UNIVERSIDADE FEDERAL DE SANTA CATARINA
CAMPUS FLORIANÓPOLIS
BACHARELADO EM MATEMÁTICA E COMPUTAÇÃO
CIENTÍFICA

Marcos Vinicios Ferreira Rosa

Modelos de Misturas Gaussianas

Florianópolis
2022

Marcos Vinicios Ferreira Rosa

Modelos de Misturas Gaussianas

Trabalho de Conclusão de Curso de Graduação em Bacharelado em Matemática e Computação Científica do Campus Florianópolis da Universidade Federal de Santa Catarina para a obtenção do título de Bacharel em Matemática.
Orientador: Prof. Edson Cilos Vargas Júnior, Dr.

Florianópolis
2022

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Ferreira Rosa, Marcos Vinicios
Modelos de Misturas Gaussianas / Marcos Vinicios Ferreira Rosa ; orientador, Dr. Prof. Edson Cilos Vargas Júnior, 2022.
55 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Centro de Ciências Físicas e Matemáticas, Graduação em Matemática e Computação Científica, Florianópolis, 2022.

Inclui referências.

1. Matemática e Computação Científica. 2. Modelos de Misturas Gaussianas. 3. Distribuições Normais. 4. Machine Learning. I. Cilos Vargas Júnior, Dr. Prof. Edson. II. Universidade Federal de Santa Catarina. Graduação em Matemática e Computação Científica. III. Título.

Marcos Vinicios Ferreira Rosa

Modelos de Misturas Gaussianas

Este Trabalho Conclusão de Curso foi julgado adequado para obtenção do Título de Bacharel em Matemática e aprovado em sua forma final pelo Curso de Bacharelado em Matemática e Computação Científica.

Florianópolis, [15] de [12] de [2022].

Prof. Silvia Martini de Holanda, Dra.
Coordenadora do Curso

Banca Examinadora:

Prof. Edson Cilos Vargas Júnior, Dr.
Orientador
Instituição UFSC

Prof. Vinícius Viana Luiz Albani, Dr.
Avaliador
Instituição UFSC

Prof. Vladimir Pestov, Dr.
Avaliador
Instituição uOttawa / UFPB

Este trabalho é dedicado aos meus queridos pais e aos amigos
que me apoiaram durante essa jornada.

AGRADECIMENTOS

Agradeço a minha familia por todo apoio durante esses anos, sem eles não seria possível ter me dedicado integralmente aos estudos e aproveitar esse precioso tempo na graduação.

Agradeço aos meus amigos, que tornaram a jornada mais leve com todas as conversas, conselhos e risadas. E especialmente a Cru Campus por me ensinar o que é viver em uma comunidade acolhedora.

Agradeço ao meu orientador, professor Edson, por toda ajuda, paciência e sabedoria compartilhada nesse último ano e pela dedicação e excelência como professor e orientador.

Agradeço aos professores do curso. E também ao PET de matemática, em especial, aos professores do pré vestibular Gauss de 2017, por me mostrarem a beleza da matemática e me inspirarem a cursar Matemática.

Por fim, agradeço a Deus pois sem Ele tudo seria em vão.

RESUMO

O Modelo de misturas gaussianas (*Gaussian Mixture Models - GMM*) é baseado em uma função de densidade de probabilidade paramétrica representada por uma soma de componentes de distribuições gaussianas. Tais modelos são comumente utilizados no contexto de *Machine Learning* para representar um conjunto de dados por meio dessas distribuições e aplicar previsões a partir dessa nova representação. No presente trabalho é desenvolvido um estudo sobre a construção matemática do GMM. Inicialmente, apresenta-se a fundamentação teórica com ferramentas matemáticas que embasam o desenvolvimento do GMM, a definição formal e a prova dos principais teoremas envolvendo a atualização dos parâmetros do modelo.

Palavras-chave: Modelos de Misturas Gaussianas. Distribuições Normais. Machine Learning.

LISTA DE FIGURAS

SUMÁRIO

1 CONCEITOS DE PROBABILIDADE

A teoria da probabilidade visa construir uma estrutura matemática que modele e descreva resultados aleatórios de eventos ou experimentos [9]. Diferentes abordagens podem ser vistas ao desenvolver essa teoria, uma abordagem muito comum é a frequentista que considera que a chance, ou a probabilidade, de um evento ocorrer é o limite de sua frequência relativa dado muitas tentativas de um experimento, ou seja, $P(E) = \lim_{N\to\infty} \frac{n(E)}{N}$, onde $P(E)$ é a probabilidade do evento E ocorrer, $n(E)$ é o número de vezes que E ocorre e N é o número total de experimentos feitos.

Uma outra construção possível, e a que seguiremos neste estudo, é a abordagem axiomática de Kolmogorov [11] que considera como verdade um conjunto de afirmações inicial sobre probabilidade para desenvolver a teoria. A seguir são apresentadas alguns conceitos precedentes a definição desses axiomas.

1.1 ESPAÇO DE PROBABILIDADE

Definição 1.1 (Espaço amostral Ω)**.** *O espaço amostral é o conjunto de todos os resultados possíveis de um experimento aleatório e normalmente é denotado por Ω. Por "resultado possível"entende-se um resultado elementar e indivisível do experimento.*

Suponha o seguinte experimento: lançar um dado equilibrado de 6 faces e observar o número da face superior. Neste caso, o espaço amostral é $\Omega = \{1, 2, 3, 4, 5, 6\}$ pois esses são os únicos possíveis resultados do experimento.

Nem sempre é simples definir o espaço amostral. Veja no caso de escolher uma pessoa ao acaso e medir sua altura, o espaço amostral poderia ser o intervalo $(0, 4)$? Ou talvez $(0, \infty)$? Por mais que no

segundo caso o conjunto possua valores impossíveis como mil ou um milhão, ainda poderia ser um candidato para Ω. Dentro da teoria, o importante é que Ω contenha todos os resultados possíveis. Por esta razão, vamos supor:

- todo resultado possível corresponde um, e somente um, ponto $\omega \in \Omega$,
- resultados distintos correspondem a pontos distintos em Ω, isto é, ω não pode representar mais de um resultado.

Veja também que ao realizar um experimento, é possível observar certos resultados desses experimentos que podem ocorrer ou não. Como, por exemplo, no experimento *lançar um dado e observar o resultado da face superior*, resultados possíveis são:

A = "observar um número par",
B = "observar um número maior que 4".

Observe que cada um destes resultados podem ser relacionados a subconjuntos de Ω, como $A = \{2, 4, 6\}$ ou $B = \{4, 5, 6\}$, tais conjuntos são chamados de eventos. Formalmente, define-se evento por:

Definição 1.2 (Evento)**.** *Seja Ω o espaço amostral do experimento. Definiremos portanto que todo subconjunto $A \subset \Omega$ será chamado evento.*

Definição 1.3. *Seja Ω o espaço amostral e A um evento associado a um experimento, isto é, um evento que seguramente irá ocorrer ou não sempre que for acontecer o experimento. Suponha ω um resultado do experimento, define-se:*

- *se A ocorre diz-se que ω é favorável a A.*
- *se A não ocorre, diz-se que ω não é favorável a A.*

Exemplo: Suponha o experimento *escolher, ao acaso, um ponto do disco de raio 1 centrado na origem.* Nesse caso,

$$\Omega = \{(x, y) \in \mathbb{R}^2 : x^2 + y^2 \leq 1\}.$$

Alguns eventos para esse experimento são:

A = "distância entre o ponto escolhido e a origem é $\leq \frac{1}{2}$",

B = "a primeira coordenada do ponto escolhido é maior que a segunda".

Assim, se $\omega = (x, y)$ for o resultado do experimento, temos

$$A = \left\{(x, y) \in \Omega : \sqrt{x^2 + y^2} \leq \frac{1}{2}\right\},$$

$$B = \{(x, y) \in \Omega : x > y\}.$$

Definição 1.4. *Seja Ω um conjunto não vazio. Uma classe $\mathcal{A}$ de subconjuntos de Ω é dita uma σ-álgebra de subconjuntos de Ω se satisfizer as seguintes propriedades:*

A1. $\Omega \in \mathcal{A}$,

A2. Se $A \in \mathcal{A}$, *então* $A^c \in \mathcal{A}$,

A3. Se $A_n \in \mathcal{A}$ *para* $n = 1, 2, \ldots,$ *então* $\bigcup_{n=1}^{\infty} A_n \in \mathcal{A}$.

Proposição 1.1. *Seja $\mathcal{A}$ uma σ-álgebra de subconjuntos de Ω. Então valem as seguintes propriedades:*

- $\emptyset \in \mathcal{A}$,
- $\forall n, \forall A_1, \ldots, A_n \in \mathcal{A}$, *temos* $\bigcup_{i=1}^{n} A_i \in \mathcal{A}$ *e* $\bigcap_{i=1}^{n} A_i \in \mathcal{A}$.

Esta proposição garante que uma álgebra é fechada para um número finito de aplicações das operações $\cup, \cap$ e ^{c}.

Proposição 1.2. *Seja $\mathcal{A}$ uma σ-álgebra de subconjuntos de Ω. Se $A_1, A_2, \cdots \in \mathcal{A}$ então $\bigcap_{n=1}^{\infty} A_n \in \mathcal{A}$.*

De tal proposição, pode-se dizer que uma σ-álgebra é fechada para uma quantidade enumerável de aplicações das operações $\cup$, $\cap$ e ^{c}.

Definição 1.5 (Borelianos)**.** *Seja Ω um espaço métrico qualquer. A menor família $\mathfrak{B}$ que contém todos os conjuntos abertos e é fechada com relação aos complementares e uniões de subfamílias enumeráveis, se chama a família de subconjuntos borelianos de Ω.*

Para a reta $\mathbb{R}$ a σ-álgebra de Borel é a menor σ-álgebra contendo todos os intervalos. Um boreliano é um conjunto que pode ser obtido de uma quantidade enumerável de intervalos aplicando-se as operações $\cup$, $\cap$ e ^{c} uma quantidade enumerável de vezes.

Definição 1.6. *Como mencionado no início do capítulo, o presente estudo seguirá a construção axiomática de Kolmogorov, logo abrimos mão de nos preocupar se uma probabilidade de fato existe e como defini-la para cada experimento. Admitiremos que tais probabilidades existem em uma certa σ-álgebra $\mathcal{A}$ de eventos, chamados eventos admissíveis e que cada $A \in \mathcal{A}$ está associado a um número real $P(A)$, chamado probabilidade de A, de modo que os seguintes axiomas sejam satisfeitos:*

Axioma 1.1. $P(A) \geq 0$.

Axioma 1.2. $P(\Omega) = 1$.

