

Super VIP Cheatsheet: Aprendizado de Máquina

Afshine AMIDI e Shervine AMIDI

13 de Outubro de 2018

Conteúdo

1	Aprendizado supervisionado	2
1.1	Introdução ao Aprendizado Supervisionado	2
1.2	Notações e conceitos gerais	2
1.3	Modelos lineares	3
1.3.1	Regressão linear	3
1.3.2	Classificação e regressão logística	3
1.3.3	Modelos Lineares Generalizados	3
1.4	Máquinas de Vetores de Suporte	3
1.5	Aprendizado Generativo	4
1.5.1	Análise Discriminante Gaussiana	4
1.5.2	Naive Bayes	4
1.6	Métodos em conjunto e baseados em árvore	4
1.7	Outras abordagens não paramétricas	5
1.8	Teoria de Aprendizagem	5
2	Aprendizado não supervisionado	6
2.1	Introdução ao aprendizado não supervisionado	6
2.2	Agrupamento	6
2.2.1	Maximização de expectativa	6
2.2.2	Agrupamento k -means	6
2.2.3	Agrupamento hierárquico	6
2.2.4	Métricas de atribuição de agrupamento	7
2.3	Redução de dimensão	7
2.3.1	Análise de componente principal	7
2.3.2	Análise de componente independente	7
3	Aprendizado profundo	8
3.1	Redes neurais	8
3.2	Redes neurais convolucionais	9
3.3	Redes neurais recorrentes	9
3.4	Aprendizado e controle reforçado	9

4	Dicas e truques de aprendizado de máquina	10
4.1	Métricas de classificação	10
4.2	Métricas de regressão	11
4.3	Seleção de modelo	11
4.4	Diagnóstico	12
5	Revisão	13
5.1	Probabilidades e Estatística	13
5.2	Álgebra Linear e Cálculo	15
5.2.1	Notações gerais	15
5.2.2	Operações de matriz	15
5.2.3	Propriedades da matriz	16
5.2.4	Cálculo com matriz	17

Traduzido por Gabriel Fonseca, Leticia Portella, Flavio Clesio e Fernando Santos

1 Aprendizado supervisionado

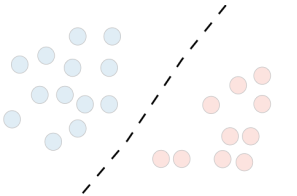
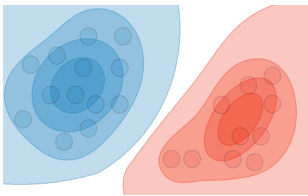
1.1 Introdução ao Aprendizado Supervisionado

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

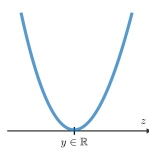
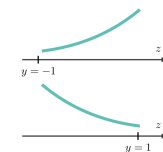
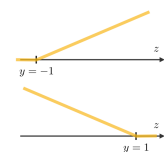
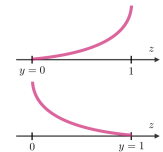
□ **Tipos de modelo** – Os diferentes modelos estão resumidos na tabela abaixo:

	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

1.2 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.

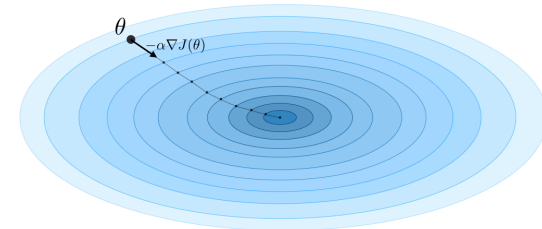
Quadrático	Logística	Hinge	Entropia cruzada
$\frac{1}{2}(y - z)^2$	$\log(1 + \exp(-yz))$	$\max(0, 1 - yz)$	$-\left[y \log(z) + (1 - y) \log(1 - z)\right]$
			
Regressão linear	Regressão logística	SVM	Rede neural

□ **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:

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

1.3 Modelos lineares

1.3.1 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:

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

1.3.2 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:

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

1.3.3 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.

1.4 Máquinas de Vetores de Suporte

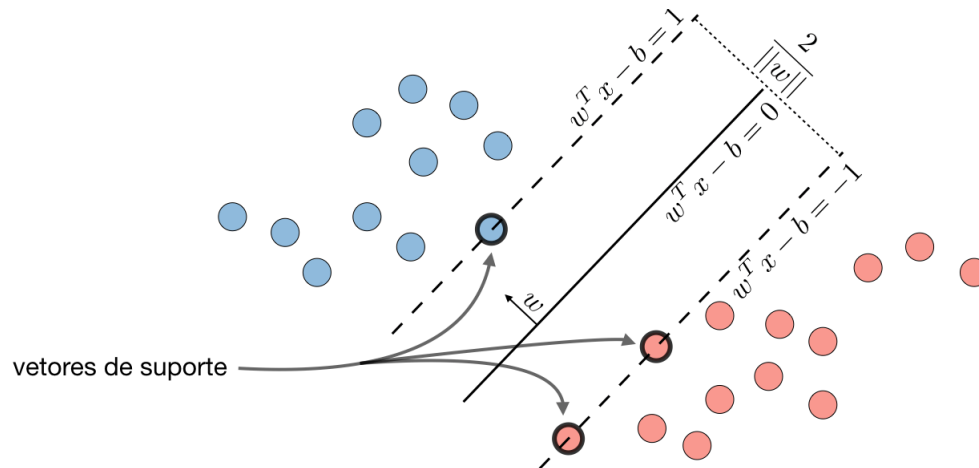
O objetivo das máquinas de vetores de suporte (support vector machines) é encontrar a linha que maximiza a distância mínima até a linha.

□ **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 $w^T x - b = 0$.

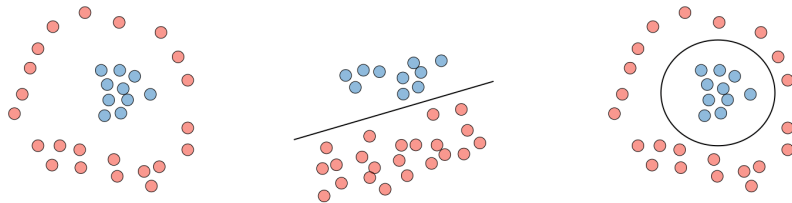
□ **Perda de Hinge** – A perda de articulação é usada na configuração das máquinas de vetores de suporte (SVMs) e é definida como:

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

□ **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 $L(w, b)$ 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.

1.5 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.

1.5.1 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:

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

1.5.2 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 $k \in \{0, 1\}, l \in [1, L]$

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

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

1.6 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 aleatórios 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.

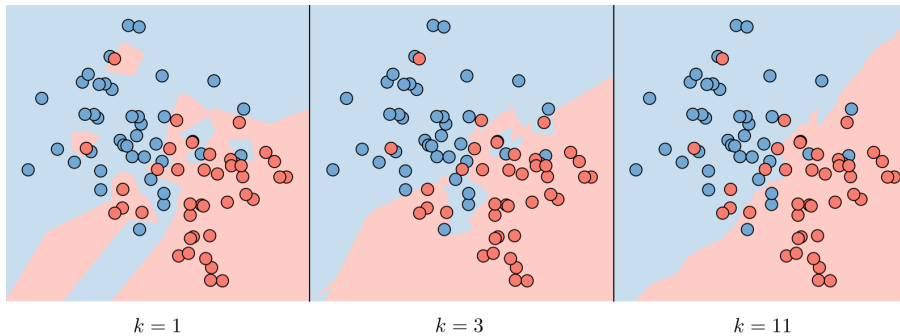
□ **Boosting** – A ideia dos métodos de boosting é combinar vários tipo de aprendizes fracos (*weak learners*) para formar um mais forte. Os principais tipos estão resumidos na tabela abaixo:

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

1.7 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.

