Maximização da margem

Limite de decisão

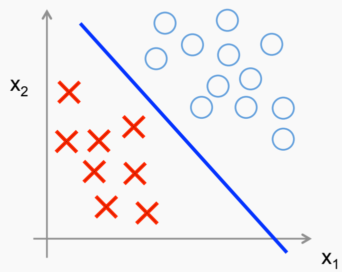
$$min_\theta \;\; C\sum_{i=1}^{m}[y^{(i)}cost_1(\theta^T x^{(i)}) + (1-y^{(i)})cost_0(\theta^T x^{(i)})] + \frac{1}{2}\sum_{j=1}^{n}\theta_j^2$$



Dados linearmente separáveis

Limite de decisão

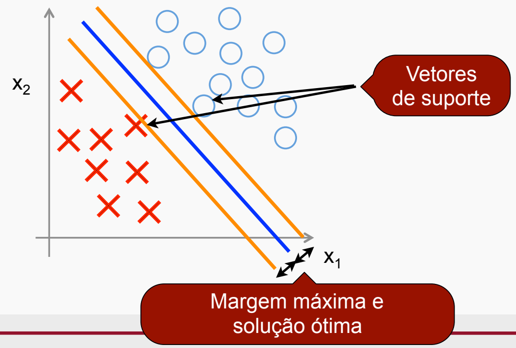
$$min_\theta \;\; C\sum_{i=1}^{m}[y^{(i)}cost_1(\theta^T x^{(i)}) + (1-y^{(i)})cost_0(\theta^T x^{(i)})] + \frac{1}{2}\sum_{j=1}^{n}\theta_j^2$$



Margem máxima

Limite de decisão

$$min_\theta \; C\sum_{i=1}^{m}[y^{(i)}cost_1(\theta^T x^{(i)}) + (1-y^{(i)})cost_0(\theta^T x^{(i)})] + \frac{1}{2}\sum_{j=1}^{n}\theta_j^2$$



Kernels

- Para gerar hipóteses mais complexas e não-lineares é necessário aumentar o grau do polinômio, ex:

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 x_2 + \theta_4 x_1 x_2^2 + \ldots \theta_9 x_1^3 x_2^3 + \ldots$$

Kernels

- Por padrão, o SVM emprega classificador linear
- Para separar dados complexos e não-lineares são usadas funções chamadas ***kernels***

Kernels

- Para gerar hipóteses mais complexas e não-lineares é necessário aumentar o grau do polinômio, ex:

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 x_2 + \theta_4 x_1 x_2^2 + \ldots \theta_9 x_1^3 x_2^3 + \ldots$$

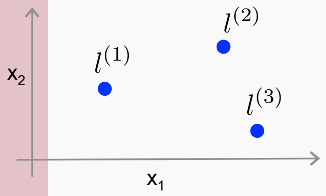
- Generalizando, cada combinação de atributos pode ser representada por uma variável f:

$$h_\theta(x) = \theta_0 + \theta_1 f_1 + \theta_2 f_2 + \theta_3 f_3 + \theta_4 f_4 + \ldots \theta_9 f_9 + \ldots$$

- Onde $f_1 = x_1, f_2 = x_2, f_3 = x_1^2 x_2, f_4 = x_1 x_2^2, \ldots, f_9 = x_1^3 x_2^3, \ldots$
- Qual é a melhor escolha para $f_1, f_2, f_3, \ldots$?

Kernels

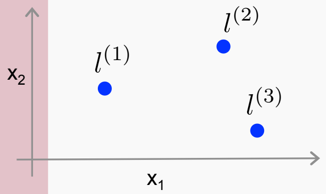
- Dada uma amostra x, é possível computar novos atributos a partir de aproximações a pontos de referências $l^{(1)}, l^{(2)}, l^{(3)}$



Com três pontos de referência, gera-se três novos atributos a partir de x

Kernels

- Dada uma amostra x, é possível computar novos atributos a partir de aproximações a pontos de referências $l^{(1)}, l^{(2)}, l^{(3)}$

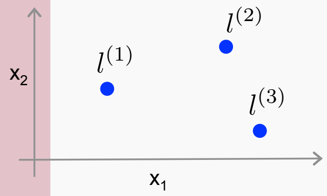


$$f_i = \text{similaridade}\,(x, l^{(i)})$$

Kernel

Kernels

- Dada uma amostra x, é possível computar novos atributos a partir de aproximações a pontos de referências $l^{(1)}, l^{(2)}, l^{(3)}$



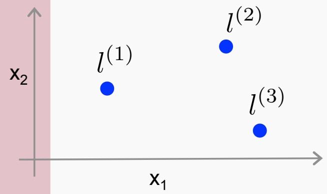
$$f_i = \text{similaridade}\,(x, l^{(i)}) = \exp\left(-\frac{||x - l^{(i)}||^2}{2\sigma^2}\right)$$

