

Dicas VIP: Aprendizado Supervisionado

Afshine AMIDI e Shervine AMIDI

13 de Outubro de 2018

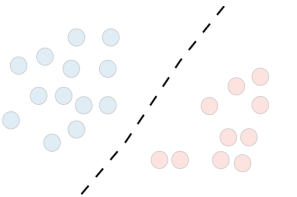
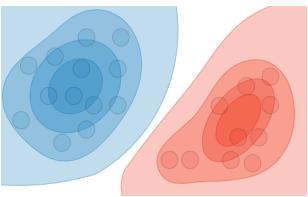
Introdução ao Aprendizado Supervisionado

Dado um conjunto de dados $\{x^{(1)}, \dots, x^{(m)}\}$ associados a um conjunto de resultados $\{y^{(1)}, \dots, y^{(m)}\}$, nós queremos construir um classificador que aprende como prever y baseado em x .

□ **Tipos de predição** – Os diferentes tipos de modelo de predição estão resumidos na tabela abaixo:

	Regressão	Classificador
Resultado	Contínuo	Classe
Exemplos	Regressão linear	Regressão logística, SVM, Naive Bayes

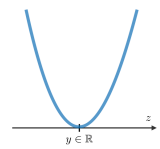
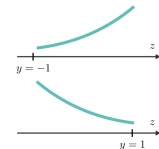
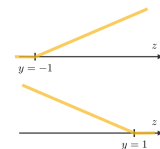
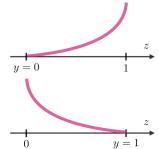
□ **Tipos de modelo** – Os diferentes modelos estão resumidos na tabela abaixo:

	Modelo discriminativo	Modelo generativo
Objetivo	Estimar diretamente $P(y x)$	Estimar $P(x y)$, deduzir $P(y x)$
O que é aprendido	Fronteira de decisão	Probabilidade da dist. dos dados
Ilustração		
Exemplos	Regressões, SVMs	GDA, Naive Bayes

Notações e conceitos gerais

□ **Hipótese** – A hipótese é denominada h_θ e é o modelo que escolhemos. Para um determinado dado de entrada $x^{(i)}$ o resultado do modelo de predição é $h_\theta(x^{(i)})$.

□ **Função de perda** – A função de perda é definida como $L: (z, y) \in \mathbb{R} \times Y \mapsto L(z, y) \in \mathbb{R}$ que recebe como entradas o valor z previsto correspondente ao valor real y e retorna o quão diferente eles são.

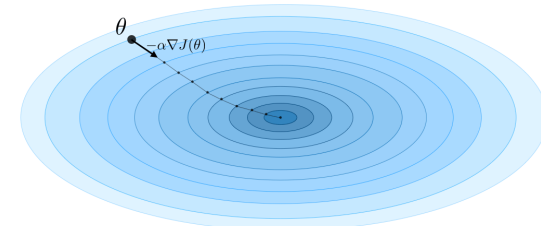
Quadrático	Logística	Hinge	Entropia cruzada
$\frac{1}{2}(y - z)^2$	$\log(1 + \exp(-yz))$	$\max(0, 1 - yz)$	$-\left[y \log(z) + (1 - y) \log(1 - z)\right]$
			
Regressão linear	Regressão logística	SVM	Rede neural

□ **Função de custo** – A função de custo J é normalmente usada para avaliar a performance de um modelo e é definida usando a função de perda L como:

$$J(\theta) = \sum_{i=1}^m L(h_\theta(x^{(i)}), y^{(i)})$$

□ **Gradiente descendente** – Definindo $\alpha \in \mathbb{R}$ como a taxa de aprendizado, a regra de atualização para o gradiente descendente é expressa usando a taxa de aprendizado e a função de custo J como:

$$\theta \leftarrow \theta - \alpha \nabla J(\theta)$$



Observação: O gradiente descendente estocástico (GDE) atualiza o parâmetro baseado em cada exemplo de treinamento e o gradiente descendente em lote em um conjunto de exemplos de treinamento.