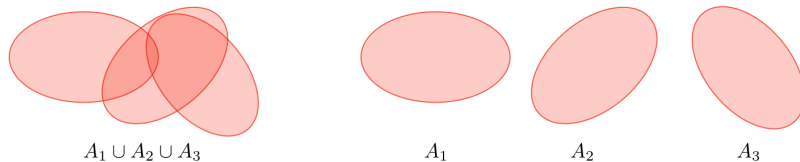
Observação: Quanto maior o parâmetro k , maior o viés, e quanto menor o parâmetro k , maior a variância.



1.8 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:

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

□ **Teorema da fronteira superior** – Seja \mathcal{H} uma classe 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:

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

2 Aprendizado não supervisionado

2.1 Introdução ao aprendizado não supervisionado

□ **Motivação** – O objetivo do aprendizado não supervisionado (*unsupervised learning*) é encontrar padrões em dados sem rótulo $\{x^{(1)}, \dots, x^{(m)}\}$.

□ **Desigualdade de Jensen** – Seja f um função convexa e X uma variável aleatória. Temos a seguinte desigualdade:

$$E[f(X)] \geq f(E[X])$$

2.2 Agrupamento

2.2.1 Maximização de expectativa

□ **Variáveis latentes** – Variáveis latentes são variáveis escondidas/não observadas que dificultam problemas de estimativa, e são geralmente indicadas por z . Aqui estão as mais comuns configurações onde há variáveis latentes:

Configuração	Variável latente z	$x z$	Comentários
Mistura de k gaussianos	Multinomial(ϕ)	$\mathcal{N}(\mu_j, \Sigma_j)$	$\mu_j \in \mathbb{R}^n, \phi \in \mathbb{R}^k$
Análise de fator	$\mathcal{N}(0, I)$	$\mathcal{N}(\mu + \Lambda z, \psi)$	$\mu_j \in \mathbb{R}^n$

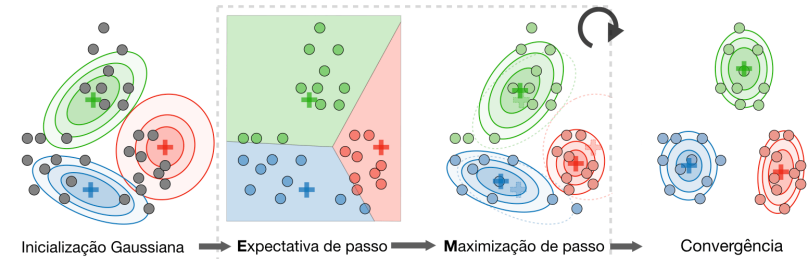
□ **Algoritmo** – O algoritmo de maximização de expectativa (*EM - Expectation-Maximization*) fornece um método eficiente para estimar o parâmetro θ através da probabilidade máxima estimada ao construir repetidamente uma fronteira inferior na probabilidade (E-step) e otimizar essa fronteira inferior (M-step) como a seguir:

- **E-step:** Avalia a probabilidade posterior $Q_i(z^{(i)})$ na qual cada ponto de dado $x^{(i)}$ veio de um grupo particular $z^{(i)}$ como a seguir:

$$Q_i(z^{(i)}) = P(z^{(i)}|x^{(i)}; \theta)$$

- **M-step:** Usa as probabilidades posteriores $Q_i(z^{(i)})$ como grupo específico de pesos nos pontos de dado $x^{(i)}$ para separadamente estimar cada modelo do grupo como a seguir:

$$\theta_i = \arg\max_{\theta} \sum_i \int_{z^{(i)}} Q_i(z^{(i)}) \log \left(\frac{P(x^{(i)}, z^{(i)}; \theta)}{Q_i(z^{(i)})} \right) dz^{(i)}$$

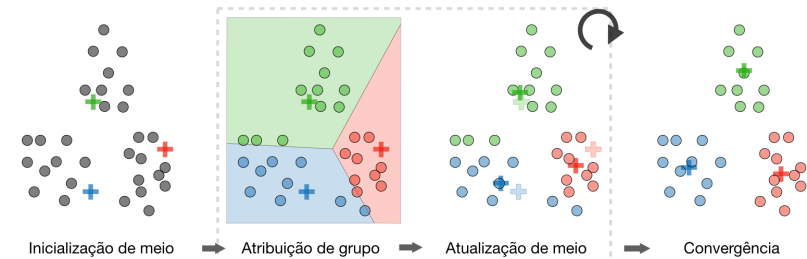


2.2.2 Agrupamento k -means

Nós indicamos $c^{(i)}$ o grupo de pontos de dados i e μ_j o centro do grupo j .

□ **Algoritmo** – Após aleatoriamente inicializar os centróides do grupo $\mu_1, \mu_2, \dots, \mu_k \in \mathbb{R}^n$, o algoritmo k -means repete os seguintes passos até a convergência:

$$c^{(i)} = \arg \min_j \|x^{(i)} - \mu_j\|^2 \quad \text{e} \quad \mu_j = \frac{\sum_{i=1}^m 1_{\{c^{(i)}=j\}} x^{(i)}}{\sum_{i=1}^m 1_{\{c^{(i)}=j\}}}$$



□ **Função de distorção** – A fim de ver se o algoritmo converge, nós olhamos para a função de distorção (*distortion function*) definida como se segue:

$$J(c, \mu) = \sum_{i=1}^m \|x^{(i)} - \mu_{c^{(i)}}\|^2$$

2.2.3 Agrupamento hierárquico

□ **Algoritmo** – É um algoritmo de agrupamento com uma abordagem hierárquica aglomerativa que constrói grupos aninhados de uma maneira sucessiva.

□ **Tipos** – Existem diferentes tipos de algoritmos de agrupamento hierárquico que objetivam a otimizar funções objetivas diferentes, os quais estão resumidos na tabela abaixo:

Ligação de vigia	Ligação média	Ligação completa
Minimizar distância dentro do grupo	Minimizar a distância média entre pares de grupos	Minimizar a distância máxima entre pares de grupos

2.2.4 Métricas de atribuição de agrupamento

Em uma configuração de aprendizado não supervisionado, é geralmente difícil acessar o desempenho de um modelo desde que não temos rótulos de verdade como era o caso na configuração de aprendizado supervisionado.

□ **Coefficiente de silhueta** – Ao indicar a e b a distância média entre uma amostra e todos os outros pontos na mesma classe, e entre uma amostra e todos os outros pontos no grupo mais próximo, o coeficiente de silhueta s para uma única amostra é definido como se segue:

$$s = \frac{b - a}{\max(a, b)}$$

□ **Índice Calinski-Harabaz** – Indicando por k o número de grupos, B_k e W_k as matrizes de dispersão entre e dentro do agrupamento respectivamente definidos como:

$$B_k = \sum_{j=1}^k n_{c(i)} (\mu_{c(i)} - \mu)(\mu_{c(i)} - \mu)^T, \quad W_k = \sum_{i=1}^m (x^{(i)} - \mu_{c(i)})(x^{(i)} - \mu_{c(i)})^T$$

o índice Calinski-Harabaz $s(k)$ indica quão bem um modelo de agrupamento define o seu grupo, tal que maior a pontuação, mais denso e bem separado os grupos estão. Ele é definido como a seguir:

$$s(k) = \frac{\text{Tr}(B_k)}{\text{Tr}(W_k)} \times \frac{N - k}{k - 1}$$

2.3 Redução de dimensão

2.3.1 Análise de componente principal

É uma técnica de redução de dimensão que encontra direções de maximização de variância em que projetam os dados.

