INSTITUTO FEDERAL
Ceará
Programa de Pós-Graduação
em Ciência da Computação

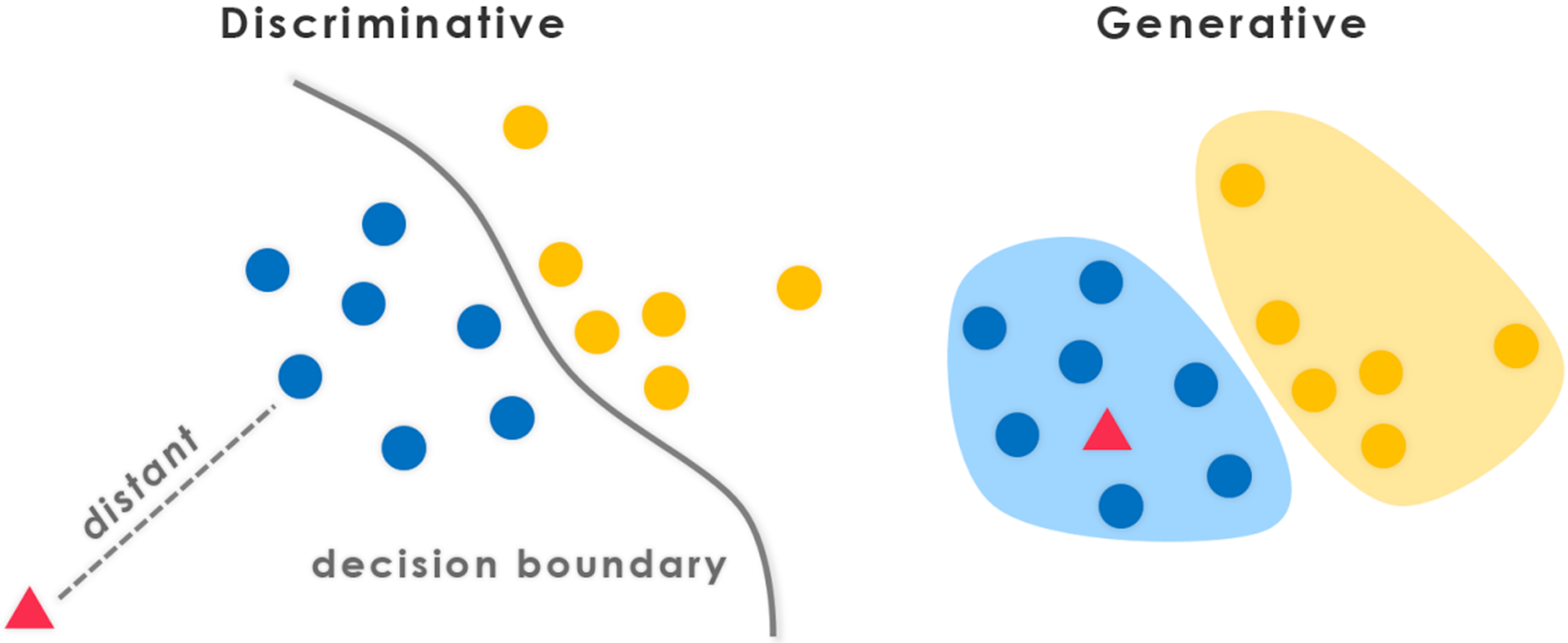
16 Revisão rápida de teoria da probabilidade

APRENDIZAGEM PRODUNDA

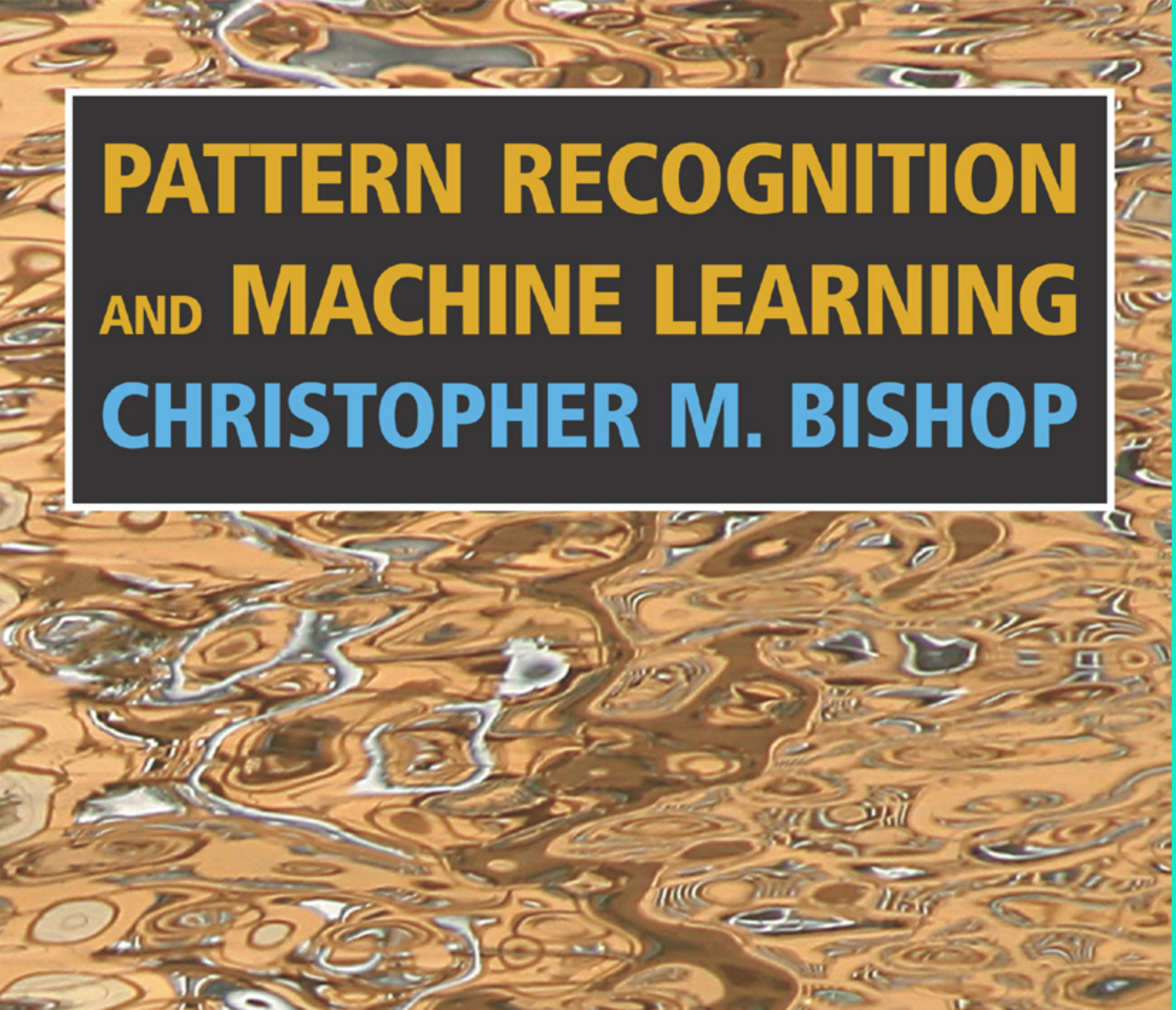
PPGCC – 2024.1

Prof. Saulo Oliveira <saulo.oliveira@ifce.edu.br>

MODELO DISCRIMINATIVO VS GENERATIVO



https://www.kdnuggets.com/2020/05/microsoft-research-three-efforts-advance-deep-generative-models.html



BISHOP, Christopher M.; NASRABADI, Nasser M. **Pattern recognition and machine learning**. New York: springer, 2006.

REVISÃO RÁPIDA DE PROBABILIDADE

Notação da Aula (variáveis contínuas)

A definição de PDF

$$p\left(x \in (a, b)\right) = \int_b^a p(x)\,\mathrm{d}x$$

Propriedades

$$p\,(x) \geq 0 \qquad \int_b^a p(x)\,\mathrm{d}x = 1$$

Variável aleatória

$$x \sim p\,(\theta) \qquad p\left(x \mid \theta\right) > 0$$

Valor esperado

$$\mathbb{E}\left[f\right] = \sum_x p(x) f(x) \text{ ou } \mathbb{E}\left[f\right] = \int p(x) f(x)\,\mathrm{d}x$$

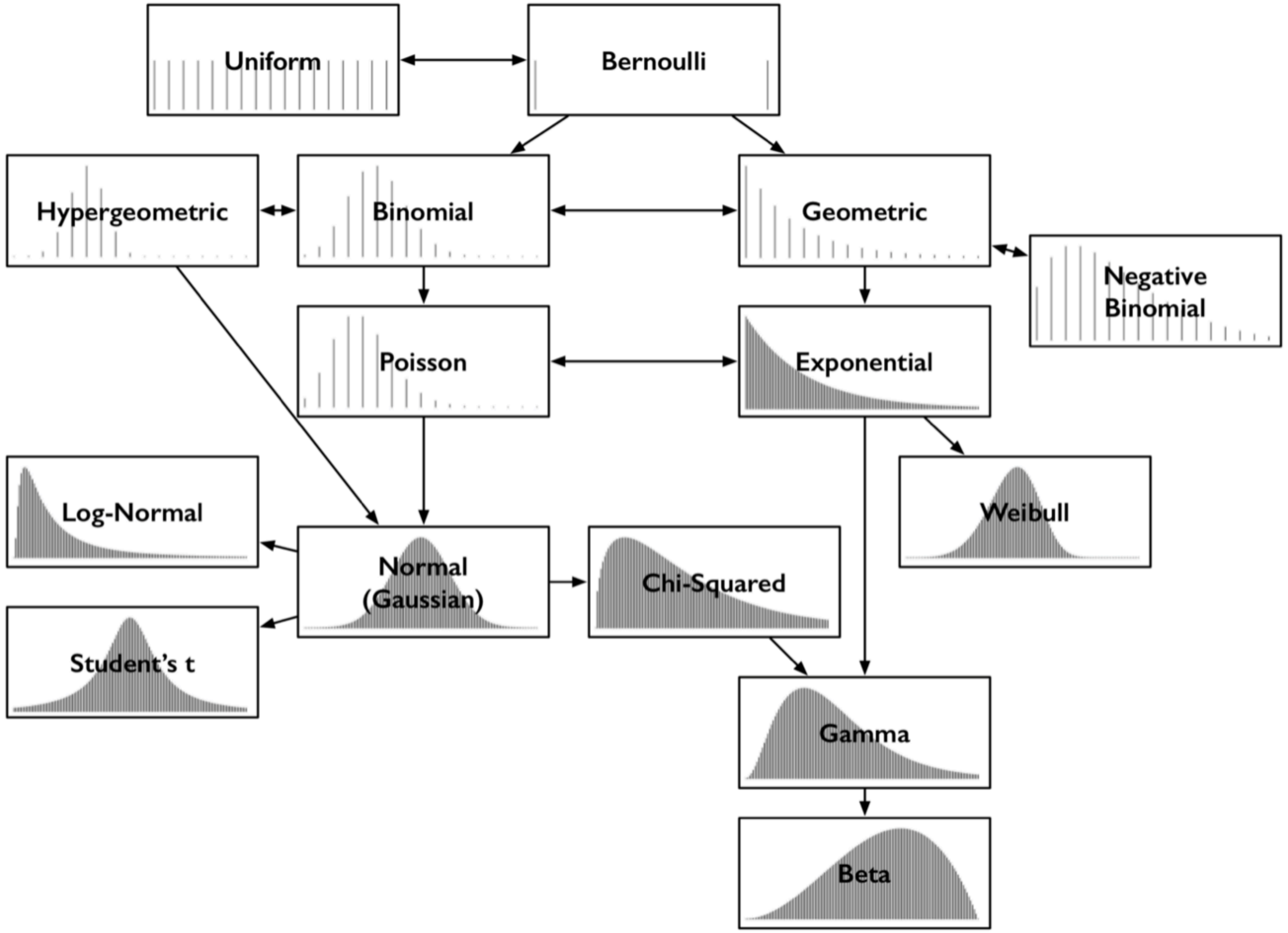
Variância

$$\mathrm{var}\left[f\right] = \mathbb{E}\left[\left(f(x) - \mathbb{E}\left[f(x)\right]\right)^2\right] = \mathbb{E}\left[f(x)^2\right] - \mathbb{E}\left[f(x)\right]^2$$

I.I.D.

$$p\left(\mathbf{x} \mid \theta\right) = \prod_{n=1}^{N} p\left(x_i \mid \theta\right)$$

Alguma distribuições...

