ICC204 - Aprendizagem de Máquina e Mineração de Dados

Classificação

(parte 3/3)

Prof. Rafael Giusti
rgiusti@icomp.ufam.edu.br

Agenda

- Parte 1/3
  - Definições
  - Teoria das probabilidades
  - Aprendizado Bayesiano e modelos probabilísticos
- Parte 2/3
  - Modelos baseados em árvores
  - Modelos baseados em regras
- Parte 3/3
  - Classificação preguiçosa: k-NN
  - Máquina de vetores de suporte

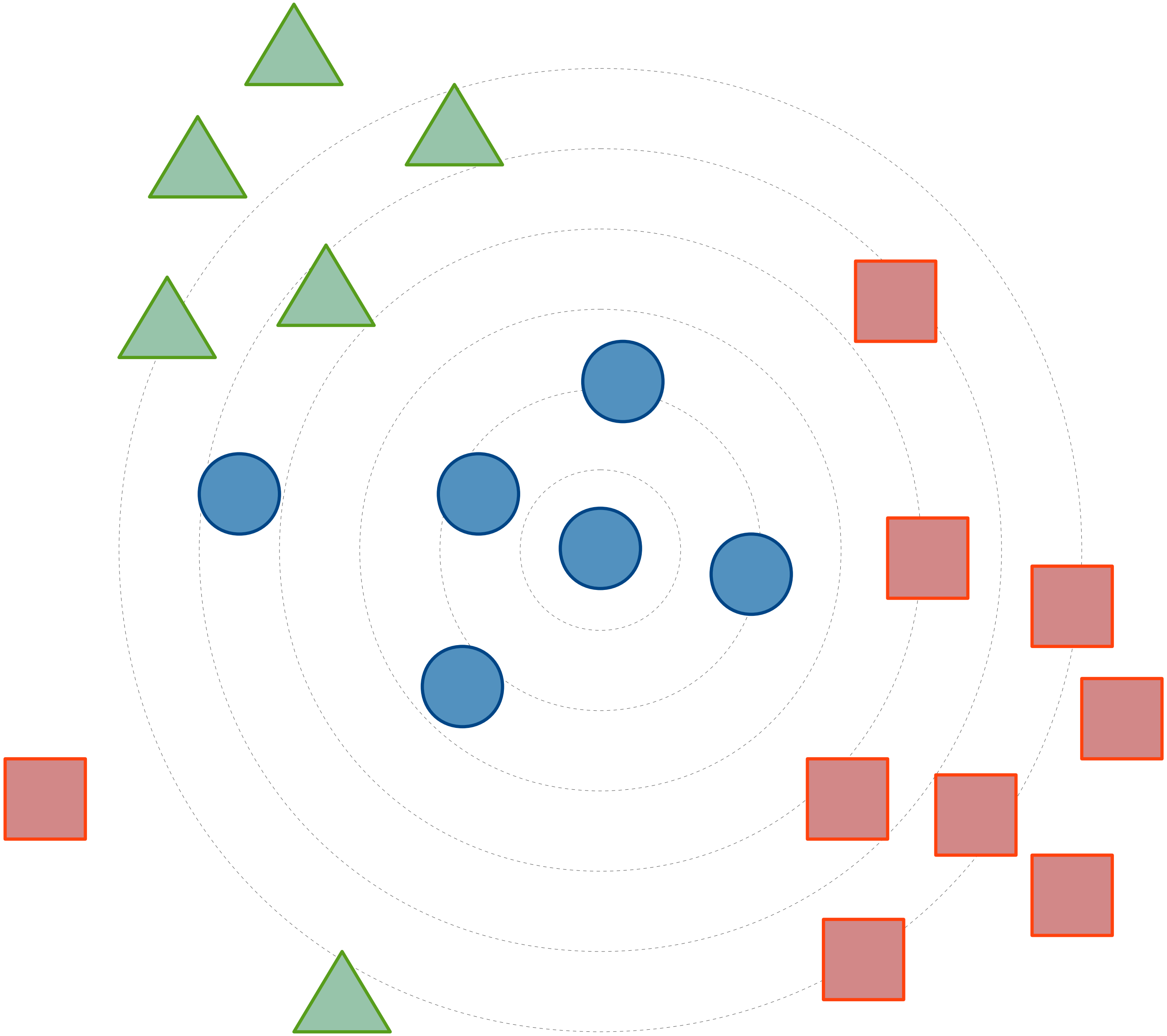
Agenda

- Definições
- Teoria das probabilidades
- Aprendizado Bayesiano e modelos probabilísticos
- Modelos baseados em árvores
- Modelos baseados em regras
- Classificação preguiçosa: k-NN
- Máquina de vetores de suporte

Princípio dos vizinhos mais próximos

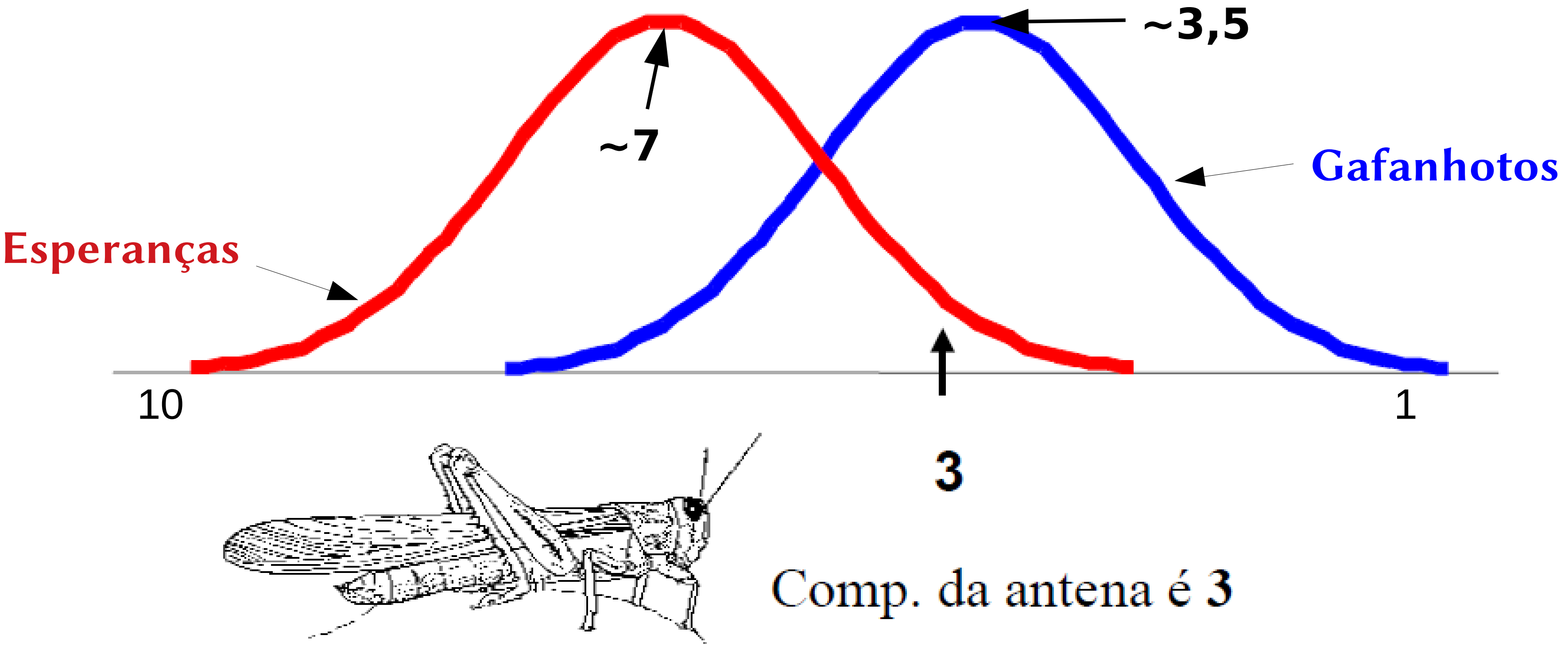
- O classificador k-NN é um classificador **baseado em instâncias** que se utiliza do princípio dos **vizinhos mais próximos**
  - A vizinhança do exemplo **x** = $(x_1, x_2, ..., x_M)$, estabelecida por uma função de distância apropriada, tenderá a ser ocupada por exemplos que pertencem à mesma categoria que **x**

Princípio dos vizinhos mais próximos



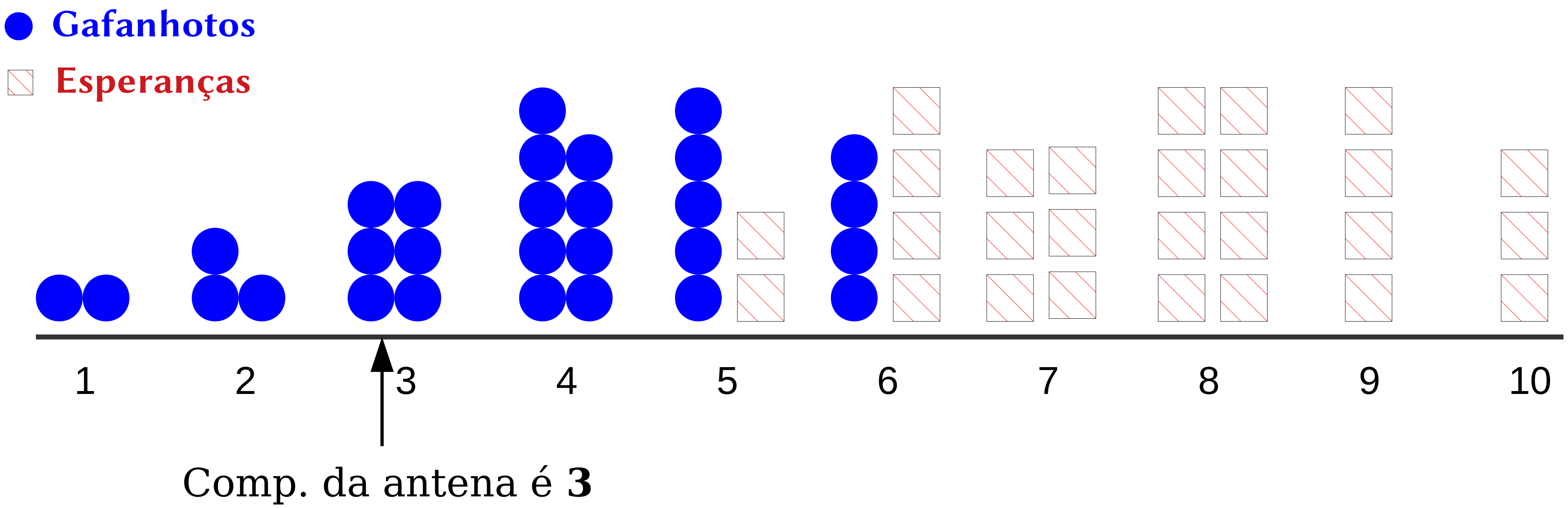
Princípio dos vizinhos mais próximos

- Recorde o caso dos gafanhotos vs. esperanças
- Se tivermos conhecimento da distribuição do atributo, podemos usar essa informação para tomar nossa decisão



Princípio dos vizinhos mais próximos

- Se não tivermos conhecimento dos parâmetros das populações, podemos observar as amostras para obter uma **aproximação**



Obs.: cada ponto equivale a um ponto do slide 36 com os valores arredondados

Classificador k-NN

- Com base nesse princípio, derivamos um classificador **preguiçoso** (*lazy*) chamado k-NN
  - k-*Nearest Neighbors* ou k-Vizinhos mais Próximos
  - Diferentemente dos modemos que vimos até agora, o k-NN **não requer treinamento**
  - Não se induz um modelo k-NN
    - O "modelo" é uma cópia do conjunto de exemplos de treinamento, denominados **protótipos**

Classificador k-NN

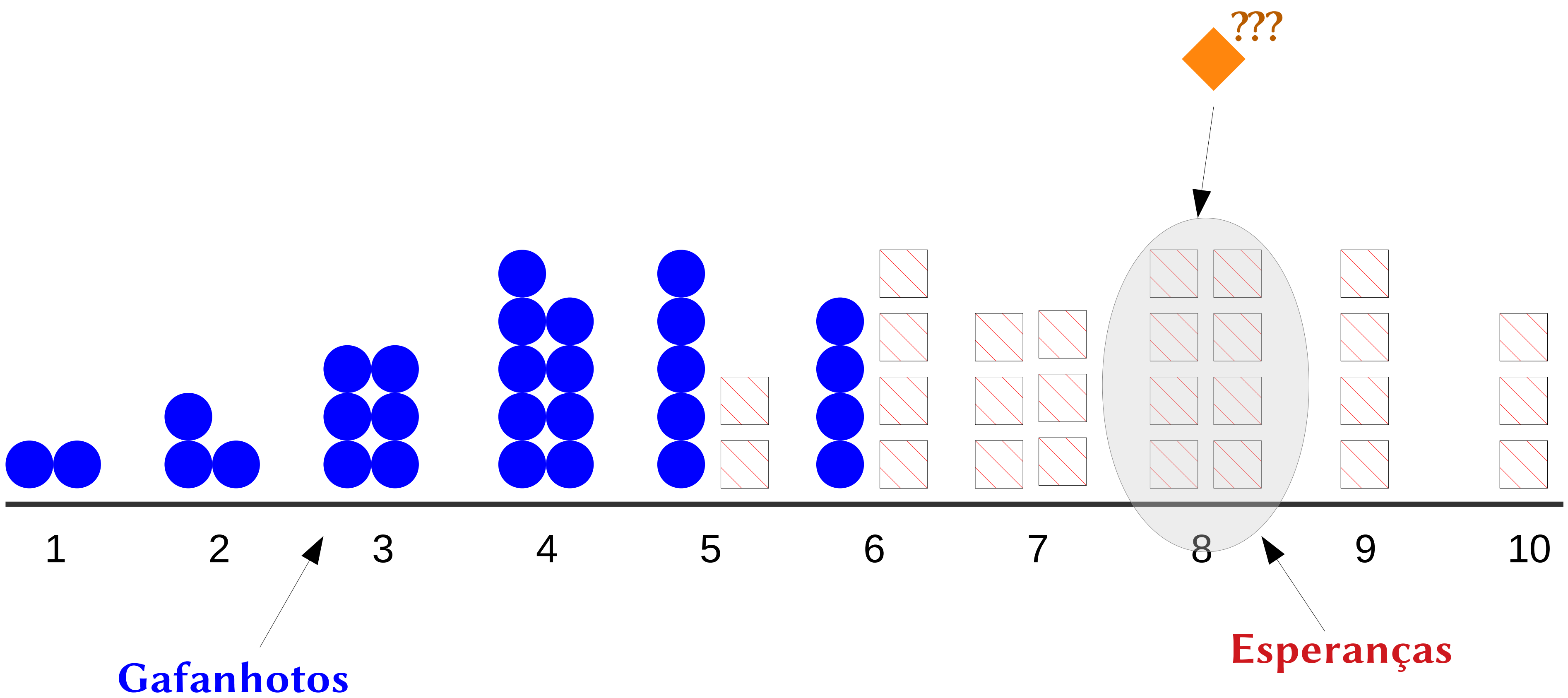
- Para classificar um novo exemplo ($\mathbf{x}_i$, y_i) cuja classe desconhecemos...
  - Selecionamos os k protótipos mais próximos de $\mathbf{x}_i$
  - Observamos as classes y_{p1}, y_{p2}, ..., y_{pk}
  - Derivamos a classe desconhecida y_i com base na informação das classes conhecidas y_{p1}, y_{p2}, ..., y_{pk}
    - Por exemplo, y_i pode ser a moda (valor mais frequente) de y_{pj}

Classificador k-NN

- O valor de *k* é um parâmetro que normalmente definimos com a ajuda de um conjunto de validação
- Quanto maior o valor de *k*, maior é a complexidade do "modelo"
  - Em específico, o classificador 1-NN ou 1NN é aquele que só utiliza um vizinho para classificar
- Em geral o *k*-NN é tipicamente considerado um método sub-ótimo
  - Mas há exceções (ex.: séries temporais)

Exemplo (k=1)

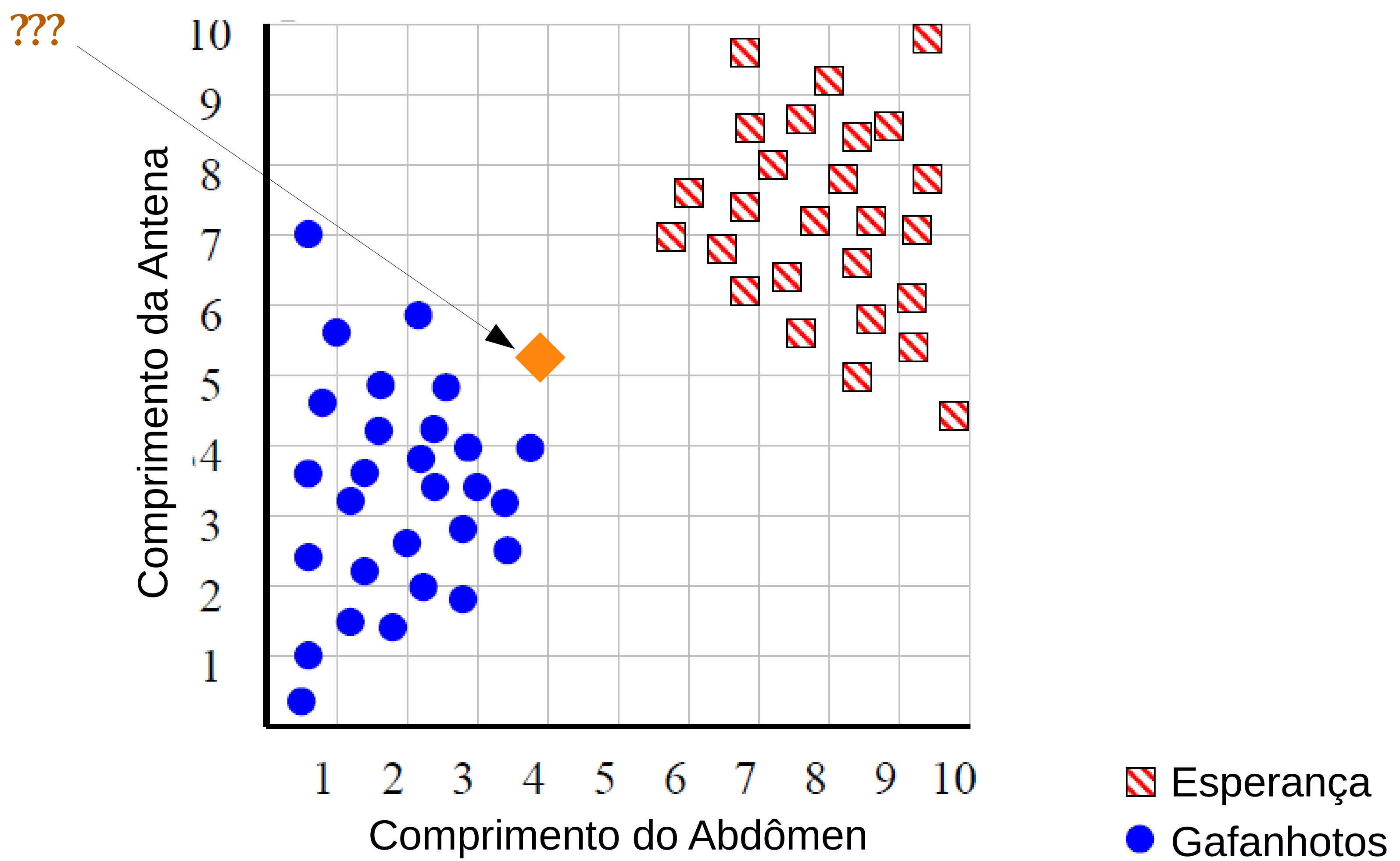
- A qual classe pertence o exemplo que cujo valor é "comprimento da antena" = 8 ?