□ **Probabilidade** – A probabilidade de um modelo $L(\theta)$ dado os parâmetros θ é usada para encontrar os parâmetros ótimos θ pela maximização da probabilidade. Na prática, é usado o logaritmo da probabilidade (log-likelihood) $\ell(\theta) = \log(L(\theta))$ que é mais simples para se otimizar. Tem-se:

$$\theta^{\text{opt}} = \arg \max_{\theta} L(\theta)$$

□ **Algoritmo de Newton** – O algoritmo de Newton é um método numérico que encontra θ tal que $\ell'(\theta) = 0$. Sua regra de atualização é:

$$\theta \leftarrow \theta - \frac{\ell'(\theta)}{\ell''(\theta)}$$

Observação: a generalização multidimensional, também conhecida como o método de Newton-Raphson, tem a seguinte regra de atualização:

$$\theta \leftarrow \theta - \left(\nabla_{\theta}^2 \ell(\theta) \right)^{-1} \nabla_{\theta} \ell(\theta)$$

Regressão linear

Assume-se que $y|x; \theta \sim \mathcal{N}(\mu, \sigma^2)$

□ **Equações normais** – Definindo X como o desenho da matriz, o valor θ que minimiza a função de custo em uma solução de forma fechada é dado por:

$$\theta = (X^T X)^{-1} X^T y$$

□ **Algoritmo MMQ** – Definindo α como a taxa de aprendizado, a regra de atualização do algoritmo de Média de Mínimos Quadrados para um conjunto de treinamento de m pontos, também conhecida como a regra de atualização de Widrow-Hoff, é dada por:

$$\forall j, \quad \theta_j \leftarrow \theta_j + \alpha \sum_{i=1}^m [y^{(i)} - h_{\theta}(x^{(i)})] x_j^{(i)}$$

Observação: a regra de atualização é um caso particular do gradiente ascendente.

□ **LWR** – Regressão Ponderada Localmente (Locally Weighted Regression), também conhecida como LWR, é uma variação da regressão linear que sempre pondera cada exemplo de treinamento em sua função de custo por $w^{(i)}(x)$, que é definida com o parâmetro $\tau \in \mathbb{R}$ como:

$$w^{(i)}(x) = \exp \left(- \frac{(x^{(i)} - x)^2}{2\tau^2} \right)$$

Classificação e regressão logística

□ **Função sigmoide** – A função sigmoide g , também conhecida como função logística, é definida como:

$$\forall z \in \mathbb{R}, \quad g(z) = \frac{1}{1 + e^{-z}} \in]0, 1[$$

□ **Regressão logística** – Se assume que $y|x; \theta \sim \text{Bernoulli}(\phi)$. Tem-se a seguinte fórmula:

$$\phi = p(y = 1|x; \theta) = \frac{1}{1 + \exp(-\theta^T x)} = g(\theta^T x)$$

Observação: não existe uma fórmula de solução fechada para o caso de regressão logística.

□ **Regressão softmax** – A regressão softmax, também chamada de regressão logística multiclasse, é usada para generalizar a regressão logística quando existem mais de 2 classes. Por convenção, definimos $\theta_K = 0$, que faz com que o parâmetro de Bernoulli ϕ_i de cada classe i seja igual a:

$$\phi_i = \frac{\exp(\theta_i^T x)}{\sum_{j=1}^K \exp(\theta_j^T x)}$$

Modelos Lineares Generalizados

□ **Família exponencial** – Uma classe de distribuições é chamada de família exponencial se ela puder ser escrita em termos de um parâmetro natural, também chamado de parâmetro canônico ou função de link η , uma estatística suficiente $T(y)$ e de uma função de partição de $\log a(\eta)$ e é dada por:

$$p(y; \eta) = b(y) \exp(\eta T(y) - a(\eta))$$

Observação: em geral tem-se $T(y) = y$. Também, $\exp(-a(\eta))$ pode ser definido como o parâmetro de normalização que garantirá que as probabilidades somem um.

Na tabela a seguir estão resumidas as distribuições exponenciais mais comuns:

Distribuição	η	$T(y)$	$a(\eta)$	$b(y)$
Bernoulli	$\log \left(\frac{\phi}{1-\phi} \right)$	y	$\log(1 + \exp(\eta))$	1
Gaussiana	μ	y	$\frac{\eta^2}{2}$	$\frac{1}{\sqrt{2\pi}} \exp \left(-\frac{y^2}{2} \right)$
Poisson	$\log(\lambda)$	y	e^{η}	$\frac{1}{y!}$
Geométrica	$\log(1 - \phi)$	y	$\log \left(\frac{e^{\eta}}{1 - e^{\eta}} \right)$	1

□ **Suposições de GLMs** – Modelos Lineares Generalizados (GLM) visa prever uma variável aleatória y através da função $x \in \mathbb{R}^{n+1}$ e conta com as 3 seguintes premissas:

$$(1) \quad y|x; \theta \sim \text{ExpFamily}(\eta) \quad (2) \quad h_{\theta}(x) = E[y|x; \theta] \quad (3) \quad \eta = \theta^T x$$

Observação: mínimos quadrados ordinários e regressão logística são casos especiais de modelos lineares generalizados.