□ **Autovalor, autovetor** – Dada uma matriz $A \in \mathbb{R}^{n \times n}$, λ é dito ser um autovalor de A se existe um vetor $z \in \mathbb{R}^n \setminus \{0\}$, chamado autovetor, tal que temos:

$$Az = \lambda z$$

□ **Teorema espectral** – Seja $A \in \mathbb{R}^{n \times n}$. Se A é simétrica, então A é diagonalizável por uma matriz ortogonal $U \in \mathbb{R}^{n \times n}$. Denotando $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$, temos:

$$\exists \Lambda \text{ diagonal, } A = U\Lambda U^T$$

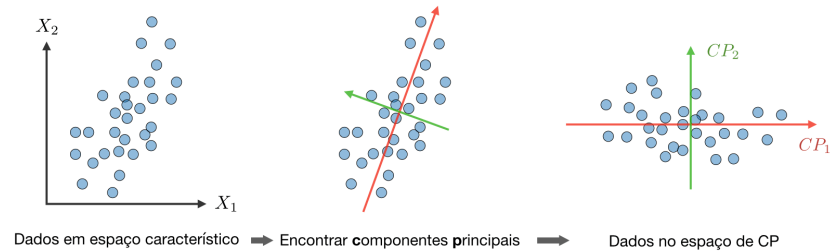
Observação: o autovetor associado com o maior autovalor é chamado de autovetor principal da matriz A .

□ **Algoritmo** – O processo de Análise de Componente Principal (*PCA - Principal Component Analysis*) é uma técnica de redução de dimensão que projeta os dados em dimensões k ao maximizar a variância dos dados como se segue:

- Etapa 1: Normalizar os dados para ter uma média de 0 e um desvio padrão de 1.

$$x_j^{(i)} \leftarrow \frac{x_j^{(i)} - \mu_j}{\sigma_j} \quad \text{ou} \quad \mu_j = \frac{1}{m} \sum_{i=1}^m x_j^{(i)} \quad \text{e} \quad \sigma_j^2 = \frac{1}{m} \sum_{i=1}^m (x_j^{(i)} - \mu_j)^2$$

- Etapa 2: Computar $\Sigma = \frac{1}{m} \sum_{i=1}^m x^{(i)} x^{(i)T} \in \mathbb{R}^{n \times n}$, a qual é simétrica com autovalores reais.
- Etapa 3: Computar $u_1, \dots, u_k \in \mathbb{R}^n$ os k principais autovetores ortogonais de Σ , i.e. os autovetores ortogonais dos k maiores autovalores.
- Etapa 4: Projetar os dados em $\text{span}_{\mathbb{R}}(u_1, \dots, u_k)$.
Esse processo maximiza a variância entre todos os espaços dimensionais k .



2.3.2 Análise de componente independente

É uma técnica que pretende encontrar as fontes de geração subjacente.

□ **Suposições** – Nós assumimos que nosso dado x foi gerado por um vetor fonte dimensional n $s = (s_1, \dots, s_n)$, onde si são variáveis aleatórias independentes, através de uma matriz A misturada e não singular como se segue:

$$x = As$$

O objetivo é encontrar a matriz $W = A^{-1}$ não misturada.

□ **Algoritmo Bell e Sejnowski ICA** – Esse algoritmo encontra a matriz W não misturada pelas seguintes etapas abaixo:

- Escreva a probabilidade de $x = As = W^{-1}s$ como:

$$p(x) = \prod_{i=1}^n p_s(w_i^T x) \cdot |W|$$

- Escreva o logaritmo da probabilidade dado o nosso dado treinado $\{x^{(i)}, i \in [1, m]\}$ e indicando g a função sigmoide como:

$$l(W) = \sum_{i=1}^m \left(\sum_{j=1}^n \log \left(g'(w_j^T x^{(i)}) \right) + \log |W| \right)$$

Portanto, a regra de aprendizagem do gradiente ascendente estocástico é tal que para cada exemplo de treinamento $x^{(i)}$, nós atualizamos W como a seguir:

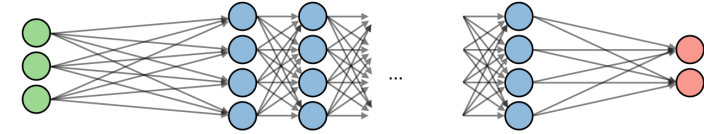
$$W \leftarrow W + \alpha \left(\begin{pmatrix} 1 - 2g(w_1^T x^{(i)}) \\ 1 - 2g(w_2^T x^{(i)}) \\ \vdots \\ 1 - 2g(w_n^T x^{(i)}) \end{pmatrix} x^{(i)T} + (W^T)^{-1} \right)$$

3 Aprendizado profundo

3.1 Redes neurais

Redes neurais são uma classe de modelos que são construídos com camadas. Os tipos de redes neurais comumente utilizadas incluem redes neurais convolucionais e recorrentes.

□ **Arquitetura** – O vocabulário em torno das arquiteturas de redes neurais é descrito na figura abaixo:



Camada de entrada Camada escondida 1 ... Camada escondida k Camada de saída

Dado que i é a i -ésima camada da rede e j a j -ésima unidade escondida da camada, nós temos:

$$z_j^{[i]} = w_j^{[i]T} x + b_j^{[i]}$$

onde é definido que w , b , z , o peso, o viés e a saída respectivamente.

□ **Função de ativação** – Funções de ativação são usadas no fim de uma unidade escondida para introduzir complexidades não lineares ao modelo. Aqui estão as mais comuns:

Sigmoide	Tanh	ReLU	Leaky ReLU
$g(z) = \frac{1}{1 + e^{-z}}$	$g(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$	$g(z) = \max(0, z)$	$g(z) = \max(\epsilon z, z)$ with $\epsilon \ll 1$

□ **Perda de entropia cruzada** – No contexto de redes neurais, a perda de entropia cruzada $L(z, y)$ é comumente utilizada e é definida como se segue:

$$L(z, y) = - \left[y \log(z) + (1 - y) \log(1 - z) \right]$$

□ **Taxa de aprendizado** – A taxa de aprendizado, frequentemente indicada por α ou às vezes η , indica a que ritmo os pesos são atualizados. Isso pode ser fixado ou alterado de modo adaptativo. O método atual mais popular é chamado Adam, o qual é um método que adapta a taxa de aprendizado.

□ **Retropropagação** – Retropropagação é um método para atualizar os pesos em uma rede neural levando em conta a saída atual e a saída desejada. A derivada relativa ao peso w é computada utilizando a regra da cadeia e é da seguinte forma:

$$\frac{\partial L(z,y)}{\partial w} = \frac{\partial L(z,y)}{\partial a} \times \frac{\partial a}{\partial z} \times \frac{\partial z}{\partial w}$$

Como resultado, o peso é atualizado como se segue:

$$w \leftarrow w - \eta \frac{\partial L(z,y)}{\partial w}$$

□ **Atualizando os pesos** – Em uma rede neural, os pesos são atualizados como a seguir:

- Passo 1 : Pegue um conjunto de dados de treinamento.
- Passo 2 : Realize propagação para frente a fim de obter a perda correspondente.
- Passo 3 : Retropropague a perda para obter os gradientes.
- Passo 4 : Use os gradientes para atualizar os pesos da rede.