Obs.: cada ponto equivale a um ponto do slide 36 com os valores arredondados

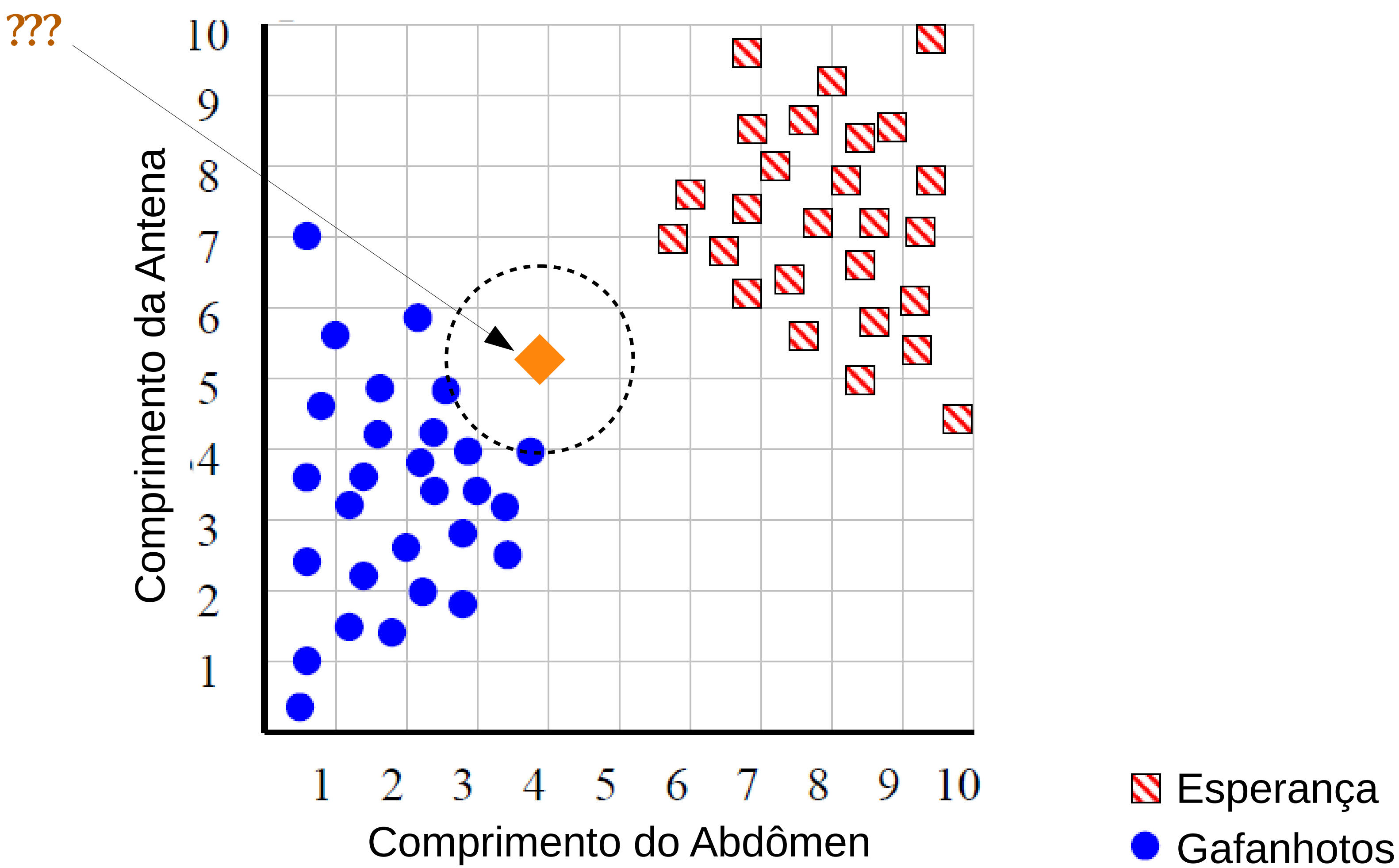
Exemplo (k=1)

- Estendendo para duas dimensões...



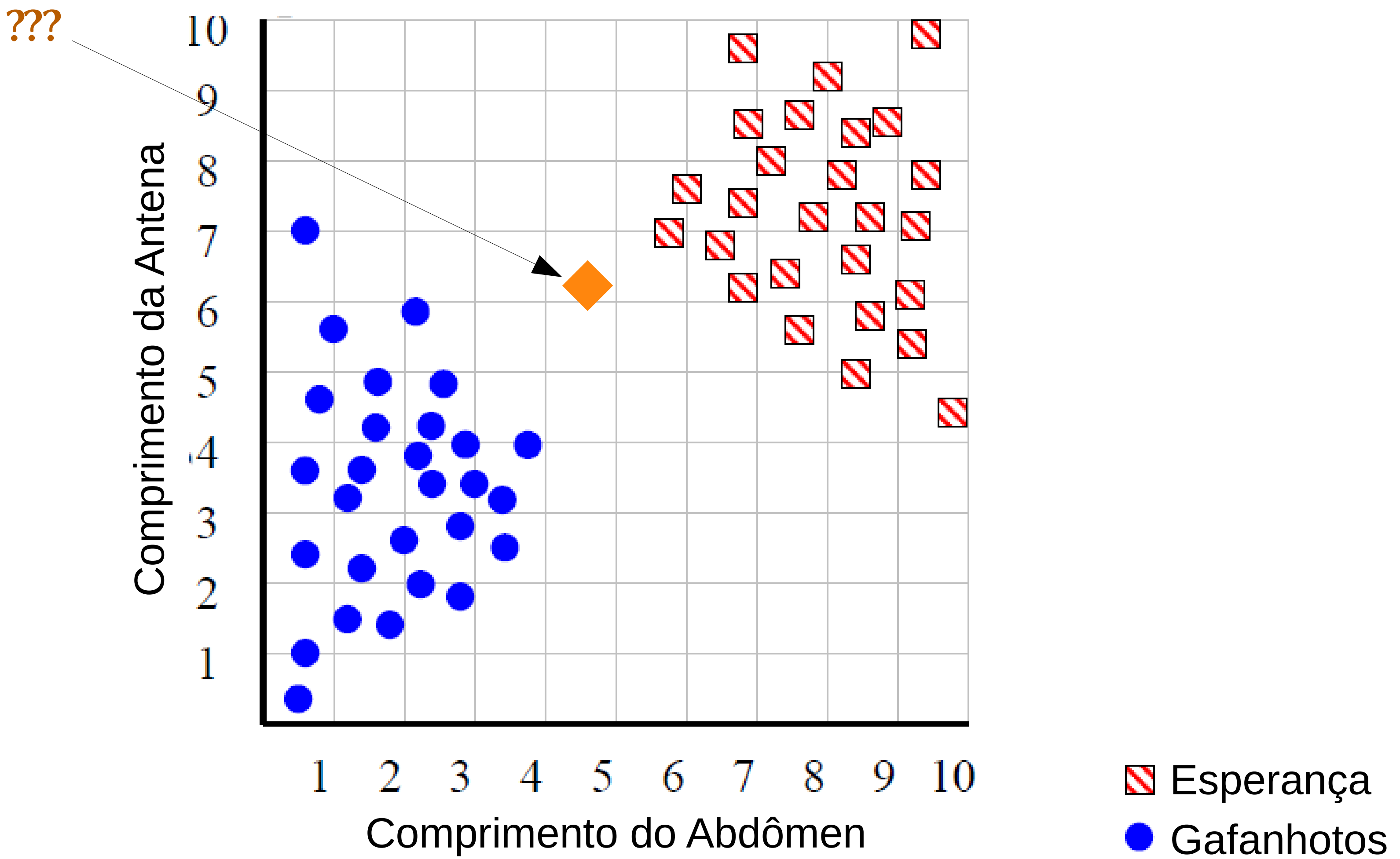
Exemplo (k=1)

- Estendendo para duas dimensões...



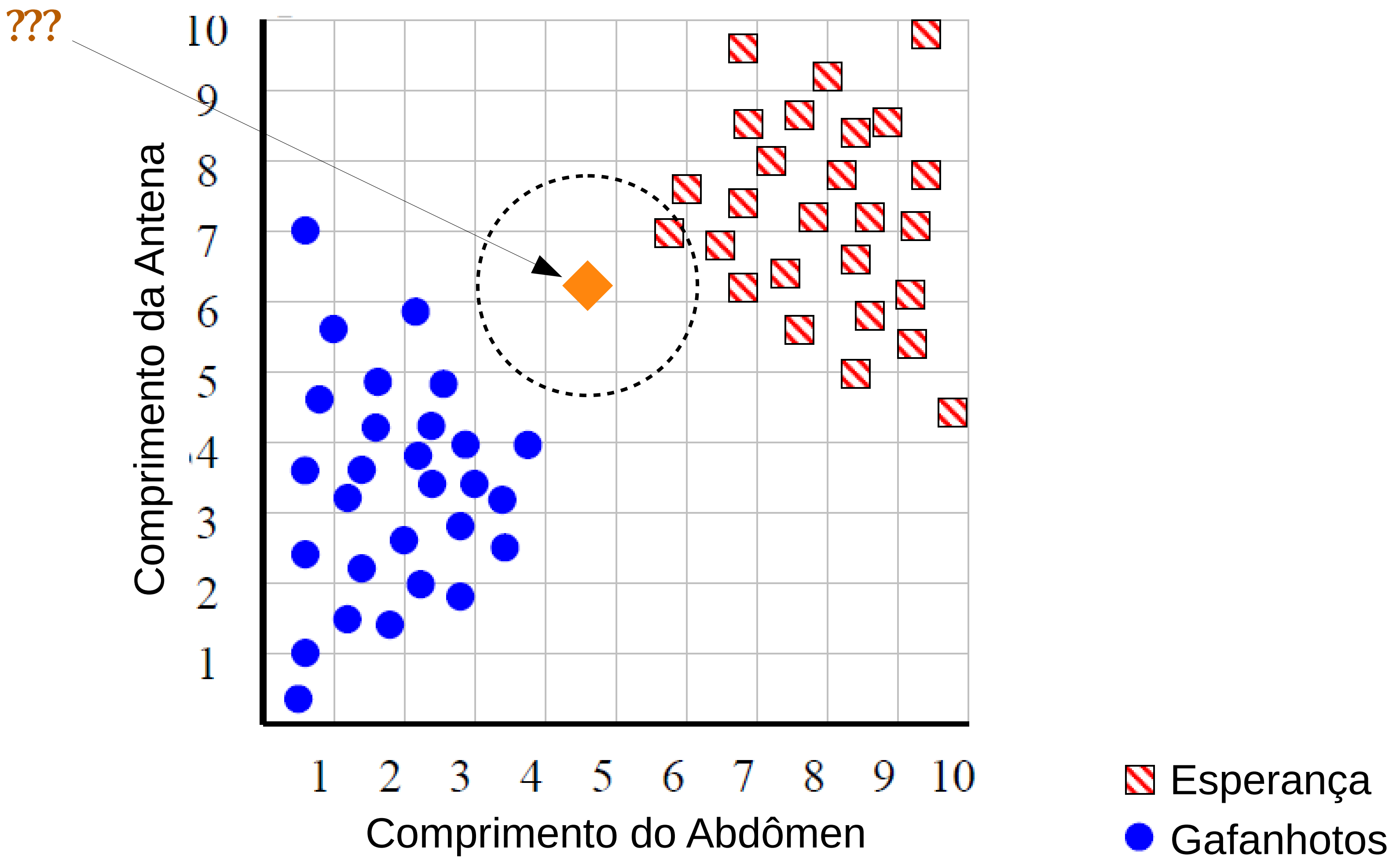
Exemplo (k=1)

- Estendendo para duas dimensões...



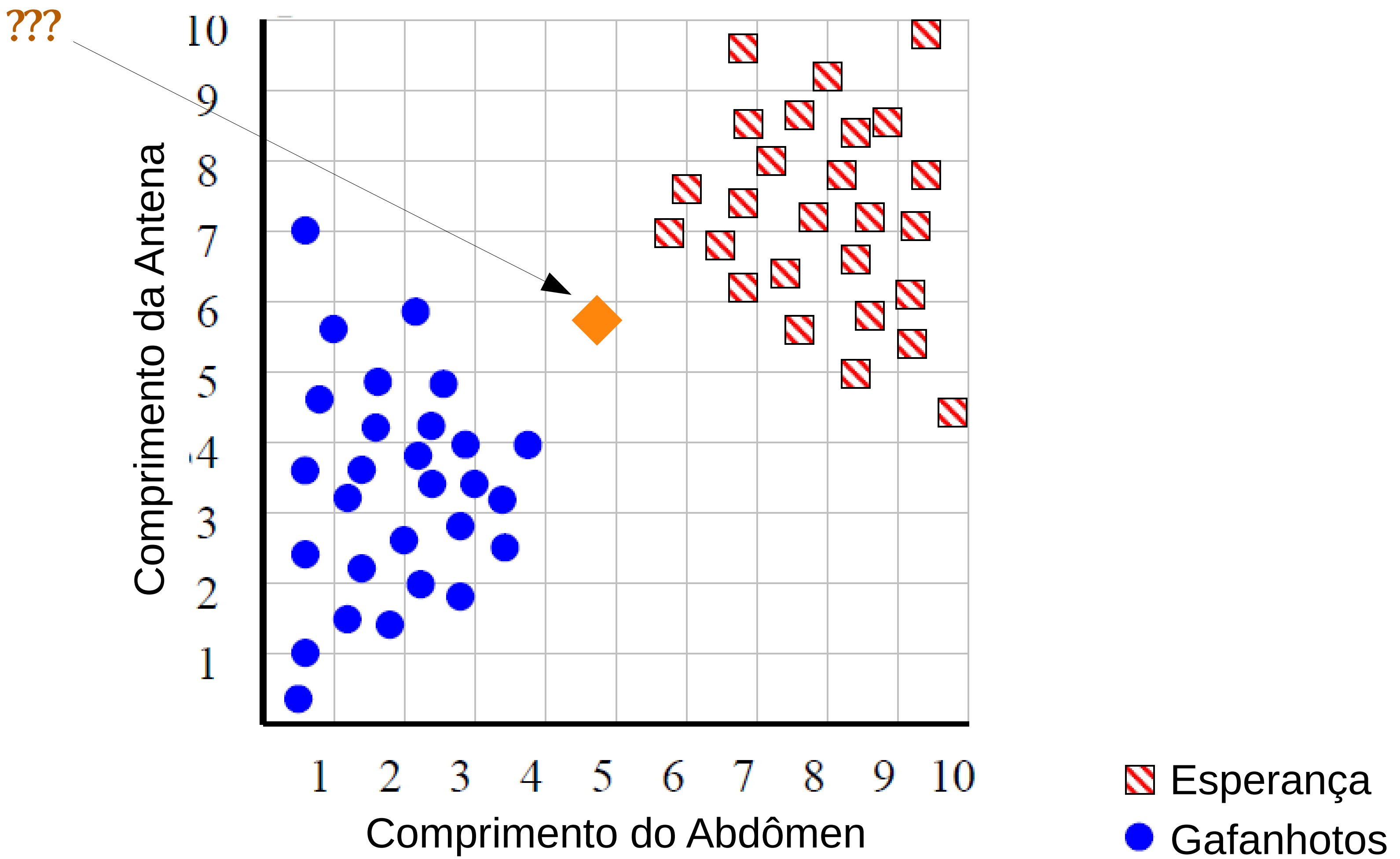
Exemplo (k=1)

- Estendendo para duas dimensões...