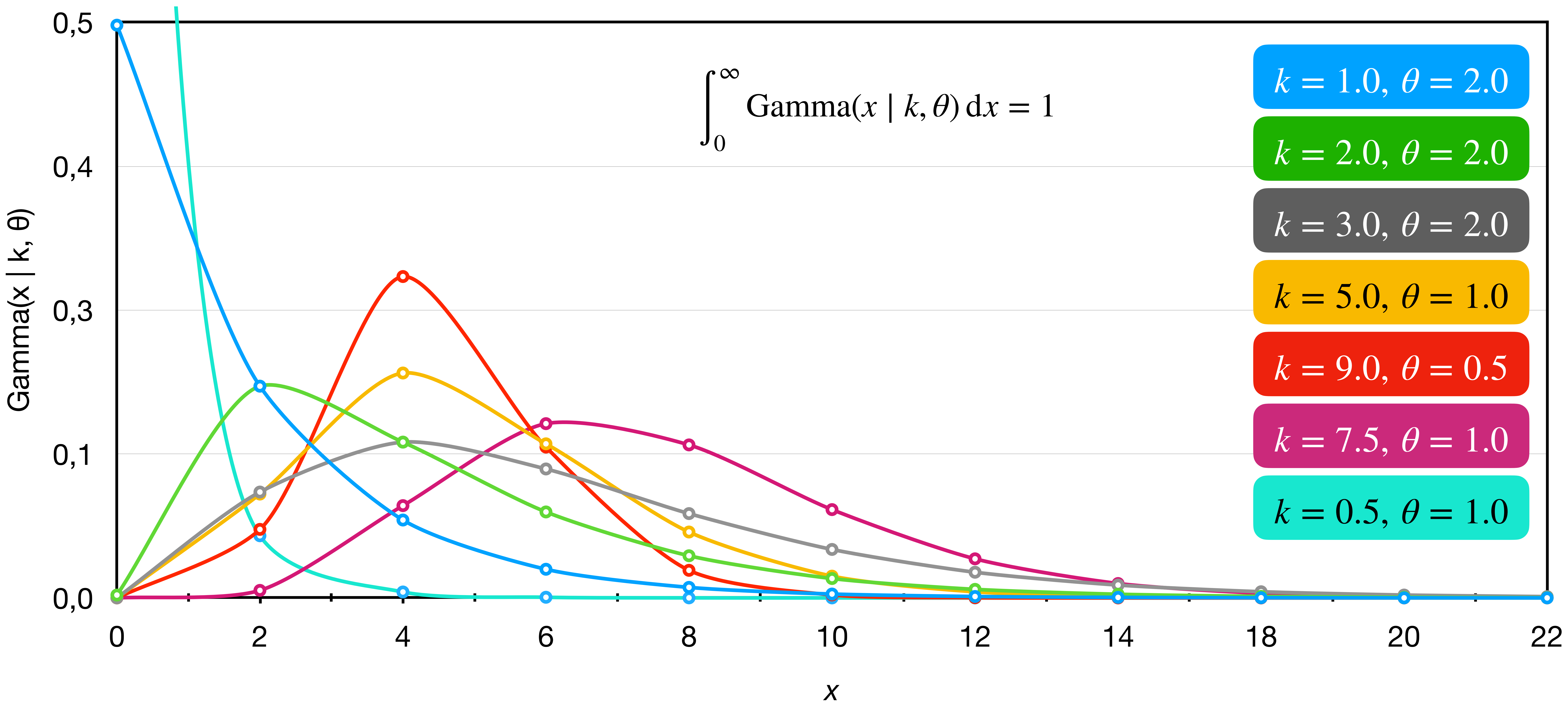


Observe:

- Teremos os dados modelados como uma variável aleatória x;
- Formato da *pdf*, $\mathrm{p}(\theta)$;
- Espaço dos dados, i.e, $\mathcal{S}, \Omega$ ou U;
- Espaço dos parâmetros θ;

https://www.johnrobininston.com/distributions/

A Distribuição Gamma



Probabilidade condicional

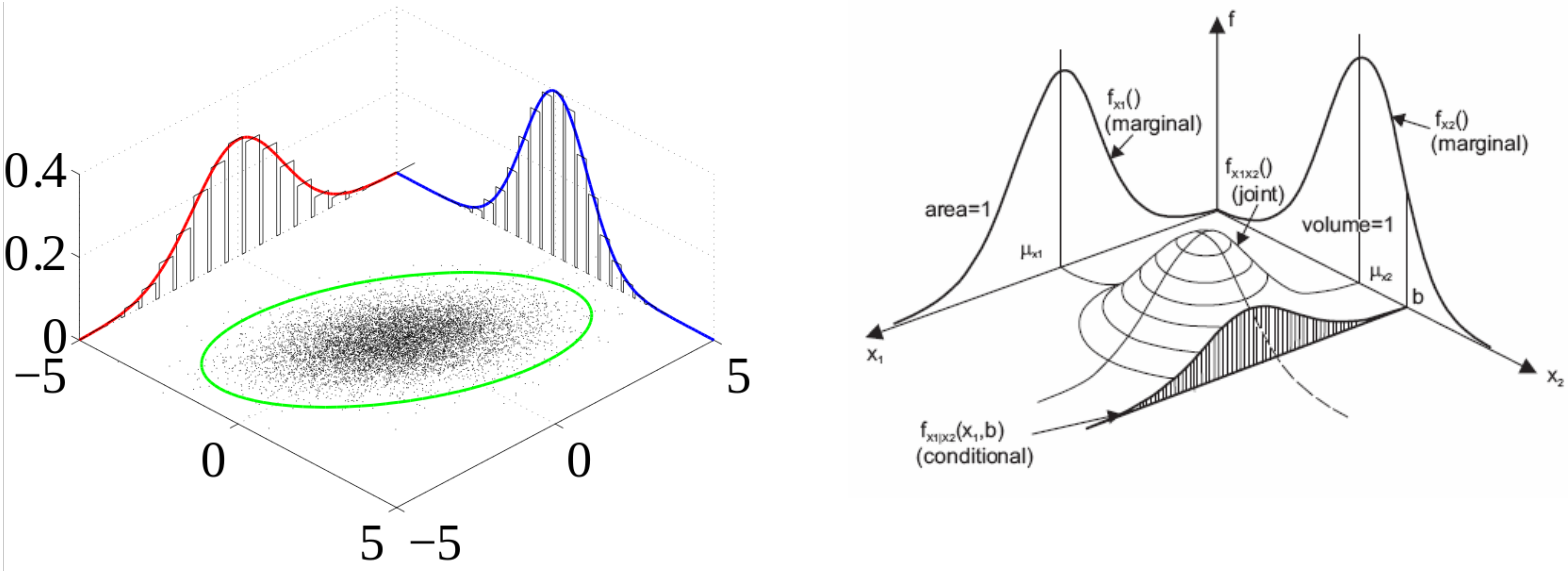
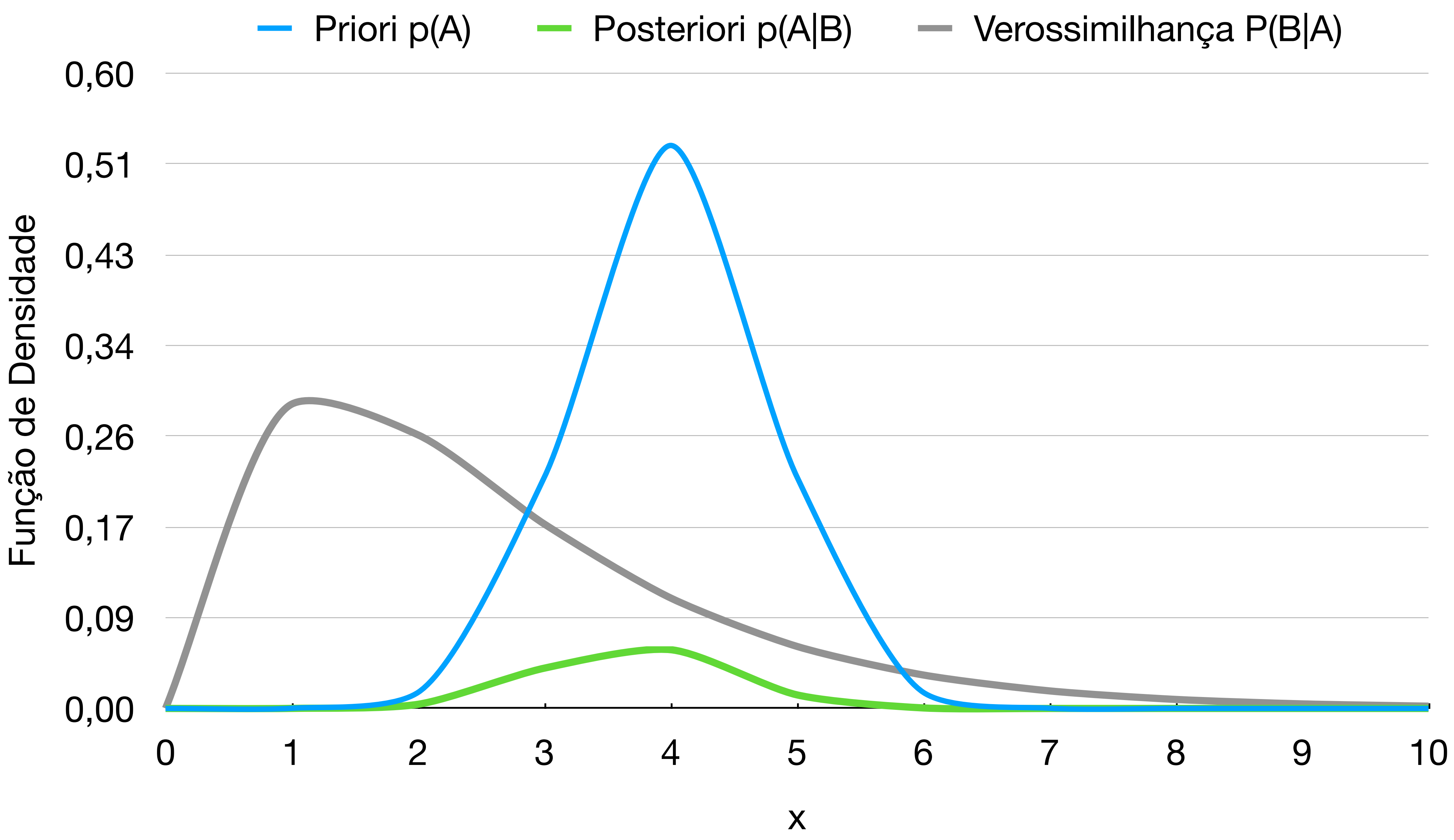
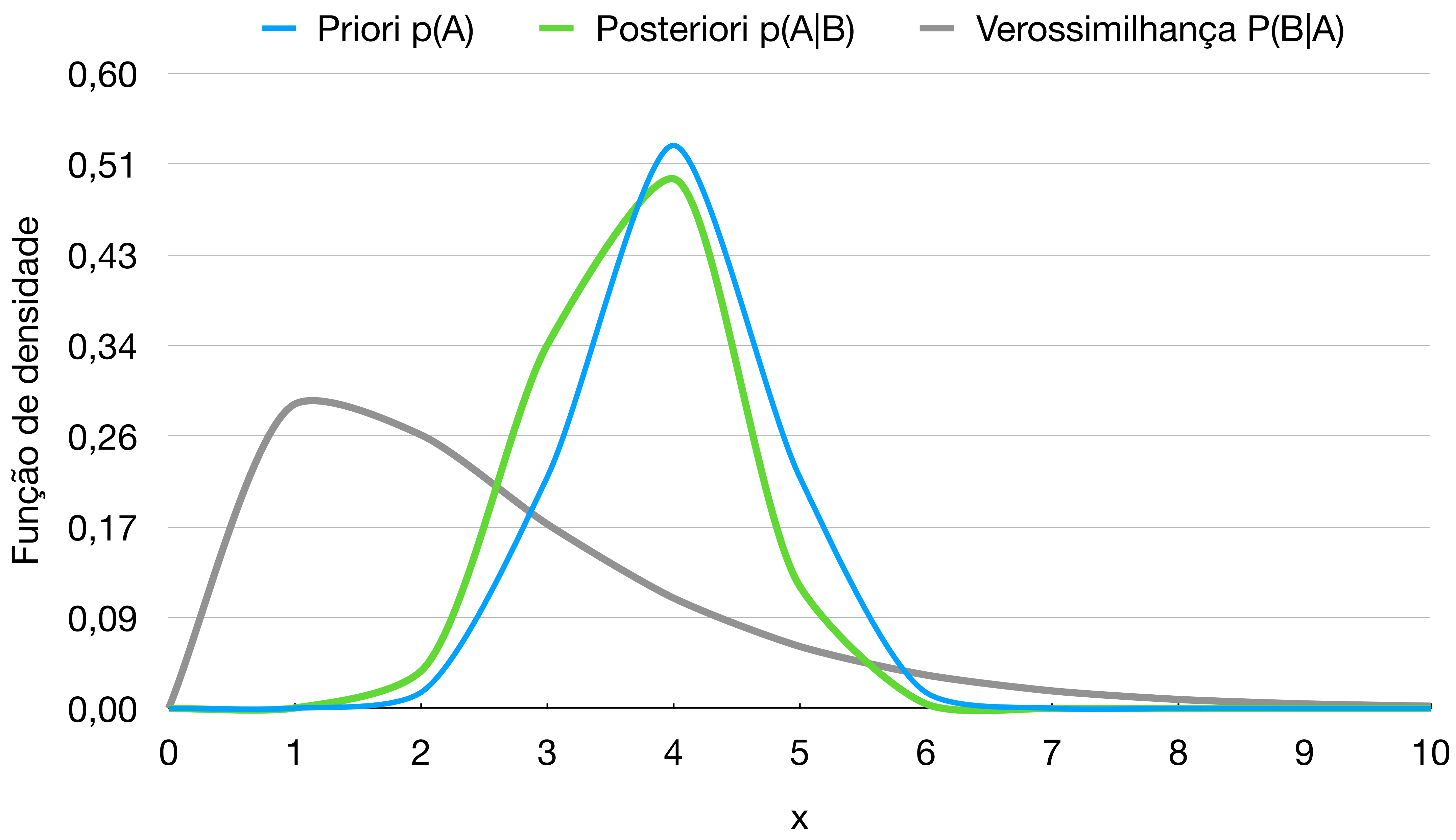