Máquinas de Vetores de Suporte

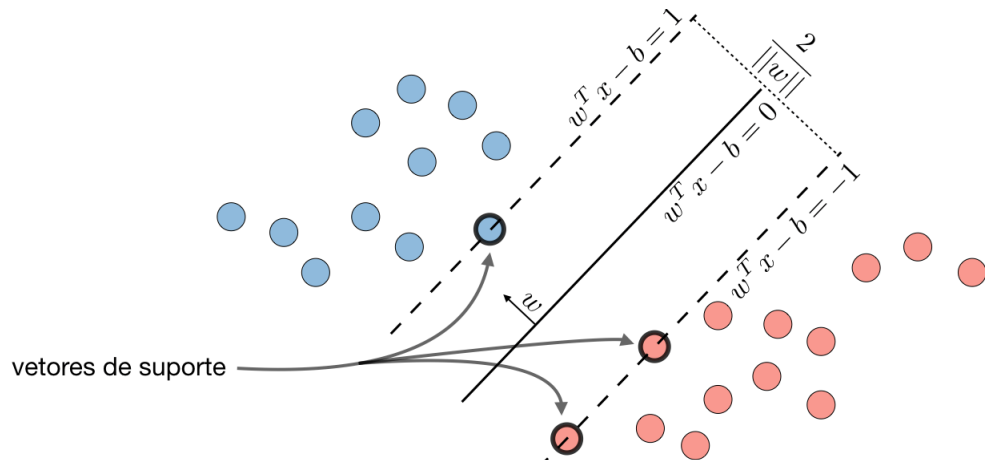
O objetivo das máquinas de vetores de suporte (support vector machines) é encontrar a linha que maximiza a distância mínima até a linha.

□ **Classificador de margem ideal** – O classificador de margem ideal h é definido por:

$$h(x) = \text{sign}(w^T x - b)$$

onde $(w, b) \in \mathbb{R}^n \times \mathbb{R}$ é a solução para o seguinte problema de otimização:

$$\min \frac{1}{2} \|w\|^2 \quad \text{tal como} \quad y^{(i)}(w^T x^{(i)} - b) \geq 1$$



Observação: a linha é definida como $w^T x - b = 0$.

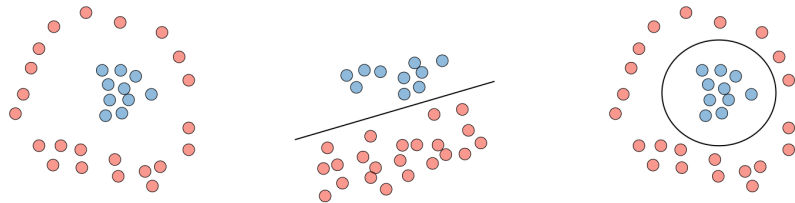
❑ **Perda de Hinge** – A perda de articulação é usada na configuração das máquinas de vetores de suporte (SVMs) e é definida como:

$$L(z, y) = [1 - yz]_+ = \max(0, 1 - yz)$$

❑ **Kernel** – Dado um mapeamento de parâmetro ϕ , o kernel K é definido como:

$$K(x, z) = \phi(x)^T \phi(z)$$

Na prática, o kernel K definido por $K(x, z) = \exp\left(-\frac{\|x - z\|^2}{2\sigma^2}\right)$ é chamado de kernel Gaussiano e é comumente usado.



Separabilidade não-linear \Rightarrow Uso de mapeamento de kernel $\phi \Rightarrow$ Limite de decisão no espaço original

Observação: é dito que é usado o "truque de kernel" (kernel trick) para calcular a função de custo usando o kernel porque na verdade não precisamos saber o mapeamento explícito de ϕ , que é muito complicado. Ao invés, apenas os valores $K(x, z)$ são necessários.

❑ **Lagrangiano** – O Lagrangiano $\mathcal{L}(w, b)$ é definido por:

$$\mathcal{L}(w, b) = f(w) + \sum_{i=1}^l \beta_i h_i(w)$$

Observação: os coeficientes β_i são chamados de multiplicadores Lagrangeanos.