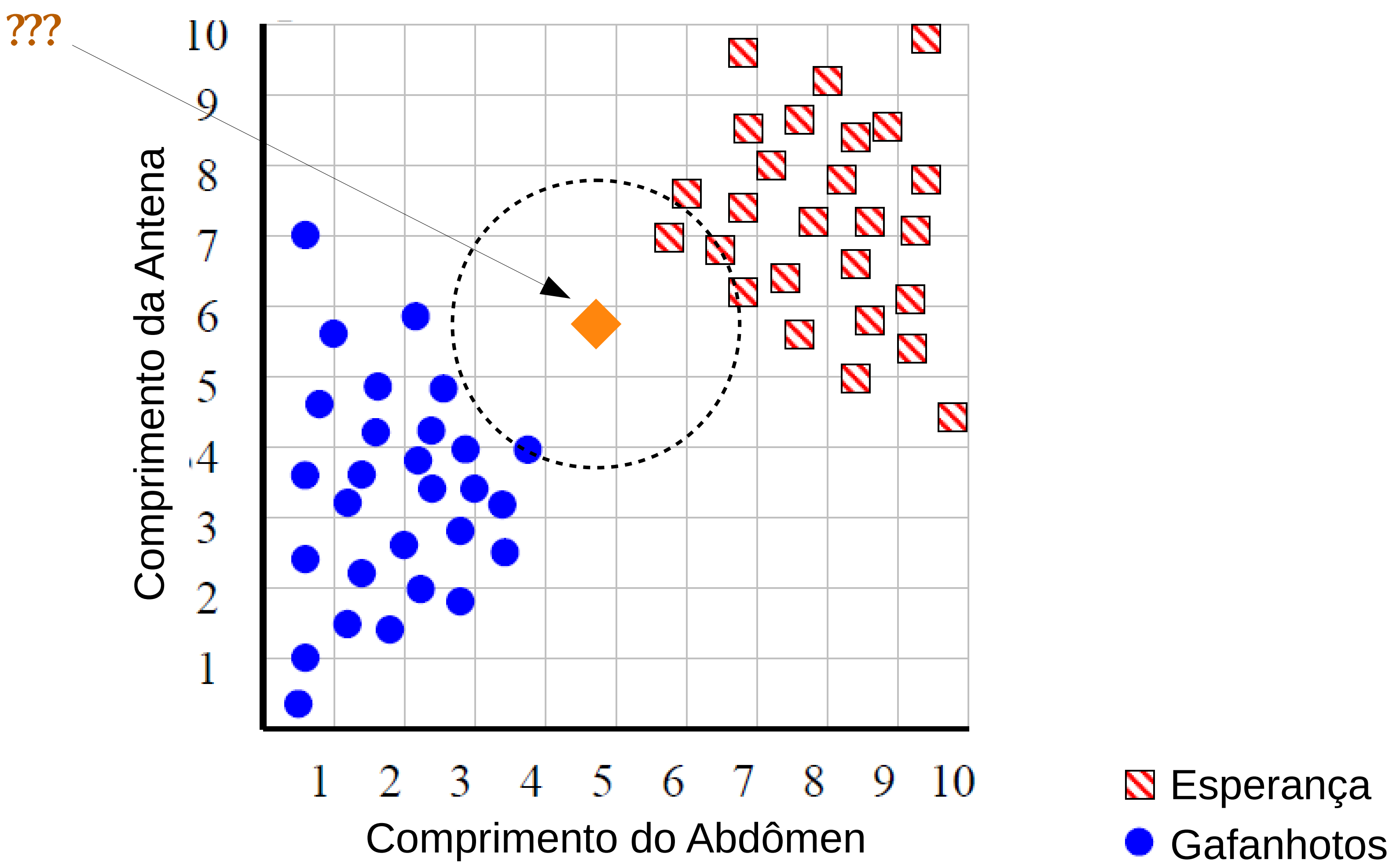
Exemplo (k=3)

- Estendendo para duas dimensões...



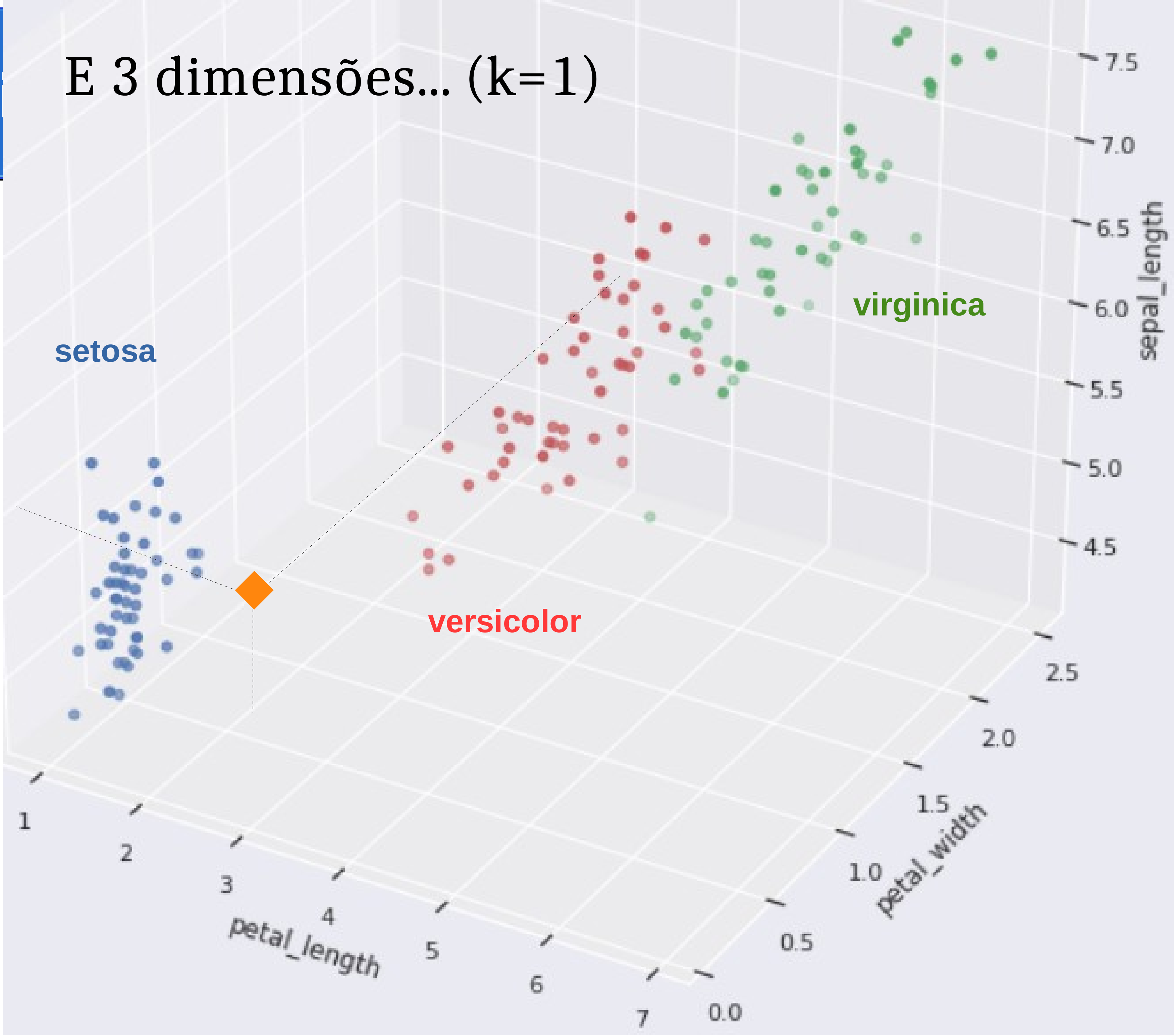
Exemplo (k=3)

- Estendendo para duas dimensões...



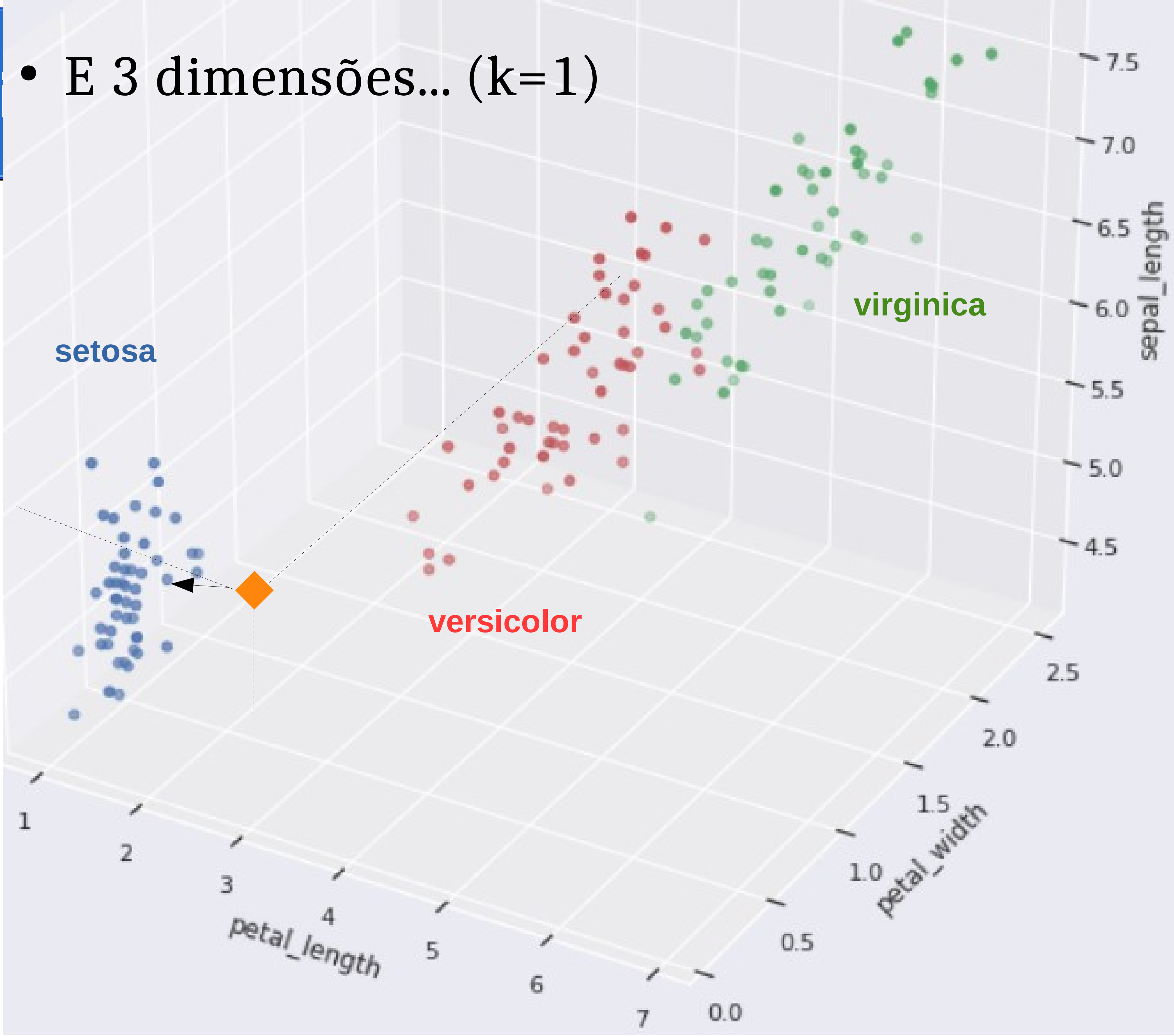
Pr

E 3 dimensões... (k=1)



Pr

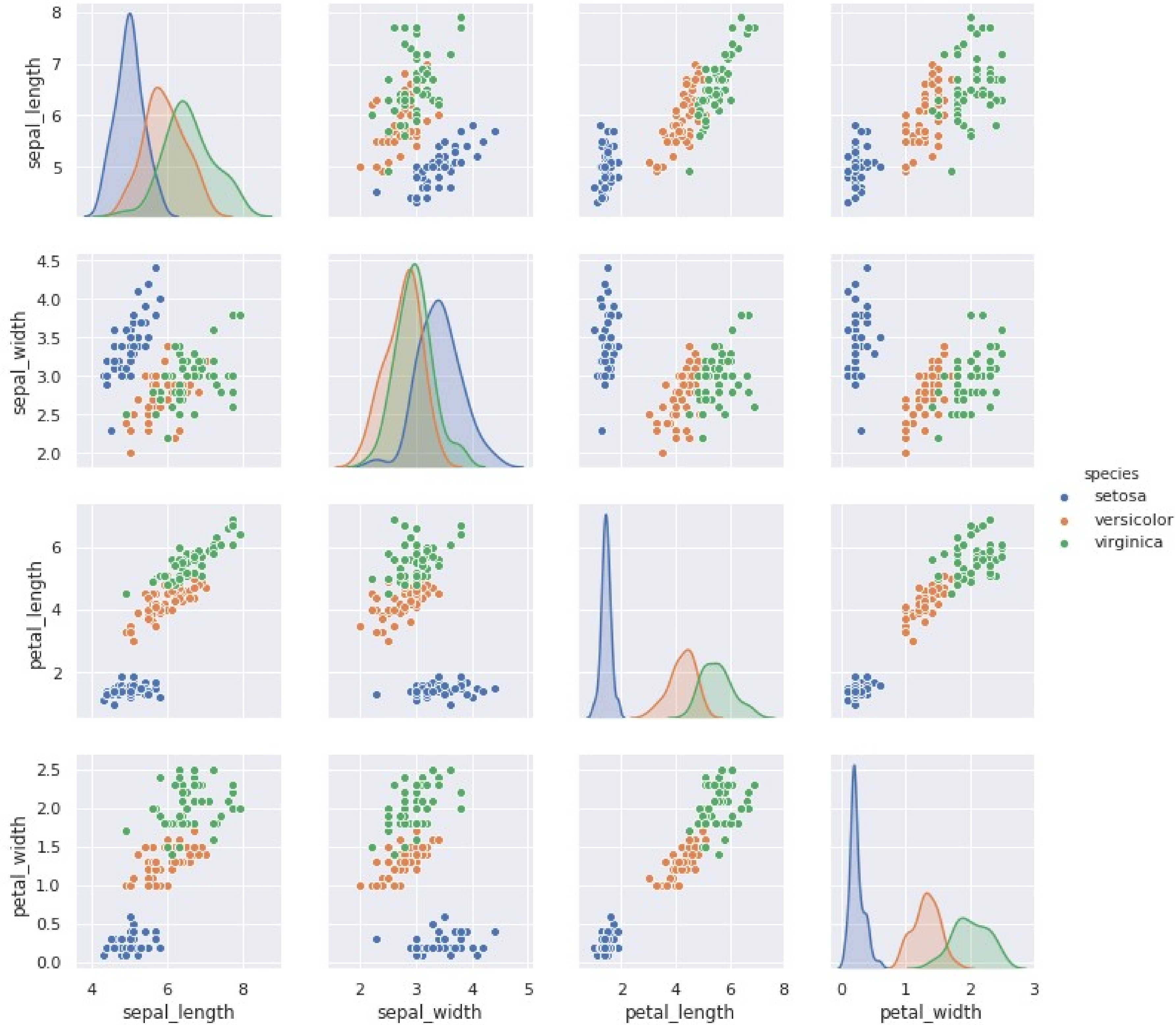
- E 3 dimensões... (k=1)



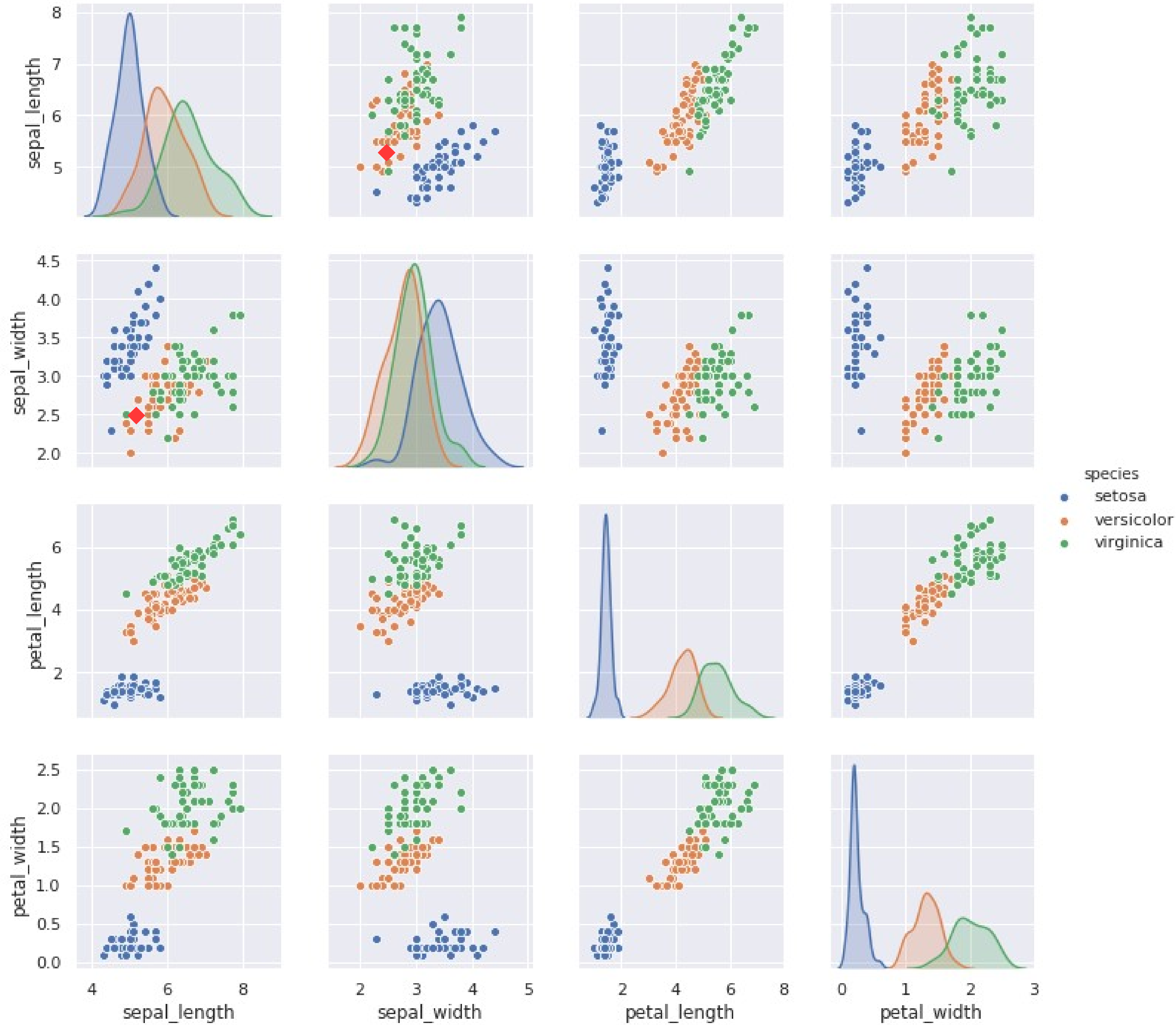
Princípio dos vizinhos mais próximos

- É impossível visualizarmos um espaço de quatro dimensões...
- Mas podemos empregar ferramentas como a visualização em pares para obter uma intuição desses espaços
  - Como podemos visualizar um exemplo de classe desconhecida com os atributos abaixo?
    - sepal_length=5.3, sepal_width=2.5, petal_length=1.4, petal_width=0.6