Figure 2.13-Joint and marginal probability density function (PDF), from Melchers (1999).

Regra de Bayes

$$p(A \mid B) = \frac{p(B \mid A) \times p(A)}{p(B)}$$





Observe:

- posteriori $\propto$ priori $\times$ verossimilhança
- posteriori $= \dfrac{\text{priori} \times \text{verossimilhança}}{\text{evidência}}$

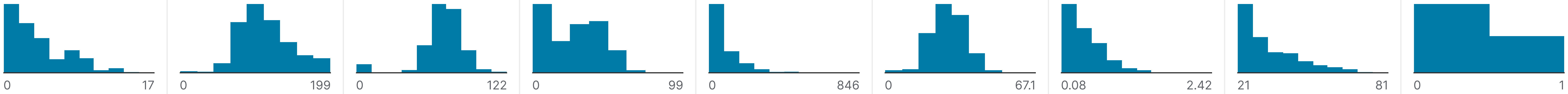
DISTRIBUIÇÃO DOS ATRIBUTOS

diabetes.csv (23.87 kB)

Detail Compact Column 9 of 9 columns

About this file

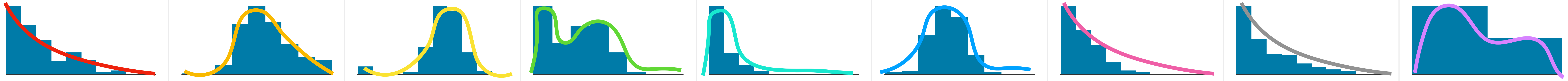
The datasets consist of several medical predictor (independent) variables and one target (dependent) variable, **Outcome**. Independent variables include the number of pregnancies the patient has had, their BMI, insulin level, age, and so on.



# Pregnancies	# Glucose	# BloodPressure	# SkinThickness	# Insulin	# BMI	# DiabetesPedigree...	# Age	# Outcome
Number of times pregnant	Plasma glucose concentration a 2 hours in an oral glucose tolerance test	Diastolic blood pressure (mm Hg)	Triceps skin fold thickness (mm)	2-Hour serum insulin (mu U/ml)	Body mass index (weight in kg/(height in m)^2)	Diabetes pedigree function	Age (years)	Class variable (0 or 1) 268 of 768 are 1, the others are 0
6	148	72	35	0	33.6	0.627	50	1
1	85	66	29	0	26.6	0.351	31	0
8	183	64	0	0	23.3	0.672	32	1
1	89	66	23	94	28.1	0.167	21	0
0	137	40	35	168	43.1	2.288	33	1
5	116	74	0	0	25.6	0.201	30	0
3	78	50	32	88	31	0.248	26	1
10	115	0	0	0	35.3	0.134	29	0
2	197	70	45	543	30.5	0.158	53	1
8	125	96	0	0	0	0.232	54	1
4	110	92	0	0	37.6	0.191	30	0

https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database?resource=download

Naïve Bayes



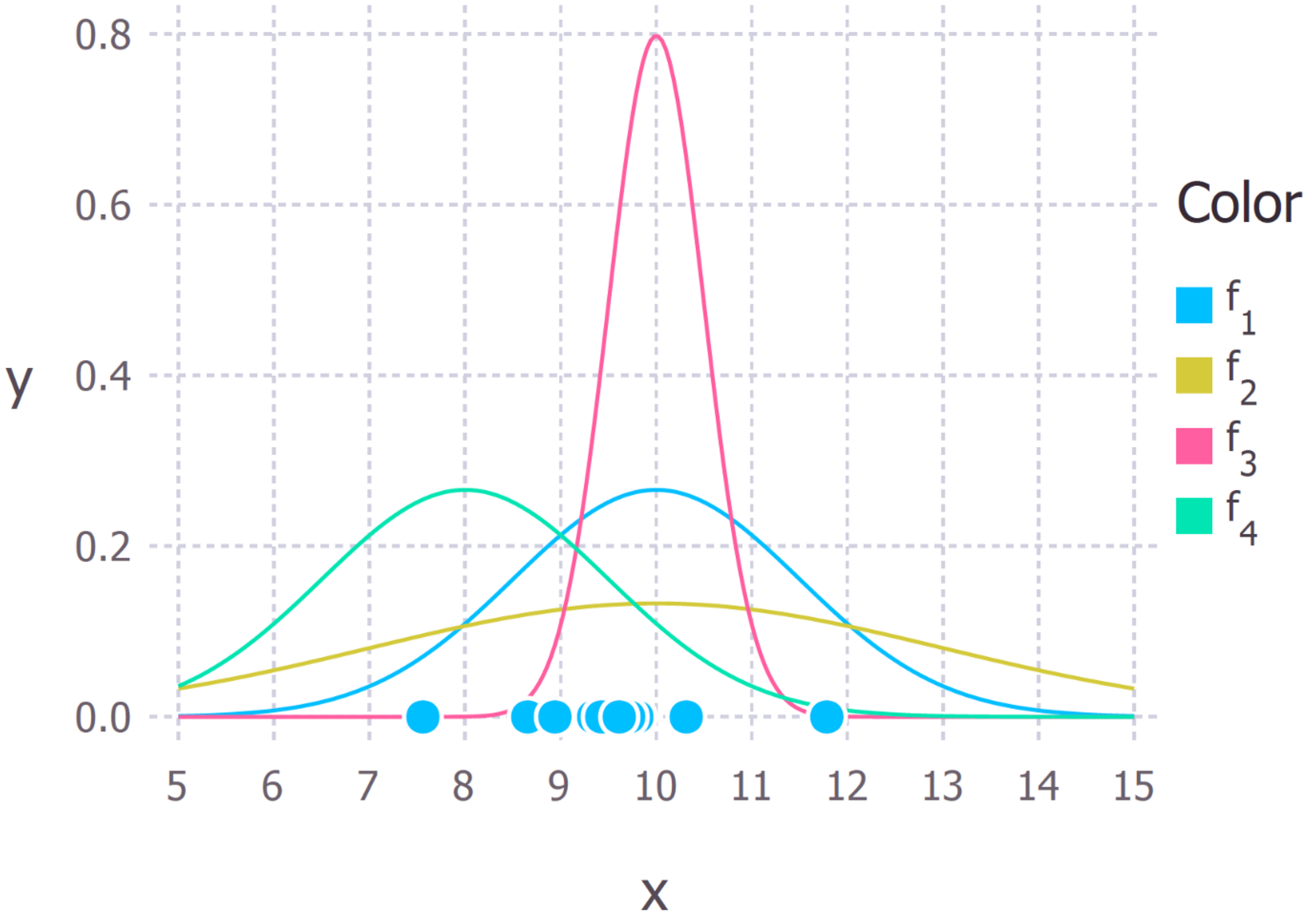
# Pregnancies	# Glucose	# BloodPressure	# SkinThickness	# Insulin	# BMI	# DiabetesPedigree...	# Age	# Outcome
Number of times pregnant	Plasma glucose concentration a 2 hours in an oral glucose tolerance test	Diastolic blood pressure (mm Hg)	Triceps skin fold thickness (mm)	2-Hour serum insulin (mu U/ml)	Body mass index (weight in kg/(height in m)^2)	Diabetes pedigree function	Age (years)	Class variable (0 or 1) 268 of 768 are 1, the others are 0
0 – 17	0 – 199	0 – 122	0 – 99	0 – 846	0 – 67.1	0.08 – 2.42	21 – 81	0 – 1
6	148	72	35	0	33.6	0.627	50	1
1	85	66	29	0	26.6	0.351	31	0
8	183	64	0	0	23.3	0.672	32	1
1	89	66	23	94	28.1	0.167	21	0
0	137	40	35	168	43.1	2.288	33	1
5	116	74	0	0	25.6	0.201	30	0
3	78	50	32	88	31	0.248	26	1
10	115	0				0.134	29	0
2	197	70				0.158	53	1
8	125	96				0.232	54	1
4	110	92				0.191	30	0

