

# Dicas VIP: Aprendizado não supervisionado

Afshine AMIDI e Shervine AMIDI

13 de Outubro de 2018

Traduzido por Gabriel Fonseca.

## 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:

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

## 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( $\phi$ )	$\mathcal{N}(\mu_j, \Sigma_j)$	$\mu_j \in \mathbb{R}^n, \phi \in \mathbb{R}^k$
Análise de fator	$\mathcal{N}(0, I)$	$\mathcal{N}(\mu + \Lambda z, \psi)$	$\mu_j \in \mathbb{R}^n$

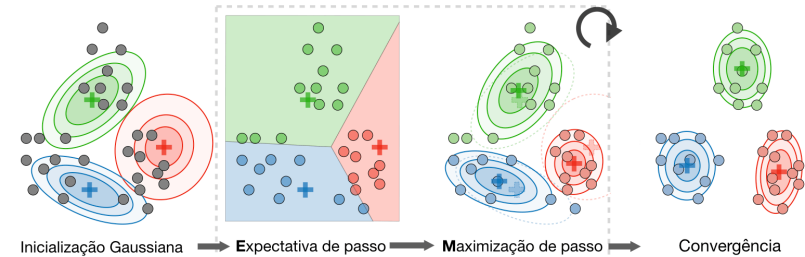
□ **Algoritmo** – O algoritmo de maximização de expectativa (*EM - Expectation-Maximization*) fornece um método eficiente para estimar o parâmetro  $\theta$  através da probabilidade máxima estimada ao construir repetidamente uma fronteira inferior na probabilidade (E-step) e otimizar essa fronteira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

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

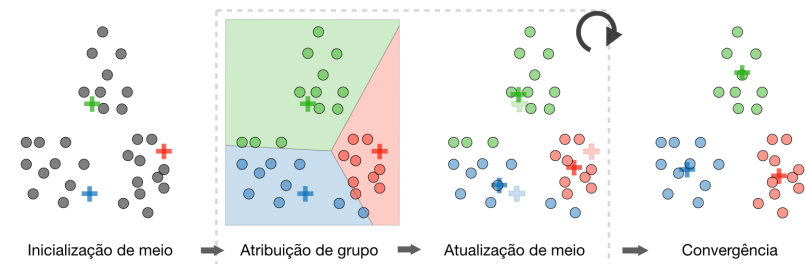


## Agrupamento $k$ -means

Nós indicamos  $c^{(i)}$  o grupo de pontos de dados  $i$  e  $\mu_j$  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:

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



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

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

## 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

### 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 definida 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:

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

### 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  $A \in \mathbb{R}^{n \times n}$ ,  $\lambda$  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:

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

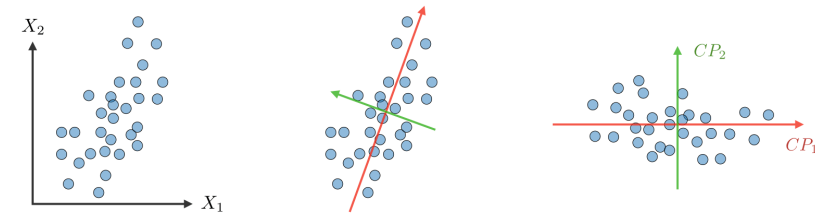
*Observação: o autovetor associado com o maior autovalor é chamado de autovetor principal da matriz  $A$ .*

□ **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  $\Sigma$ , 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 espaços dimensionais  $k$ .



Dados em espaço característico → Encontrar componentes principais → Dados no espaço de CP

### 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:

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