Aprendizado Generativo

Um modelo generativo primeiro tenta aprender como o dado é gerado estimando $P(x|y)$, o que pode ser usado para estimar $P(y|x)$ usando a regra de Bayes.

Análise Discriminante Gaussiana

❑ **Configuração** – A Análise Discriminante Gaussiana assume que y e $x|y = 0$ e $x|y = 1$ são tais que:

$$y \sim \text{Bernoulli}(\phi)$$

$$x|y = 0 \sim \mathcal{N}(\mu_0, \Sigma) \quad \text{et} \quad x|y = 1 \sim \mathcal{N}(\mu_1, \Sigma)$$

❑ **Estimativa** – A tabela a seguir resume as estimativas que encontramos ao maximizar a probabilidade:

$\hat{\phi}$	$\hat{\mu}_j \quad (j = 0, 1)$	$\hat{\Sigma}$
$\frac{1}{m} \sum_{i=1}^m 1_{\{y^{(i)}=1\}}$	$\frac{\sum_{i=1}^m 1_{\{y^{(i)}=j\}} x^{(i)}}{\sum_{i=1}^m 1_{\{y^{(i)}=j\}}}$	$\frac{1}{m} \sum_{i=1}^m (x^{(i)} - \mu_{y^{(i)}})(x^{(i)} - \mu_{y^{(i)}})^T$

Naive Bayes

❑ **Premissas** – O modelo de Naive Bayes assume que os parâmetros (features) de cada dado do conjunto são independentes:

$$P(x|y) = P(x_1, x_2, \dots | y) = P(x_1|y)P(x_2|y)\dots = \prod_{i=1}^n P(x_i|y)$$

❑ **Soluções** – Maximizar o logaritmo da probabilidade nos dá as seguintes soluções, com $k \in \{0, 1\}, l \in \llbracket 1, L \rrbracket$

$$P(y = k) = \frac{1}{m} \times \#\{j | y^{(j)} = k\} \quad \text{et} \quad P(x_i = l | y = k) = \frac{\#\{j | y^{(j)} = k \text{ et } x_i^{(j)} = l\}}{\#\{j | y^{(j)} = k\}}$$

Observação: Naive Bayes é amplamente utilizado para classificação de texto e detecção de spam.

Métodos em conjunto e baseados em árvore

Esses métodos podem ser usados tanto para problemas de regressão quanto de classificação.

❑ **CART** – Árvores de Classificação e Regressão (CART), normalmente conhecida como árvores de decisão (decision trees), podem ser representadas como árvores binárias. Elas tem a vantagem de serem facilmente interpretadas.

❑ **Floresta aleatória** – É uma técnica baseada em árvore que usa um grande número de árvores de decisão construídas a partir de um conjunto aleatório de parâmetros. Ao contrário de uma

simples árvore de decisão, esta técnica é de difícil interpretação mas geralmente alcança uma boa performance, sendo um algoritmo popular.

Observação: florestas aleatórias são um tipo de métodos de conjunto.

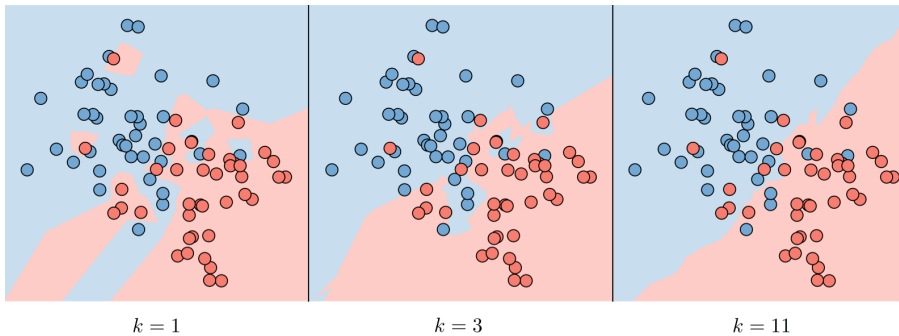
□ **Boosting** – A ideia dos métodos de boosting é combinar vários tipo de aprendizes fracos (*weak learners*) para formar um mais forte. Os principais tipos estão resumidos na tabela abaixo:

Boosting adaptativo	Gradiente de boosting
- De grands coefficients sont mis sur les erreurs pour s'améliorer à la prochaine étape de boosting - Connus sous le nom d'Adaboost	- Les modèles faibles sont entraînés sur les erreurs résiduelles

Outras abordagens não paramétricas

□ **k -vizinhos próximos** – O algoritmo de k -vizinhos próximos, normalmente conhecido como k -NN, é uma abordagem não paramétrica onde a resposta do dado é determinada pela natureza dos seus k vizinhos no conjunto de treinamento. Ele pode ser usado tanto em configurações de classificação como regressão.