$$\hat{y} = \underset{k\in\{1,\ldots,K\}}{\arg\max}\ p\left(C_k\right)\prod_{i=1}^{n} p\left(x_i \mid C_k\right)$$

$$\hat{y} = \underset{k\in\{1,\ldots,K\}}{\arg\max}\ p\left(C_k\right)\quad p\left(x_1 \mid C_k\right)\quad p\left(x_2 \mid C_k\right)\quad p\left(x_3 \mid C_k\right)\quad p\left(x_4 \mid C_k\right)\quad p\left(x_5 \mid C_k\right)\quad p\left(x_6 \mid C_k\right)\quad p\left(x_7 \mid C_k\right)\quad p\left(x_8 \mid C_k\right)\quad p\left(x_9 \mid C_k\right)$$

ESTIMAÇÃO DE PARÂMETROS

Máxima verossimilhança



- Achar o conjunto de parâmetros que maximize a função de verossimilhança, i.e.,

$$\boldsymbol{\theta}^{\star} = \underset{\boldsymbol{\theta} \in \boldsymbol{\Theta}}{\arg\max}\ \mathscr{L}(\mathbf{X}, \boldsymbol{\theta})$$

- Achar o conjunto de parâmetros mais prováveis segundo os dados observados, i.e.,

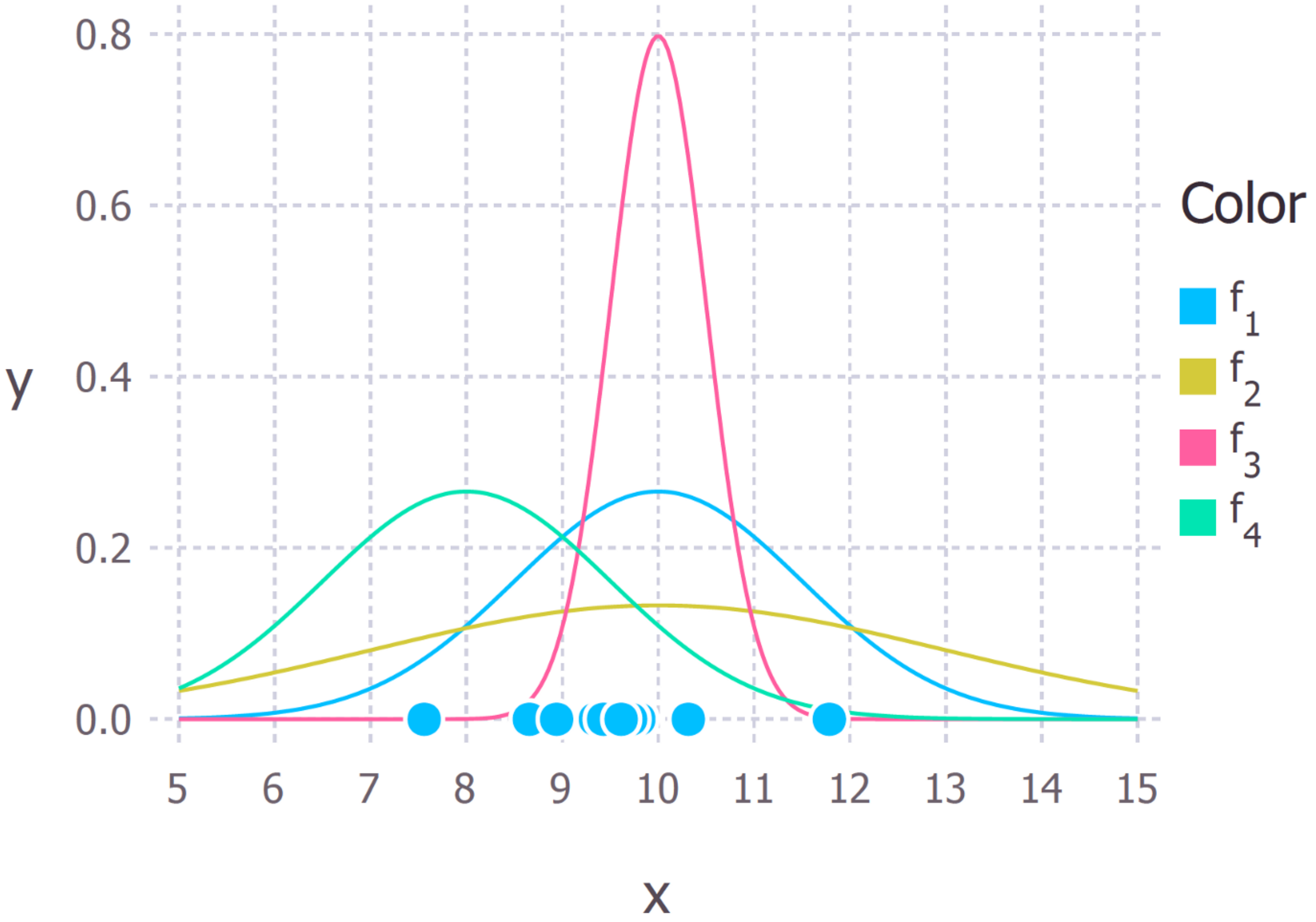
$$\mathscr{L}(\mathbf{X}, \boldsymbol{\theta}) = \prod_{i=1}^{N} f_n(\mathbf{x}_i \mid \boldsymbol{\theta})$$

- Na prática, muitas vezes é conveniente trabalhar com o logaritmo natural da função de verossimilhança, i.e.,

$$\ell(\mathbf{X}, \boldsymbol{\theta}) = \log \mathscr{L}(\mathbf{X}, \boldsymbol{\theta})$$

$$\ell(\mathbf{X}, \boldsymbol{\theta}) = \frac{1}{N} \sum_{i=1}^{N} \log f_n(\mathbf{x}_i \mid \boldsymbol{\theta})$$

Máxima a posteriori



- Achar o conjunto de parâmetros que maximize a função a posteriori, i.e.,

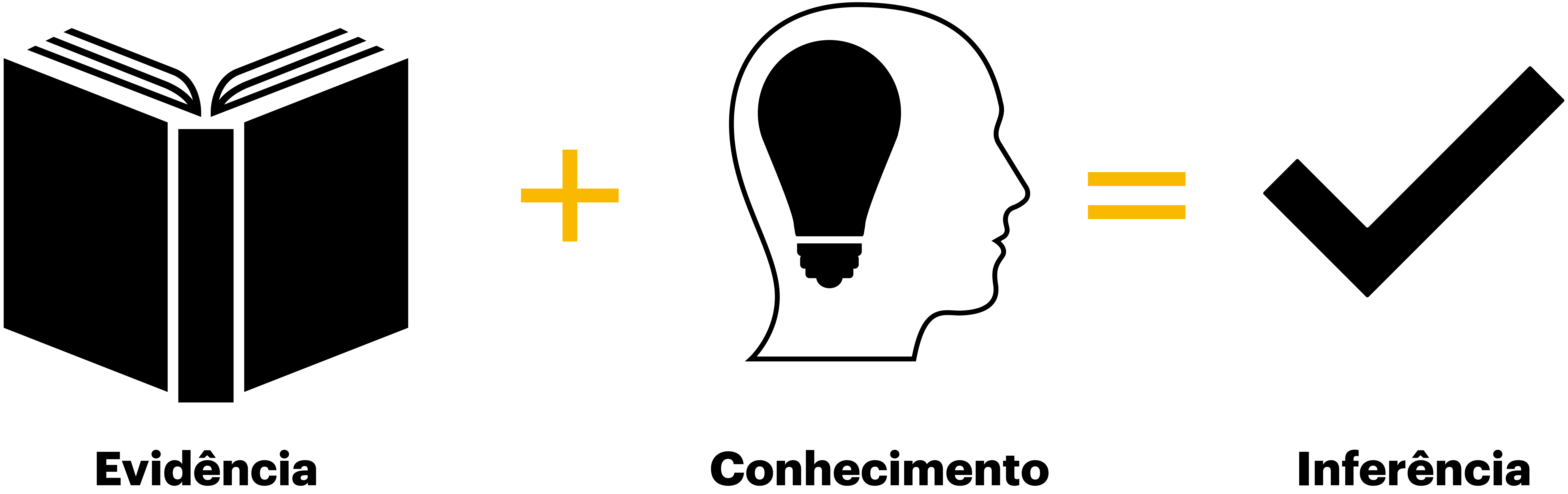
$$\boldsymbol{\theta}^\star = \underset{\boldsymbol{\theta} \in \boldsymbol{\Theta}}{\arg\max}\ \mathscr{MAP}(\mathbf{X}, \boldsymbol{\theta})$$

- Achar o conjunto de parâmetros mais prováveis segundo os dados observados, i.e.,