□ **Abandono** – Abandono (*dropout*) é uma técnica que pretende prevenir o sobreajuste dos dados de treinamento abandonando unidades na rede neural. Na prática, neurônios são ou abandonados com a probabilidade p ou mantidos com a probabilidade $1 - p$.

3.2 Redes neurais convolucionais

□ **Requisito de camada convolucional** – Dado que W é o tamanho do volume de entrada, F o tamanho dos neurônios da camada convolucional, P a quantidade de preenchimento de zeros, então o número de neurônios N que cabem em um dado volume é tal que:

$$N = \frac{W - F + 2P}{S} + 1$$

□ **Normalização em lote** – É uma etapa de hiperparâmetro γ, β que normaliza o lote $\{x_i\}$. Dado que μ_B, σ_B^2 são a média e a variância daquilo que queremos conectar ao lote, isso é feito como se segue:

$$x_i \leftarrow \gamma \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} + \beta$$

Isso é usualmente feito após de uma totalmente conectada/camada concolucional e antes de uma camada não linear e objetiva permitir maiores taxas de aprendizado e reduzir a forte dependência na inicialização.

3.3 Redes neurais recorrentes

□ **Tipos de portas** – Aqui estão os diferentes tipos de portas (*gates*) que encontramos em uma rede neural recorrente típica:

Porta de entrada	Porta esquecida	Porta	Porta de saída
Escrever?	Apagar?	Quanto escrever?	Quanto revelar?

□ **LSTM** – Uma rede de memória de longo prazo (LSTM) é um tipo de modelo de rede neural recorrente (RNN) que evita o problema do desaparecimento da gradiente adicionando portas de ‘esquecimento’.

3.4 Aprendizado e controle reforçado

O objetivo do aprendizado reforçado é fazer um agente aprender como evoluir em um ambiente.

□ **Processos de decisão de Markov** – Um processo de decisão de Markov (MDP) é uma tupla de 5 elementos $(S, \mathcal{A}, \{P_{sa}\}, \gamma, R)$ onde:

- S é o conjunto de estados
- \mathcal{A} é conjunto de ações
- $\{P_{sa}\}$ são as probabilidade de transição de estado para $s \in S$ e $a \in \mathcal{A}$
- $\gamma \in [0, 1[$ é o fator de desconto
- $R : S \times \mathcal{A} \rightarrow \mathbb{R}$ ou $R : S \rightarrow \mathbb{R}$ é a função de recompensa que o algoritmo quer maximizar

□ **Diretriz** – Uma diretriz π é a função $\pi : S \rightarrow \mathcal{A}$ que mapeia os estados a ações.

Observação: dizemos que executamos uma dada diretriz π se dado um estado s nós tomamos a ação $a = \pi(s)$.

□ **Função de valor** – Para uma dada diretriz π e um dado estado s , nós definimos a função de valor V^π como a seguir:

$$V^\pi(s) = E \left[R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \dots | s_0 = s, \pi \right]$$

□ **Equação de Bellman** – As equações de Bellman ótimas caracterizam a função de valor V^{π^*} para a ótima diretriz π^* :

$$V^{\pi^*}(s) = R(s) + \max_{a \in \mathcal{A}} \gamma \sum_{s' \in S} P_{sa}(s') V^{\pi^*}(s')$$

Observação: definimos que a ótima diretriz π^ para um dado estado s é tal que:*

$$\pi^*(s) = \operatorname{argmax}_{a \in \mathcal{A}} \sum_{s' \in S} P_{sa}(s') V^*(s')$$

□ **Algoritmo de iteração de valor** – O algoritmo de iteração de valor é realizado em duas etapas:

- Inicializamos o valor:

$$V_0(s) = 0$$

- Iteramos o valor baseado nos valores anteriores:

$$V_{i+1}(s) = R(s) + \max_{a \in \mathcal{A}} \left[\sum_{s' \in S} \gamma P_{sa}(s') V_i(s') \right]$$

□ **Máxima probabilidade estimada** – A máxima probabilidade estimada para o estado de transição de probabilidades como se segue:

$$P_{sa}(s') = \frac{\text{\#vezes que a ação } a \text{ entrou no estado } s \text{ e obteve } s'}{\text{\#vezes que a ação } a \text{ entrou no estado } s}$$

□ **Aprendizado Q** – Aprendizado Q é um modelo livre de estimativa de Q, o qual é feito como se segue:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[R(s,a,s') + \gamma \max_{a'} Q(s',a') - Q(s,a) \right]$$

4 Dicas e truques de aprendizado de máquina

4.1 Métricas de classificação

Em um contexto de classificação binária, essas são as principais métricas que são importantes acompanhar para avaliar a desempenho do modelo.

□ **Matriz de confusão** – A matriz de confusão (*confusion matrix*) é usada para termos uma cenário mais completa quando estamos avaliando o desempenho de um modelo. Ela é definida conforme a seguir:

		Classe prevista	
		+	-
Classe real	+	TP True Positives	FN False Negatives Type II error
	-	FP False Positives Type I error	TN True Negatives

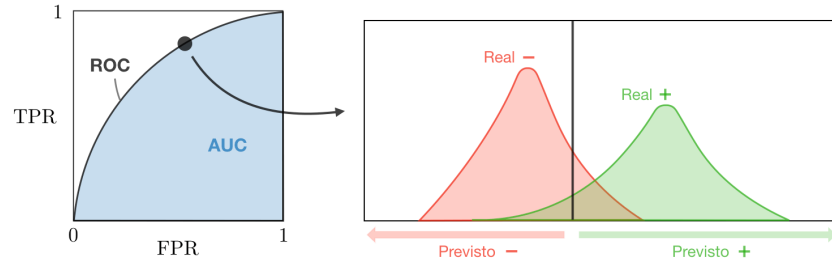
□ **Principais métricas** – As seguintes métricas são comumente usadas para avaliar o desempenho de modelos de classificação:

Métrica	Fórmula	Interpretação
Acurácia	$\frac{TP + TN}{TP + TN + FP + FN}$	Desempenho geral do modelo
Precisão	$\frac{TP}{TP + FP}$	Quão precisas são as predições positivas
Revocação Sensibilidade	$\frac{TP}{TP + FN}$	Cobertura da amostra positiva real
Specificity	$\frac{TN}{TN + FP}$	Cobertura da amostra negativa real
F1 score	$\frac{2TP}{2TP + FP + FN}$	Métrica híbrida útil para classes desequilibradas

□ **ROC** – A curva de operação do receptor, também chamada ROC (*Receiver Operating Characteristic*), é a área de TPR versus FPR variando o limiar. Essa métricas estão resumidas na tabela abaixo:

Métrica	Fórmula	Equivalente
True Positive Rate TPR	$\frac{TP}{TP + FN}$	Revocação, sensibilidade
False Positive Rate FPR	$\frac{FP}{TN + FP}$	1-specificity

□ **AUC** – A área sob a curva de operação de recebimento, também chamado AUC ou AUROC, é a área abaixo da ROC como mostrada na figura a seguir:



Uma vez que o modelo é escolhido, ele é treinado no conjunto inteiro de dados e testado no conjunto de dados de testes não vistos. São representados na figura abaixo:



□ **Validação cruzada** – Validação cruzada, também chamada de CV (*Cross-Validation*), é um método utilizado para selecionar um modelo que não depende muito do conjunto de treinamento inicial. Os diferentes tipos estão resumidos na tabela abaixo:

4.2 Métricas de regressão

□ **Métricas básicas** – Dado um modelo de regressão f , as seguintes métricas são geralmente utilizadas para avaliar o desempenho do modelo:

S. total dos quadrados	S. explicada dos quadrados	S. residual dos quadrados
$SS_{\text{tot}} = \sum_{i=1}^m (y_i - \bar{y})^2$	$SS_{\text{reg}} = \sum_{i=1}^m (f(x_i) - \bar{y})^2$	$SS_{\text{res}} = \sum_{i=1}^m (y_i - f(x_i))^2$

□ **Coefficiente de determinação** – O coeficiente de determinação, frequentemente escrito como R^2 ou r^2 , fornece uma medida de quão bem os resultados observados são replicados pelo modelo e é definido como se segue:

$$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}$$

□ **Principais métricas** – As seguintes métricas são comumente utilizadas para avaliar o desempenho de modelos de regressão, levando em conta o número de variáveis n que eles consideram:

Cp de Mallow	AIC	BIC	R^2 ajustado
$\frac{SS_{\text{res}} + 2(n+1)\hat{\sigma}^2}{m}$	$2[(n+2) - \log(L)]$	$\log(m)(n+2) - 2\log(L)$	$1 - \frac{(1-R^2)(m-1)}{m-n-1}$

onde L é a probabilidade e $\hat{\sigma}^2$ é uma estimativa da variância associada com cada resposta.

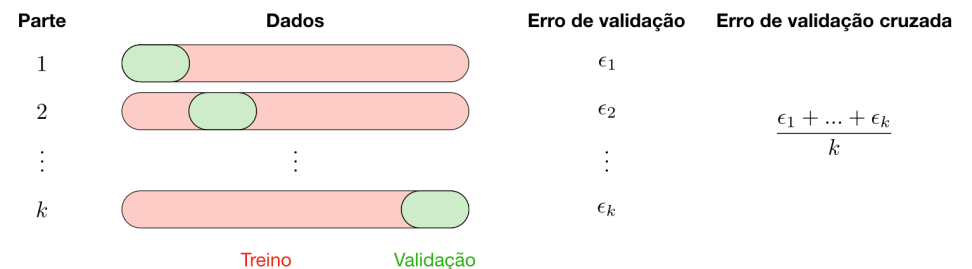
4.3 Seleção de modelo

□ **Vocabulário** – Ao selecionar um modelo, nós consideramos 3 diferentes partes dos dados que seguimos conforme a seguir:

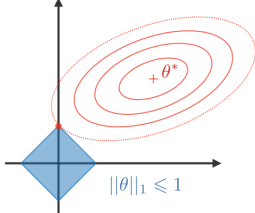
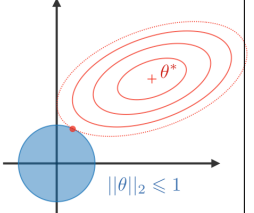
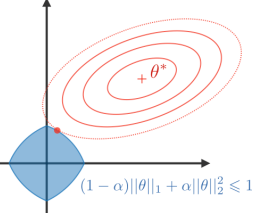
Conjunto de treino	Conjunto de validação	Conjunto de teste
- Modelo é treinado - Geralmente 80% do conjunto de dados	- Modelo é avaliado - Geralmente 20% do conjunto de dados Também chamado de hold-out	- Modelo fornece previsões - Dados não vistos

k -fold	Leave- p -out
- Treino em $k-1$ partes e teste sobre o restante - Geralmente $k=5$ ou 10	- Treino em $n-p$ observações e teste sobre p restantes - Caso $p=1$ é chamado <i>leave-one-out</i>

O método mais comumente usado é chamado k -fold cross validation e divide os dados de treinamento em k partes enquanto treina o modelo nas outras $k-1$ partes, todas estas em k vezes. O erro é então calculado sobre as k partes e é chamado erro de validação cruzada (*cross-validation error*).



□ **Regularização** – O procedimento de regularização (*regularization*) visa evitar que o modelo sobreajuste os dados e portanto lide com os problemas de alta variância. A tabela a seguir resume os diferentes tipos de técnicas de regularização comumente utilizadas:

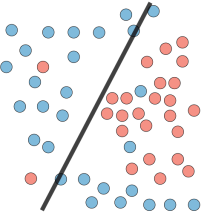
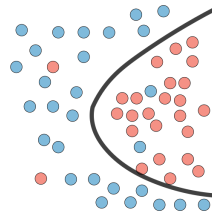
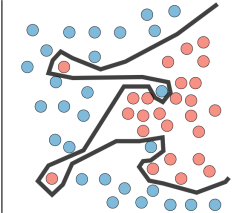
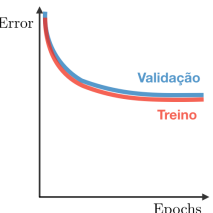
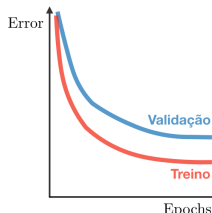
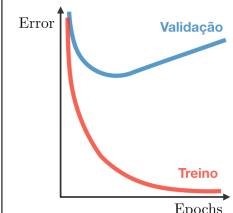
LASSO	Ridge	Elastic Net
<ul style="list-style-type: none"> - Diminui coeficientes para 0 - Bom para seleção de variáveis 	Faz o coeficiente menor	Balço entre seleção de variáveis e coeficientes pequenos
		
$\dots + \lambda \ \theta\ _1$ $\lambda \in \mathbb{R}$	$\dots + \lambda \ \theta\ _2^2$ $\lambda \in \mathbb{R}$	$\dots + \lambda \left[(1 - \alpha) \ \theta\ _1 + \alpha \ \theta\ _2^2 \right]$ $\lambda \in \mathbb{R}, \alpha \in [0, 1]$

4.4 Diagnóstico

□ **Viés** – O viés (*bias*) de um modelo é a diferença entre a predição esperada e o modelo correto que nós tentamos prever para determinados pontos de dados.

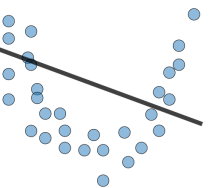
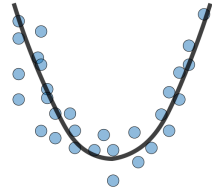
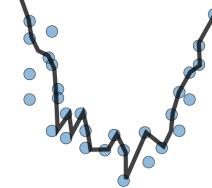
□ **Variância** – A variância (*variance*) de um modelo é a variabilidade da previsão do modelo para determinados pontos de dados.

□ **Balço viés/variância** – Quanto mais simples o modelo, maior o viés e, quanto mais complexo o modelo, maior a variância.

	Classificação		
			
Deep Learning			
Remédios	<ul style="list-style-type: none"> - Modelo de complexificação - Adicionar mais recursos - Treinar mais 		<ul style="list-style-type: none"> - Executar a regularização - Obter mais dados

□ **Análise de erro** – Análise de erro (*error analysis*) é a análise da causa raiz da diferença no desempenho entre o modelo atual e o modelo perfeito.

□ **Análise ablativa** – Ablative analysis (*ablative analysis*) é a análise da causa raiz da diferença no desempenho entre o modelo atual e o modelo base.