Axioma 1.3 (σ-aditividade)**.** *Se $A_1, A_2, \cdots \in \mathcal{A}$ são mutuamente exclusivos, então*

$$P\left(\bigcup_{n=1}^{\infty} A_n\right) = \sum_{n=1}^{\infty} P(A_n). \tag{1}$$

Definição 1.7. *Uma função P definida numa σ-álgebra $\mathcal{A}$ e satisfazendo os axiomas 1.1, 1.2 e 1.3 chama-se uma medida de probabilidade em $\mathcal{A}$ ou simplesmente uma probabilidade de $\mathcal{A}$.*

Propriedades de probabilidade: Seja P uma probabilidade de uma σ-álgebra $\mathcal{A}$. Suponha $A \in \mathcal{A}$, então as seguintes propriedades são consequência dos axiomas:

P1. $P(A^c) = 1 - P(A)$.

P2. $0 \leq P(A) \leq 1$.

P3. $A_1 \subset A_2 \Rightarrow P(A_1) \leq P(A_2)$.

P5. $P\left(\bigcup_{i=1}^{\infty} A_i\right) \leq \sum_{i=1}^{\infty} P(A_i)$.

P6. (Continuidade de probabilidade) Se $A_n \uparrow A$, então $P(A_n) \uparrow P(A)$. Se $A_n \downarrow A$, então $P(A_n) \downarrow P(A)$.

Definição 1.8 (Evento aleatório)**.** *Um evento A ao qual atribuímos uma probabilidade será chamado evento aleatório.*

Definição 1.9. *Um espaço de probabilidade é um trio $(\Omega, \mathcal{A}, P)$, onde Ω é um conjunto não vazio, $\mathcal{A}$ é uma σ-álgebra de subconjuntos de Ω e P é uma probabilidade em $\mathcal{A}$.*

Temos desenvolvido portanto o conceito de modelo probabilístico que é constituído por um conjunto não vazio de resultados possíveis dito espaço amostral, uma σ-álgebra $\mathcal{A}$ de eventos admissíveis e uma probabilidade P definida em $\mathcal{A}$. E também, o conceito de espaço de probabilidade sendo a tripla $(\Omega, \mathcal{A}, P)$ definida em (1.9).

Veja que todo modelo probabilístico é um espaço de probabilidade e reciprocamente, o espaço de probabilidade $(\Omega, \mathcal{A}, P)$ pode ser considerado um modelo para o experimento "selecionar um ponto de Ω conforme a probabilidade P". A partir desse momento, apesar de mantermos a linguagem de experimentos e eventos, tudo será estudado em espaços de probabilidade.

1.2 PROBABILIDADE CONDICIONAL

Definição 1.10. *Seja $(\Omega, \mathcal{A}, P)$ um espaço de probabilidade. Se $A, B \in \mathcal{A}$ e $P(B) > 0$, a probabilidade condicional de A dado B é definida por*

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)}. \tag{2}$$

Observe que diretamente dessa definição segue que $P(A \cap B) = P(B)P(A \mid B)$, por indução é possível provar que essa igualdade se generaliza para casos com mais de 2 conjunto, sendo assim, temos o seguinte teorema:

Teorema 1.1. *Seja $(\Omega, \mathcal{A}, P)$ um espaço de probabilidade. Então*

- $P(A \cap B) = P(A)P(B \mid A) = P(B)P(A \mid B), \forall A, B \in \mathcal{A}$.
- $P(A_1 \cap A_2 \cap \cdots \cap A_n) = P(A_1)P(A_2 \mid A_1)P(A_3 \mid A_1 \cap A_2) \ldots P(A_n \mid A_1 \cap \cdots \cap A_{n-1}), \forall A_1, \ldots, A_n \in \mathcal{A}, \forall n = 2, 3, \ldots$.

Definição 1.11. *Sejam $A_1, A_2, \ldots$ eventos aleatórios tais que $A_i \neq A_j$, para $i \neq j$ e $\bigcup A_i = \Omega$, neste caso dizemos que os A_i formam uma partição do espaço amostral Ω.*

Teorema 1.2 (Teorema da probabilidade total)**.** *Se a sequência, finita ou enumerável, de eventos aleatórios $A_1, A_2, \ldots$ formar uma partição*

de Ω*, então*

$$P(B) = \sum_i P(A_i)P(B \mid A_i), \forall B \in \mathcal{A}. \tag{3}$$

Por meio desse teorema, é possível calcular a probabilidade de A_i dada a ocorrência de B:

$$P(A_i \mid B) = \frac{P(A_i \cap B)}{P(B)} \Rightarrow P(A_i \mid B) = \frac{P(A_i)P(B \mid A_i)}{\sum_j P(A_j)P(B \mid A_j)}. \tag{4}$$

Esta é chamada da fórmula de Bayes. Ela é útil quando se sabe as probabilidades dos A_i e a probabilidade condicional de B dado A_i, mas não a probabilidade de B diretamente. Este é um caso muito comum em *machine learning*, portanto essa é uma fórmula de grande importância nesse contexto.

Em *machine learning* normalmente se está interessado em fazer inferências sobre variáveis aleatórias não observadas a partir de outras variáveis já observadas. Assumindo algum conhecimento $p(x)$ sobre uma variável aleatória x não observada e uma relação $p(y \mid x)$ entre x e uma segunda variável y que conseguimos observar. Observando y, é possível utilizar o teorema de Bayes para inferir conclusões sobre x, assim

$$p(x \mid y) = \frac{p(y \mid x)p(x)}{p(y)},$$

onde $p(x \mid y)$ é dito posteriori, $p(x)$ é priori, $p(y)$ evidência e $p(y \mid x)$ é chamada de verossimilhança.

1.3 INDEPENDÊNCIA

Definição 1.12. *Seja* $(\Omega, \mathcal{A}, P)$ *um espaço de probabilidade. Os eventos aleatórios* A *e* B *são independentes se*

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

Proposição 1.3. *A é independente de si mesmo se, e somente se,* $P(A) = 0$ *ou* $P(A) = 1$.

Proposição 1.4. *Se A e B são independentes, então A e* B^c, A^c *e B,* A^c *e* B^c *também são independentes.*

Definição 1.13. *1. Os eventos* $A_1, \ldots, A_n (n \geq 2)$ *são chamados independentes se*

$$\begin{aligned} &P(A_{i_1} \cap A_{i_2} \cap \cdots \cap A_{i_m}) = P(A_{i_1})P(A_{i_2}) \ldots P(A_{i_m}), \\ &\forall 1 \leq i_1 \leq i_2 \leq i_m \leq n, \forall m = 2, 3, \ldots, n. \end{aligned} \tag{5}$$

2 . Os eventos $A_1, A_2, A_3, \ldots$ *são independentes se* $\forall n \geq 2$, A_1, $A_2, \ldots, A_n$ *são independentes.*

3. Seja I um conjunto de índices tal que $\#I \geq 2$. *Os eventos* A_i, *onde* $i \in I$, *são independentes se* $A_{i_1}, A_{i_2}, \ldots, A_{i_m}$ *são independentes para toda combinação* $i_1, i_2, \ldots, i_m$ *de elementos de I,* $\forall m = 2, 3, \ldots$

1.4 VARIÁVEIS ALEATÓRIAS

Definição 1.14. *Uma variável aleatória X em um espaço de probabilidade* $(\Omega, \mathcal{A}, P)$ *é uma função real definida no espaço* Ω *tal que* $[X \leq x]$ *é evento aleatório para todo* $x \in \mathbb{R}$, *ou seja,* $X : \Omega \to \mathbb{R}$ *é variável aleatória se* $[X \leq x] \in \mathcal{A}, \forall x \in \mathbb{R}$. *Onde define-se* $[X \leq x] = \{\omega \in \Omega : X(\omega) \leq x\}$.

Exemplos: Escolher um ponto ao acaso em $[0, 1]$. Seja X o quadrado do valor obtido, então

$$\Omega = [0, 1]$$

e

$$X(\omega) = \omega^2.$$

Quando o resultado é um ponto no plano, pode-se considerar como o valor de um par de variáveis aleatórias, por exemplo: escolher um ponto

ao acaso no circulo unitário, neste caso sendo X e Y as coordenadas do resultado. Então

$$\Omega = \{(x, y) : x^2 + y^2 \leq 1\}$$

e, com $\omega = (x, y)$, temos $X(\omega) = x, Y(\omega) = y$ e $(X(\omega), Y(\omega)) = (x, y) = \omega$. X e Y são ditas funções coordenadas.

Definição 1.15. *A função de distribuição da variável aleatória X, representada por F_X ou simplesmente por F, é definida por*

$$F_X(x) = P(X \leq x), x \in R. \tag{6}$$

Exemplo: Uma variável aleatória que segue uma distribuição gaussiana é denotada por $X \sim \mathcal{N}(\mu, \sigma^2)$ possui função de distribuição dada por:

$$F_X(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2}, \tag{7}$$

onde $X : \Omega \to \mathbb{R}$

Proposição 1.5. *Seja X uma variável aleatória, a função de distribuição F da variável aleatória X tem as seguintes propriedades:*

- $x \leq y \Rightarrow F(x) \leq F(y)$*, ou seja, F é não decrescente.*
- $\lim_{x_n \to x^+} F(x_n) = F(x)$*, ou seja, F é contínua a direita.*
- $\lim_{x_n \to -\infty} F(x_n) = 0$ *e* $\lim_{x_n \to +\infty} F(x_n) = 1$.

Proposição 1.6. *Toda função F que satisfaz as três propriedades anteriores é uma função de distribução de alguma variável aleatória. Assim, toda F que satisfizer essa propriedade será chamada de função de distribuição.*

1.5 TIPOS DE VARIÁVEIS ALEATÓRIAS

Definição 1.16. *A variável aleatória X é discreta se toma um número finito ou enumerável de valores, ou seja, se existe um conjunto finito ou enumerável $\{x_1, x_2, \dots\} \subset \mathbb{R}$ tal que $X(\omega) \in \{x_1, x_2, \dots\}, \forall \omega \in \Omega$. A função $p(x_i) = P(X = x_i), i = 1, 2, \dots,$ é chamada função de probabilidade.*

Definição 1.17. *A variável aleatória X é absolutamente contínua se existe uma função $f(x) \geq 0$ tal que*

$$F_X(x) = \int_{-\infty}^{x} f(t)dt, \forall x \in \mathbb{R} \tag{8}$$

Neste caso, dizemos que f é a função de densidade de probabilidade de X ou simplesmente densidade de X. Observa-se que pela proposição 1.5. para qualquer função de densidade tem-se que

$$\int_{-\infty}^{+\infty} f(x)dx = 1.$$

O escopo do estudo sobre Modelos de Mistura Gaussiana utiliza apenas conceitos relacionados a variáveis contínuas, por esta razão focaremos em tais variáveis no restante da seção.