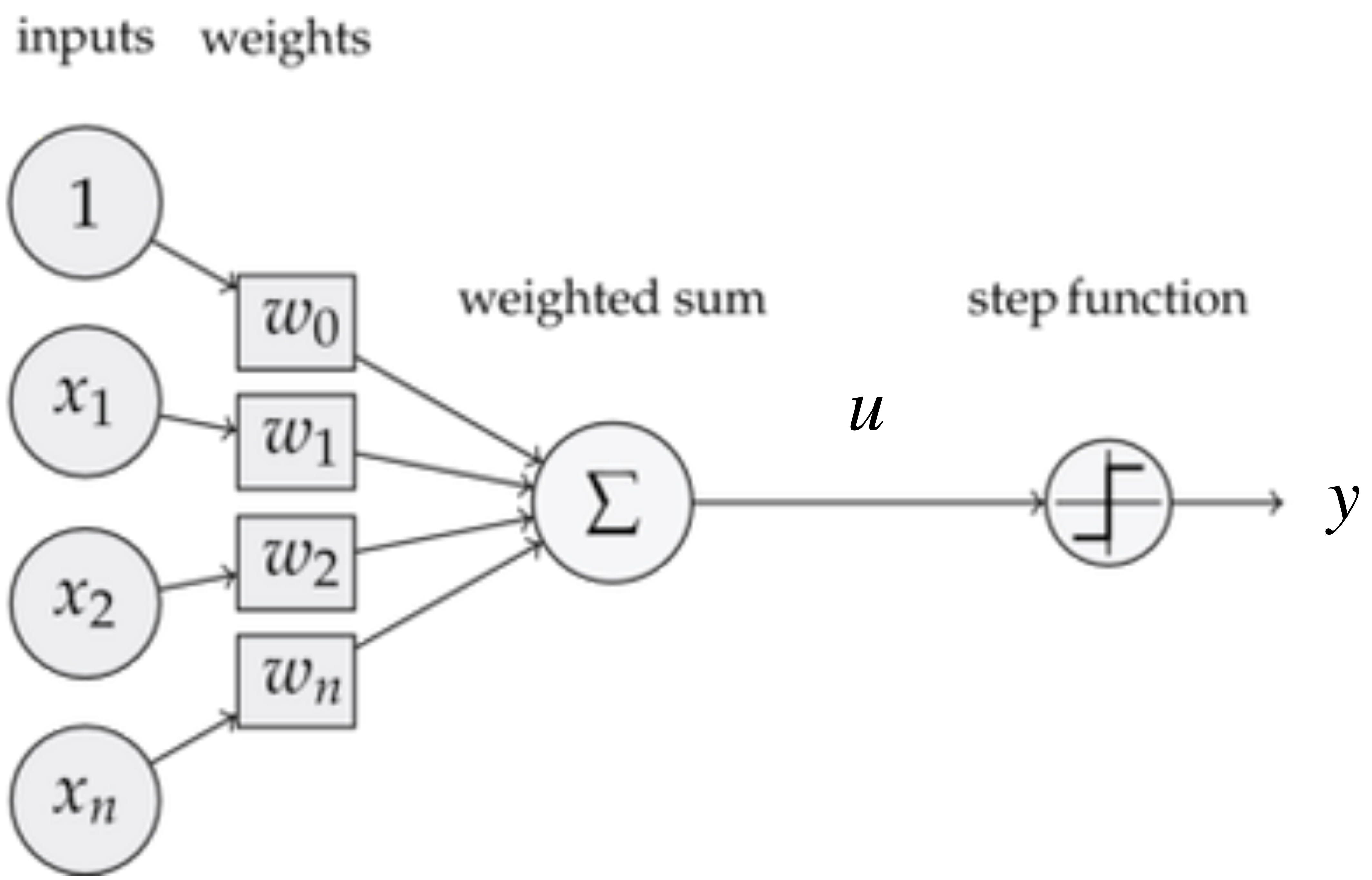
$$\begin{aligned}\mathscr{MAP}(\mathbf{X}, \boldsymbol{\theta}) &= \prod_{i=1}^{N} f_n(\boldsymbol{\theta} \mid \mathbf{x}_i) \\ &= \prod_{i=1}^{N} \frac{f_n(\mathbf{x}_i \mid \boldsymbol{\theta}) g(\boldsymbol{\theta})}{f(\mathbf{x})} \\ &= \prod_{i=1}^{N} f_n(\mathbf{x}_i \mid \boldsymbol{\theta}) g(\boldsymbol{\theta})\end{aligned}$$

INFERÊNCIA

Como fazer inferências?



Neurônio artificial



- Entrada: Recebem as informações de entrada;
- Pesos sinápticos: Ponderam as informações de entrada;
- Junção aditiva: Combina (soma) as informações ponderadas;
- Função de ativação: Despenha o papel de excitação/inibição da informação processada;
- Saída: Ponto de conexão para outros neurônios.

$$u = w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_n x_n = \mathbf{w}^\intercal \mathbf{x} = \mathbf{x}^\intercal \mathbf{w} = \sum_i x_i w_i$$

$$y = \phi(u) = \phi(\mathbf{x}^\intercal \mathbf{w})$$

Modelo Linear Geral

- Encontrar coeficientes pros dados, os pesos, $\mathbf{w}$ que minimize o erro empírico, i.e., o de treinamento:

$$\mathbf{w}^\star = \arg\min_{\mathbf{w}} L\left(f(\mathbf{X}), \mathbf{y}\right), \text{ de modo que } f(\mathbf{x}) = \mathbf{w}^\intercal \mathbf{x};$$

- Geralmente, escolhe-se $L\left(f(\mathbf{X}), \mathbf{y}\right) = \sum_{i=1}^{N} \left(f(\mathbf{x}_i) - y_i\right)^2 = \sum_{i=1}^{N} (e_i)^2$;
- E ainda, assume-se que os erros são i.i.d. e pertecem a uma distribuição normal com $\mu = 0$ e σ^2 desconhecida, .i.e., $e_i \sim \mathcal{N}\left(0, \sigma^2\right)$;
- Assim, vem um truque de re-parametrização (por padrão):

$$e \sim \mathcal{N}\left(\mathbf{w}^\intercal \mathbf{x} - y, \sigma^2\right) \text{ ou ainda } y \sim \mathcal{N}\left(\mathbf{w}^\intercal \mathbf{x}, \sigma^2\right);$$

- Assim estabelecemos a seguinte função de verossimilhança (multivariada) $p\left(\mathbf{y} \mid \mathbf{X}, \mathbf{w}, \sigma^2\right)$.

Modelo Linear Geral

- O problema de achar o erro mínimo agora vira o de achar o $\mathbf{w}$ mais provável que transforme todo $\mathbf{x}$ em y;
- Lembre-se que estamos trabalho com essa função de verossimilhança $p\left(\mathbf{y} \mid \mathbf{X}, \mathbf{w}, \sigma^2\right)$;

$$p\left(\mathbf{y} \mid \mathbf{X}, \mathbf{w}, \sigma^2\right) = \prod_{i=1}^{N} \mathcal{N}\left(y_i \mid \mathbf{w}^\intercal \mathbf{x}_i, \sigma^2\right)$$

$$\ln p\left(\mathbf{y} \mid \mathbf{X}, \mathbf{w}, \sigma^2\right) = \sum_{i=1}^{N} \ln \mathcal{N}\left(y_i \mid \mathbf{w}^\intercal \mathbf{x}_i, \sigma^2\right)$$

- $$= \frac{N}{2} \ln \sigma^{-2} - \frac{N}{2} \ln(2\pi) - \frac{1}{\sigma^2} \frac{1}{2} \sum_{i=1}^{N} \left(y_i - \mathbf{w}^\intercal \mathbf{x}_i\right)^2$$

Ver a seção 3.1.1 Maximum likelihood and least squares do livro do Bishop

Modelo Linear Geral

- Encontrar coeficientes pros dados, os pesos, $\mathbf{w}$ que minimize o erro empírico, i.e., o erro de treinamento:

$$\mathbf{w}^\star = \arg\min_{\mathbf{w}} \, L\left(\mathbf{X}\mathbf{w}^\intercal, \mathbf{y}\right).$$

- Partindo de $\mathbf{y} \sim \mathcal{N}\left(\mathbf{X}\mathbf{w}^\intercal, \sigma^2\right)$. Dessa conta não temos $\mathbf{w}$ e nem σ;
- $p\left(\mathbf{y} \mid \mathbf{X}, \mathbf{w}, \sigma^2\right) \propto \left(\sigma^2\right)^{-n/2} \exp\left(-\frac{1}{2\sigma^2}\left(\mathbf{y} - \mathbf{X}\mathbf{w}\right)^\intercal \left(\mathbf{y} - \mathbf{X}\mathbf{w}\right)\right)$
- $0 = \sum_{i=1}^{N} y_i \mathbf{x}^\intercal - \mathbf{w}^\intercal \left(\sum_{i=1}^{N} \mathbf{x}\mathbf{x}^\intercal\right)$
- $\mathbf{w}_{\text{ML}} = (\mathbf{X}^\intercal\mathbf{X})^{-1}\mathbf{X}^\intercal\mathbf{y} = \mathbf{X}^\dagger\mathbf{y}$

Esperança Condicional

Esperança

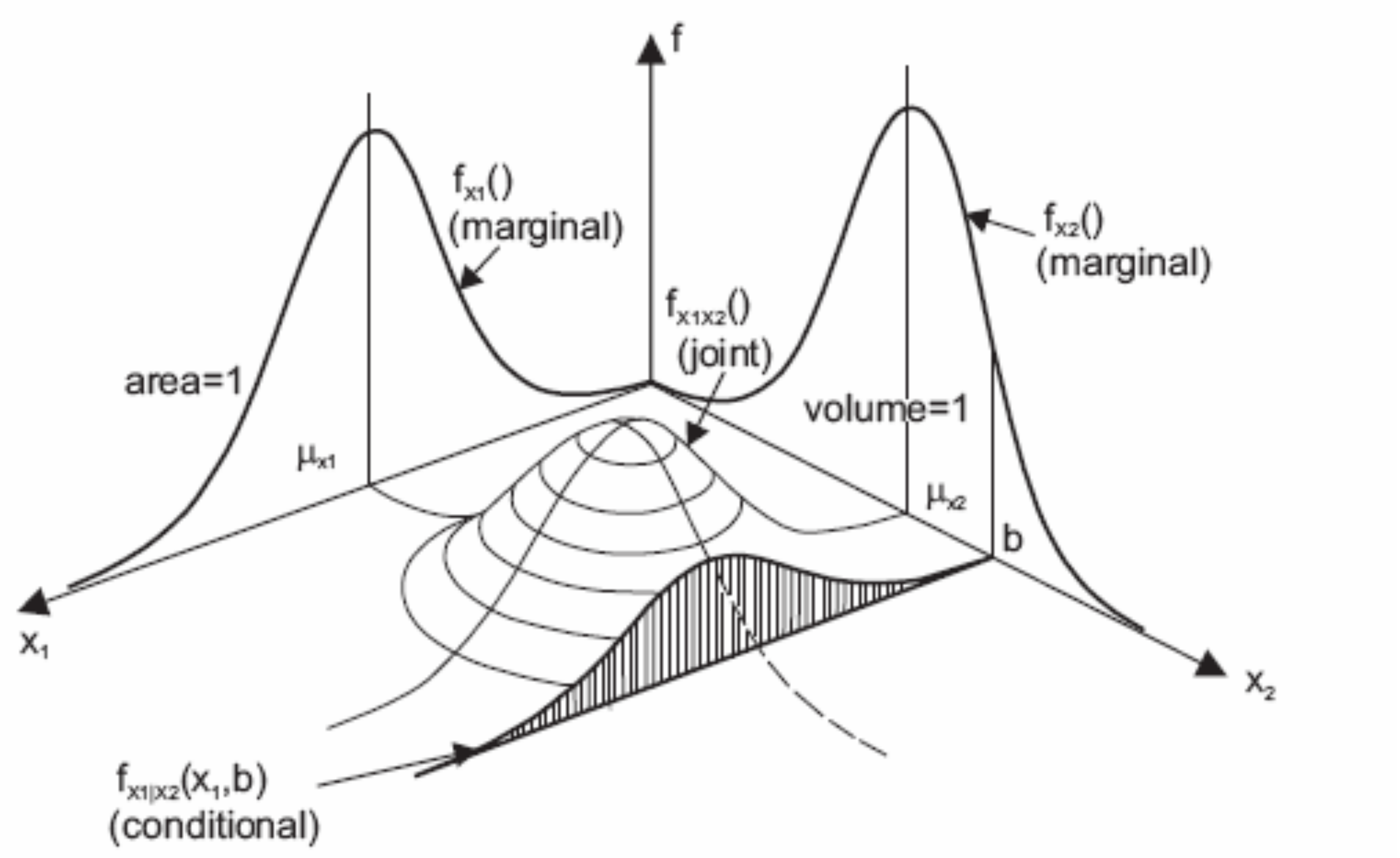
$$\mathbb{E}[x] = \int x\,\mathrm{p}(x)\,\mathrm{d}x$$