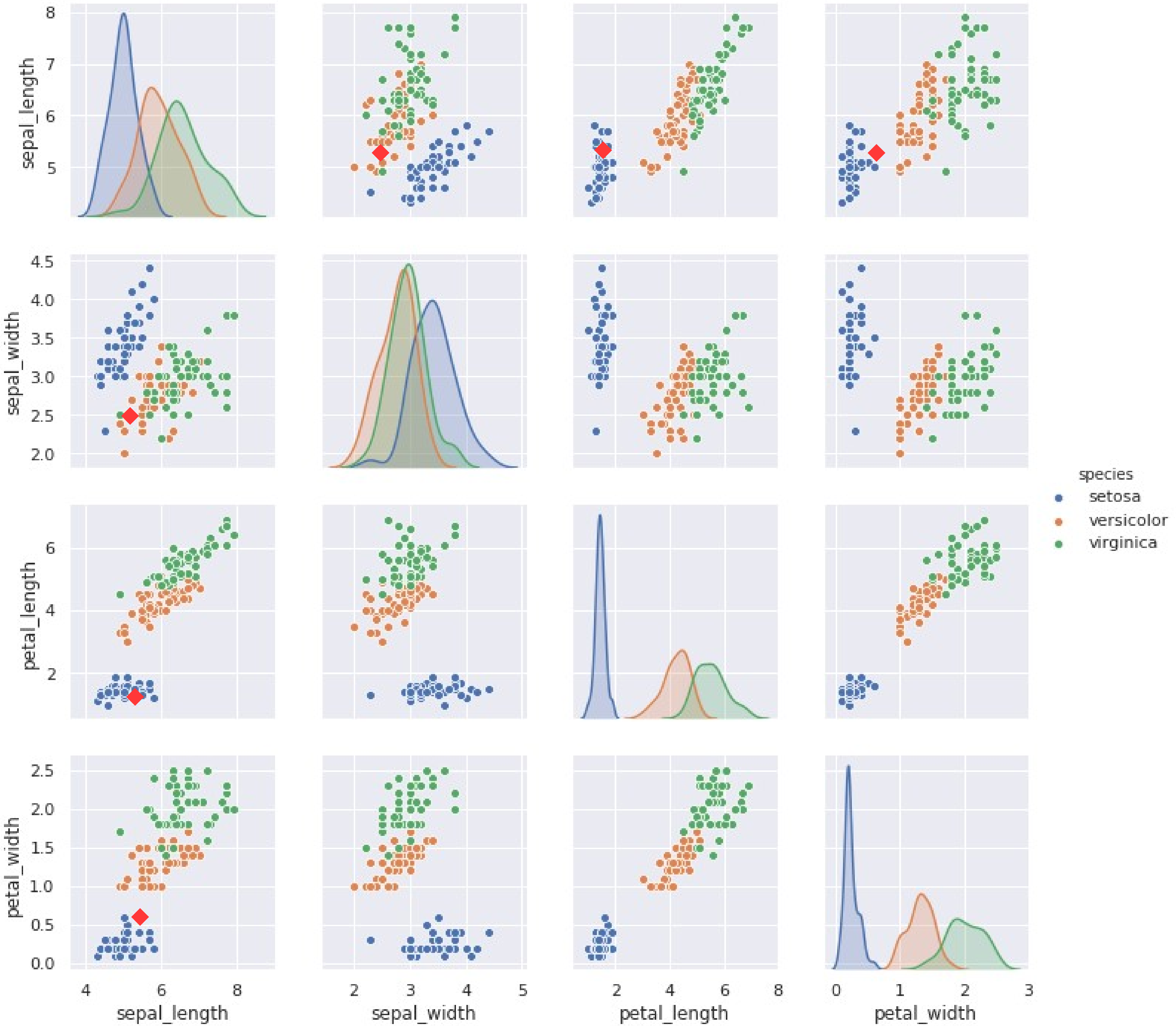
Pri


sepal_length
sepal_width
petal_length
petal_width
species
setosa
versicolor
virginica

Pri

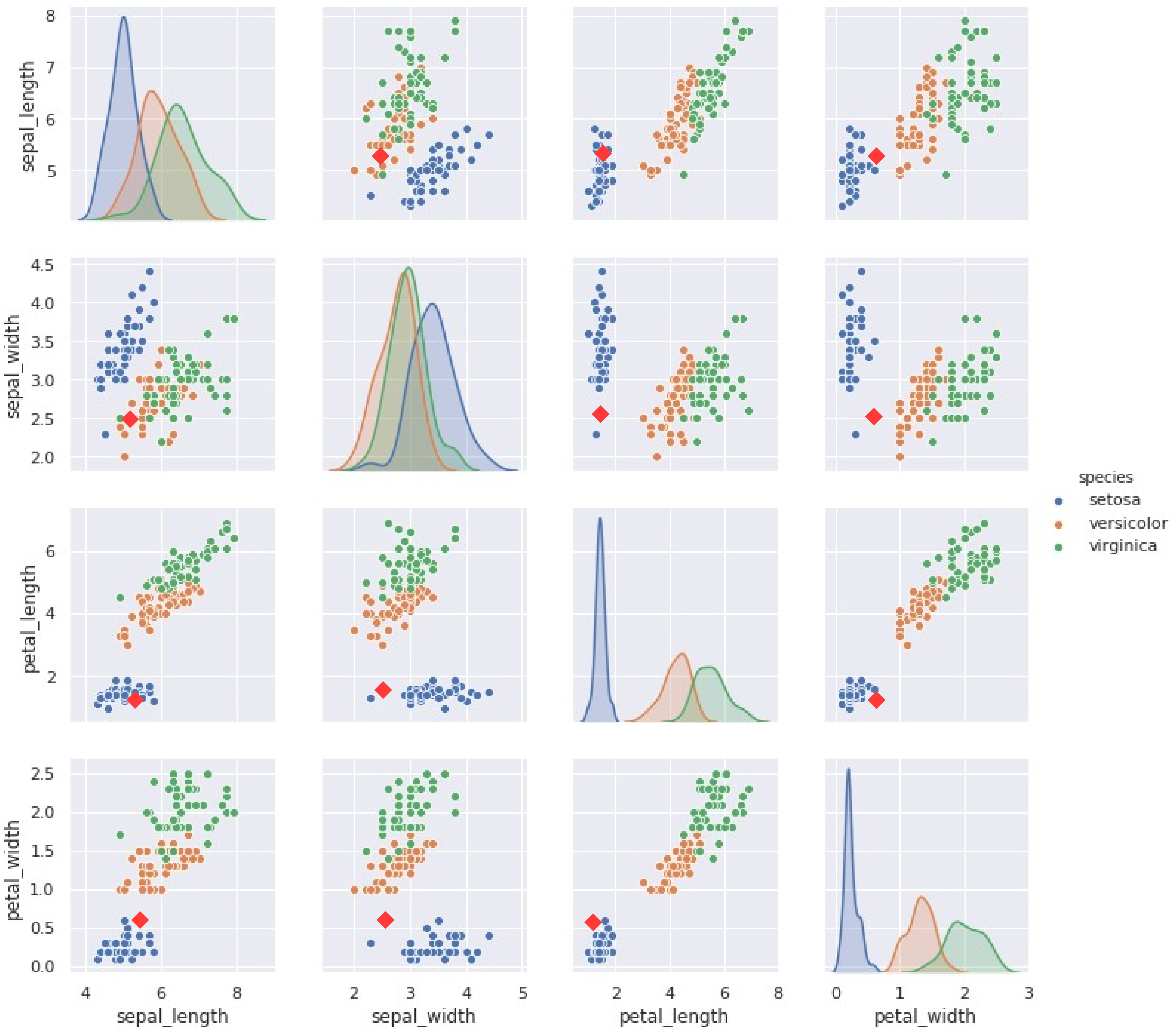

sepal_length
sepal_width
petal_length
petal_width
species
setosa
versicolor
virginica

Pri



sepal_length
sepal_width
petal_length
petal_width
species
setosa
versicolor
virginica

Pri


sepal_length
sepal_width
petal_length
petal_width
species
setosa
versicolor
virginica

Função de distância

- O k-NN utiliza uma **função de distância** ou uma **função de similaridade** para estabelecer o conceito de vizinhança
- Exemplos de distâncias:
  - Família de Minkowski (norma Lp): distância euclidiana, Manhattan, Chebyshev
  - Métrica de Tanimoto
  - Distância de Mahalanobis

Função de distância

- O k-NN utiliza uma **função de distância** ou uma **função de similaridade** para estabelecer o conceito de vizinhança
- Exemplos de funções de similaridade:
  - Cosseno
  - Produto escalar

Distância vs. métrica

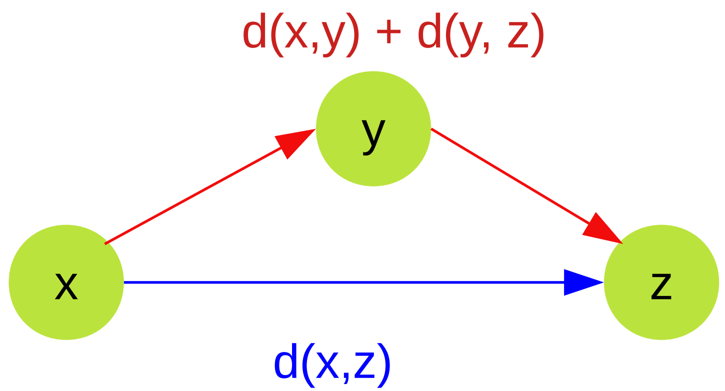
- Normalmente, distância e métrica são sinônimos, mas no contexto de AM, normalmente considera-se
  - Distância: uma função que mede a diferença entre dois objetos (ex.: Canberra)
  - Métrica: uma função de distância que respeita as propriedades de uma métrica (ex.: Manhattan e distância euclidiana)
  - Similaridade: uma função que mede a semelhança entre dois objetos (ex.: cosseno)

Distância vs. métrica

- Propriedades de uma métrica:
  - Não negatividade
    - $\forall x, \forall y, d(x, y) \geq 0$
  - Simetria
    - $\forall x, \forall y, d(x, y) = d(y, x)$
  - Identidade
    - $\forall x, \forall y, d(x, y) = 0 \Leftrightarrow x = y$

Distância vs. métrica

- Propriedades de uma métrica:
  - Desigualdade triangular
    - $\forall x, \forall y, \forall z, d(x, z) \leq d(x, y) + d(y, z)$



Métricas de Minkowski

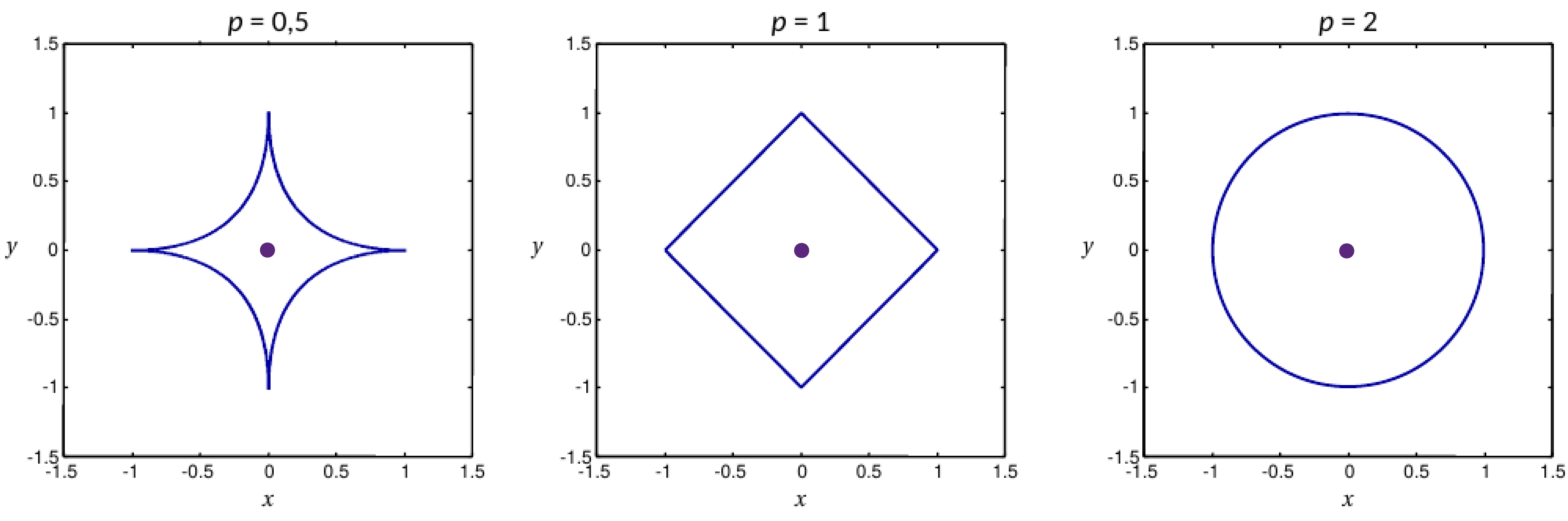
- Generalização da distância euclidiana
- Também conhecidas como normas Lp

$$Lp(x, y) = \sqrt[p]{\sum_{i=1}^{N} (x_i - y_i)^p}$$

  - p = 2 → distância euclidiana
  - p = 1 → distância Manhattan

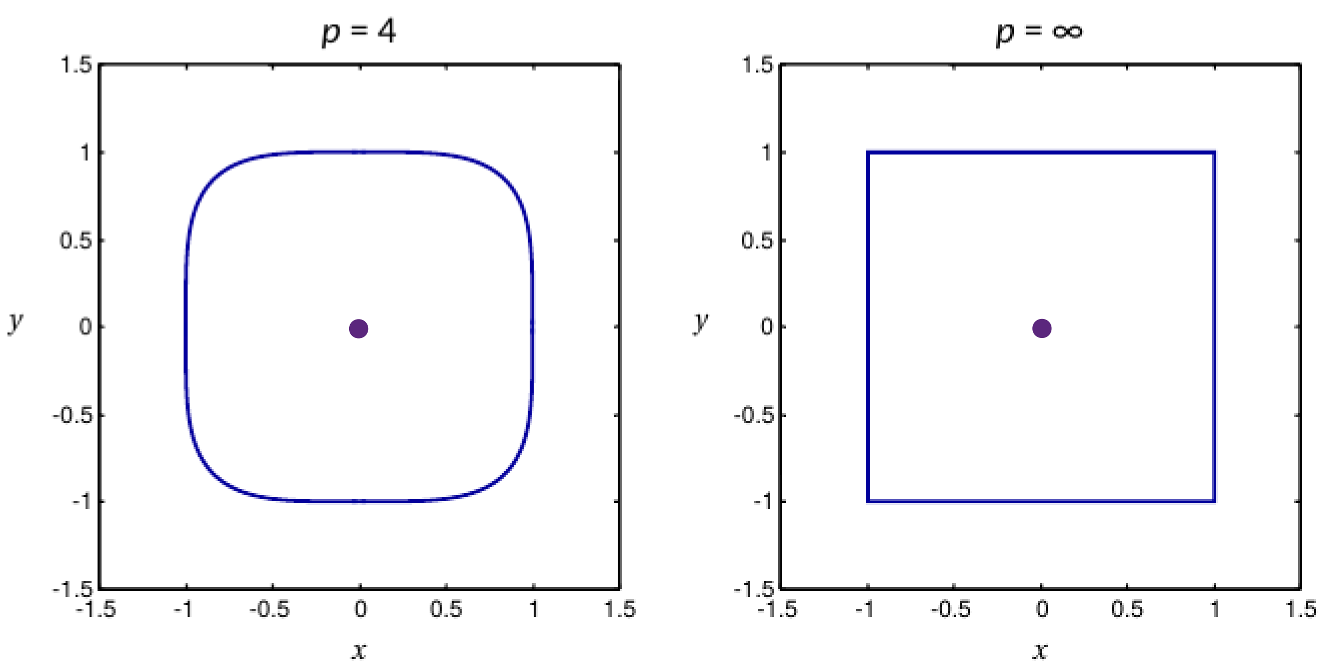
Métricas de Minkowski

- Uma visualização comum da métrica de Minkowski é a coleção de pontos a distância Lp(c, p) = 1 de algum ponto central
  - "Raio" unitário