Kernel

Exemplo: *kernel* Gaussiano. Retorna 1, se $x = l^{(i)}$ e 0, se x estiver longe de $l^{(i)}$

Kernels

- Dada uma amostra x, é possível computar novos atributos a partir de aproximações a pontos de referências $l^{(1)}, l^{(2)}, l^{(3)}$



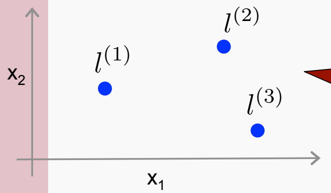
Como escolher $l^{(1)}, l^{(2)}, l^{(3)}$, ... ?

$$f_i = \text{similaridade}\,(x, l^{(i)}) = \exp\left(-\frac{||x - l^{(i)}||^2}{2\sigma^2}\right)$$

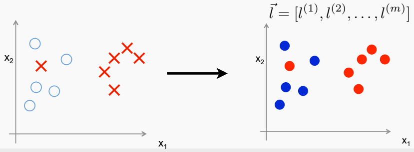
Kernel

Kernels

- Dada uma amostra x, é possível computar novos atributos a partir de aproximações a pontos de referências $l^{(1)}, l^{(2)}, l^{(3)}$



Como escolher $l^{(1)}, l^{(2)}, l^{(3)}$, ... ?
Resposta: definir um ponto de referência para cada amostra da base de dados



Kernels

- Dado conjunto: $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(m)}, y^{(m)})$, definir $l^{(1)} = x^{(1)}, l^{(2)} = x^{(2)}, \ldots, l^{(m)} = x^{(m)}$.

- Dado um exemplo qualquer $x^{(i)}, y^{(i)}$:

$$x^{(i)} \rightarrow \begin{array}{l} f_1^{(i)} = similaridade(x^{(i)}, l^{(1)}) \\ f_2^{(i)} = similaridade(x^{(i)}, l^{(2)}) \\ \vdots \\ f_m^{(i)} = similaridade(x^{(i)}, l^{(m)}) \end{array} \longrightarrow f^{(i)} = \begin{bmatrix} f_0^{(i)} = 1 \\ f_1^{(i)} \\ f_2^{(i)} \\ \vdots \\ f_m^{(i)} \end{bmatrix}$$

Kernel

Kernels

- **Classificação**
  - Dado x, calcular termos $f \in \mathbb{R}^{m+1}$
  - Predição y = 1, se $\theta^T f \geq 0$

Kernels

- **Classificação**
  - Dado x, calcular termos $f \in \mathbb{R}^{m+1}$
  - Predição y = 1, se $\theta^T f \geq 0$

- **Treinamento**

$$\min_\theta C \sum_{i=1}^{m} y^{(i)} cost_1(\theta^T f^{(i)}) + (1 - y^{(i)}) cost_0(\theta^T f^{(i)}) + \frac{1}{2} \sum_{j=1}^{n} \theta_j^2$$

Parâmetros

- C (= $\frac{1}{\lambda}$)
  - C grande (λ pequeno): baixo viés, alta variância
  - C pequeno (λ grande): alto viés, baixa variância

Usando o SVM

- Empregar biblioteca do SVM (libSVM, Shogun, libLinear, SVM^{light}, ...) para minimizar a função custo e ajustar os parâmetros θ
- É necessário:
  - Setar o fator de regularização C
  - Escolher o *kernel* e setar seus parâmetros (se houver):
    - Linear (ausência de *kernel*): y = 1, se $\theta^T x \geq 0$
    - Gaussiano (setar σ^2)
    - Polinomial
    - ...

Recomendável realizar normalização dos dados

Parâmetros

- C (= $\frac{1}{\lambda}$)
  - C grande (λ pequeno): baixo viés, alta variância
  - C pequeno (λ grande): alto viés, baixa variância
- σ^2
  - Valor alto: atributos f_i variam mais suavemente
    - alto viés, baixa variância
  - Valor baixo: atributos f_i variam menos suavemente
    - baixo viés, alta variância

SVM x Regressão Logística

- n = qtde de atributos, m = qtde de amostras
  - Se n >> m
    - SVM Linear ou Regressão Logística
  - Se m >> n, porém m não é muito grande
    - SVM com *kernel* Gaussiano
  - Se m >> n e m é muito grande
    - Adicionar/criar mais atributos e usar Regressão Logística ou SVM Linear

SVM x Regressão Logística

- n = qtde de atributos, m = qtde de amostras
  - Se $n >> m$
    - SVM Linear ou Regressão Logística
  - Se $m >> n$, porém m não é muito grande
    - SVM com *kernel* Gaussiano
  - Se $m >> n$ e m é muito grande
    - Adicionar/criar mais atributos e usar Regressão Logística ou SVM Linear

Redes neurais podem ser usadas em qualquer situação, porém costumam usar mais recursos computacionais (mais lentas)