1.6 DISTRIBUIÇÃO DE UMA VARIÁVEL ALEATÓRIA

Proposição 1.7. *Se X é variável aleatória em $(\Omega, \mathcal{A}, P)$, então o evento*

$$[X \in B] = \{\omega \in \Omega : X(\omega) \in B\}$$

é um evento aleatório para todo boreliano B, isto é

$$[X \in B] \in \mathcal{A}, \forall B \in \mathfrak{B} = \sigma\text{-álgebra de Borel.}$$

Em outras palavras, o evento $[X \in B]$ é aleatório e $P(X \in B)$ está definida se existe x tal que $B = (-\infty, x]$, já que por definição $[X \leq x] \in \mathcal{A}$ para todo $x \in \mathbb{R}$

Definição 1.18. *A probabilidade P_x, definida na σ-álgebra de borel por $P_x(B) = P(X \in B)$, é chamada distribuição de X.*

Para o caso contínuo, podemos descrever a distribuição por meio da função de densidade:

Proposição 1.8. *Se X é absolutamente contínua com densidade $f(x)$, então*

$$P_X(B) = \int_B f(x)dx, \forall B \in \mathfrak{B}$$

Exemplo: A variável aleatória X possui distribuição normal padrão, denotada por $X \sim \mathcal{N}(0,1)$, se X tem densidade

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{\frac{-x^2}{2}}, x \in \mathbb{R}. \tag{9}$$

Logo

$$P_X(B) = \int_B \frac{1}{\sqrt{2\pi}} e^{\frac{-x^2}{2}}, \text{ para } B \in \mathbb{R}. \tag{10}$$

1.7 VETORES ALEATÓRIOS

Definição 1.19. *É dito um vetor aleatório, ou variável aleatória n-dimensional, um vetor $X = (X_1, \ldots, X_n)$ cujo os componentes são variáveis aleatórias todas definidas no mesmo espaço de probabilidade $(\Omega, \mathcal{A}, P)$.*

Definição 1.20. *A função de distribuição $F = F_X = F_{X_1,\ldots,X_n}$ de um vetor aleatório $X = (X_1, \ldots, X_n)$ é assim definida:*

$$F(x) = F(x_1, \ldots, x_n) = P(X_1 \leq x_1, \ldots, X_n \leq x_n)$$

para todo $(x_1, \ldots, x_n) \in \mathbb{R}^2$. Onde

$$[X_1 \leq x_1, \ldots, X_n \leq x_n] = \bigcap_{i=1}^{n} [X_i \leq x_i].$$

1.8 ESPERANÇA MATEMÁTICA E MATRIZ DE COVARIÂNCIA

Definição 1.21. *Seja X uma variável aleatória e F sua função de distribuição. A esperança de X é definida por*

$$E[X] = \int_{\mathbb{R}} x f(x) dx \tag{11}$$

quando a integral está bem definida. Sendo f(x) é a densidade de X. Caso $E[X]$ seja finita, dizemos que X é integrável.

Definição 1.22. *Sejam X e Y vetores aleatórios integráveis. Então a covariância entre X e Y é definida por*

$$Cov(X, Y) = E[(X - E[X])(Y - E[Y])], \tag{12}$$

se esta esperança existe. Se $Cov(X, Y) = 0$ dizemos que X e Y são não-correlacionadas.

Definição 1.23. *Seja $X = (X_1, \ldots, X_n)$ vetor aleatório. Definimos a matriz de covariância como*

$$Cov(X) = \begin{bmatrix} Cov(X_1, X_1) & Cov(X_1, X_2) & \cdots & Cov(X_1, X_n) \\ Cov(X_2, X_1) & Cov(X_2, X_2) & \cdots & Cov(X_2, X_n) \\ \vdots & \vdots & \ddots & \vdots \\ Cov(X_n, X_1) & Cov(X_n, X_2) & \cdots & Cov(X_n, X_n) \end{bmatrix} \tag{13}$$

Na maioria dos casos pode ser difícil, ou até mesmo impossível, calcular o valor esperado e, por consequência, a matriz de covariância de um vetor aleatório. Isto porque calcular esses valores dependem de saber como é a medida de probabilidade no espaço em questão, e isto, na prática, é extremamente difícil e inviável na grande parte dos casos. Contudo, podemos aproximar esses valores pela média e covariância empírica.

Definição 1.24. *Sejam $x^1, \ldots, x^N$ amostras de um vetor aleatório $X = (X_1, X_2, \ldots, X_M)$, ou seja, para cada $i = 1, 2, \ldots, N$, tem- se $x^i = (x_1^i, x_2^i, \ldots, x_M^i)$. Define-se o vetor médio empírico pela média aritmética das amostras que é definida por*

$$\overline{x} = \frac{1}{N} \sum_{i=1}^{N} x^i. \tag{14}$$

Definição 1.25. *A matriz de covariância empírica é uma matriz $M \times M$ definida por*

$$\Sigma = \frac{1}{N-1} \sum_{i=1}^{N} (x^i - \overline{x})^\intercal (x^i - \overline{x}). \tag{15}$$

2 CONCEITOS DE MACHINE LEARNING

Essa seção é dedicada a ilustrar como os conceitos matemáticos e estatísticos apresentados até aqui são utilizados para a construção de modelos de *machine learning*.

Vamos começar definindo, o que é *machine learning*? Segundo Arthur Samuel *machine learning é o campo de estudo que da aos computadores a habilidade de aprender sem serem explicitamente programados* [16].

Uma outra definição possível, segundo Tom Mitchell, diz que *um programa de computador aprende pelo experimento E com respeito a alguma tarefa T e uma medida de performance P, se a sua performance em T, sendo medida por P, melhora com a experiência E* [13].

Vamos ilustrar com um exemplo. Suponha que se queira construir um programa capaz de decidir se uma imagem contém uma árvore.

Idealmente este deveria se comportar como nós aprendemos: através de estrutura(s) pré-estabelecida(s) (no nosso caso, cérebro), após um período de aprendizado (que pode ser entendido como um

"ajuste"ou "adaptação"da(s) estrutura(s) aos dados) é possível compreender o conceito, no sentido de que é possível aplicar o modelo construído para novas situações, com uma boa taxa de sucesso. Da mesma forma que nós vimos um número finito de árvores e somos capazes de identificar qualquer árvore, mesmo sem ter visto todas as que existem, um bom modelo de *machine learning* será aquele capaz de generalizar o seu conjunto de dados usado para a sua construção, isto é, terá bom poder preditivo em dados semelhantes porém desconhecidos.

Se apresentarmos essa imagem para uma criança de 3 anos e perguntarmos se na imagem existe uma árvore ou não, teríamos com certeza uma reposta correta. Agora se a pergunta fosse qual é a definição de uma árvore, mesmo a alguém de 30 anos, provavelmente a reposta seria inconclusiva. Isso porque não aprendemos o que é uma árvore (e muitos outros conceitos) por estudar a sua definição biológica do que é uma árvore, mas sim olhando e observando árvores. Em outras palavras, aprendemos por meio de dados [1].

Para o caso do exemplo ilustrado, um bom modelo seria aquele que obtivesse uma boa taxa de acerto, isto é, uma boa acurácia. Por outro lado, em um contexto mais geral o sentido de "boa performance" vai depender do problema. Por exemplo, um modelo de previsão de fraude que indica todos os clientes de uma empresa como "fraude", terá 100% de acerto nas suas previsões, porém esse tipo de análise não possui valor algum para o problema. Desta forma é indispensável definir métricas relevantes para o problema para avaliar como o modelo se comporta em relação ao *ground truth*[1], quando disponível e, na sua

[1] Informações disponíveis a priori, tidas como verdadeiras ou reais, fruto da observação ou medição direta. No exemplo das árvores, o *ground truth* poderia ser uma lista com diversas imagens com ou sem árvores. As imagens com árvore(s) receberiam um rótulo positivo, enquanto que imagens sem árvore(s) teriam um rótulo negativo. A partir desse *ground truth* seria possível construir um modelo preditivo.

ausência, o "quão bem o modelo representa os dados".

2.1 DADOS E MODELOS

Nessa seção, iremos descrever dois conceitos essenciais para o aprendizado em Machine Learning que são os dados e os modelos. Ilustraremos esses conceitos com o seguinte exemplo, suponha que um banco receba milhares de pedido de análise de crédito todos os dias e que tal banco deseja automatizar esse processo de análise e aprovação. Em geral, alguém que já trabalhe nessa área e conheça os padrões de pessoas que pedem análise de crédito, saberia dizer se é uma boa ideia ou não aprovar determinadas pessoas baseado nas informações coletadas no cadastro do cliente e possivelmente de uma conversa com ele. Porém, dificilmente essa pessoa conseguiria descrever e criar uma fórmula que possa apontar claramente se o crédito deveria ser aprovado ou não.

Entretanto, esse banco já aprovou e negou diversas análises anteriores e possui muitos dados. Portanto, isso torna possível aprender por esses dados uma boa fórmula para aprovação de crédito.

Em geral, dados são tabulares, isto é, estruturados em linhas e colunas, em que cada linha pode ser vista como uma instância ou uma amostra de algum experimento e as colunas representam características dessas instâncias. Porém, existem situações mais desafiadoras em que temos um grande volume de dados sem essa estrutura tabular, também chamados de dados não estruturados. Alguns exemplos dessa situação seriam textos, vídeos e imagens de redes sociais ou sinais gerados por dispositivos da "Internet das Coisas"(IoT) [17].

Portanto, um grande desafio é encontrar uma representação apropriada para os diversos tipos de dados. Em problemas mais difíceis, encontrar uma boa representação exige um vasto conhecimento da área de estudo. Contudo, em geral, o objetivo final será representar cada

amostra dos dados como um vetor de números reais D-dimensional $\mathbf{X_n} = (\mathbf{x_n^1}, \mathbf{x_n^2}, \ldots, \mathbf{x_n^D})$. Cada uma dessas amostras, ou vetores, serão uma linha na estrutura tabular e cada entrada dos vetores refere-se a uma característica específica das amostras.

Voltando ao exemplo de aprovação de crédito, cada amostra representa o próprio cliente e cada coluna armazena uma informação desse cliente referente a análise de crédito, como salário anual, tempo na residência em que mora, empréstimos pendentes, etc. O banco, nesse caso, também mantém registros para saber se cada aprovação de crédito foi uma boa ou má ideia , tendo assim um par de (características do cliente com crédito aprovado, aprovação foi positiva ou negativa para a empresa) que permite traçar perfis de clientes bons e ruins para a aprovação.