	Underfitting	Just right	Overfitting
Sintomas	<ul style="list-style-type: none"> - Erro de treinamento elevado - Erro de treinamento próximo ao erro de teste - Viés elevado 	<ul style="list-style-type: none"> - Erro de treinamento ligeiramente menor que erro de teste 	<ul style="list-style-type: none"> - Erro de treinamento muito baixo - Erro de treinamento muito menor que erro de teste - Alta variância
Regressão			

5 Revisão

5.1 Probabilidades e Estatística

Introdução a Probabilidade e Combinatória

□ **Espaço amostral** – O conjunto de todos os resultados possíveis é chamado de espaço amostral do experimento e é denotado por S .

□ **Evento** – Qualquer subconjunto E do espaço amostral é chamado de evento. Isso é, um evento é um conjunto de possíveis resultados do experimento. Se o resultado do experimento está contido em E , então é dito que o evento ocorreu.

□ **Axiomas de probabilidade** – Para cada evento E , denotamos $P(E)$ a probabilidade do evento E ocorrer.

$$(1) \quad 0 \leq P(E) \leq 1 \quad (2) \quad P(S) = 1 \quad (3) \quad P\left(\bigcup_{i=1}^n E_i\right) = \sum_{i=1}^n P(E_i)$$

□ **Permutação** – A permutação é um arranjo de r objetos de um conjunto de n objetos, em uma determinada ordem. O número desses arranjos é dado por $P(n, r)$, definido como:

$$P(n, r) = \frac{n!}{(n-r)!}$$

□ **Combinação** – A combinação de um arranjo de r objetos de um conjunto de n objetos, onde a ordem não importa. O número desses arranjos é dado por $C(n, r)$, definido como:

$$C(n, r) = \frac{P(n, r)}{r!} = \frac{n!}{r!(n-r)!}$$

Observação: dado que $0 \leq r \leq n$, então temos que $P(n, r) \geq C(n, r)$.

Probabilidade Condicional

□ **Regra de Bayes** – Para eventos A e B tal que $P(B) > 0$, temos que:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Observação: temos que $P(A \cap B) = P(A)P(B|A) = P(A|B)P(B)$.

□ **Partição** – Dado que $\{A_i, i \in [1, n]\}$ seja tal que para todo i , $A_i \neq \emptyset$. Dizemos que $\{A_i\}$ é uma partição se temos:

$$\forall i \neq j, A_i \cap A_j = \emptyset \quad \text{e} \quad \bigcup_{i=1}^n A_i = S$$

Observação: para qualquer evento B no espaço amostral temos que $P(B) = \sum_{i=1}^n P(B|A_i)P(A_i)$.

□ **Extensão da regra de Bayes** – Seja $\{A_i, i \in [1, n]\}$ uma partição do espaço amostral. Temos que:

$$P(A_k|B) = \frac{P(B|A_k)P(A_k)}{\sum_{i=1}^n P(B|A_i)P(A_i)}$$

□ **Independência** – Dois eventos A e B são independentes se e apenas se tivermos:

$$P(A \cap B) = P(A)P(B)$$

Variável aleatória

□ **Variável aleatória** – Uma variável aleatória, normalmente denominada X , é uma função que mapeia todo elemento em um espaço amostral para uma linha verdadeira.

□ **Função de distribuição cumulativa (CDF)** – A função de distribuição cumulativa F , que é monotonicamente não decrescente e é tal que

$$\lim_{x \rightarrow -\infty} F(x) = 0 \quad \text{e} \quad \lim_{x \rightarrow +\infty} F(x) = 1$$

é definida como:

$$F(x) = P(X \leq x)$$

Lembrete: temos que $P(a < X \leq b) = F(b) - F(a)$.

□ **Função densidade de probabilidade (PDF)** – A função densidade de probabilidade f é a probabilidade de que X assumia valores entre duas realizações adjacentes da variável aleatória.

□ **Relações envolvendo a PDF e a CDF** – Aqui estão as propriedades mais importantes que se deve conhecer dos casos discretos (D) e contínuos (C).

Caso	CDF F	PDF f	Propriedades da PDF
(D)	$F(x) = \sum_{x_i \leq x} P(X = x_i)$	$f(x_j) = P(X = x_j)$	$0 \leq f(x_j) \leq 1$ e $\sum_j f(x_j) = 1$
(C)	$F(x) = \int_{-\infty}^x f(y)dy$	$f(x) = \frac{dF}{dx}$	$f(x) \geq 0$ e $\int_{-\infty}^{+\infty} f(x)dx = 1$

□ **Variância** – A variância de uma variável aleatória, normalmente denominada $\text{Var}(X)$ ou σ^2 , é a medida do espalhamento da sua função de distribuição. Ela é determinada por:

$$\text{Var}(X) = E[(X - E[X])^2] = E[X^2] - E[X]^2$$

□ **Desvio padrão** – O desvio padrão de uma variável aleatória, normalmente denominado σ , é a medida do espalhamento da sua função de distribuição que é compatível com a unidade da variável aleatória. Ele é determinado por:

$$\sigma = \sqrt{\text{Var}(X)}$$

□ **Expectativas e Momentos da Distribuição** – Aqui estão as expressões do valor esperado $E[X]$, do valor esperado generalizado $E[g(X)]$, do k -ésimo momento $E[X^k]$ e função característica $\psi(\omega)$ para os casos discretos e contínuos:

Caso	$E[X]$	$E[g(X)]$	$E[X^k]$	$\psi(\omega)$
(D)	$\sum_{i=1}^n x_i f(x_i)$	$\sum_{i=1}^n g(x_i) f(x_i)$	$\sum_{i=1}^n x_i^k f(x_i)$	$\sum_{i=1}^n f(x_i) e^{i\omega x_i}$
(C)	$\int_{-\infty}^{+\infty} x f(x) dx$	$\int_{-\infty}^{+\infty} g(x) f(x) dx$	$\int_{-\infty}^{+\infty} x^k f(x) dx$	$\int_{-\infty}^{+\infty} f(x) e^{i\omega x} dx$

Remarque: on a $e^{i\omega x} = \cos(\omega x) + i \sin(\omega x)$.

□ **Transformação das variáveis aleatórias** – Sejam as variáveis X e Y ligadas por alguma função. Ao denotador f_X e f_Y para as funções de distribuição de X e de Y respectivamente, temos que:

$$f_Y(y) = f_X(x) \left| \frac{dx}{dy} \right|$$

□ **Regra integral de Leibniz** – Seja g uma função de x e possivelmente de c , e a, b fronteiras que podem depender de c . Temos que:

$$\frac{\partial}{\partial c} \left(\int_a^b g(x) dx \right) = \frac{\partial b}{\partial c} \cdot g(b) - \frac{\partial a}{\partial c} \cdot g(a) + \int_a^b \frac{\partial g}{\partial c}(x) dx$$

□ **Desigualdade de Chebyshev** – Seja X uma variável aleatória com valor esperado μ . Para $k, \sigma > 0$, temos a seguinte desigualdade:

$$P(|X - \mu| \geq k\sigma) \leq \frac{1}{k^2}$$

Variáveis aleatórias distribuídas conjuntamente

□ **Densidade condicional** – A densidade condicional de X com respeito a Y , normalmente denotada como $f_{X|Y}$, é definida como:

$$f_{X|Y}(x) = \frac{f_{XY}(x, y)}{f_Y(y)}$$

□ **Independência** – Duas variáveis aleatórias X e Y são ditas independentes se:

$$f_{XY}(x, y) = f_X(x) f_Y(y)$$

□ **Densidade marginal e distribuição cumulativa** – A partir da função de probabilidade de densidade conjunta f_{XY} , temos que:

Caso	Densidade marginal	Função cumulativa
(D)	$f_X(x_i) = \sum_j f_{XY}(x_i, y_j)$	$F_{XY}(x, y) = \sum_{x_i \leq x} \sum_{y_j \leq y} f_{XY}(x_i, y_j)$
(C)	$f_X(x) = \int_{-\infty}^{+\infty} f_{XY}(x, y) dy$	$F_{XY}(x, y) = \int_{-\infty}^x \int_{-\infty}^y f_{XY}(x', y') dx' dy'$

□ **Covariância** – Definimos covariância de duas variáveis aleatórias X e Y , que chamamos de σ_{XY}^2 ou mais comumente de $\text{Cov}(X, Y)$, como:

$$\text{Cov}(X, Y) \triangleq \sigma_{XY}^2 = E[(X - \mu_X)(Y - \mu_Y)] = E[XY] - \mu_X \mu_Y$$

□ **Correlação** – Dado que σ_X, σ_Y são os desvios padrão de X e Y , definimos a correlação entre as variáveis aleatórias X e Y , denominada ρ_{XY} , como:

$$\rho_{XY} = \frac{\sigma_{XY}^2}{\sigma_X \sigma_Y}$$

Observação 1: é definido que para qualquer variáveis aleatórias X, Y temos que $\rho_{XY} \in [-1, 1]$.
Observação 2: Se X e Y são independentes, então $\rho_{XY} = 0$.

□ **Distribuições principais** – Aqui estão as principais distribuições que não devem ser esquecidas:

Tipo	Distribuição	PDF	$\psi(\omega)$	$E[X]$	$\text{Var}(X)$
(D)	$X \sim \mathcal{B}(n, p)$ Binomial	$P(X = x) = \binom{n}{x} p^x q^{n-x}$ $x \in \llbracket 0, n \rrbracket$	$(pe^{i\omega} + q)^n$	np	npq
	$X \sim \text{Po}(\mu)$ Poisson	$P(X = x) = \frac{\mu^x}{x!} e^{-\mu}$ $x \in \mathbb{N}$	$e^{\mu(e^{i\omega} - 1)}$	μ	μ
(C)	$X \sim \mathcal{U}(a, b)$ Uniform	$f(x) = \frac{1}{b-a}$ $x \in [a, b]$	$\frac{e^{i\omega b} - e^{i\omega a}}{(b-a)i\omega}$	$\frac{a+b}{2}$	$\frac{(b-a)^2}{12}$
	$X \sim \mathcal{N}(\mu, \sigma)$ Gaussian	$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$ $x \in \mathbb{R}$	$e^{i\omega\mu - \frac{1}{2}\omega^2\sigma^2}$	μ	σ^2
	$X \sim \text{Exp}(\lambda)$ Exponential	$f(x) = \lambda e^{-\lambda x}$ $x \in \mathbb{R}_+$	$\frac{1}{1 - i\omega/\lambda}$	$\frac{1}{\lambda}$	$\frac{1}{\lambda^2}$

Estimativa de parâmetro

□ **Amostra aleatória** – Uma amostra aleatória é uma coleção de n variáveis aleatórias X_1, \dots, X_n que são independentes e igualmente distribuídas com X .

□ **Estimador** – Um estimador é uma função dos dados que é usada para inferir o valor de um parâmetro desconhecido em um modelo estatístico.

□ **Viés** – O viés de um estimador $\hat{\theta}$ é definido como a diferença entre o valor esperado da distribuição de $\hat{\theta}$ e o seu real valor, i.e.:

$$\text{Bias}(\hat{\theta}) = E[\hat{\theta}] - \theta$$

Observação: um estimador é chamado de imparcial (unbiased) quando $E[\hat{\theta}] = \theta$.

□ **Média da amostra** – A média da amostra de uma amostra aleatória é usada para estimar a verdadeira média μ de uma distribuição, e é denominada \bar{X} e é definida como:

Observação: a média da amostra é imparcial, i.e $E[\bar{X}] = \mu$.

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i$$

□ **Amostra da variância** – A amostra da variância de uma amostra aleatória é usada para estimar a verdadeira variância σ^2 da distribuição, e é normalmente denominada s^2 ou $\hat{\sigma}^2$ e definida por:

Observação: a variância da amostra é imparcial, i.e $E[s^2] = \sigma^2$.

$$s^2 = \hat{\sigma}^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2$$

□ **Teorema do Limite Central** – Dado que temos uma amostra aleatória X_1, \dots, X_n seguindo uma determinada distribuição com a média μ e a variância σ^2 , temos que:

$$\bar{X} \underset{n \rightarrow +\infty}{\sim} \mathcal{N}\left(\mu, \frac{\sigma}{\sqrt{n}}\right)$$

5.2 Álgebra Linear e Cálculo

5.2.1 Notações gerais

□ **Vetor** – Indicamos por $x \in \mathbb{R}^n$ um vetor com n elementos, onde $x_i \in \mathbb{R}$ é o $i^{\text{ésimo}}$ elemento:

$$x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \in \mathbb{R}^n$$

□ **Matriz** – Indicamos por $A \in \mathbb{R}^{m \times n}$ uma matriz com m linhas e n colunas, onde $A_{i,j} \in \mathbb{R}$ é o elementos localizado na $i^{\text{ésima}}$ linha e $j^{\text{ésima}}$ coluna:

$$A = \begin{pmatrix} A_{1,1} & \cdots & A_{1,n} \\ \vdots & & \vdots \\ A_{m,1} & \cdots & A_{m,n} \end{pmatrix} \in \mathbb{R}^{m \times n}$$

Observação: o vetor x defindo acima pode ser visto como uma matriz $n \times 1$ e é mais particularmente chamado de vetor coluna.

□ **Matriz identidade** – A matriz identidade $I \in \mathbb{R}^{n \times n}$ é uma matriz quadrada com uns na sua diagonal e zeros nas demais posições:

$$I = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix}$$

Observação: para todas as matrizes $A \in \mathbb{R}^{n \times n}$, nós temos $A \times I = I \times A = A$.

□ **Matriz diagonal** – Uma matriz diagonal $D \in \mathbb{R}^{n \times n}$ é uma matriz quadrada com valores não nulos na sua diagonal e zeros nas demais posições:

$$D = \begin{pmatrix} d_1 & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & d_n \end{pmatrix}$$

Observação: nós também indicamos D como $\text{diag}(d_1, \dots, d_n)$.

5.2.2 Operações de matriz

□ **Vetor-vetor** – Há dois tipos de produtos vetoriais:

- Produto interno: para $x, y \in \mathbb{R}^n$, temos:

$$x^T y = \sum_{i=1}^n x_i y_i \in \mathbb{R}$$

- Produto tensorial: para $x \in \mathbb{R}^m, y \in \mathbb{R}^n$, temos :

$$xy^T = \begin{pmatrix} x_1 y_1 & \cdots & x_1 y_n \\ \vdots & & \vdots \\ x_m y_1 & \cdots & x_m y_n \end{pmatrix} \in \mathbb{R}^{m \times n}$$

□ **Matriz-vetor** – O produto de uma matriz $A \in \mathbb{R}^{m \times n}$ e um vetor $x \in \mathbb{R}^n$ é um vetor de tamanho \mathbb{R}^m , de tal modo que:

$$Ax = \begin{pmatrix} a_{r,1}^T x \\ \vdots \\ a_{r,m}^T x \end{pmatrix} = \sum_{i=1}^n a_{c,i} x_i \in \mathbb{R}^m$$

onde $a_{r,i}^T$ são vetores linhas e $a_{c,j}$ vetores colunas de A , e x_i são os elementos de x .