Métricas de Minkowski

- Uma visualização comum da métrica de Minkowski é a coleção de pontos a distância Lp(c, p) = 1 de algum ponto central
  - "Raio" unitário

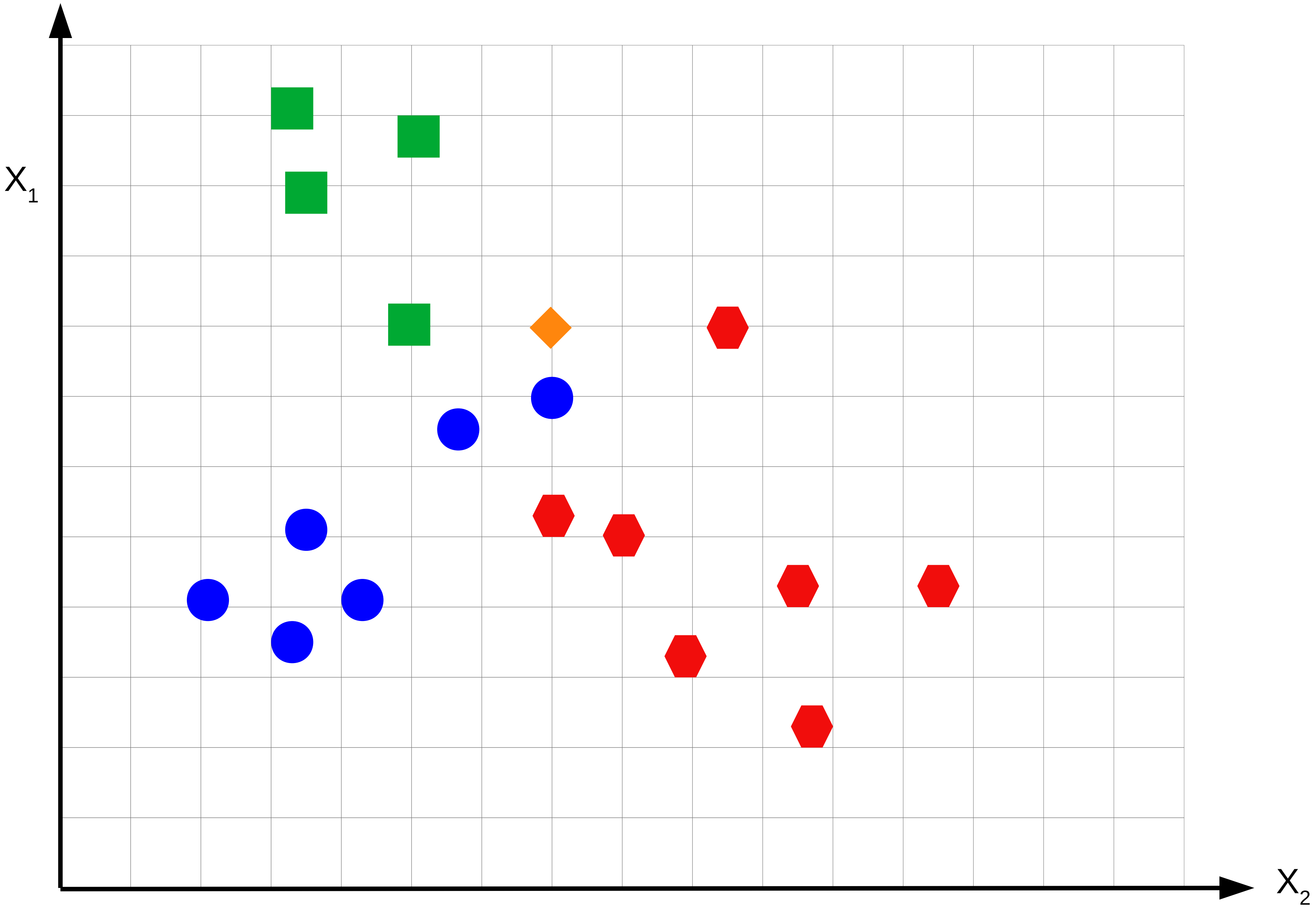


Distância Manhattan

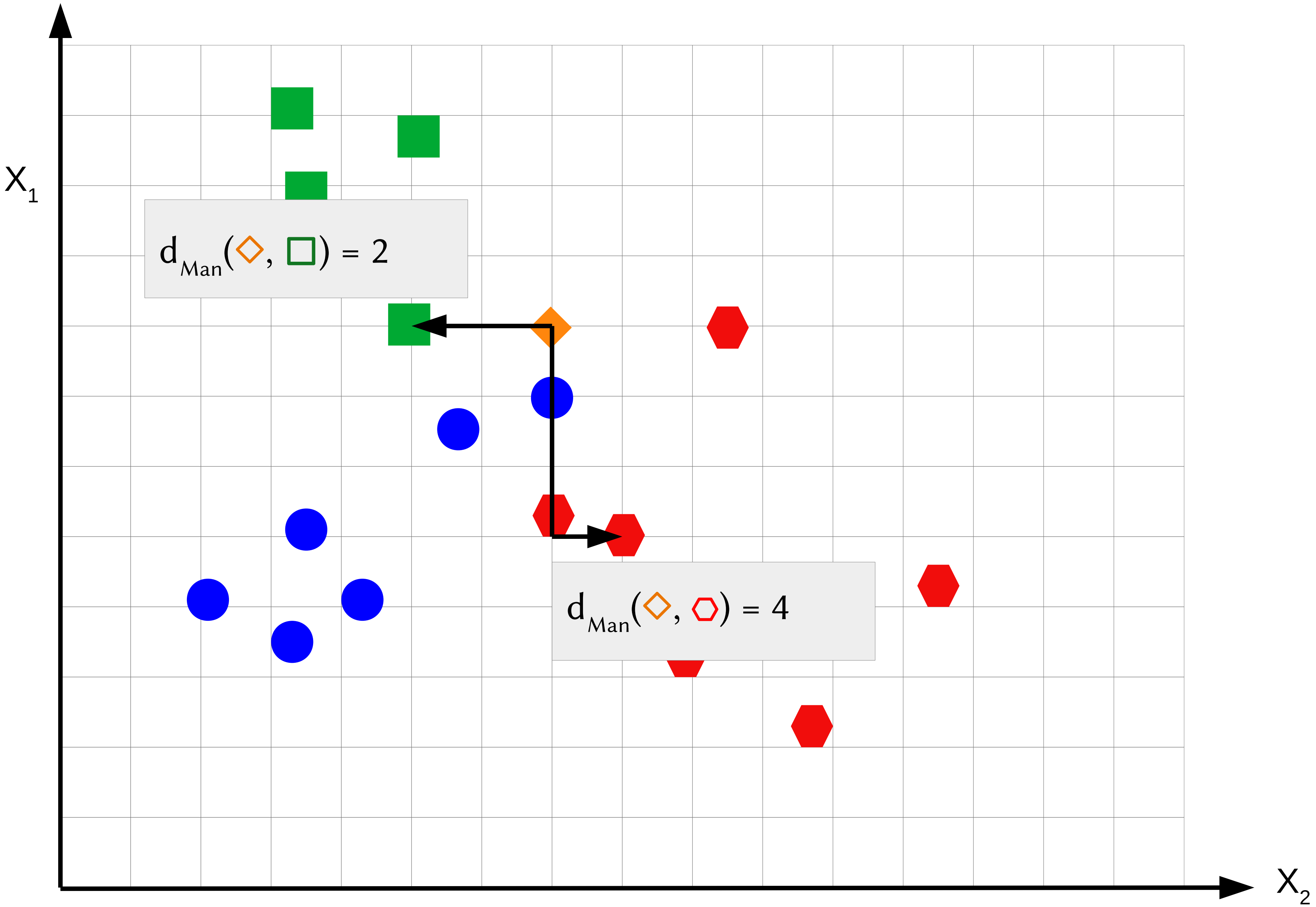
- Norma Lp com p = 1
  - Também conhecida como distância "*city block*"
  - É a distância ao longo dos eixos