Veja que, o conjunto de dados poderá ser representado por uma matriz $\mathbf{X}$, de ordem NXD, sendo N o número de amostras e D o número de características. Tal abordagem proporciona que boa parte do ferramental e dos conceitos de álgebra linear possam ser aplicados, sendo possível, por exemplo, definir o produto interno entre dois dados (vetores) e assim, conseguir calcular a distância entre dois pontos de dados. Permite também que tais vetores sejam manipulados para encontrar as melhores representações dos dados, como por exemplo encontrando a representação SVD da matriz X, que inicialmente representa os dados, para encontrar matrizes de projeções para espaços menores, de forma que os dados podem ser representado em dimensões menores.

Uma vez encontrada a melhor representação vetorial dos dados, parte-se para a construção de um preditor. Um preditor é uma função que fará uma inferência a respeito dos dados, ou seja, uma função que ao receber uma amostra produz um resultado baseado nas informações dessa amostra. Um caso particular são funções lineares

$$f(x) = \theta \cdot \mathbf{x} + \theta_0, \tag{16}$$

na qual θ e θ_0 são variáveis desconhecidas.

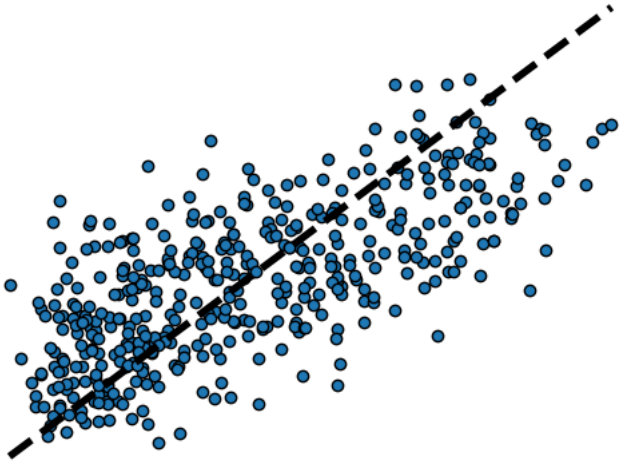

Figura 1 – Fonte: Scikit-learn

Suponha que ao invés de querer saber se o crédito deve ser aprovado ou não, o banco quer avaliar o **quanto** de crédito cada cliente pode receber e para simplificar nesse exemplo, vamos dizer que essa variável dependa apenas do salário anual de cada cliente. Assim, em 1 pode-se considerar os dados (pontos azuis), como um par (x = salário anual, y = crédito liberado), e com o modelo linear 16 queremos encontrar a reta que melhor se ajusta aos dados. Ou seja, com os dados $(x_n, y = f(x)$, para $n = 1, 2, ..., N$, deseja-se encontrar os parâmetros θ e θ_0 que minimizem uma função de perda (*loss function*). No caso da regressão linear, a função de perda mais comum para encontrar tais parâmetros é *Mean Square Error* (MSE) [7] definida por:

$$MSE(X, \theta) = \frac{1}{m} \sum_{i=1}^{m} (\theta^{\intercal} x^{(i)} - y^{(i)})^2 \tag{17}$$

Assim, tem-se um modelo que para cada "salário anual"possível na reta real em $\mathbf{x}$, tem-se um valor de crédito "ideal"a ser liberado. Veja que tal exemplo é apenas ilustrativo e, em um contexto real, a variável objetivo dependeria de diversas outras características, sendo necessária a construção do problema em N dimensões, envolvendo separações em hiperplanos e análises mais complexas.

Preditores podem ser também modelos probabilísticos, isto é, modelos descrevendo distribuições probabilísticas de funções possíveis. Neste contexto, considera-se um conjunto de dados como sendo amostras com ruídos gerados de um experimento com distribuição implícita desconhecida. E utilizando técnicas de probabilidade e estatística é possível não só ter um preditor que apresenta um resultado possível das amostras, mas também o grau de confiabilidade desse resultado.

2.2 TIPOS DE ALGORITMOS DE APRENDIZADO

A premissa básica de aprender com os dados é a utilização de um conjunto de observações para descobrir um padrão subjacente no processo que gerou os dados. Tal premissa é bastante abrangente e difícil de adaptar a uma única metodologia, e por esse motivo, diversos paradigmas de aprendizagem surgiram para lidar com as diferentes situações e suposições de cada problema [1]. Os algoritmos de *machine learning* são organizados em uma taxonomia baseada no resultado esperado dos algoritmos [2]. Algumas classificações comuns de tipos de algoritmos são:

- Aprendizado supervisionado (*supervised learning*): quando o conjunto de dados de treino possui exemplos específicos de qual é o

resultado esperado do algoritmo para determinada entrada, tais exemplos são chamados de **rótulos**. Assim, o algoritmo gera um função que mapeia as entradas a cada um dos rótulos. Um exemplo de aprendizado supervisionado é o exemplo em (16)

- aprendizado por reforço (*reinforcement learning*): Neste caso, há um agente que deve aprender a "se comportar"por interações de tentativa e erro por meio de um ambiente dinâmico. o algoritmo gera algum resultado, inicialmente aleatório, e recebe uma "nota"do quão bom ou ruim foi essa tentativa. Essa avaliação é feito por uma **politica** veja [10].

Uma outra classe importante de algoritmos é os de aprendizado não supervisionado (*unsupervised learning*). Esse aprendizado é de grande relevância para *machine learning*, e, em particular, para esse trabalho, dado que os modelos de mistura gaussiana são deste tipo.

Esses algoritmos permitem desenvolver modelos quando não há rótulos disponíveis no conjunto de dados. Na maioria dos casos, rotular amostras de um conjunto é uma tarefa onerosa e implica gastos inviáveis de tempo e dinheiro, sendo assim, esse tipo de algoritmo, quando bem aplicado, tem um grande valor para a resolução de problemas com dados.

Em geral, o aprendizado supervisionado pode ser visto como a estratégia de encontrar padrões e estruturas dos conjuntos de dados. Isso é feito buscando similaridades entre os dados e agrupando-os em classes ou subconjuntos do conjunto de dados total. Por exemplo, se você quer classificar uma pilha de livros em tópicos como comédia, drama, romance, etc., uma forma é buscar semelhanças entre as propriedades dos livros, como observar o padrão das palavras utilizadas ao longo dos livros, já que, em geral, assume-se que livros que se utilizam da mesma linguagem narrativa são da mesma classe literária.

Assim, o algoritmo agrupará livros semelhantes baseado nessas propriedades. Observe que esse tipo de algoritmo não dirá se um livro é de romance ou ficção, mas apenas que um grupo de livros estão no mesmo tópico, e partir disso, análises mais detalhadas desses grupos podem ser desenvolvidas.

2.3 MÁXIMA VEROSSIMILHANÇA

A premissa por detrás da estimativa da máxima verossimilhança é definir uma função dos parâmetros que possibilite encontrar um modelo que se ajuste bem aos dados. O problema de estimativa se concentra na função verossimilhança, ou mais precisamente no seu logaritmo negativo (log-likelihood negativo) [5].

Considere inicialmente uma distribuição de probabilidade *discreta*, através de um modelo paramétrico. O objetivo em questão é inferir os parâmetros deste modelo, dada uma observação dos dados. Em outras palavras, visa-se encontrar a distribuição que melhor se encaixa nos dados informados. Para $\mathbf{x} = (x_1, x_2, ..., x_n)$ conjunto de observações feitas (isto é, $\mathbf{x}$ está fixo no problema), e θ a variável que representa o parâmetro (possivelmente vetorial), a verosimilhança (*Likelihood*) para o caso discreto é a seguinte função na variável θ:

$$L_x(\theta) = L(\theta|x) := P(X = x, \Theta = \theta),$$

no qual $P(X, \Theta)$ é a probabilidade conjunta entre a variáveis aleatórias X e Θ. Nesse caso $L_x(\theta)$ nos diz o quão "verossímil" é observar a nossa distribuição com parâmetro θ, uma vez que conhecemos os dados $\mathbf{x}$. Dessa forma, queremos encontrar o θ que maximiza a verossimilhança, processo também chamado de Maximum Likelihood Estimation (MLE). No caso discreto, $P(X = x, \Theta = \theta) = p(x, \theta)$, isto é, $L(\theta|x)$ é exatamente $p(\mathbf{x}, \theta)$, que é a densidade de massa. Para o caso contínuo podemos entender de forma análoga, trocando a densi-

dade de massa pela densidade de probabilidade. Assim, fica motivada a seguinte definição:

Definição 2.1. *Seja $p(\mathbf{x}, \theta)$ a densidade de probabilidade conjunta entre $\mathbf{x}$ e θ, seja contínua ou discreta, podemos definir a Likelihood como sendo a* ***função*** *em θ:*

$$L_x(\theta) := p(\mathbf{x}, \theta).$$

Dessa forma, o MLE tem por objetivo encontrar o máximo: $\hat{\theta} = \max_{\theta} p(\mathbf{x}, \theta)$, sendo a distribuição de probabilidade e os dados observados conhecidos no problema.

Considere um conjunto de amostras $(\mathbf{x_1}, y_1), \ldots, (\mathbf{x_N}, y_N)$ independentes e identicamente distribuídos (i.i.d.). A independência implica que a verossimilhança de todo o conjunto de dados ($\mathcal{Y} = \{y_1, y_2, \ldots, y_N\}$ e $\mathcal{X} = \{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_N}\}$) se separa no produtório de cada amostra individual, ou seja,

$$p(\mathcal{Y} \mid \mathcal{X}, \theta) = \prod_{n=1}^{N} p(y_n \mid \mathbf{x_n}, \theta),$$

sendo que as amostras serem identicamente distribuídas significa que cada termo no produtório é da mesma distribuição e todos compartilham os mesmos parâmetros.

Normalmente é mais fácil otimizar funções que são decompostas em somatórios de funções do que em produtórios. Assim, em *machine learning*, em geral considerarmos o log-likelihood negativo

$$\mathcal{L}(\theta) = -log\ p(\mathcal{Y} \mid \mathcal{X}, \theta) = -\sum_{n=1}^{N} log\ p(y_n \mid \mathbf{x_n}, \theta).$$

Desta forma, encontrar o MLE para uma amostra i.i.d é equivalente a minimizar o log-likelihood.

Exemplo: Seguiremos com o exemplo da função linear já apresentada em 16. Considere o conjunto de dados $\mathcal{X} = ((\mathbf{x}^1, y^1), (\mathbf{x}^2, y^1),$

$\ldots, (\mathbf{x}^n, y^n))$. Por simplicidade, vamos considerar o exemplo unidimensional representado pelos pontos roxos em (2), contudo a construção segue análoga para $\mathbb{R}^n$.