$$\mathbb{E}[y|x] = \int y\,\mathrm{p}\left(y \mid x\right)\,\mathrm{d}y$$

$$= \int y\,\frac{\mathrm{p}\left(y, x\right)}{\mathrm{p}\left(x\right)}\mathrm{d}y$$

$$= \int y\,\frac{\sum_{i=1}^{N} \mathbb{K}_h(x - x_i)\,\mathbb{K}_h(y - y_i)}{\sum_{i=1}^{N} \mathbb{K}_h(x - x_i)}\mathrm{d}y$$

$$= \frac{\sum_{i=1}^{N} \mathbb{K}_h(x - x_i)\int y\,\mathbb{K}_h(y - y_i)\,\mathrm{d}y}{\sum_{i=1}^{N} \mathbb{K}_h(x - x_i)} = \frac{\sum_{i=1}^{N} y_i\,\mathbb{K}_h(\mathbf{x} - \mathbf{x}_i)}{\sum_{i=1}^{N} \mathbb{K}_h(\mathbf{x} - \mathbf{x}_i)}$$



Nataraya-Watson kernel estimator

Propriedades

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

$$\mathrm{p}\left(y, x\right) \approx \sum_{i=1}^{N} \mathbb{K}_h(x - x_i)\,\mathbb{K}_h(y - y_i)$$

$$\mathrm{p}\left(x\right) \approx \sum_{i=1}^{N} \mathbb{K}_h(x - x_i)$$

Função de kernel

$$\int \mathbb{K}(u)\,\mathrm{d}u = 1 \quad \text{(normalização)}$$

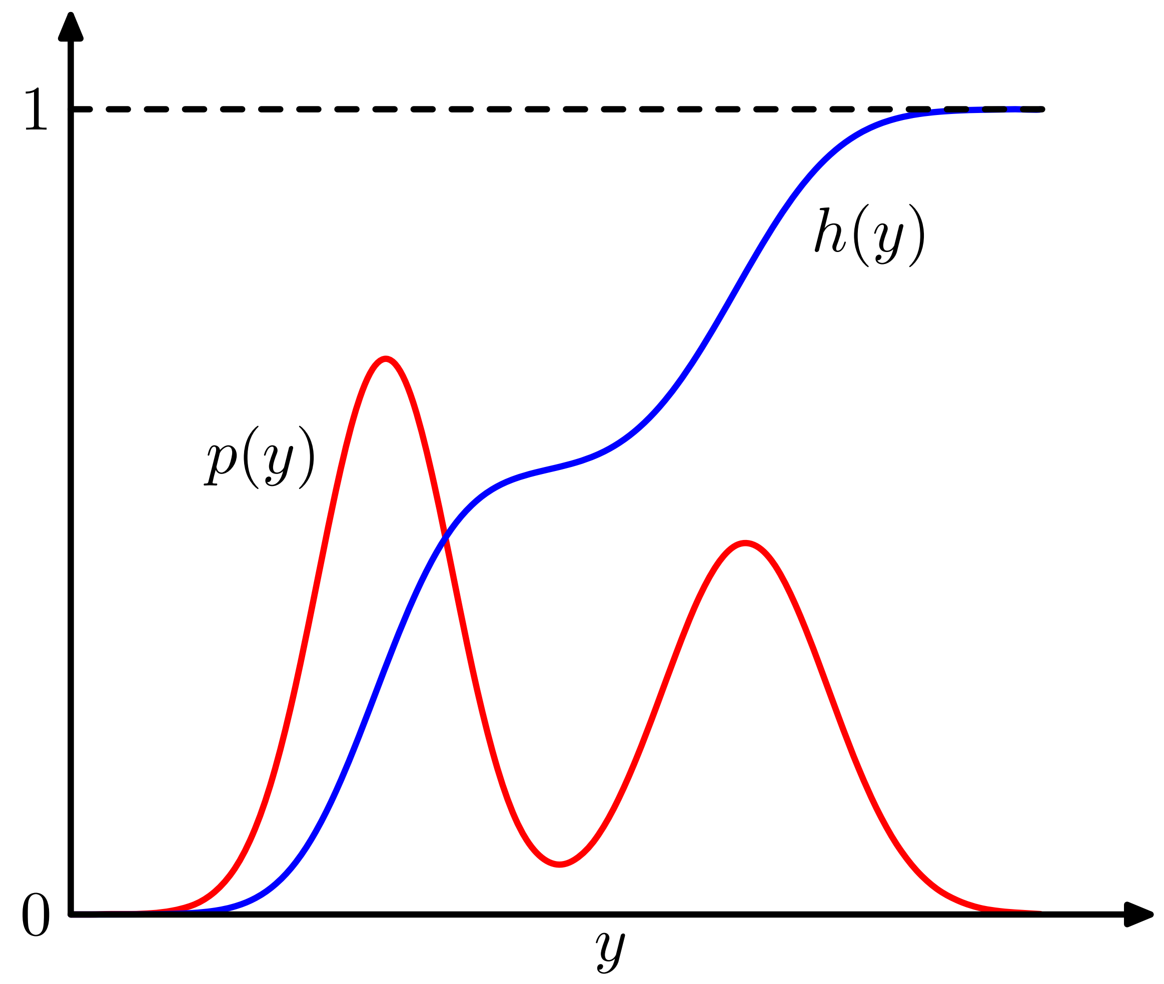
$$\mathbb{K}(u) = \mathbb{K}(-u) \quad \text{(simetria)}$$

AMOSTRAGEM

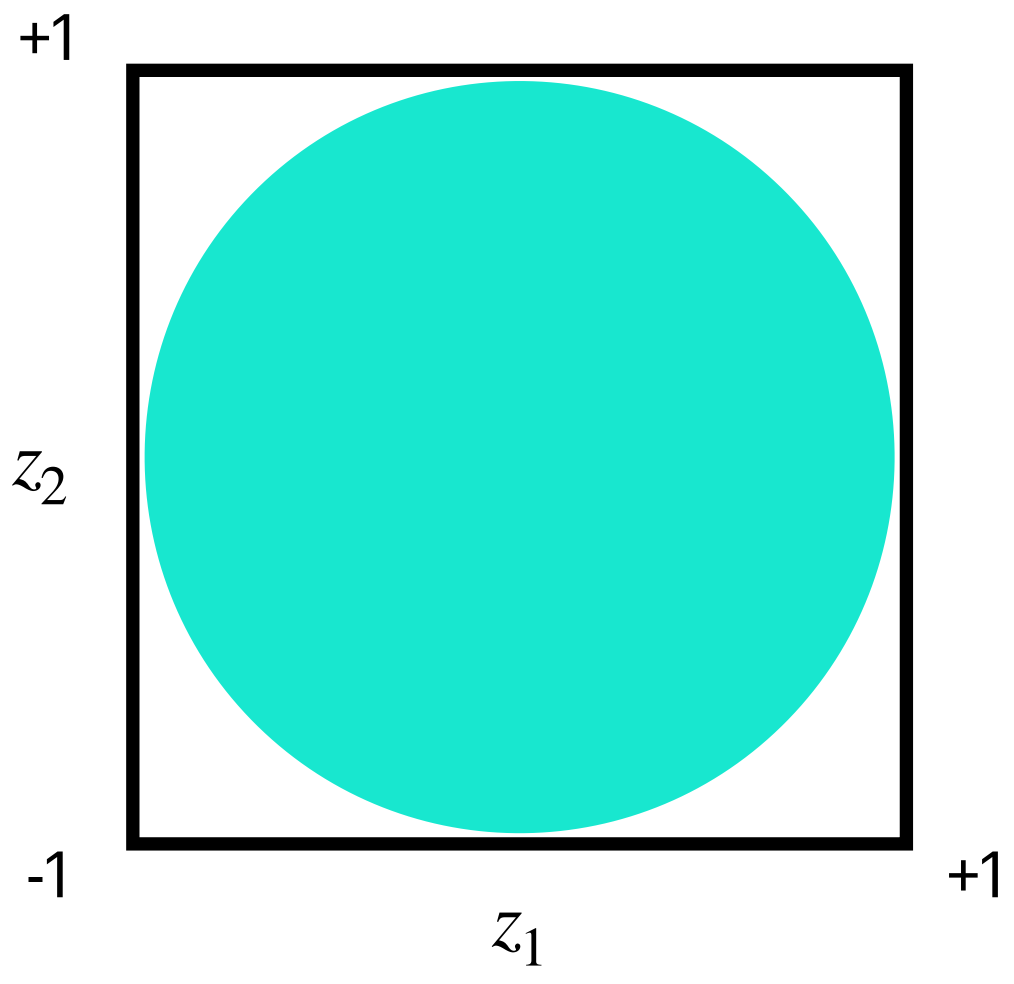
Amostragem por transformação

Usando as clássicas

- $z \sim \mathcal{U}(0,1)$, então procura-se uma função $f(z) = y$, sendo y uma v.a. de distribuição conhecida;
- Exemplo:
  - Se $y \sim \text{Exp}(\lambda)$,
  - Tem-se $p(y) = \lambda \exp(-\lambda y)$,
  - Logo $y = -\lambda^{-1}\ln(1-z)$.