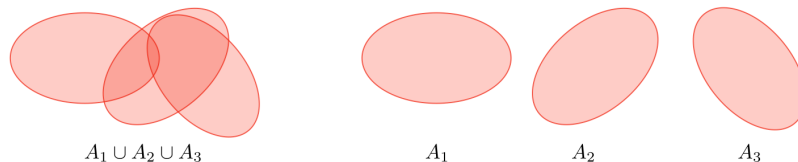
Observação: Quanto maior o parâmetro k , maior o viés, e quanto menor o parâmetro k , maior a variância.



Teoria de Aprendizagem

□ **Limite de união** – Dado que A_1, \dots, A_k são k eventos. Temos que:

$$P(A_1 \cup \dots \cup A_k) \leq P(A_1) + \dots + P(A_k)$$



□ **Desigualdade de Hoeffding** – Dado que Z_1, \dots, Z_m são m iid variáveis extraídas de uma distribuição de Bernoulli do parâmetro ϕ . Seja $\hat{\phi}$ a média amostral deles e fixado $\gamma > 0$. Temos que:

$$P(|\phi - \hat{\phi}| > \gamma) \leq 2 \exp(-2\gamma^2 m)$$

Observação: essa desigualdade também é chamada de fronteira Chernoff.

□ **Erro de treinamento** – Para um dado classificador h , é definido o erro de treinamento $\hat{\epsilon}(h)$, também conhecido como o risco ou o erro empírico, como:

$$\hat{\epsilon}(h) = \frac{1}{m} \sum_{i=1}^m 1_{\{h(x^{(i)}) \neq y^{(i)}\}}$$

□ **Provavelmente Aproximadamente Correto (PAC)** – PAC é uma estrutura (framework) em que numerosos resultados da teoria de aprendizagem foram provados, e tem o seguinte conjunto de premissas:

- o conjunto de treino e teste seguem a mesma distribuição
- os exemplos de treinamento foram extraídos de forma independente

□ **Shattering** – Dado um conjunto $S = \{x^{(1)}, \dots, x^{(d)}\}$, e um conjunto de classificadores \mathcal{H} , diz-se que \mathcal{H} destrói (shatters) S se para qualquer conjunto de rótulos $\{y^{(1)}, \dots, y^{(d)}\}$, temos:

$$\exists h \in \mathcal{H}, \quad \forall i \in [1, d], \quad h(x^{(i)}) = y^{(i)}$$

□ **Teorema da fronteira superior** – Seja \mathcal{H} uma class de hipótese finita tal que $|\mathcal{H}| = k$ e seja δ e o tamanho da amostra m fixado. Então, com a probabilidade de ao menos $1 - \delta$, temos:

$$\epsilon(\hat{h}) \leq \left(\min_{h \in \mathcal{H}} \epsilon(h) \right) + 2 \sqrt{\frac{1}{2m} \log \left(\frac{2k}{\delta} \right)}$$

□ **Dimensão VC** – A dimensão Vapnik-Chervonenkis (VC) de uma classe de hipótese infinita \mathcal{H} , denominada $VC(\mathcal{H})$ é o tamanho do maior conjunto que é destruído (shattered) por \mathcal{H} .

Observação: a dimensão VC de $\mathcal{H} = \{\text{set of linear classifiers in 2 dimensions}\}$ é 3



□ **Teorema (Vapnik)** – Dado \mathcal{H} , com $VC(\mathcal{H}) = d$ e m o número de exemplos de treinamento. Com a probabilidade de ao menos $1 - \delta$, temos que:

$$\epsilon(\hat{h}) \leq \left(\min_{h \in \mathcal{H}} \epsilon(h) \right) + O \left(\sqrt{\frac{d}{m} \log \left(\frac{m}{d} \right)} + \frac{1}{m} \log \left(\frac{1}{\delta} \right) \right)$$