$$d_{\mathrm{Man}}(x, y) = \sum_{i=1}^{N} |x_i - y_i|$$

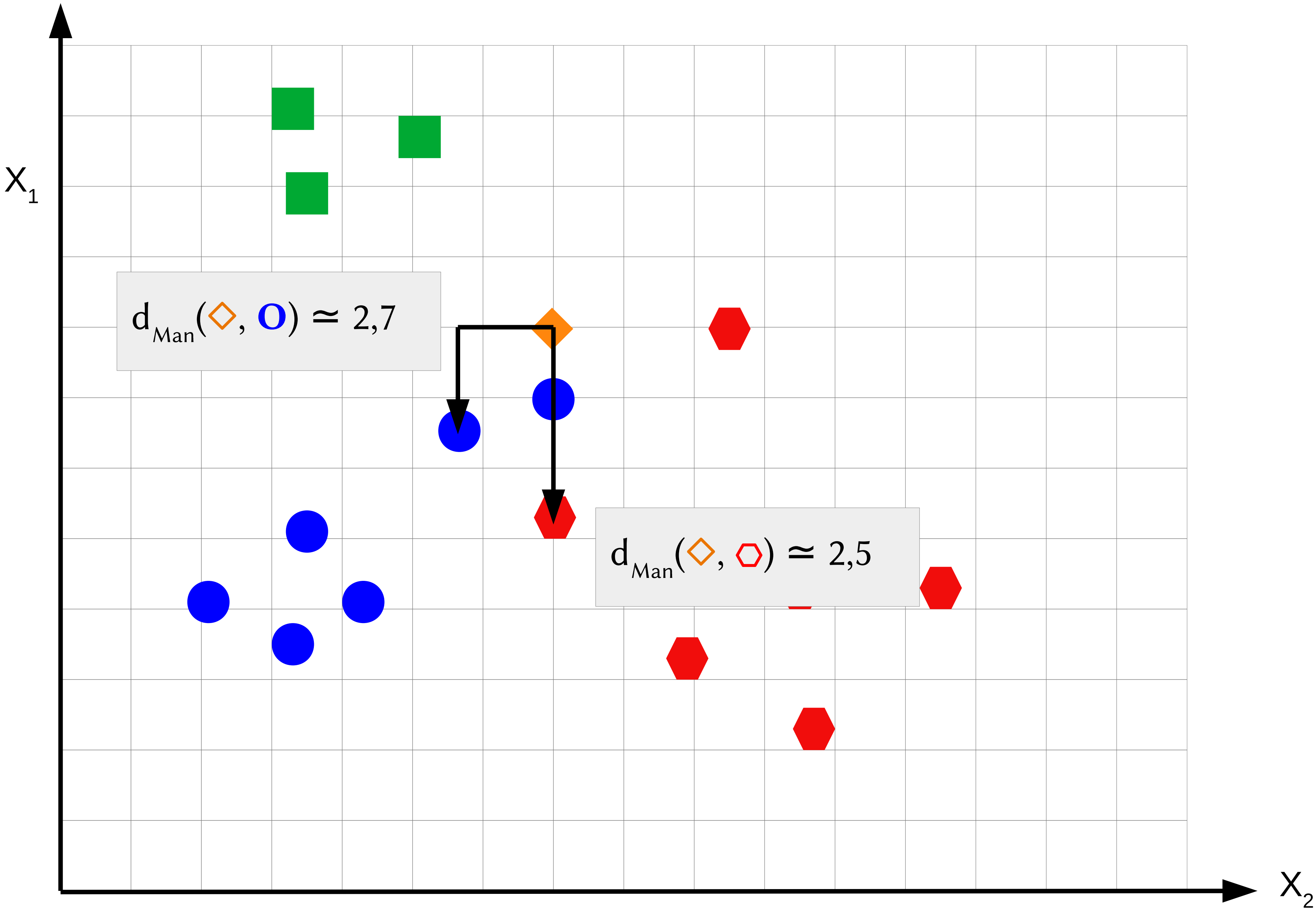
Distância Manhattan



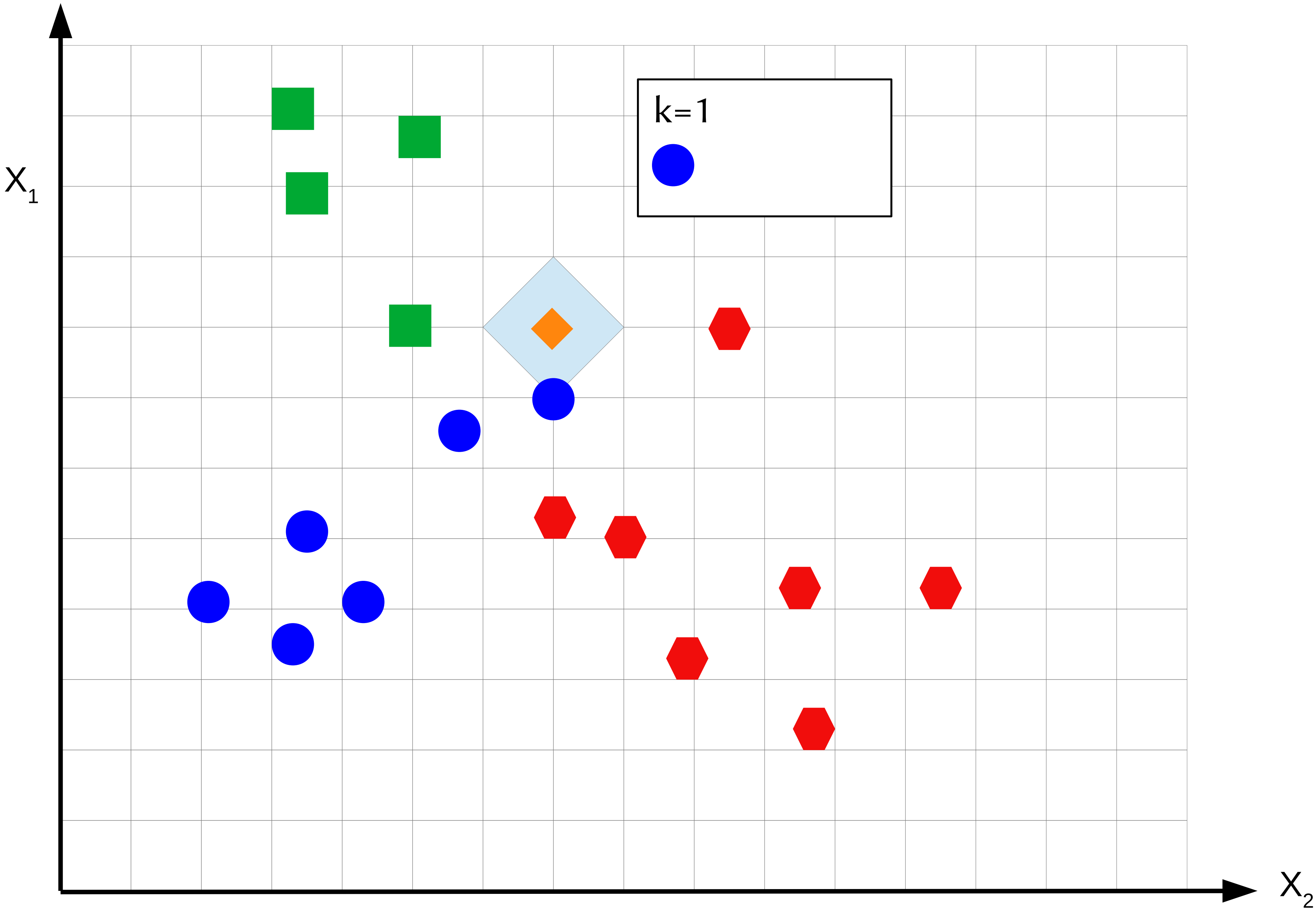
Distância Manhattan



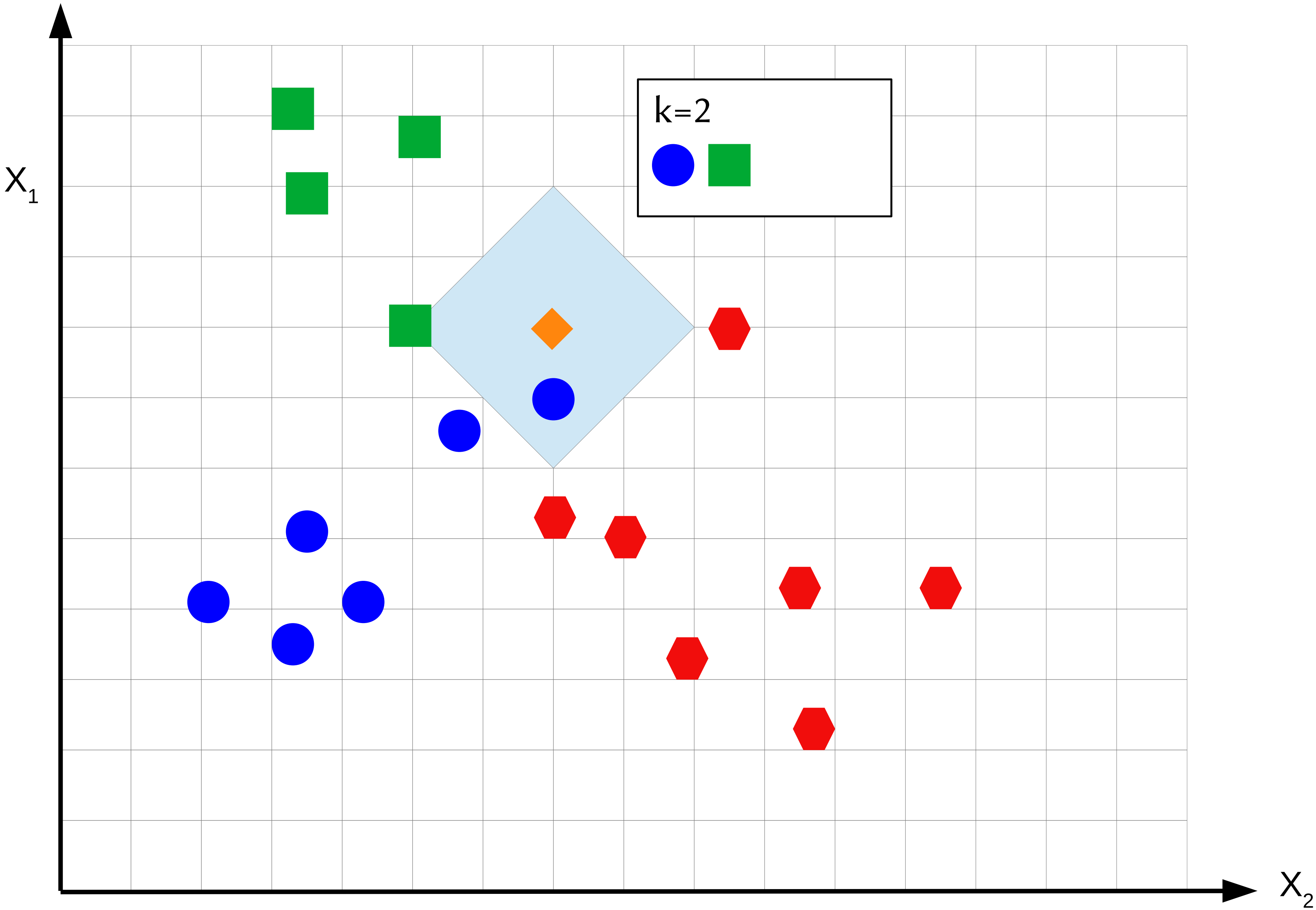
Distância Manhattan



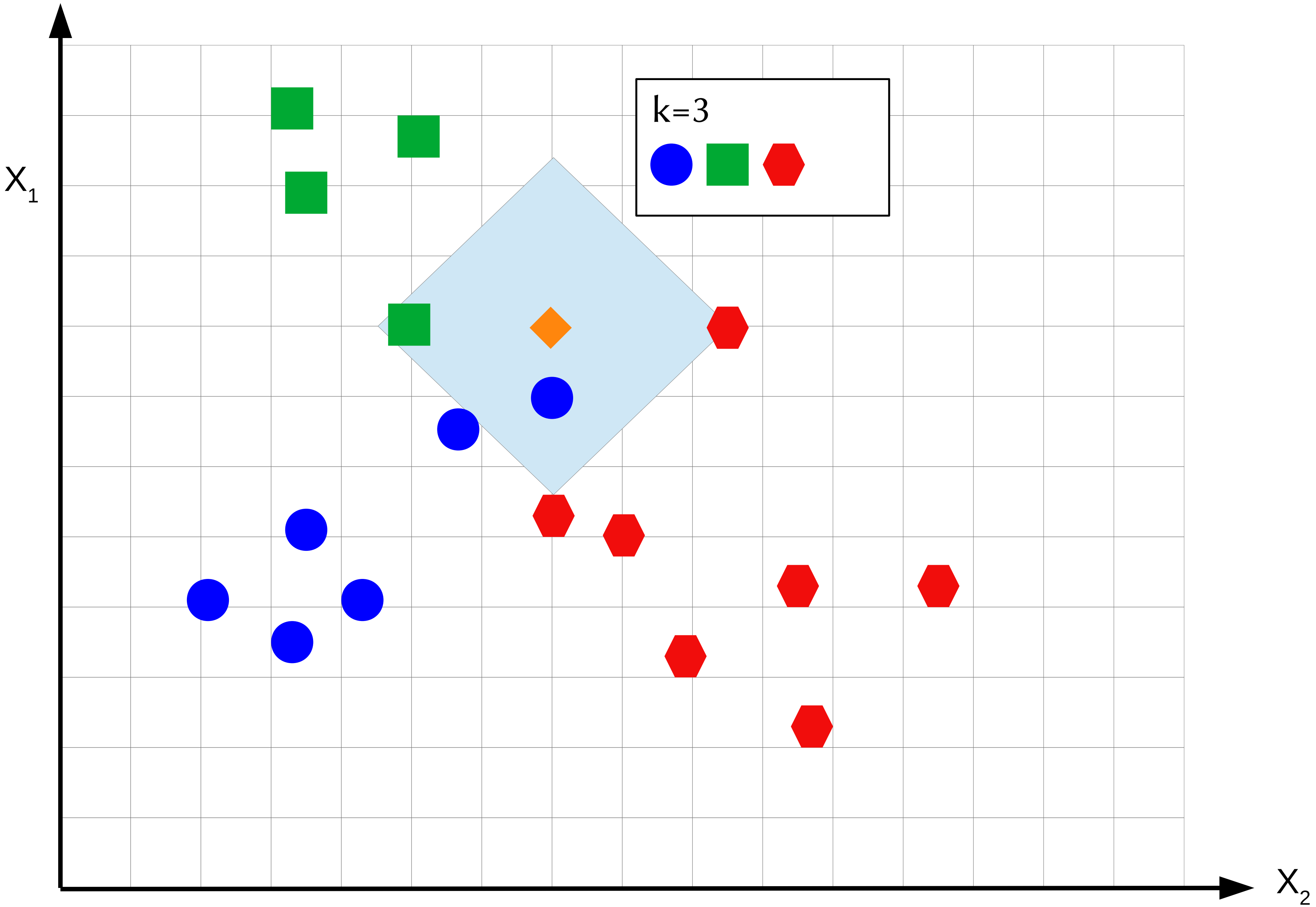
Distância Manhattan



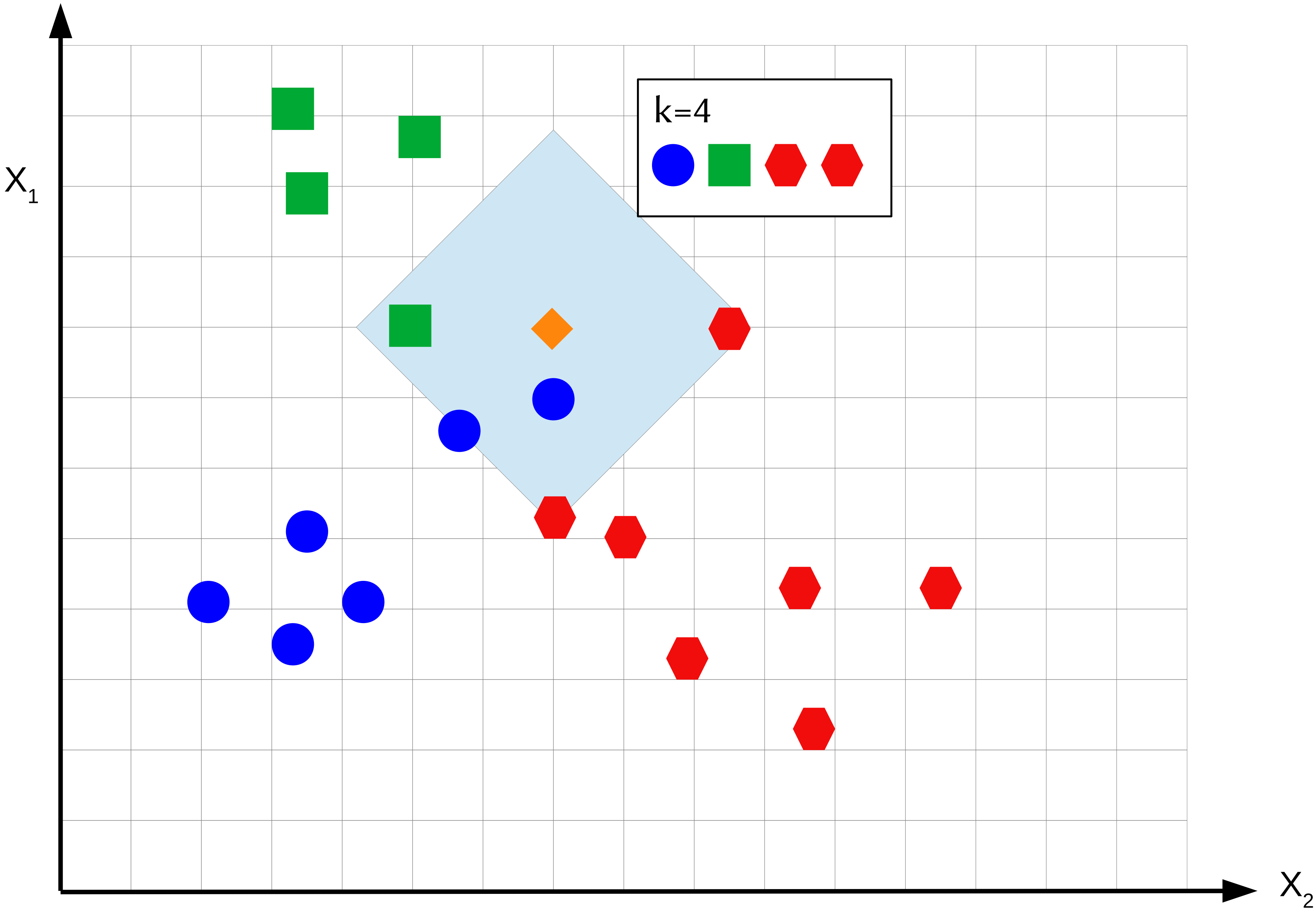
Distância Manhattan



Distância Manhattan



Distância Manhattan

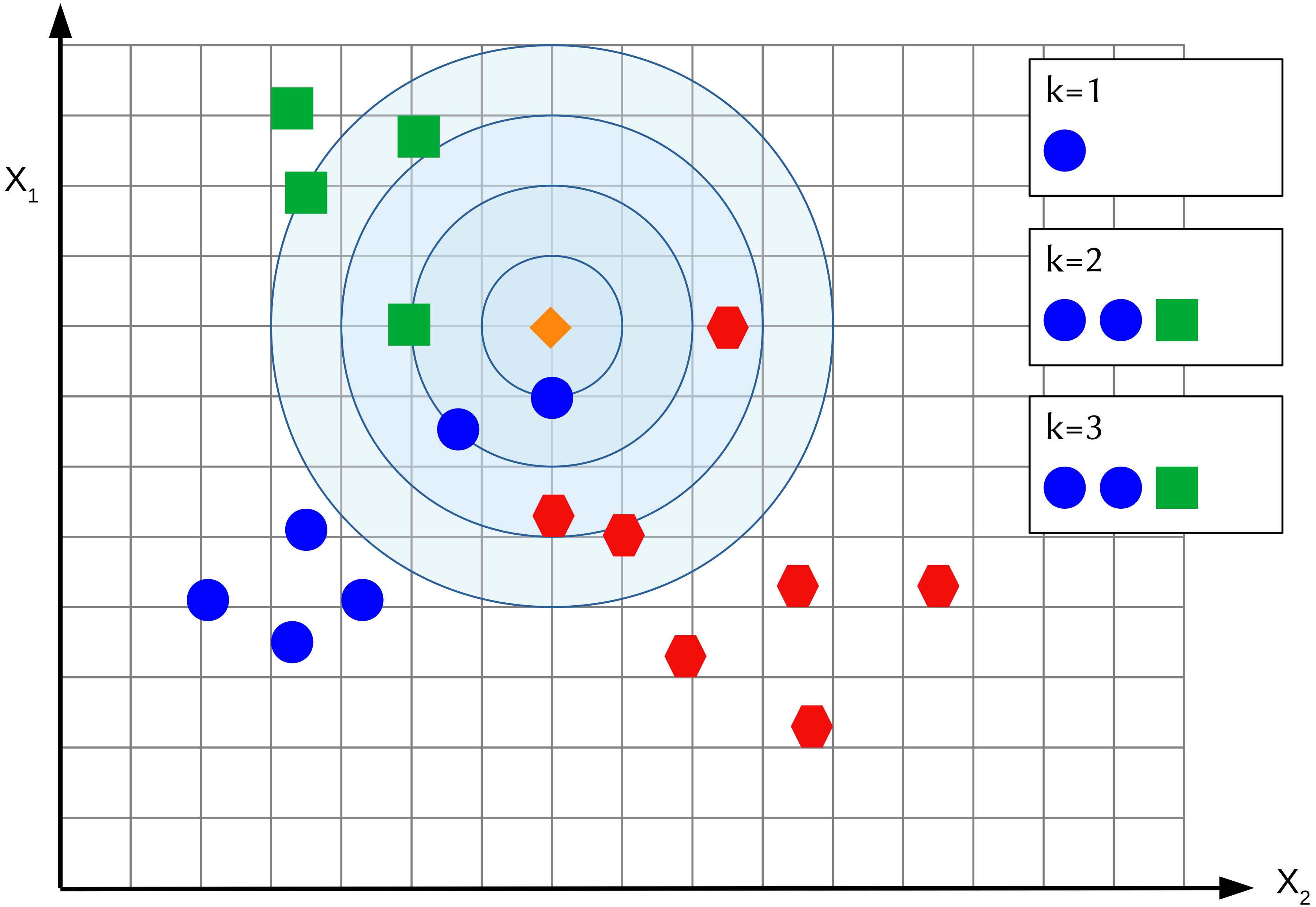


Distância euclidiana

- Norma Lp com p = 2
  - Distância entre vetores na geometria euclidiana
  - Reflete o conceito cotidiano de distância

$$d_{\mathrm{Euc}}(x, y) = \sqrt{\sum_{i=1}^{N} (x_i - y_i)^2}$$

Distância euclidiana



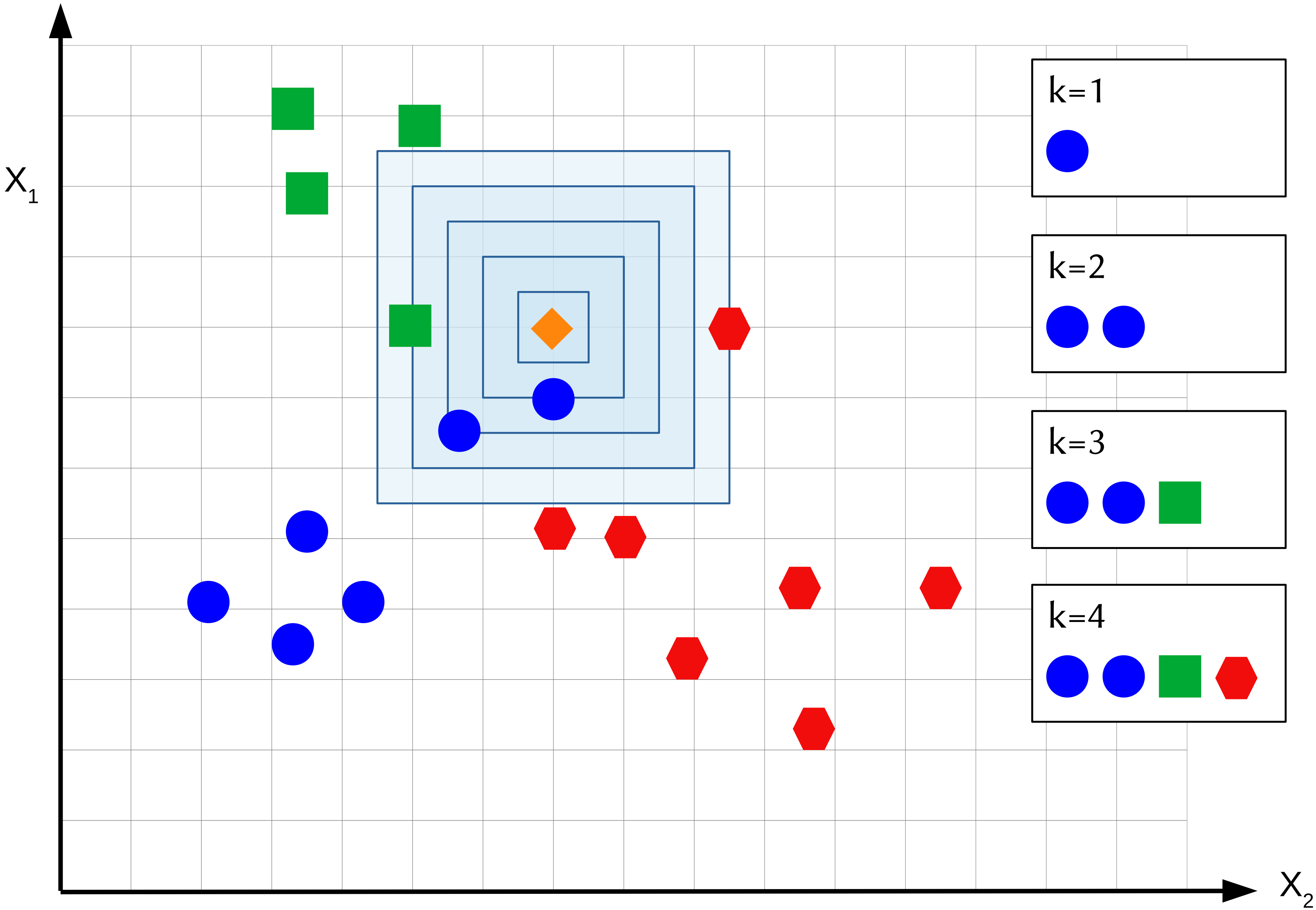
Distância Chebyshev

- Norma Lp com p $\longrightarrow\infty$
  - Apenas a diferença mais significativa entre os atributos determina a distância