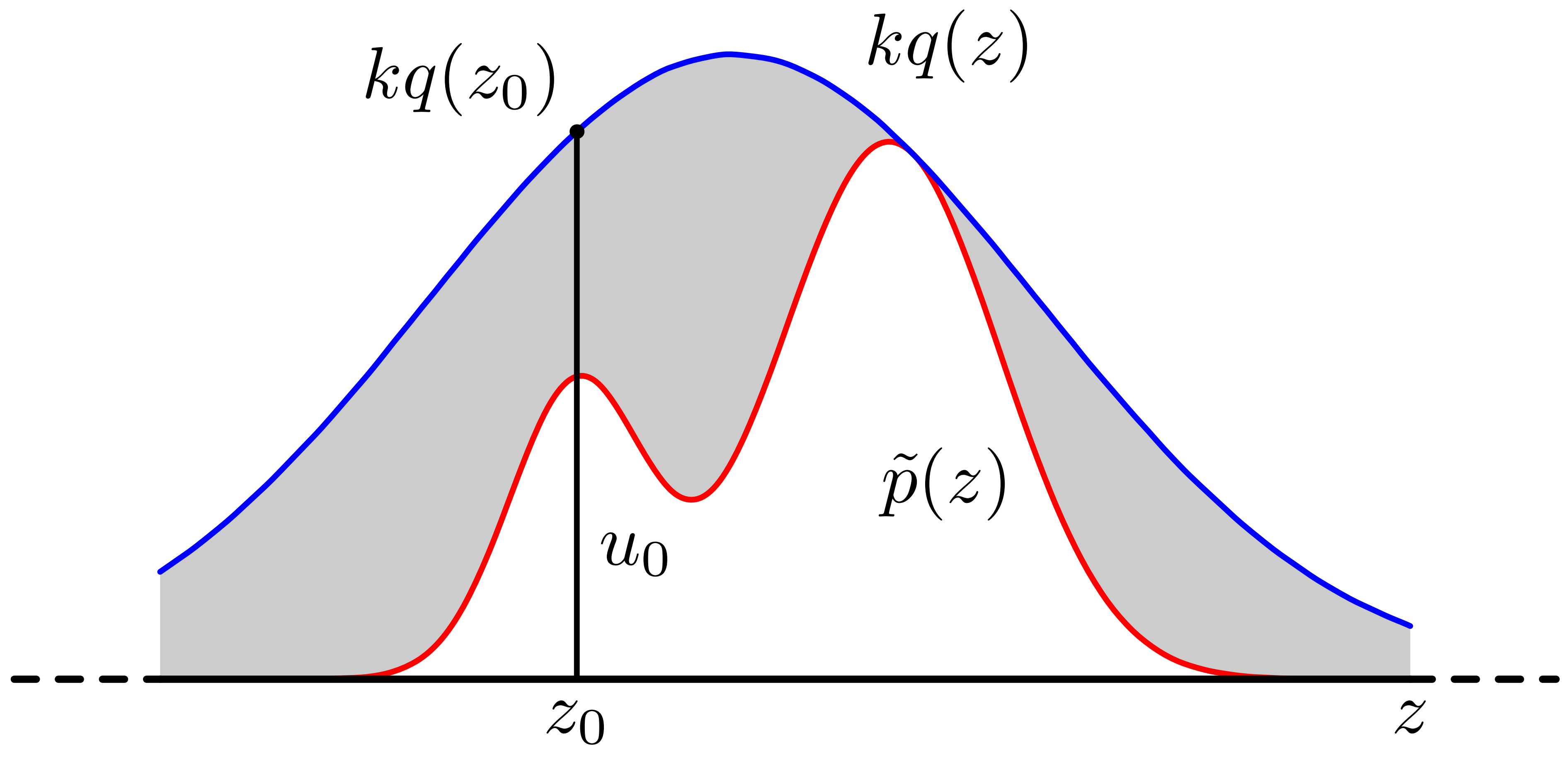
h(y) é a CDF.



Método de Box-Muller

Amostragem por Rejeição

- Suponha que $p(z)$ não é uma das clássicas, de difícil amostragem, mas fácil de verificar;
- Suponha uma outra distribuição $q(\,\cdot\,)$ que é de fácil amostragem;
- Damos o nome de distribuição de propósito a $q(\,\cdot\,)$;
- Depois, escolhe-se um k constante, de modo que $kq(z) \geq p(z)$;



- São geradas duas amostras aleatórias;
  - z_0 é gerado a partir de $q(z)$;
  - $u_0 \sim \text{Uniform}\big(0, kq(z_0)\big)$;
- Só se aceita a amostra se $p(z_0) \geq u_0$.

IMPUTAÇÃO

Pré-conceitos

	col1	col2	col3	col4	col5
0	2	5.0	3.0	6	NaN
1	9	NaN	9.0	0	7.0
2	19	17.0	NaN	9	NaN

↓

	col1	col2	col3	col4	col5
0	2.0	5.0	3.0	6.0	7.0
1	9.0	11.0	9.0	0.0	7.0
2	19.0	17.0	6.0	9.0	7.0

MCAR Missing completely at Random

O fato de os dados estarem faltando é independente dos dados observados e não observados. Em outras palavras, não existem diferenças sistemáticas entre amostras com dados faltantes e aqueles com dados completos.

MNAR Missing Not at Random

O fato de os dados estarem faltando está sistematicamente relacionado aos dados não observados, ou seja, a falta está relacionada a eventos ou fatores que não são medidos pelo pesquisador.

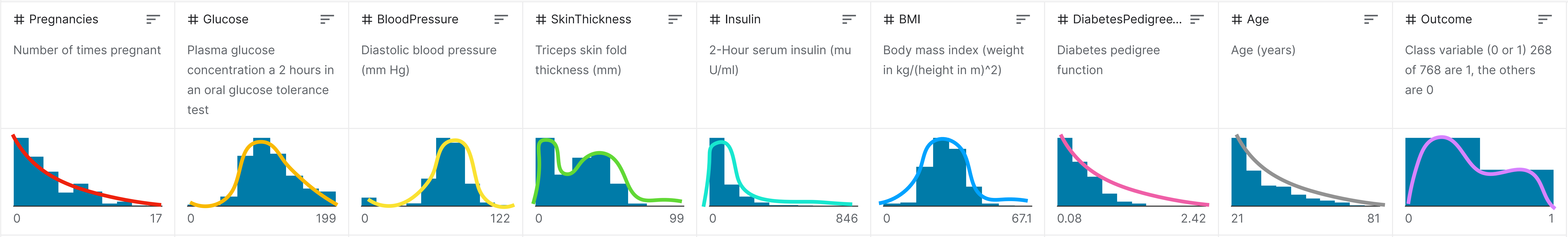
MAR Missing at Random

O fato de os dados estarem ausentes está sistematicamente relacionado aos dados observados, mas não aos não -observados.

https://www.ncbi.nlm.nih.gov/books/NBK493614/

Imputação por Valor Médio

A *Condicional Mean Imputation* (CMI) consiste em estimar uma distribuição de probabilidade para as entradas faltantes em um vetor e usar seu valor médio para preencher as entradas.



$$X_i \sim p_i(\theta_i)$$
$$x_k = \mathbb{E}[X_i]$$

Imputação pelo Vizinho Mais Próximo

A ideia por trás do *Incomplete-case k Nearest Neighbors imputation* (ICkNNI) é usar a distância euclidiana incompleta para selecionar os k vizinhos totalmente observados mais próximos para cada vetor incompleto, preenchendo as entradas faltantes com a média dos vizinhos selecionados.

$$x_{i,d} = \frac{1}{k} \sum_{\mathbf{x}_j} x_{j,d} \text{ de modo que } \mathbf{x}_j \text{ não possui dados faltando na componente } k.$$

Adota-se, para isso, uma função de distância incompleta:

$$d(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} \sqrt{\sum_{d \in O(\mathbf{x}_i)} \left(x_{i,d} - x_{j,d}\right)^2} & \text{se } O(\mathbf{x}_i) \subseteq O(\mathbf{x}_j) \\ \infty, & \text{caso contrário.} \end{cases}$$

A função $O(\,\cdot\,)$ retorna o conjunto de índices observados.

Imputação por Regressão

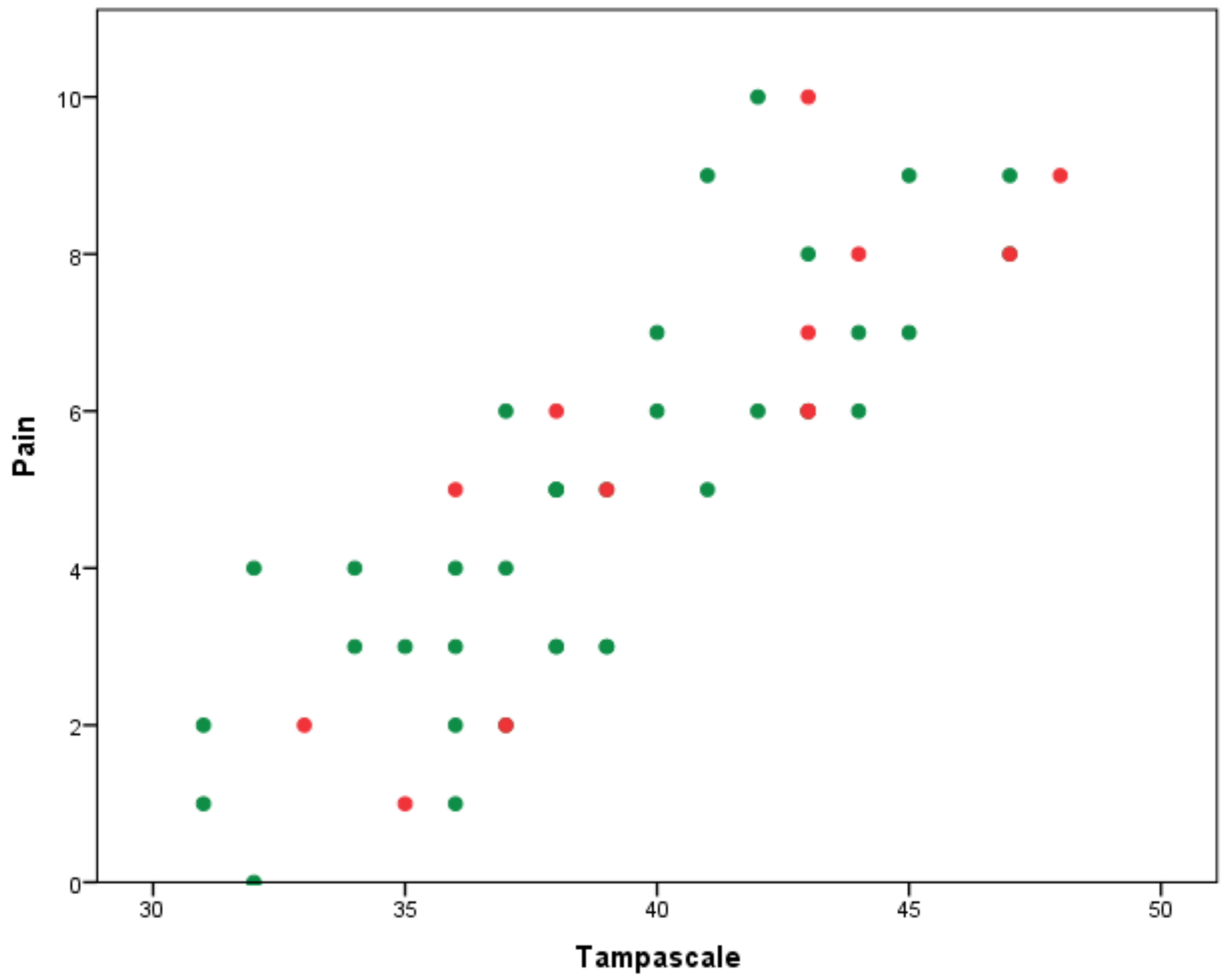
Um modelo de regressão é estimado para prever valores observados de uma variável com base em outras variáveis, e esse modelo é então usado para imputar valores nos casos em que o valor dessa variável está ausente.