□ **Matriz-matriz** – O produto das matrizes $A \in \mathbb{R}^{m \times n}$ e $B \in \mathbb{R}^{n \times p}$ é uma matriz de tamanho $\mathbb{R}^{m \times p}$, de tal modo que:

$$AB = \begin{pmatrix} a_{r,1}^T b_{c,1} & \cdots & a_{r,1}^T b_{c,p} \\ \vdots & & \vdots \\ a_{r,m}^T b_{c,1} & \cdots & a_{r,m}^T b_{c,p} \end{pmatrix} = \sum_{i=1}^n a_{c,i} b_{r,i}^T \in \mathbb{R}^{m \times p}$$

onde $a_{r,i}^T, b_{r,i}^T$ são vetores linhas e $a_{c,j}, b_{c,j}$ vetores colunas de A e B respectivamente.

□ **Transposta** – A transposta de uma matriz $A \in \mathbb{R}^{m \times n}$, indicada por A^T , é tal que suas linhas são trocadas por suas colunas:

$$\forall i, j, \quad A_{i,j}^T = A_{j,i}$$

Observação: para matrizes A, B , temos $(AB)^T = B^T A^T$.

□ **Inversa** – A inversa de uma matriz quadrada inversível A é indicada por A^{-1} e é uma matriz única de tal modo que:

$$AA^{-1} = A^{-1}A = I$$

Observação: nem todas as matrizes quadrada são inversíveis. Também, para matrizes A, B , temos $(AB)^{-1} = B^{-1}A^{-1}$.

□ **Traço** – O traço de uma matriz quadrada A , indicado por $\text{tr}(A)$, é a soma dos elementos de sua diagonal:

$$\text{tr}(A) = \sum_{i=1}^n A_{i,i}$$

Observação: para matrizes A, B , temos $\text{tr}(A^T) = \text{tr}(A)$ e $\text{tr}(AB) = \text{tr}(BA)$.

□ **Determinante** – A determinante de uma matriz quadrada $A \in \mathbb{R}^{n \times n}$, indicada por $|A|$ ou $\det(A)$ é expressa recursivamente em termos de $A_{\setminus i, \setminus j}$, a qual é a matriz A sem a sua $i^{\text{ésima}}$ linha e $j^{\text{ésima}}$ coluna, como se segue:

$$\det(A) = |A| = \sum_{j=1}^n (-1)^{i+j} A_{i,j} |A_{\setminus i, \setminus j}|$$

Observação: A é inversível se e somente se $|A| \neq 0$. Além disso, $|AB| = |A||B|$ e $|A^T| = |A|$.

5.2.3 Propriedades da matriz

□ **Decomposição simétrica** – Uma dada matriz A pode ser expressa em termos de suas partes simétricas e assimétricas como a seguir:

$$A = \underbrace{\frac{A + A^T}{2}}_{\text{Simétrica}} + \underbrace{\frac{A - A^T}{2}}_{\text{Assimétrica}}$$

□ **Norma** – Uma norma é uma função $N : V \rightarrow [0, +\infty[$ onde V é um vetor espaço, e de tal modo que para todo $x, y \in V$, nós temos:

- $N(x + y) \leq N(x) + N(y)$
- $N(ax) = |a|N(x)$ para a escalar
- se $N(x) = 0$, então $x = 0$

Para $x \in V$, as mais comumente utilizadas normas estão resumidas na tabela abaixo:

Norma	Notação	Definição	Caso de uso
Manhattan, L^1	$\ x\ _1$	$\sum_{i=1}^n x_i $	LASSO
Euclidean, L^2	$\ x\ _2$	$\sqrt{\sum_{i=1}^n x_i^2}$	Ridge
p -norme, L^p	$\ x\ _p$	$\left(\sum_{i=1}^n x_i^p\right)^{\frac{1}{p}}$	Inégalité de Hölder
Infini, L^∞	$\ x\ _\infty$	$\max_i x_i $	Convergence uniforme

□ **Dependência linear** – Um conjunto de vetores é dito ser linearmente dependente se um dos vetores no conjunto puder ser definido como uma combinação linear dos demais.

Observação: se nenhum vetor puder ser escrito dessa maneira, então os vetores são ditos serem linearmente independentes.

□ **Posto da matriz** – O posto de uma dada matriz A é indicada por $\text{rank}(A)$ e é a dimensão do vetor espaço gerado por suas colunas. Isso é equivalente ao número máximo de colunas linearmente independentes de A .

□ **Matriz positiva semi-definida** – Uma matriz $A \in \mathbb{R}^{n \times n}$ é positiva semi-definida (PSD) e é indicada por $A \succeq 0$ se tivermos:

$$A = A^T \quad \text{e} \quad \forall x \in \mathbb{R}^n, \quad x^T A x \geq 0$$

Observação: de forma similar, uma matriz A é dita ser positiva definida, e é indicada por $A \succ 0$ se ela é uma matriz (PSD) que satisfaz todo vetor x não nulo, $x^T A x > 0$.

□ **Autovalor, autovetor** – Dada uma matriz $A \in \mathbb{R}^{n \times n}$, λ é dita ser um autovalor de A se existe um vetor $z \in \mathbb{R}^n \setminus \{0\}$, chamado autovetor, nós temos:

$$Az = \lambda z$$

□ **Teorema spectral** – Seja $A \in \mathbb{R}^{n \times n}$. Se A é simétrica, então A é diagonalizável por uma matriz ortogonal $U \in \mathbb{R}^{n \times n}$. Indicando $\Lambda = \text{diag}(\lambda_1, \dots, \lambda_n)$, nós temos:

$$\exists \Lambda \text{ diagonal, } A = U \Lambda U^T$$

□ **Decomposição em valor singular** – Para uma dada matriz A de dimensões $m \times n$, a decomposição em valor singular (SVD) é uma técnica de fatorização que garante a existência de matrizes unitária $U \in \mathbb{R}^{m \times m}$, diagonal $\Sigma \in \mathbb{R}^{m \times n}$ e unitária $V \in \mathbb{R}^{n \times n}$, de tal modo que:

$$A = U \Sigma V^T$$

5.2.4 Cálculo com matriz

□ **Gradiente** – Seja $f : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}$ uma função e $A \in \mathbb{R}^{m \times n}$ uma matriz. O gradiente de f a respeito a A é a matriz $m \times n$, indicada por $\nabla_A f(A)$, de tal modo que:

$$\left(\nabla_A f(A) \right)_{i,j} = \frac{\partial f(A)}{\partial A_{i,j}}$$

Observação: o gradiente de f é somente definido quando f é uma função que retorna um escalar.

□ **Hessiano** – Seja $f : \mathbb{R}^n \rightarrow \mathbb{R}$ uma função e $x \in \mathbb{R}^n$ um vetor. O hessiano de f a respeito a x uma matriz simétrica $n \times n$, indicada por $\nabla_x^2 f(x)$, de tal modo que:

$$\left(\nabla_x^2 f(x) \right)_{i,j} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}$$

Observação: o hessiano de f é somente definido quando f é uma função que retorna um escalar.

□ **Operações com gradiente** – Para matrizes A, B, C , as seguintes propriedades de gradiente valem a pena ter em mente:

$$\nabla_A \text{tr}(AB) = B^T$$

$$\nabla_{A^T} f(A) = (\nabla_A f(A))^T$$

$$\nabla_A \text{tr}(ABA^T C) = CAB + C^T AB^T$$

$$\nabla_A |A| = |A| (A^{-1})^T$$