$$d_{\mathrm{Cheb}}(x,y) = \lim_{p\to\infty} Lp$$

$$= \max_{i=1}^{N} |x_i - y_i|$$

Distância Chebyshev



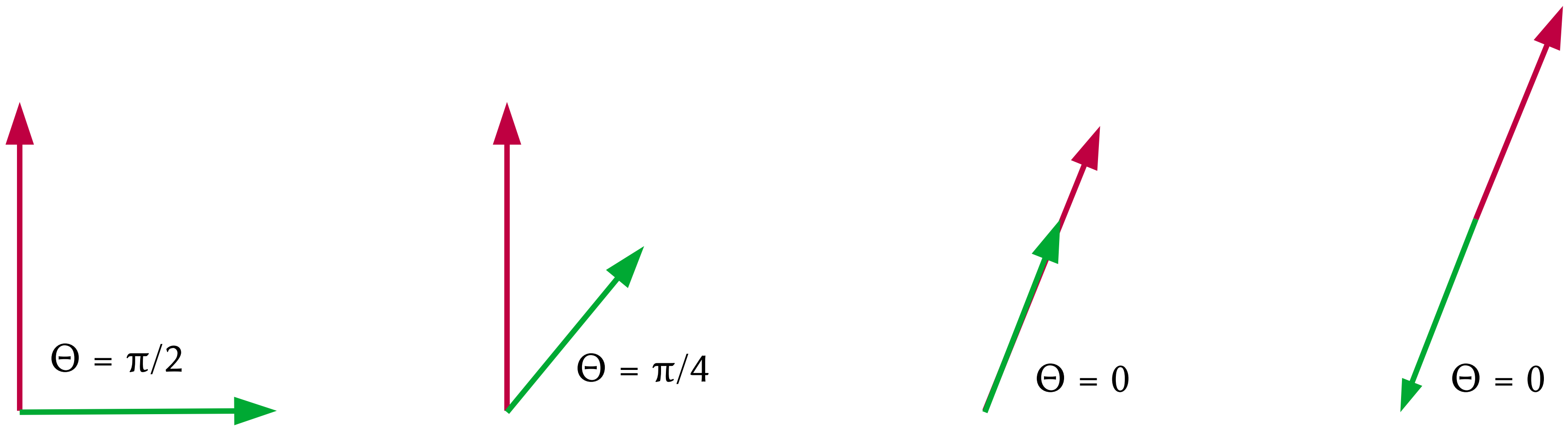
Similaridade cosseno

- Derivada da relação entre magnitude dos vetores e seu produto escalar
  - Desconsidera a magnitude dos atributos
  - Considera a proporção relativa entre os atributos do exemplo
  - O vetor de características $\mathbf{x}_1 = (1, 1)$ é mais similar a $\mathbf{x}_2 = (2, 2)$ do que a $\mathbf{x}_3 = (1, 0)$
  - O vetor de características $\mathbf{x}_1 = (1, 1)$ é mais similar a $\mathbf{x}_2 = (10, 10)$ do do que $\mathbf{x}_3 = (1, 0)$

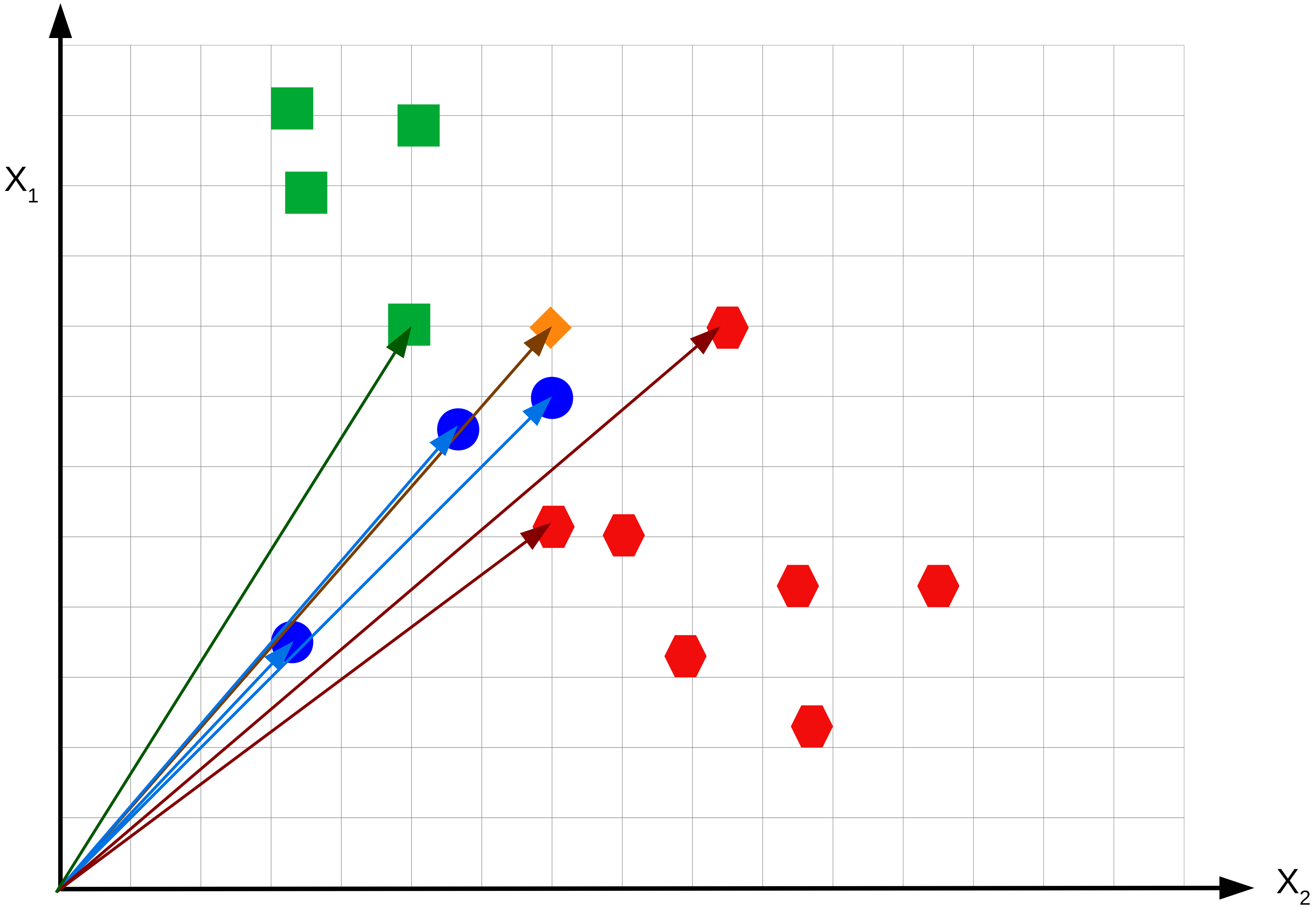
Similaridade cosseno

$$\mathbf{x} \cdot \mathbf{y} = ||\mathbf{x}|| \, ||\mathbf{y}|| \cos \theta$$

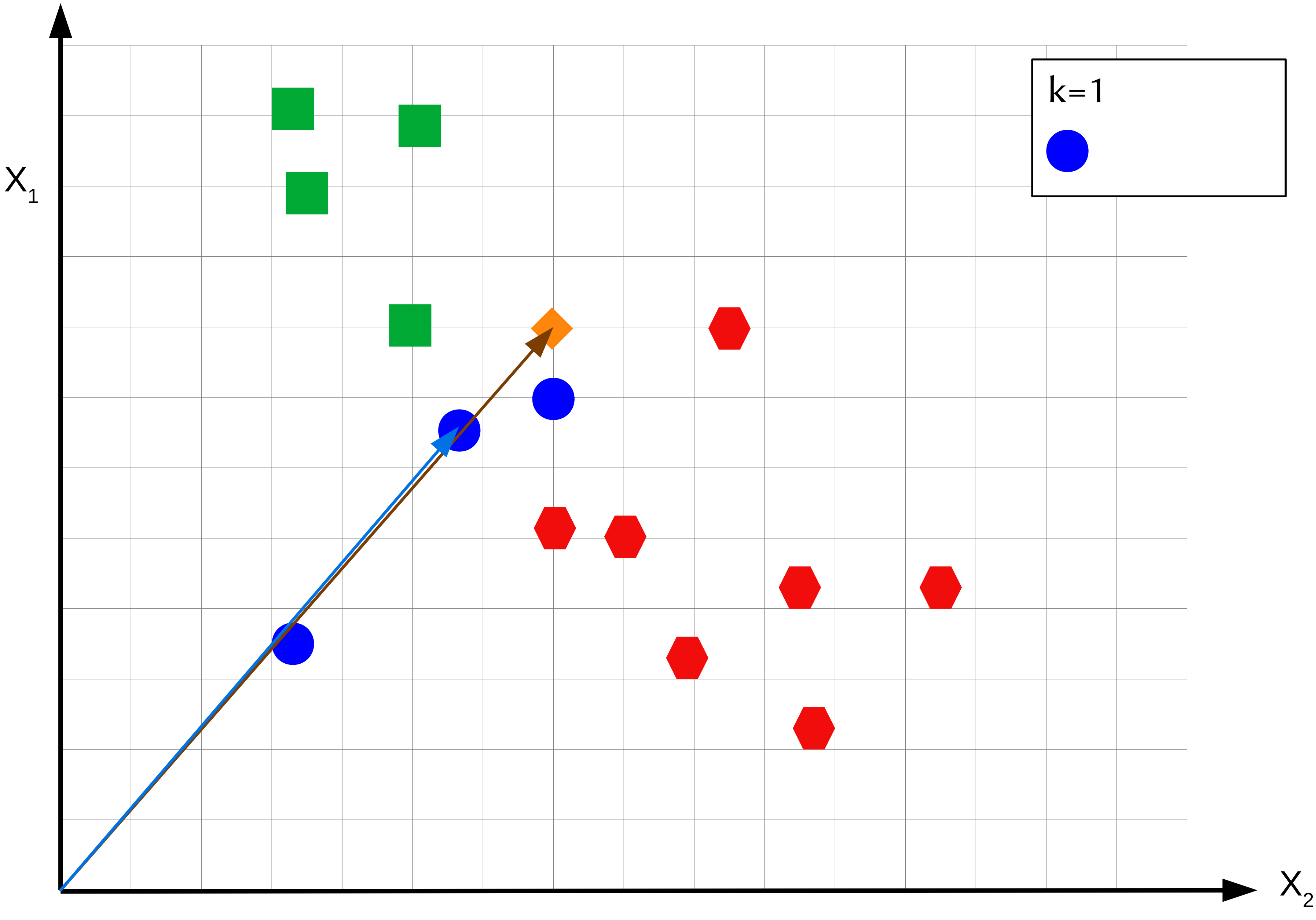
$$s_{\cos}(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x} \cdot \mathbf{y}}{||\mathbf{x}|| \times ||\mathbf{y}||} = \frac{\sum x_i y_i}{\sqrt{\sum x_i^2} \sqrt{\sum y_i^2}}$$