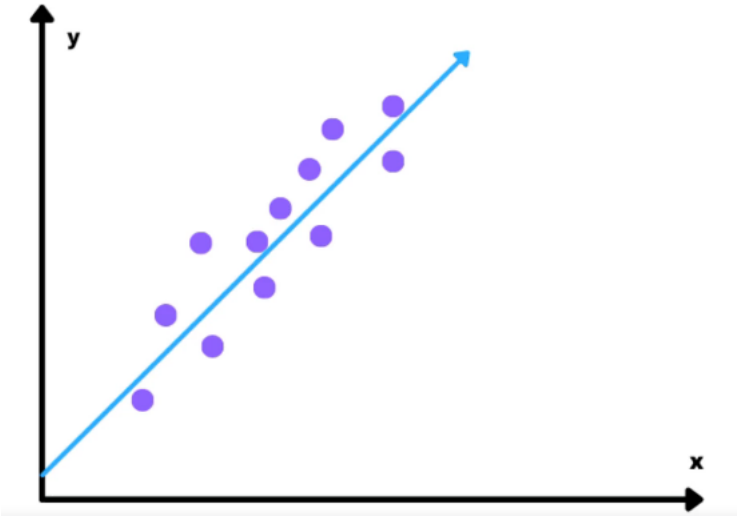

Figura 2 – Fonte: [6]

e que se quer encontrar parâmetros $\theta = (\theta_0, \theta_1, \ldots, \theta_n)$ que satisfaçam a equação $\mathbf{y}^{(n)} = \mathbf{x}^{(n)} \cdot \theta + \epsilon$. Sendo ϵ o ruído (erro) da modelagem, já que as construções de *machine learning* são à luz da probabilidade, logo dizemos que a menos de um erro, o modelo é $\mathbf{y}^{(n)} = \mathbf{x}^{(n)} \cdot \theta$ e o parâmetro ϵ está modelando esse erro.

Assumindo que para esse problema temos $p(y \mid x, \theta) = N(y \mid \mathbf{x}^\intercal \cdot \theta, \sigma^2)$, ou seja, que a probabilidade condicional dos rótulos dado as amostras é uma distribuição gaussiana. E tal suposição é equivalente a dizer que $\mathbf{y}^{(n)} = \mathbf{x}^{(n)} \cdot \theta + \epsilon$, onde $\epsilon_n \sim \mathcal{N}(0, \sigma^2)$. Isto é, é possível explicar o erro deste problema por uma distribuição normal de média zero e variância σ^2. Portanto, tendo a densidade de probabilidade

$$y \sim \mathcal{N}(y \mid \mathbf{x}^\intercal \cdot \theta, \sigma^2) = p(y \mid \mathbf{x}, \theta).$$

é possível calcular o *log-likelihood* negativo e encontrar o parâmetro θ da seguinte forma:

$$\begin{aligned}
\mathcal{L}(\theta) &= -\sum_{n=1}^{N} log\ p(y_n, \mathbf{x_n}, \theta) \ = -\sum_{n=1}^{N} log\ \mathcal{N}(y_n \mid \mathbf{x_n}^\intercal \theta, \sigma) \\
&= -\sum_{n=1}^{N} log \frac{1}{\sqrt{2\pi\sigma^2}}\ exp\left(-\frac{(y_n - \mathbf{x}_n^\intercal \theta)^2}{2\sigma^2} \right) \\
&= -\sum_{n=1}^{N} log\ exp\left(-\frac{(y_n - \mathbf{x}_n^\intercal \theta)^2}{2\sigma^2} \right) - \sum_{n=1}^{N} log \frac{1}{\sqrt{2\pi\sigma^2}} \\
&= \frac{1}{2\sigma^2} \sum_{n=1}^{N} (y_n - \mathbf{x}_n^\intercal \theta)^2 - \sum_{n=1}^{N} log \frac{1}{\sqrt{2\pi\sigma}}
\end{aligned} \tag{18}$$

Agora veja que o objetivo é encontrar o θ que minimize 18, logo minimizar

$$\mathcal{L}(\theta) = \frac{1}{2\sigma^2} \sum_{n=1}^{N} (y_n - \mathbf{x}_n^\intercal \theta)^2 - \sum_{n=1}^{N} log \frac{1}{\sqrt{2\pi\sigma}} \tag{19}$$

é equivalente a minimizar

$$\mathcal{L}(\theta) = \frac{1}{2\sigma^2} \sum_{n=1}^{N} (y_n - \mathbf{x}_n^\intercal \theta)^2 = ||y - X\theta||^2 \tag{20}$$

pois a segunda parcela de (19) não depende de θ. Por fim, temos que encontrar o θ_{ML} que resulta na máxima verossimilhança, o que é dado por

$$\theta_{ML} = arg\ \min_{\theta} \big(-ln(p(\mathcal{Y} \mid \mathcal{X}, \theta)) \big) = arg\ \min_{\theta} ||y - x \cdot \theta||^2 \tag{21}$$

2.4 AJUSTE DE MODELOS

Considere um conjunto de dados onde se está interessado em ajustar um modelo parametrizado ao dados. "Ajustar"tipicamente refere-se ao processo de aprender/otimizar os parâmetros do modelo de forma a minimizar uma função perda (*loss function*), por exemplo, *log-likelihood* negativo ou função de mérito.

A parametrização do modelo define uma classe M_θ em que são feitas operações para encontrar o melhor conjunto de parâmetros. Por exemplo, em uma configuração de regressão linear, define-se a relação entre as amostras (*inputs*) x e os rótulos y (*outputs*) sendo $y = ax + b$ onde $\theta := a, b$ são os parâmetros do modelo. Nesse caso, os parâmetros θ definem a família de funções afins, isto é, linhas retas com inclinação a e deslocadas b unidades da origem.

Assumindo que os dados são gerados de um modelo M^* desconhecido. Para um dado conjunto de amostras, otimiza-se θ de modo que M_θ esteja o mais próximo possível de M^*, onde "mais próximo"é definido por uma função objetivo que é otimizada. Após essa otimização, isto é, quando obtém-se os melhores parâmetros possíveis dado a classe M_θ, identifica-se três possíveis casos: sobreajuste (*overfitting*), subajuste (*underfitting*) e bem ajustado.

Basicamente, *overfitting* ocorre quando a classe de modelos parametrizados é muito abrangente para o modelo que gerou o conjunto de dados, i.e., a classe M_θ poderia modelar conjuntos de dados muito mais complexos. O problema de utilizar essas classes muito abrangentes e flexíveis em casos simples é porque o modelo utilizará toda a sua capacidade para diminuir o erro no conjunto de treinamento, porém se o dado gerado possui muito ruído, o modelo irá compreender esses ruídos como informações relevantes e isso causará uma má generalização do modelo para os dados desconhecidos.

Já para o *underfitting* é o oposto, ocorre quando a classe M_θ

não é rica o suficiente. Por exemplo, se os dados são gerados por uma função senoidal, mas θ parametriza apenas linhas retas, mesmo que os parâmetros sejam otimizados e seja encontrado a melhor linha reta que modela os dados, ainda não será próximo do modelo alvo que gerou os dados.

O terceiro caso é quando a classe de modelos é suficientemente abrangente. Então o modelo se ajusta bem aos dados, de forma que não é simples demais para os dados do problema, mas também não se ajusta excessivamente ao ponto de considerar os ruídos. Tais modelos são ideias para os problemas de *machine learning* por possuírem boas propriedades de generalização.

2.5 VALIDAÇÃO CRUZADA

Para entender o quão bem um algoritmo está generalizando para novos casos, é necessário de fato aplicar o modelo em dados não vistos durante o treinamento. Para alguns casos em que se quer verificar a curva de aprendizado ou apenas uma compreensão inicial da performance, é possível testar no conjunto de treino. Porém, conceitualmente, testar uma função preditora no mesmo conjunto de dados em que foi treinada é um erro metodológico, e nenhuma informação sobre o erro de generalização deve ser considerada nesses casos.

Separa-se o conjunto de dados em uma parte de treinamento e uma de teste seguindo a estratégia *holdout*. Isso para que, ao final do processo, possa se ter uma avaliação não enviesada do erro de generalização. A estratégia *holdout* permite que o algoritmo seja testado em um conjunto "externo", totalmente desconhecido do algoritmo. Isso pode ser feito utilizando ferramentas matemáticas como a inequação de Hoeffding's [8] , teoria de Vapnik-Chervonenki [18] ou técnicas de inferência estatística usuais. [12]

Além disso, todo algoritmo de machine learning possui um

conjunto de parâmetros que precisam ser predeterminados e não são aprendidos durante a fase de treino, esses são os chamados hiperparâmetros. Uma abordagem possível para encontrar a melhor combinação de hiperparâmetros é o GridSearch, onde diversas combinações de hiperâmetros são testados e é escolhida a combinação que permite o modelo performar melhor baseado em métricas relevantes para o domínio do problema.

Quando avaliando as diferentes combinações dos hiperparâmetros para os estimadores, se testarmos cada combinação diretamente no conjunto de teste, há um risco de *overfitting* no teste, pois os parâmetros podem ser ajustar até que o estimador esteja performando de forma ótima. Assim, conhecimento sobre o conjunto de teste pode "vazar"para o modelo e as métricas de avaliação não relatarão mais uma performance de generalização.

Para solucionar esse problema, utiliza-se novamente o *holdout* e além de separar uma parte do conjunto de dados para teste, é separada também uma parte para validação. Assim, é possível manter o conjunto de teste totalmente a parte até a avaliação final, apenas para se ter uma boa perspectiva do erro de generalização.

Entretanto, um novo problema pode surgir, pois separando o conjunto de dados em 3 partes, diminui-se drasticamente o número de amostras que se tem no conjunto de treino.

Uma solução para esse caso é utilizar a técnica de cross-validação (veja figura 3). Nessa técnica, o conjunto de teste (*test data*) continua sem ser utilizado para a avaliação final, porém o conjunto de treino é divido em K folhas (*folds* e a cada iteração desse algoritmo é definido uma separação (*split*) onde uma folha é selecionada como conjunto de validação (folha em azul em 3) e para cada K-ésima folha, segue o seguinte procedimento:

- O algoritmo é treinado utilizando todas as $K - 1$ folhas do

conjunto de treinamento.

- O resultado do modelo é avaliado e validado utilizando uma medida de performance na K-ésima folha que sobrou.