$$x_k = w0 + \sum_{d \neq k}^{D} w_d x_d$$

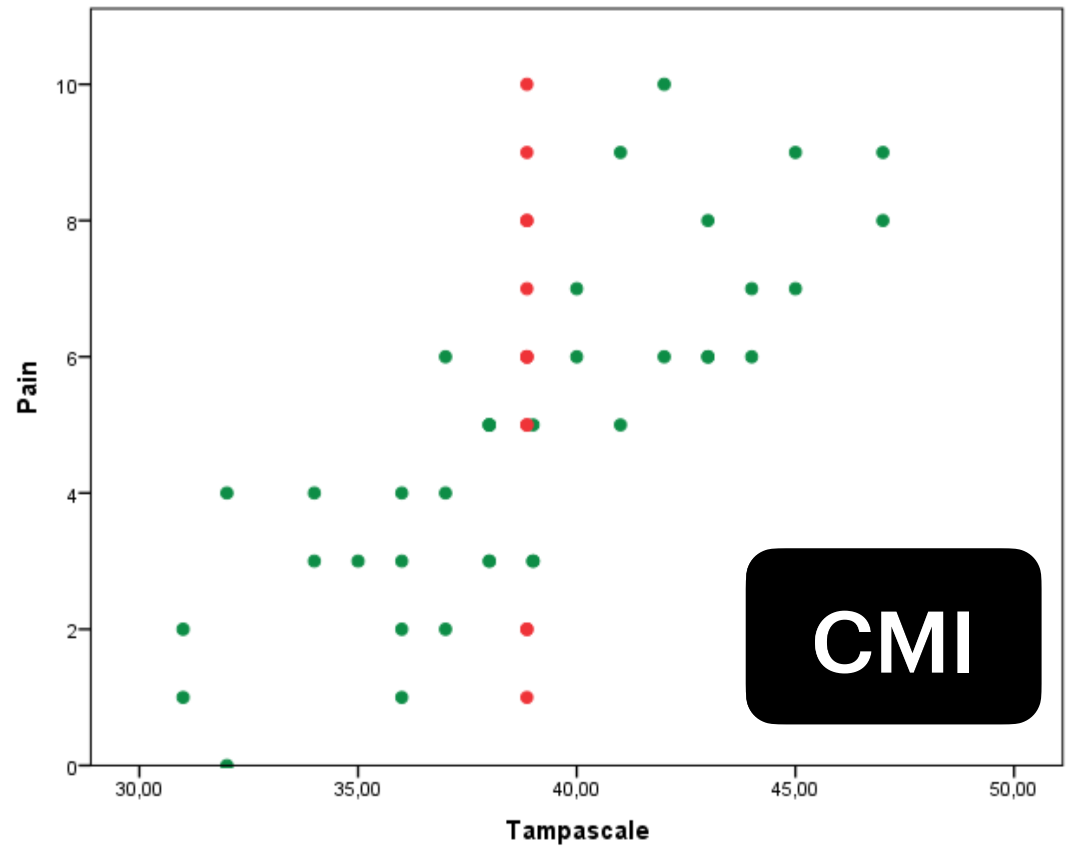
- Treina-se um modelo com os dados observados
- Não há termo de erro na modelagem:
  - Relaciona sejam superidentificados e sugiram maior precisão nos valores imputados do que o garantido;
  - Acaba prevendo o valor mais provável de dados ausentes, mas não fornece incerteza sobre esse valor;
  - Pode-se ajustar ao adicionar um ruído $\mathrm{Normal}\left(0, \sigma^2\right)$).

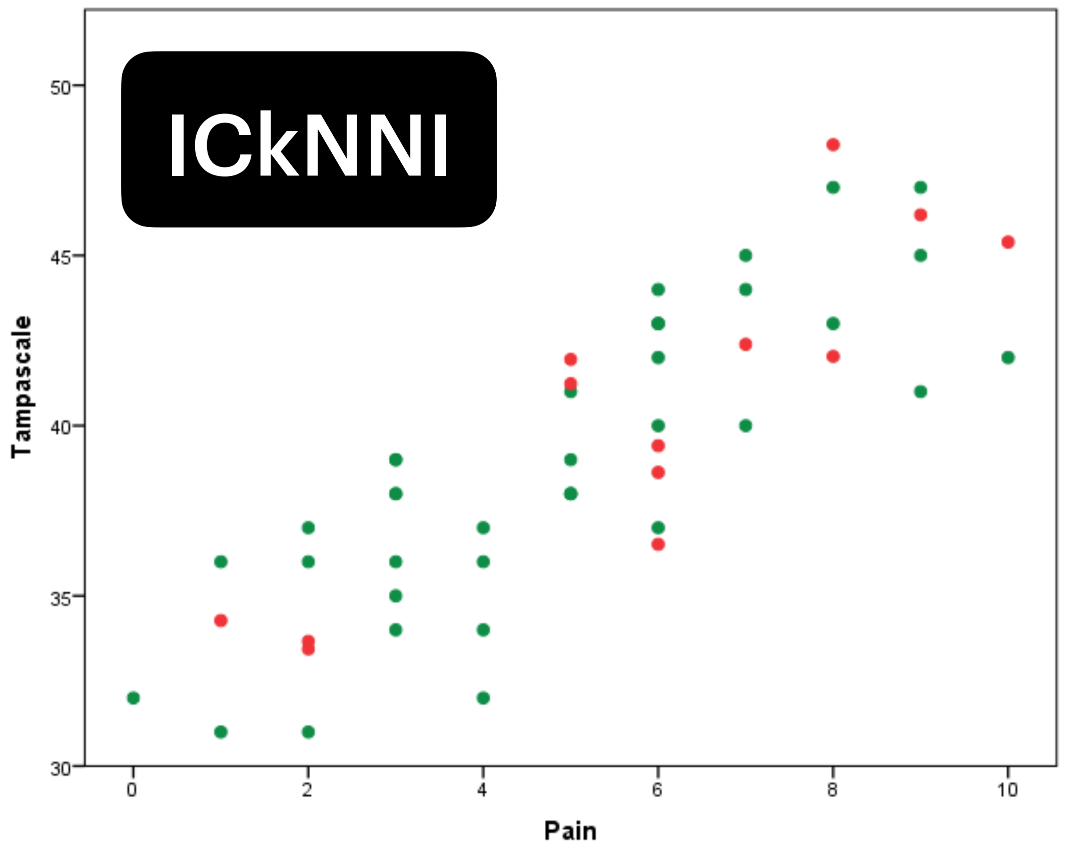
CASOS

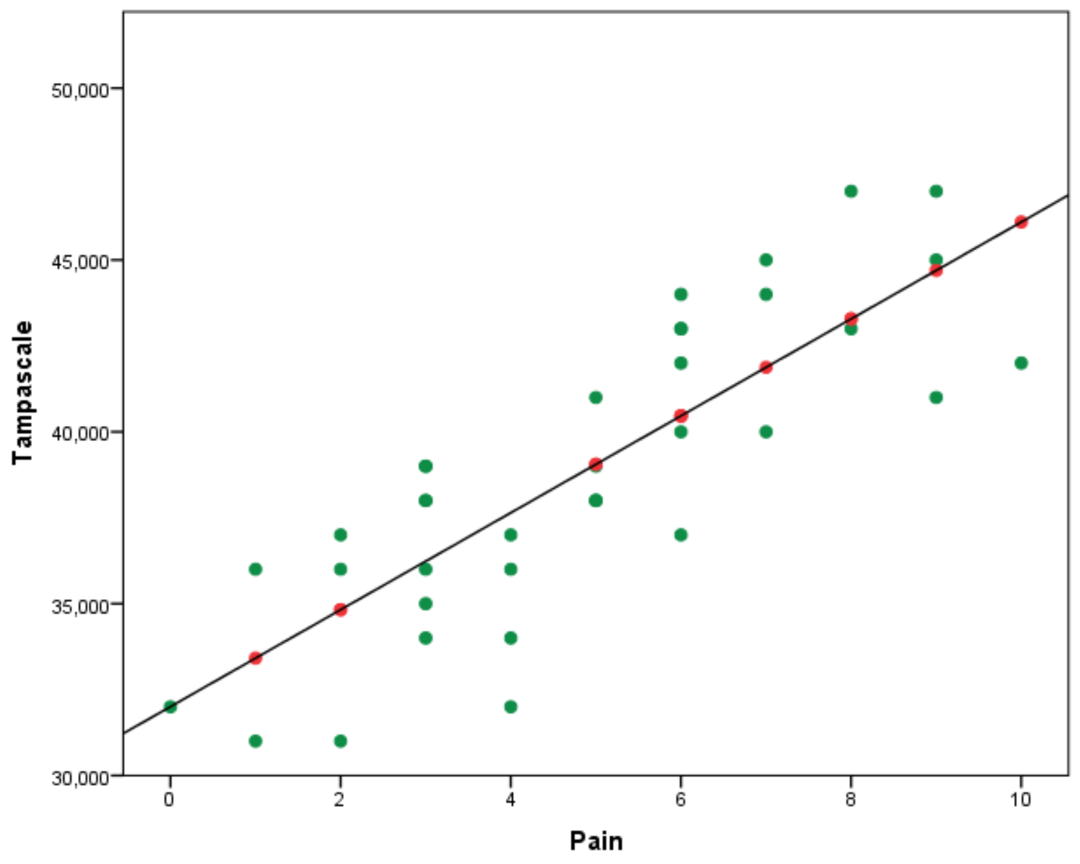
Exemplos de Imputação: Tampascale



- Amostras em vermelho, com dados faltando;
- Amostras em verde são observações completas;







Regresão

https://bookdown.org/mwheymans/bookmi/single-missing-data-imputation.html#regression-imputatio

Referências

- BISHOP, Christopher M.; NASRABADI, Nasser M. **Pattern recognition and machine learning**. New York: springer, 2006.
- Fernando J. Von Zuben. **Notas de Aula: Redes Neurais Artificiais**. DCA/FEEC/Unicamp, 2014. Disponível em: https://www.dca.fee.unicamp.br/~vonzuben/ia013_1s14/notas_de_aula/topico2_IA013_1s2014_Parte2.pdf. Acessado em: 14 de maio de 2022.
- Richard O. Duda, Peter E. Hart, David G. Stork. **Pattern Classification**. John Wiley & Sons, 2012.