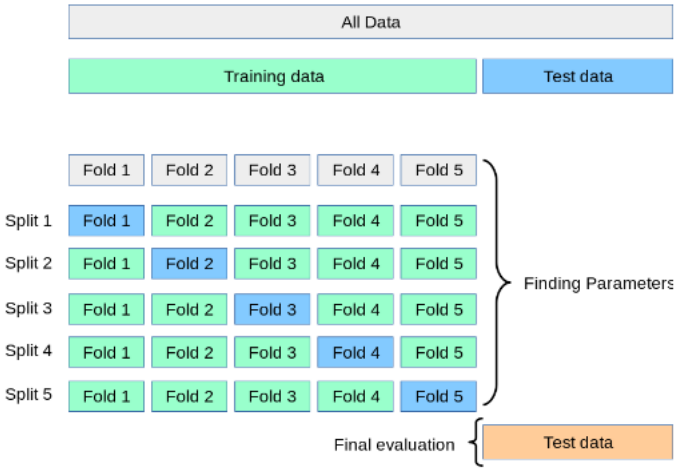

Figura 3 – Fonte: [15]

No exemplo apresentado, o conjunto é divido em 5 folhas e a cross-validação itera até que cada folha gerada tenha servido como conjunto de validação, ou seja, ocorre 5 iterações (*split* 1 até 5) e por fim, após os melhores parâmetros encontrados, o modelo final com esses parâmetros é avaliado no conjunto de teste (*test data*) para entender o quão bem o modelo está generalizando para dados não vistos antes. O valor de performance final reportado por esse método será a média dos scores em cada uma das iterações.

Esse algoritmo de cross-validação busca apenas treinar e encontrar os melhores parâmetros para do modelo, para uma busca melhor

por hiperparâmetros é necessário uma outra abordagem, como, por exemplo, o *random search* [3].

2.6 HARD ASSIGNMENT VS SOFT ASSIGNMENT

Nesta seção apresenta-se o conceito de *hard assignment* e *soft assignment*. Vamos iniciar com uma ilustração intuitiva desses conceitos. Considere um problema de classificar um dado que em uma classe A ou B. Considerando o *Hard Assignment*, esse dado pode pertencer somente a classe A ou somente a classe B. Esse dado é 100% da classe atribuida a ele. Já para o *soft assignment* o dado pode pertencer "parcialmente"a diferentes classes. Por exemplo o dado em questão poderia pertencer 70% a classe A e 30% a classe B. Isso é feito atribuindo um score/valor de probabilidade desse dado pertencer a cada conjunto.

Veja o seguinte exemplo:

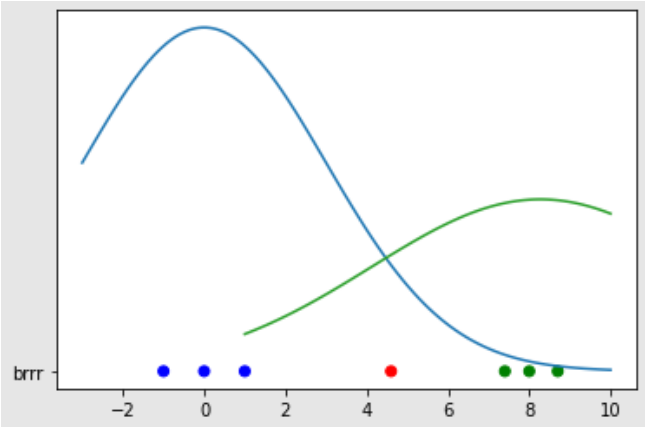

Figura 4 – Fonte: O autor.

No exemplo em questão, cada linha é uma distribuição gaussi-

ana e representa uma classe. Para os pontos em azul, a probabilidade de pertencer à classe representada pela distribuição em azul é maior do que do que a verde, esses dados seriam considerados, por exemplo, 95% classe azul e 5% classe verde. Já os pontos em verde, analogamente tem mais probrabilidade de pertencer a classe verde.

Observe agora o ponto vermelho, este está na intersecção entre essas distribuições, tal ponto seria considerado 50% azul e 50% verde.

No contexto do presente estudo, o modelo de mistura gaussiana tem a flexibilidade de ser trabalhado com *hard* ou *soft assignment*

3 FORMULAÇÃO MATEMÁTICA DO MODELO

Nesta seção apresentamos a construção teórica dos Modelos de Mistura Gaussiana.

Seja $\mathcal{X} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \cdots, \mathbf{x}^{(N)}\}$ um conjunto de dados não rotulados, onde $\mathbf{x}^{(i)} = (x_1^{(i)}, x_2^{(i)}, \cdots, x_n^{(i)}) \in \mathbb{R}^n$. Supõe-se que esses dados são amostras i.i.d de uma distribuição desconhecida $p(x \mid \theta)$.

Considere $K \in \mathbb{N}$ distribuições gaussianas $\mathcal{N}(\mathbf{x} \mid \mu_k, \Sigma_k)$, onde $k = 1, \ldots, K$. Um modelo de mistura gaussiana é um modelo de densidade tal que

$$p(\mathcal{X} \mid \theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x \mid \mu_k, \Sigma_k), \tag{22}$$

$$0 \leq \pi_k \leq 1, \qquad \sum_{k=1}^{K} \pi_k = 1, \tag{23}$$

na qual $\theta = \{\mu_k, \Sigma_k, \pi_k : k = 1, \cdots, K\}$ é a coleção de parâmetros do modelo. Mais especificamente, a média μ e covariância Σ são definidas como:

$$\mu = \frac{1}{N}(\mathbf{x}^{(1)} + \mathbf{x}^{(2)} + \cdots + \mathbf{x}^{(N)}), \tag{24}$$

$$\Sigma = \begin{bmatrix} Var(x_1) & Cov(x_1, x_2) & \cdots & Cov(x_1, x_n) \\ Cov(x_2, x_1) & Var(x_2) & \cdots & Cov(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ Cov(x_n, x_1) & Cov(x_n, x_2) & \cdots & Var(x_n) \end{bmatrix}. \tag{25}$$

Assim, o nosso objetivo é estimar os parâmetros μ_k, Σ_k, isto é, θ, para representar $\mathcal{X}$ assumindo que esta variável aleatória segue uma distribuição de mistura gaussiana.

Observa-se primeiramente que sob a hipótese dos dados i.i.d tem-se que:

$$p(\mathcal{X}, \theta) = \prod_{n=1}^{N} p(x_n, \theta) = \prod_{n=1}^{N} \left(\sum_{k=1}^{K} \pi_k \mathcal{N}(x_n \mid \mu_k, \Sigma_k) \right). \tag{26}$$

e aplicando a função *log* em (26), temos

$$\begin{aligned} log\ p(\mathcal{X} \mid \theta) &= log \left(\prod_{n=1}^{N} p(x_n, \theta) \right) \\ &= \sum_{n=1}^{N} log\ p(x_n \mid \theta) \\ &= \sum_{n=1}^{N} log \left(\sum_{n=1}^{K} \pi_k \mathcal{N}(x_n \mid \mu_k, \Sigma_k) \right) \end{aligned} \tag{27}$$

.

por fim, toma-se o negativo de 27

$$\mathcal{L}(\mu_k, \Sigma_k, \pi_k = -log\ p(\mathcal{X} \mid \theta) \tag{28}$$

Portanto, tem-se a função *negative log-likelihood* (veja seção 2.3) para as misturas gaussianas. Conforme apresentado nas seções anteriores, nosso objetivo é encontrar a máxima verossimilhança do modelo de mistura, e da construção de 28 tem-se que maximar a verossimilhança é equivalente a minimizar 28, a função *negative log-likelihood* é também uma função convexa, ou seja, encontrar um ponto crítico da função é equivalente a encontrar um ponto de mínimo. Assim, maximizar a verossimilhança é equivalente a encontrar o ponto onde a derivada da função *negative log-likelihood* é zero.

Assim, propriedades do cálculo garantem que em qualquer ponto ótimo local de uma função ocorre necessariamente que o seu gradiente com respeito aos parâmetros seja zero. No caso desse estudo,

é possível descrever as seguintes condições necessárias quando otimizamos a função log-likelihood com respeito aos parâmetros μ_k, Σ_k, π_k:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \mu_k} = 0^\intercal \Leftrightarrow \sum_{n=1}^{N} \frac{\partial \ log \ p(x_n \mid \theta)}{\partial \mu_k} = 0^\intercal, \\
\frac{\partial \mathcal{L}}{\partial \Sigma_k} = 0^\intercal \Leftrightarrow \sum_{n=1}^{N} \frac{\partial \ log \ p(x_n \mid \theta)}{\partial \Sigma_k} = 0^\intercal, \\
\frac{\partial \mathcal{L}}{\partial \pi_k} = 0 \Leftrightarrow \sum_{n=1}^{N} \frac{\partial \ log \ p(x_n \mid \theta)}{\partial \pi_k} = 0.
\end{aligned} \tag{29}$$

3.1 RESPONSABILIDADES

Nesta seção será apresentada uma medida essencial para o desenvolvimento do modelo de misturas gaussianas: as responsabilidades.

Define-se:

$$r_{nk} = \frac{\pi_k \mathcal{N}(x_n \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n \mid \mu_j, \Sigma_j)}, \tag{30}$$

como sendo a responsabilidade da k-ésima componente de mistura para o n-ésimo ponto do conjunto de dados.

A responsabilidade r_{nk} da k-ésima componente de mistura para o ponto x_n é proporcional a verossimilhança

$$p(x_n \mid \pi_k, \mu_k, \Sigma_k) = \pi_k \mathcal{N}(x_n, \mu_k, \Sigma_k). \tag{31}$$

Portanto, fixado uma amostra $x^{(n)}$, r_{nk} indica a responsabilidade (ou contribuição) da k-ésima componente. Em outras palavras, componentes de mistura possuem mais responsabilidade quanto maior for a plausibilidade de representar determinada amostra como pertencente àquela componente.

Além disso, note que o vetor

$$r_n = [r_{n1}, r_{n2}, \cdots, r_{nK}]^\intercal$$

é um vetor de probabilidade normalizado, ou seja,

$$\sum_k r_{nk} = 1,$$

com $r_{nk} \geq 0$.

O vetor r_n distribui uma massa ou score de probabilidade entre as K componentes de mistura. A entrada com maior valor indica a componente mais plausível para a amostra, representando um *hard assignment*. Por outro lado, r_n é o vetor de *soft assignment* (2.6) para o dado x_n em relação às K componentes.

É importante observar que $p(x_n \mid \pi_k, \mu_k, \Sigma_k)$ é diferente de $p(x_n, \theta)$, uma vez que $p(x_n, \theta)$ considera θ com todos os parâmetros qualquer que seja o índice k, já $p(x_n \mid \pi_k, \mu_k, \Sigma_k)$ considera apenas os k-ésimos parâmetros.

3.2 TEOREMAS IMPORTANTES

Para encontrar os parâmetros do modelo (22) que melhor descrevem os dados, é utilizada uma abordagem iterativa em que os valores dos parâmetros μ_k, Σ_k, π_k são atualizados a cada passo.

Os teoremas apresentados a seguir mostram que cada um dos parâmetros do modelo depende das responsabilidades (30), o que torna impossível uma solução de forma fechada para o problema de estimar a máxima verossimilhança. Para lidar com este problema, utilizamos a seguinte abordagem: dadas as responsabilidades iniciais, que podem ser quaisquer, calcula-se e atualiza-se um parâmetro do modelo por vez, mantendo os outros fixos. Com esses novos valores dos parâmetros as responsabilidades serão computadas novamente e, com essas novas

responsabilidades, calcula-se os parâmetros e esse processo continua até alcançar a convergência ou até se chegar a um número máximo de iterações.

Teorema 3.1. *A atualização dos parâmetros $\mu_k, k = 1, \cdots, K$, do modelo de mistura gaussiana é dada por:*

$$\mu_k^{new} = \frac{\sum_{n=1}^{N} r_{nk}\, x_n}{\sum_{n=1}^{N} r_{nk}}. \tag{32}$$

Demonstração. De (29) tem-se que o gradiente da função *log-likelihood* com respeito aos parâmetros $\mu_k, k = 1, 2, \cdots, K$ requer que a seguinte derivada parcial seja computada:

$$\begin{aligned} \frac{\partial p(x_n \mid \theta)}{\partial \mu_k} &= \sum_{j=1}^{K} \pi_j \frac{\partial \mathcal{N}(x_n \mid \mu_j, \Sigma_j)}{\partial \mu_k} \\ &= \pi_k \frac{\partial \mathcal{N}(x_n \mid \mu_k, \Sigma_k)}{\mu_k} \\ &= \pi_k (x_n - \mu_k)^{\intercal} \Sigma_k^{-1} \mathcal{N}(x_n \mid \mu_k, \Sigma_k), \end{aligned} \tag{33}$$

logo, apenas a k-ésima componente da mistura depende de μ_k. E da regra da cadeia tem-se que

$$\frac{\partial log\ p(x_n \mid \theta)}{\partial \mu_k} = \frac{1}{p(x_n \mid \theta)} \frac{\partial p(x_n \mid \theta)}{\partial \mu_k}. \tag{34}$$

Portanto, calculando a derivada parcial de $\mathcal{L}$ (27) com respeito a μ_k, tem-se que

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \mu_k} &= \sum_{n=1}^{N} \frac{\partial log\ p(x_n \mid \theta)}{\partial \mu_k} \\
&\overset{(34)}{=} \sum_{n=1}^{N} \frac{1}{p(x_n \mid \theta)} \frac{\partial\ p(x_n \mid \theta)}{\partial \mu_k} \\
&\overset{(33)}{=} \sum_{n=1}^{N} (x_n - \mu_k)^\intercal \Sigma_k^{-1} \pi_k \mathcal{N}(x_n \mid \mu_k, \Sigma_k) \frac{1}{p(x_n \mid \theta)} \\
&\overset{(22)}{=} \sum_{n=1}^{N} (x_n - \mu_k)^\intercal \Sigma_k^{-1} \pi_k \mathcal{N}(x_n \mid \mu_k, \Sigma_k) \frac{1}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n \mid \mu_j, \Sigma_j)} \\
&\overset{(30)}{=} \sum_{n-1}^{N} r_{nk} (x_n - \mu_k)^\intercal \Sigma_k^{-1}. \qquad (35)
\end{aligned}
$$

Agora fazendo $\frac{\partial L(\mu_k^{new})}{\partial \mu_k} = 0^\intercal$ e resolvendo (35) para μ_k^{new}, obtem-se

$$
\sum_{n=1}^{N} r_{nk} x_n = \sum_{n=1}^{N} r_{nk} \mu_k^{new} \quad \Leftrightarrow \quad \mu_k^{new} = \frac{\sum_{n=1}^{N} r_{nk} x_n}{\sum_{n=1}^{N} r_{nk}}, \qquad (36)
$$

□

Intuitivamente, Pode-se dizer que a média μ_k é "puxada"em direção ao ponto de dado x_n com intensidade dada por r_{nk}. As médias são puxadas com maior intensidade para os pontos de dados em que as componentes de misturas gaussianas correspondentes tem maior probabilidade, isto é, uma alta verossimilhança. A atualização da média também pode ser vista pelo valor esperado de todos os pontos de dados com distribuição dada por

$$
r_k = \frac{1}{N} [r_{1k}, \ldots, r_{Nk}]^\intercal \qquad (37)
$$

que é um vetor de probabilidade.

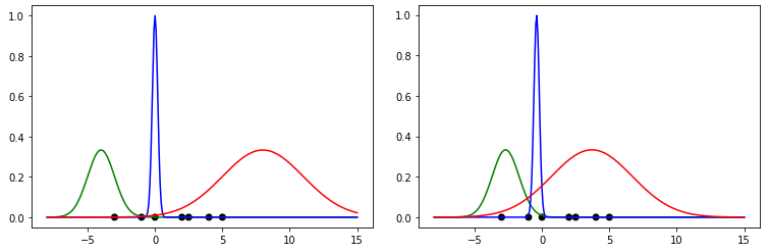

Figura 5 – Imagem a esquerda apresenta componentes de uma mistura com parâmetros iniciais, antes das atualizações dos parâmetros. Segunda imagem apresenta as distribuições dado uma iteração para atualização dos parâmetros. Fonte: O autor baseado em [5]

Vejamos o seguinte exemplo: Considere 3 componentes de misturas $\mathcal{N}(x \mid -4, 1)$, $\mathcal{N}(x \mid 0, 0.2)$ e $\mathcal{N}(x \mid 8, 3)$ e os dados distribuídos como na figura 5. Dada uma aplicação da atualização das médias μ_k, temos as seguintes médias

$$\mu_1 : -4 \Leftrightarrow -2,7$$
$$\mu_2 : 0 \Leftrightarrow -0,4$$
$$\mu_3 : 8 \Leftrightarrow 3,7$$

observe da figura 5 que ao aplicar a atualização dos parâmetros as médias das componentes se movem em direção aonde os dados se concentram, veja como a primeira e terceira componente de mistura se comportam após uma iteração.

A atualização das médias parece bem direta, porém observe que as responsabilidades r_{nk} são funções que dependem de π_j, μ_j, Σ_j para todo $j = 1, 2, \ldots, K$. Assim, essa atualização depende de todos os parâmetros do GMM e uma solução fechada não pode ser obtida.

Teorema 3.2. *: As atualizações dos parâmetros Σ_k são dados por:*

$$\Sigma_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} r_{nk}(x_n - \mu_k)(x_n - \mu_k)^\intercal, \tag{38}$$

sendo $N_k := \sum_{n=1}^{N} r_{nk}$.

Demonstração. Inicialmente, observa-se que

$$\frac{\partial \mathcal{L}}{\partial \Sigma_k} = \sum_{n=1}^{N} \frac{\partial log\ p(x_n \mid \theta)}{\partial \Sigma_k} = \sum_{n=1}^{N} \frac{1}{p(x_n \mid \theta)} \frac{\partial p(x_n \mid \theta)}{\partial \Sigma_k}, \tag{39}$$

derivando a distribuição gaussiana $p(x_n \mid \theta)$ em relação a Σ_k, obtém-se

$$\begin{aligned}
&\frac{\partial p(x_n \mid \theta)}{\Sigma_k} \\
&= \frac{\partial}{\partial \Sigma_k} \left(\pi_k (2\pi)^{-\frac{D}{2}} det(\Sigma_k)^{-\frac{1}{2}}\ exp\left(-\frac{1}{2}(x_n - \mu_k)^\intercal \Sigma_k^{-1}(x_n - \mu_k) \right) \right) \\
&= \pi_k (2\pi)^{-\frac{D}{2}} \Bigg[\\
&\left(\frac{\partial}{\partial \Sigma_k} det(\Sigma_k)^{-\frac{1}{2}} \right)\ exp\left(-\frac{1}{2}(x_n - \mu_k)^\intercal \Sigma_k^{-1}(x_n - \mu_k) \right) \\
&+ det(\Sigma_k)^{-\frac{1}{2}} \left(\frac{\partial}{\partial \Sigma_k} exp\left(-\frac{1}{2}(x_n - \mu_k)^\intercal (x_n - \mu_k) \right) \right) \Bigg].
\end{aligned} \tag{40}$$

Utilizando identidades úteis para o cálculo de gradientes em (40) (veja seção 5.5 de [5]) e refatorando essa equação, temos

$$\begin{aligned}
\frac{\partial p(x_n \mid \theta)}{\partial \Sigma_k} =& \pi_k \mathcal{N}(x_n \mid \mu_k, \Sigma_k) \Big[\\
&- \frac{1}{2}(\Sigma_k^{-1} - \Sigma_k^{-1}(x_n - \mu_k)(x_n - \mu_k)^\intercal \Sigma_k^{-1}) \Big].
\end{aligned} \tag{41}$$

Agora substituindo (22) e (41) em (39)

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \Sigma_k} &= \sum_{n=1}^{N} \frac{1}{p(x_n \mid \theta)} \frac{\partial p(x_n \mid \theta)}{\partial \Sigma_k} \\ &= \sum_{n=1}^{N} \frac{\pi_k \mathcal{N}(x_n \mid \mu_k, \Sigma_k}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_j \mid \mu_j, \Sigma_j)} \Big[\\ &- \frac{1}{2}(\Sigma_k^{-1} - \Sigma_k^{-1}(x_n - \mu_k)(x_n - \mu_k)^\intercal \Sigma_k^{-1})\Big] \\ &= -\frac{1}{2} \sum_{n=1}^{N} r_{nk}(\Sigma_k^{-1} - \Sigma_k^{-1}(x_n - \mu_k)(x_n - \mu_k)^\intercal \Sigma_k^{-1}) \\ &= -\frac{1}{2}\Sigma_k^{-1} \sum_{n=1}^{N} r_{nk} + \frac{1}{2}\Sigma_k^{-1}\Big(\\ &\sum_{n=1}^{N} r_{nk}(x_n - \mu_k)(x_n - \mu_k)^\intercal\Big)\Sigma_k^{-1}.\end{aligned} \tag{42}$$

Fazendo $\frac{\partial \mathcal{L}}{\partial \Sigma_k} = 0$, de (29) obtém-se a condição necessária para a otimização

$$\begin{aligned}\Sigma_k^{-1} \sum_{n=1}^{N} r_{nk} &= \Sigma_k^{-1} \left(\sum_{n=1}^{N} r_{nk}(x_n - \mu_k)(x_n - \mu_k)^\intercal \right) \Sigma_k^{-1} \\ \Leftrightarrow \quad IN_k &= \left(\sum_{n=1}^{N} r_{nk}(x_n - \mu_k)(x_n - \mu_k)^\intercal \right) \Sigma_k^{-1}.\end{aligned} \tag{43}$$

Neste último passo foi utilizado que $\sum_{n=1}^{N} r_{nk} = N_k$ e I é a matriz identidade. Por fim, resolvendo (43) para Σ_k, tem-se

$$\Sigma_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} r_{nk}(x_n - \mu_k)(x_n - \mu_k)^\intercal. \tag{44}$$

□

Semelhante a atualização de μ_k em 32, pode-se interpretar a atualização da covariância em (38) como um valor esperado ponderado pela im-

portância do quadrado do dado centralizado $X_k := \{x_1 - \mu_k, \ldots, x_N - \mu_k\}$.

Teorema 3.3. *As atualizações dos parâmetros π_k são dadas por*

$$\pi_k^{new} = \frac{1}{N}\sum_{n=1}^{N} r_{nk}, \qquad k = 1, \cdots, K, \tag{45}$$

na qual N é o número de dados no conjunto de dados.

Demonstração. Da definição (23), tem-se nessa situação um problema de otimização com restrições. Neste caso utiliza-se os multiplicadores de Lagrange para encontrar a solução do problema [4]. Portanto, o lagrangiano se da por

$$\begin{aligned} \mathbf{L} &= \mathcal{L} + \lambda\left(\sum_{k=1}^{K}\pi_k - 1\right) \\ &= \sum_{n=1}^{N} log\left(\sum_{n=1}^{K}\pi_k \mathcal{N}(x_n \mid \mu_k, \Sigma_k)\right) + \lambda\left(\sum_{k=1}^{K}\pi_k - 1\right), \end{aligned} \tag{46}$$

na qual $\mathcal{L}$ é a função *negative log-likelihood* definida em (28) e o segundo termo de (46) é dado pela restrição $\sum_{k=1}^{N}\pi_k = 1$. Assim, calculando a derivada de $\mathbf{L}$ com respeito a π_k, tem-se

$$\begin{aligned} \frac{\partial \mathbf{L}}{\partial \pi_k} &= \sum_{n=1}^{N}\frac{\mathcal{N}(x_n \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{N}\pi_j \mathcal{N}(x_n \mid \mu_j, \Sigma_j)} + \lambda \\ &= \frac{1}{\pi_k}\sum_{n=1}^{N}\frac{\pi_k \mathcal{N}(x_n \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{N}\pi_j \mathcal{N}(x_n \mid \mu_j, \Sigma_j)} + \lambda. \end{aligned} \tag{47}$$

Considerando $N_k = \sum_{n=1}^{N} r_{nk}$ e da definição de responsabilidade dada em (30), tem-se de (47)

$$\frac{\partial \mathbf{L}}{\partial \pi_k} = \frac{N_k}{\pi_k} + \lambda. \tag{48}$$

Observa-se também que de (46) a derivada de $\mathbf{L}$ em relação a λ é dada por

$$\frac{\partial \mathbf{L}}{\partial \lambda} = \sum_{k=1}^{K} \pi_k - 1. \tag{49}$$

Pela condição necessária de otimização, as equações (48) e (49) devem ser igualadas a 0, logo

$$\pi_k = -\frac{N_k}{\lambda}, \tag{50}$$

$$1 = \sum_{n=1}^{K} \pi_k. \tag{51}$$

Utilizando essas equações e resolvendo para π_k, obtém-se que

$$\Sigma_{n=1}^{K} \pi_k = 1 \Leftrightarrow -\sum_{n=1}^{K} \frac{N_k}{\lambda} = 1 \Leftrightarrow -\frac{N}{\lambda} = 1 \Leftrightarrow \lambda = -N. \tag{52}$$

considerando $\sum_{n=1}^{N} N_k = N$. Pois por definição $N_k = \sum_{n=1}^{N} r_{nk}$ e de (23), temos da definição de (30)

$$\sum_{k=1}^{K} r_{nk} = \sum_{k=1}^{K} \frac{\pi_k \mathcal{N}(x_n \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n \mid \mu_j, \Sigma_j)} = \frac{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_n \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n \mid \mu_j, \Sigma_j)} = 1$$

, logo

$$\sum_{k=1}^{K} N_k = \sum_{k=1}^{K} \sum_{n=1}^{N} r_{nk} = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} = \sum_{n=1}^{N} 1 = N$$

Portanto, substituindo o valor de λ em (50) constata-se que

$$\pi_k^{new} = \frac{N_k}{N}.$$

□

É possível identificar os pesos em (45) como a razão da responsabilidade total do k-ésimo componente e o número total de dados. Como $N = \sum_k N_k$, a quantidade de dados também pode ser interpretada como a responsabilidade total de todas as componentes de mistura juntas, de tal forma que π_k é a importância relativa da k-ésima componente de mistura.

Observação: Como $N_k = \sum_{i=1}^{N} r_{nk}$, a atualização em (45) para os pesos também depende de todos os parâmetros $\pi_j, \mu_j, \Sigma_j, j = 1, \ldots, K$ por causa das responsabilidades r_{nk}

3.3 ALGORITMO *EXPECTATION MAXIMIZATION*

Observa-se que as atualizações em (32), (38) e (45) não constituem uma solução com fórmula fechada para a atualização dos parâmetros μ_k, Σ_k, π_k do modelo de mistura porque as responsabilidade r_{nk} dependem desses parâmetros de forma complexa. Entretanto, os resultados sugerem um esquema simples de iterações para encontrar uma solução para o problema de estimar os parâmetros através da máxima verossimilhança.

O algoritmo *expectation maximization* (EM), proposto por Dempster et al. (1977), é um método iterativo geral para aprender parâmetros em modelos de mistura.

No caso dos modelos de misturas gaussianas, escolhe-se valores iniciais aleatórios para μ_k, Σ_k, π_k e alterna-se entre os seguintes passos até alcançar a convergência:

- **Passo 1 (*Expectation*)**: Calcule os valores de r_{nk} (probabilidade posterior do n-ésimo dado pertencente a k-ésima componente de mistura).
- **Passo 2 (*Maximization*)**: Utilize as responsabilidades atualizadas do passo anterior para reestimar os parâmetros μ_k, Σ_k, π_k

cada passo do algoritmo EM aumenta a função *log-likelihood* [14]. Para a convergência, pode-se observar a *log-likelihood* ou diretamente os parâmetros. Um exemplo mais específico da implementação do algoritmo é apresentada a seguir:

1. Inicie com μ_k, Σ_k, π_k aleatórios.
2. *expectation*: Avalie as responsabilidades r_{nk} para cada dado x_n utilizando os parâmetros μ_k, Σ_k, π_k atuais:
$$r_{nk} = \frac{\pi_k \mathcal{N}(x_n \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n \mid \mu_j, \Sigma_j)},$$
3. Reestime os parâmetros μ_k, Σ_k, π_k utilizando as responsabilidades r_{nk} do passo anterior:

$$\mu_k = \frac{\sum_{n=1}^{N} r_{nk}\, x_n}{\sum_{n=1}^{N} r_{nk}},$$
$$\Sigma_k = \frac{1}{N_k} \sum_{n=1}^{N} r_{nk}(x_n - \mu_k)(x_n - \mu_k)^{\intercal},$$
$$\pi_k = \frac{1}{N} \sum_{n=1}^{N} r_{nk}, \qquad k = 1, \cdots, K.$$

4 CONCLUSÃO

Neste trabalho apresentou-se a construção do Modelo de Misturas Gaussianas, uma ferramenta bastante útil em Matemática Aplicada, em especial, em *Machine Learning*. Apesar de não apresentadas neste trabalho, as aplicações deste modelo são vastas e podem ser exploradas em trabalhos futuros. O GMM torna possível representar dados por meio de componentes de distribuições normais e por ser uma distribuição de probabilidade, permite retirar amostras dessa distribuição gerando dados sintéticos. Esta é uma abordagem bem relevante uma vez que se possui um conjunto de dados com poucas amostras de cada classe. Seguindo essa abordagem é incentivada, por exemplo, a utilização do GMM para gerar amostras sintéticas do conjunto de dados de [12] com o objetivo de disponibilizar um número de amostras mais representativas para que algoritmos performem melhor na classificação de microplásticos no oceano.

BIBLIOGRAFIA

[1] Yaser S. Abu-Mostafa, M. Magdon-Ismail e H.T. Lin. *Learning from Data: A Short Course*. AMLBook.com, 2012.

[2] Taiwo Oladipupo Ayodele. "Types of machine learning algorithms". Em: *New advances in machine learning* 3 (2010), pp. 19–48.

[3] James Bergstra e Yoshua Bengio. "Random Search for Hyper-Parameter Optimization". Em: *J. Mach. Learn. Res.* 13 (2012), pp. 281–305.

[4] C.M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer New York, 2016. Disp. em: `https://books.google.com.br/books?id=kOXDtAEACAAJ` (acesso em 11/12/2022).

[5] M.P. Deisenroth, A.A. Faisal e C.S. Ong. *Mathematics for Machine Learning*. Cambridge University Press, 2020.

[6] Edson Cilos Vargas. *Machine Learning | Curso completo em Python | 2022*. 2022. Disp. em: `https://www.udemy.com/course/edson-cilos-ml/` (acesso em 11/12/2022).

[7] A. Géron. *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, 2017.

[8] Wassily Hoeffding. “Probability inequalities for sums of bounded random variables”. Em: *The collected works of Wassily Hoeffding*. Springer, 1994, pp. 409–426.

[9] B.R. James. *Probabilidade: um curso em nível intermediário*. Projeto Euclides. Instituto de Matemática Pura e Aplicada, 2004.

[10] Leslie Pack Kaelbling, Michael L Littman e Andrew W Moore. “Reinforcement learning: A survey”. Em: *Journal of artificial intelligence research* 4 (1996), pp. 237–285.

[11] Andrei N. Kolmogorov. “Foundations of the theory of probability”. Em: 1960.

[12] Henrique de Medeiros Back, Edson Cilos Vargas Junior, Orestes Estevam Alarcon e Daphiny Pottmaier. “Training and evaluating machine learning algorithms for ocean microplastics classification through vibrational spectroscopy”. Em: *Chemosphere* 287 (2022), p. 131903.

[13] T.M. Mitchell. *Machine Learning*. McGraw-Hill international editions - computer science series. McGraw-Hill Education.

[14] Radford M. Neal e Geoffrey E. Hinton. “A View of the Em Algorithm that Justifies Incremental, Sparse, and other Variants”. Em: *Learning in Graphical Models*. Dordrecht: Springer Netherlands, 1998, pp. 355–368.

[15] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. Em: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[16] A. L. Samuel. “Some Studies in Machine Learning Using the Game of Checkers. II—Recent Progress”. Em: *IBM Journal of Research and Development* 11.6 (1967), pp. 601–617.

[17] “The Internet of Things: A survey”. Em: *Computer Networks* 54.15 (2010), pp. 2787–2805.

[18] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 1999.