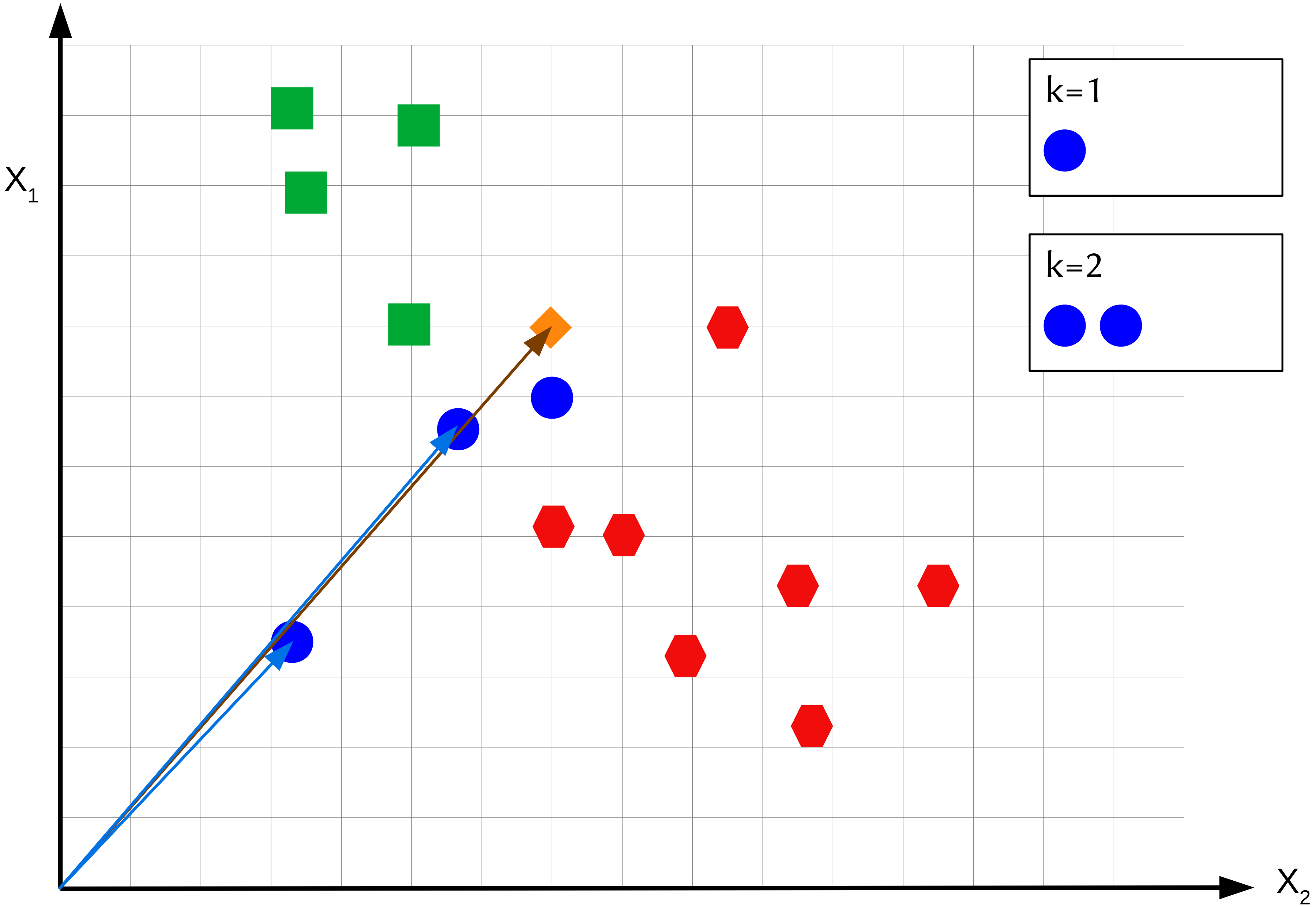
Similaridade cosseno



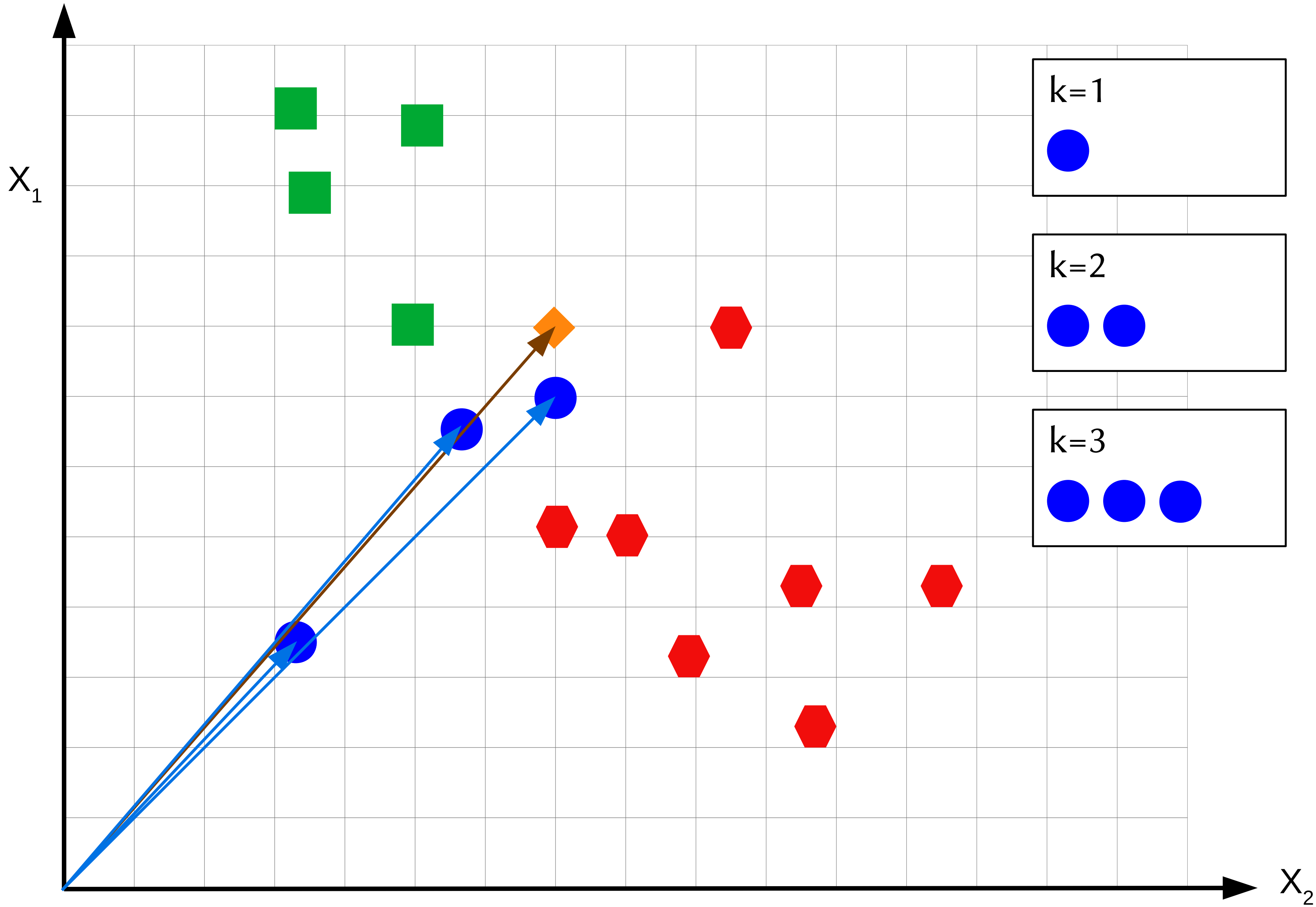
Similaridade cosseno



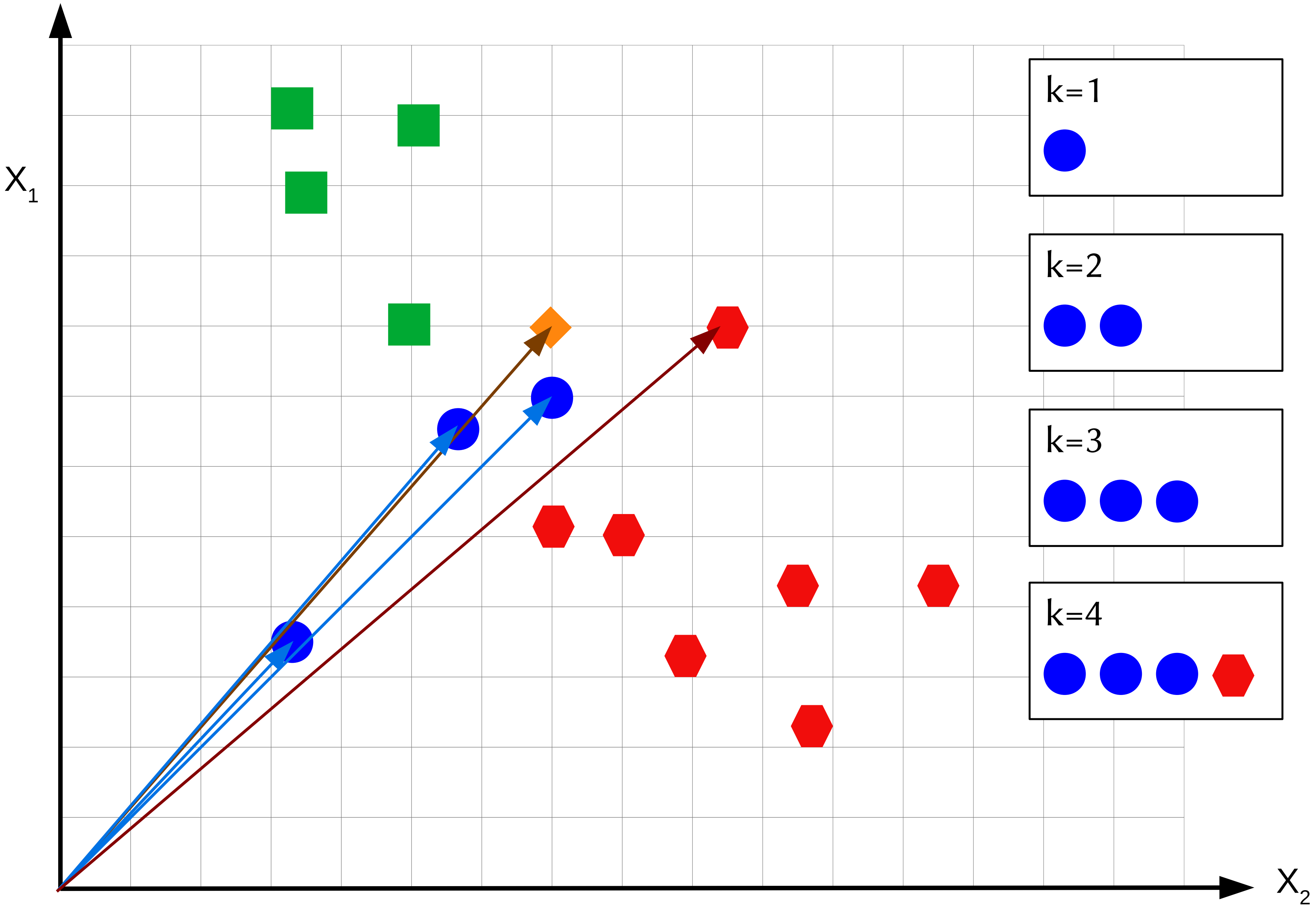
Similaridade cosseno



Similaridade cosseno



Similaridade cosseno



Similaridade cosseno

- É uma medida muito utilizada em recuperação de documentos
  - Documentos são representados como **sacos-de--palavras** (BoW, *bag of words*)
    - Os atributos são as palavras que aparecem na coleção de documentos
    - Cada documento se torna um exemplo
    - O valor de cada característica é a frequência da palavra

Representação saco-de-palavras

- Documentos

Assunto: Promoções de abril! Inscreva-se em cursos top a partir de R$21,99.

Aprenda do seu jeito
Explore nosso conteúdo top
Compre novos cursos

Assunto: Reunião marcada

Caros, nossa próxima reunião ficou marcada para o dia 18. Até lá! Confirmem recebimento, por favor!

Assunto: Confirmação de compra

Sua compra para a próxima viagem foi efetuada!

- Atributos:
  - X_1: 18
    X_2: a
    X_3: abril
    X_4: aprenda
    X_5: assunto
    X_6: até
    X_7: caros
    X_8: compra
    X_9: comprar
    X_{10}: confirmação
    X_{11}: confirmem
    X_{12}: conteúdo
    ...

Representação

Remoção de palavras muito frequentes (***stopwords***) que pouco contribuem para identificar documentos, tais como números, preposições, conjunções, artigos etc.

- Documentos

~~Assunto~~: Promoções ~~de~~ abril! Inscreva-se ~~em~~ cursos top ~~a partir de R$21,99~~.

Aprenda ~~do seu~~ jeito
Explore ~~nosso~~ conteúdo top
Compre novos cursos

~~Assunto~~: Reunião marcada

Caros, ~~nossa~~ próxima reunião ficou marcada ~~para o~~ dia ~~18~~. ~~Até lá~~! Confirmem recebimento, por favor!

~~Assunto~~: Confirmação ~~de~~ compra

~~Sua~~ compra ~~para a~~ próxima viagem foi efetuada!

- Atributos:
  - X_1: abril
    X_2: aprenda
    X_3: caros
    X_4: compra
    X_5: comprar
    X_6: confirmação
    X_7: confirmem
    X_8: conteúdo
    X_9: cursos
    X_{10}: dia
    X_{11}: efetuada
    X_{12}: explore
    ...

Rep[...] palavras

Lematização ou
redução à **forma canônica**.

- Documentos

~~Assunto~~: Promoções ~~de~~ abril! Inscreva-se ~~em~~ cursos top ~~a partir de R$21,99~~.

Aprenda ~~do seu~~ jeito
Explore ~~nosso~~ conteúdo top
Compre novos cursos

~~Assunto~~: Reunião marcada

Caros, ~~nossa~~ próxima reunião ficou marcada ~~para o~~ dia ~~18~~. ~~Até lá~~! Confirmem recebimento, por favor!

~~Assunto~~: Confirmação ~~de~~ compra

~~Sua~~ compra ~~para a~~ próxima viagem foi efetuada!

- Atributos:
  - X_1: abril
    X_2: aprender
    X_3: caro
    X_4: compra
    X_5: comprar
    X_6: confirmação
    X_7: confirmar
    X_8: conteúdo
    X_9: curso
    X_{10}: dia
    X_{11}: efetuar
    X_{12}: explorar
    ...

Re e-palavras

Stemização: reduz as palavras a suas componentes fundamentais

- Documentos

~~Assunto~~: Promoções ~~de~~ abril! Inscreva-se ~~em~~ cursos top ~~a partir de R$21,99~~.

Aprenda ~~do seu~~ jeito
Explore ~~nosso~~ conteúdo top
Compre novos cursos

~~Assunto~~: Reunião marcada

Caros, ~~nossa~~ próxima reunião ficou marcada ~~para o~~ dia ~~18~~. ~~Até lá~~! Confirmem recebimento, por favor!

~~Assunto~~: Confirmação ~~de~~ compra

~~Sua~~ compra ~~para a~~ próxima viagem foi efetuada!

- Atributos:

X_1: abril
X_2: aprend
X_3: car
X_4: compr
X_5: confirm
X_6: cont
X_7: curs
X_8: dia
X_9: efet
X_{10}: explor
X_{11}: fic
...

Represe

O objetivo da **stemmização** é identificar palavras que transmitem ideias semelhantes, sem distinção de suas funções sintáticas (verbo, substantivo etc.)

- Documentos

~~Assunto~~: Promoções ~~de~~ abril! Inscreva-se ~~em~~ cursos top ~~a partir de R$21,99~~.

Aprenda ~~do seu~~ jeito
Explore ~~nosso~~ conteúdo top
Compre ~~novos~~ cursos

~~Assunto~~: Reunião marcada

Caros, ~~nossa~~ próxima reunião ficou marcada ~~para o~~ dia ~~18~~. ~~Até lá~~! Confirmem recebimento, por favor!

~~Assunto~~: Confirmação ~~de~~ compra

~~Sua~~ compra ~~para~~ ~~a~~ próxima viagem foi efetuada!

- Atributos:

X_1: abril
X_2: aprend
X_3: car
X_4: compr
X_5: confirm
X_6: cont
X_7: curs
X_8: dia
X_9: efet
X_{10}: explor
X_{11}: fic
...

Represe

O objetivo da **stemmização** é identificar palavras que transmitem ideias semelhantes, sem distinção de suas funções sintáticas (verbo, substantivo etc.)

- Documentos

~~Assunto~~: Promoções ~~de~~ abril! Inscreva-se ~~em~~ cursos top ~~a partir de R$21,99~~.

Aprenda ~~do seu~~ jeito
Explore ~~nosso~~ conteúdo top
Compre novos cursos

~~Assunto~~: Reunião marcada

Caros, ~~nossa~~ próxima reunião ficou marcada ~~para o~~ dia ~~18~~. ~~Até lá~~! Confirmem recebimento, por favor!

~~Assunto~~: Confirmação ~~de~~ compra

~~Sua~~ compra ~~para a~~ próxima viagem foi efetuada!

- Atributos:

X_1: abril
X_2: aprend
X_3: car
X_4: compr
X_5: confirm
X_6: cont
X_7: curs
X_8: dia
X_9: efet
X_{10}: explor
X_{11}: fic
...

Representação saco-de-palavras

- O saco-de-palavras será a matriz atributo-valor
  - Cada documento é um exemplo
  - A característica x_{ij} é a quantidade de vezes que o termo X_j aparece no exemplo E_i

X_2: aprend (aprenda ⟶ aprender ⟶ aprend)

X_1	X_2	X_3	X_4	X_5	X_6	X_7	..
1	1	0	1	0	1	2	...
0	0	1	0	1	0	0	...
0	0	0	1	1	0	0	...

X_5: confirm (confirmem ⟶ confirmar ⟶ confirm e também confirmação ⟶ confirm)

Representação saco-de-palavras

- Outras operações comuns no pré-processamento
  - Redução de bigramas: palavras que ocorrem juntas com frequência são consideradas um mesmo termo
    - Vice_versa, bom_dia, lava_jato
  - A redução pode ser feita em qualquer passo: antes da remoção de *stopwords*, depois da lematização, depois da *stemmização*...
    - lava_jato ou lav_jat,   padrao_vida ou padr_vi

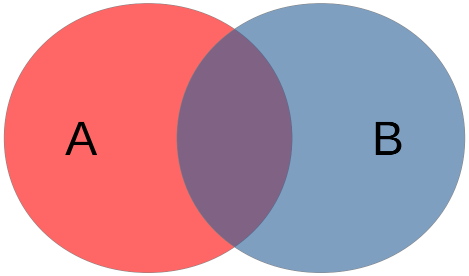
Representação saco-de-palavras

- Além dos bigramas, existem também trigramas, 4-gramas etc. (n-gramas)
  - Exemplos:
    - Padrão de vida (2-grama)
    - Supremo tribunal federal (3-grama)

Métrica de Tanimoto

- Distância apropriada para valores binários
  - **A** é o conjunto dos atributos em *x* com valor 1
  - **B** é o conjunto dos atributos em *y* com valor 1

$$d_{\text{Tan}}(\mathbf{A},\mathbf{B}) = \frac{|\mathbf{A}| + |\mathbf{B}| - 2|\mathbf{A}\cap\mathbf{B}|}{|\mathbf{A}| + |\mathbf{B}| - |\mathbf{A}\cap\mathbf{B}|}$$



Outros pontos

- Distância entre valores categóricos
- Normalização
  - Atributos em escalas diferentes podem dominar a distância ou serem irrelevantes
- Emprego da vizinhança como uma medida de probabilidade de classificação correta
  - Probabilidade de ser classe c_j pode ser estimada com base na fração de exemplos de c_j na vizinhança
    - $p(c_j \mid x_i) = k_j / k$

Normalização

- Exemplo da base de dados ***breast cancer*** [1]

Idade	Glicose	Insulina	Câncer
48	70	2,71	não
83	92	3,12	não
34	78	3,47	não
48	112	10,40	sim
82	199	12,16	sim

$d_{Euc}^2 = 40^2 + 22^2 + 0{,}41^2 = 1600 + 484 + 0{,}17$

$d_{Euc}^2 = 0^2 + 42^2 + 7{,}69^2 = 0 + 1764 + 59{,}13$

[1] https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Coimbra (verificado em 2019-04-25)

Normalização

- Exemplo da base de dados ***breast cancer*** [1]

Idade	Glicose	Insulina	Câncer
48	70	2,71	não
83	92	3,12	não
34	78	3,47	não
48	112	10,40	sim
82	199	12,16	sim

$d_{Man}^{2} = 40 + 22 + 0{,}41$

$d_{Man}^{2} = 0 + 42 + 7{,}69$

[1] https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Coimbra (verificado em 2019-04-25)

Normalização

- Para cada atributo X_i

$$x_i = \frac{x_i - \min(X_i)}{\max(X_i) - \min(X_i)}$$

Idade	Glicose	Insulina	Câncer
48	70	2,71	não
83	92	3,12	não
34	78	3,47	não
48	112	10,40	sim
82	199	12,16	sim
[34, 83]	[70, 199]	[2,71; 12,16]	

Idade	Glicose	Insulina	Câncer
0,29	0,00	0,00	não
1,00	0,17	0,04	não
0,00	0,06	0,08	não
0,29	0,33	0,81	sim
0,98	1,00	1,00	sim

Normalização

- Para um novo exemplo

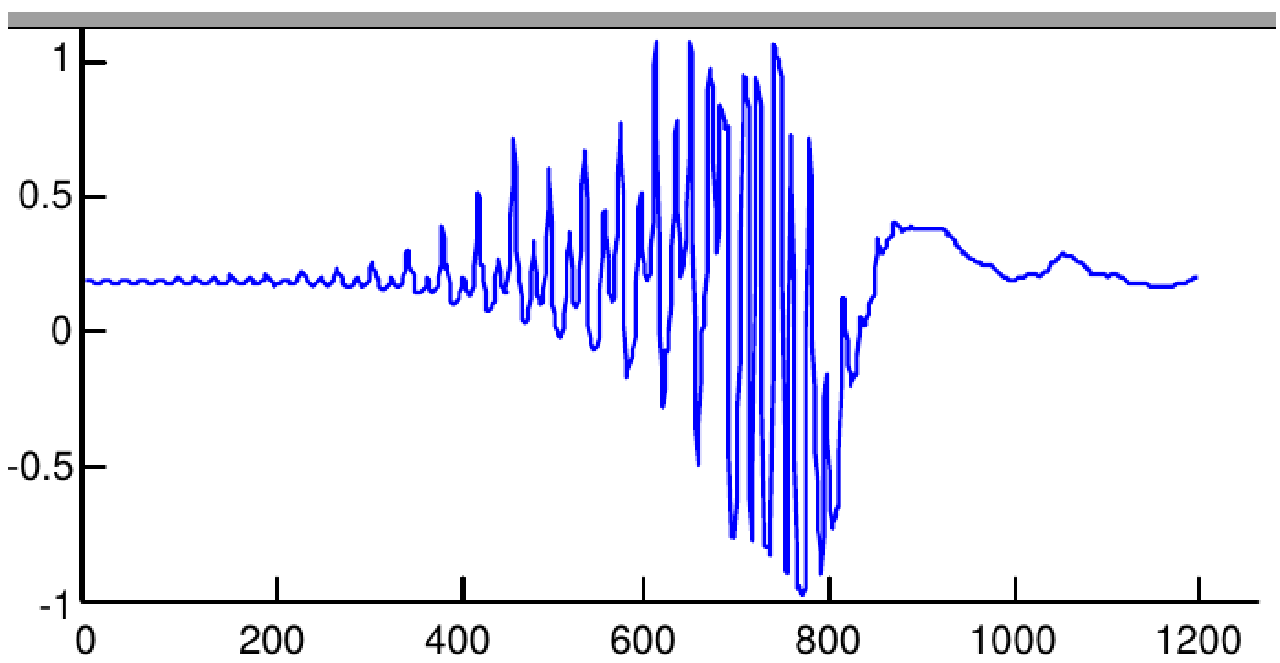
Idade	Glicose	Insulina	Câncer
0,29	0,00	0,00	não
1,00	0,17	0,04	não
0,00	0,06	0,08	não
0,29	0,33	0,81	sim
0,98	1,00	1,00	sim
[34, 83]	[70, 199]	[2,71; 12,16]	

Idade	Glicose	Insulina
30	88	4,6
35	130	8,7
46	92	1,05

Idade	Glicose	Insulina
-0,08	0,14	0,20
0,98	1,12	1,08
0,24	0,17	-0,18

Séries temporais

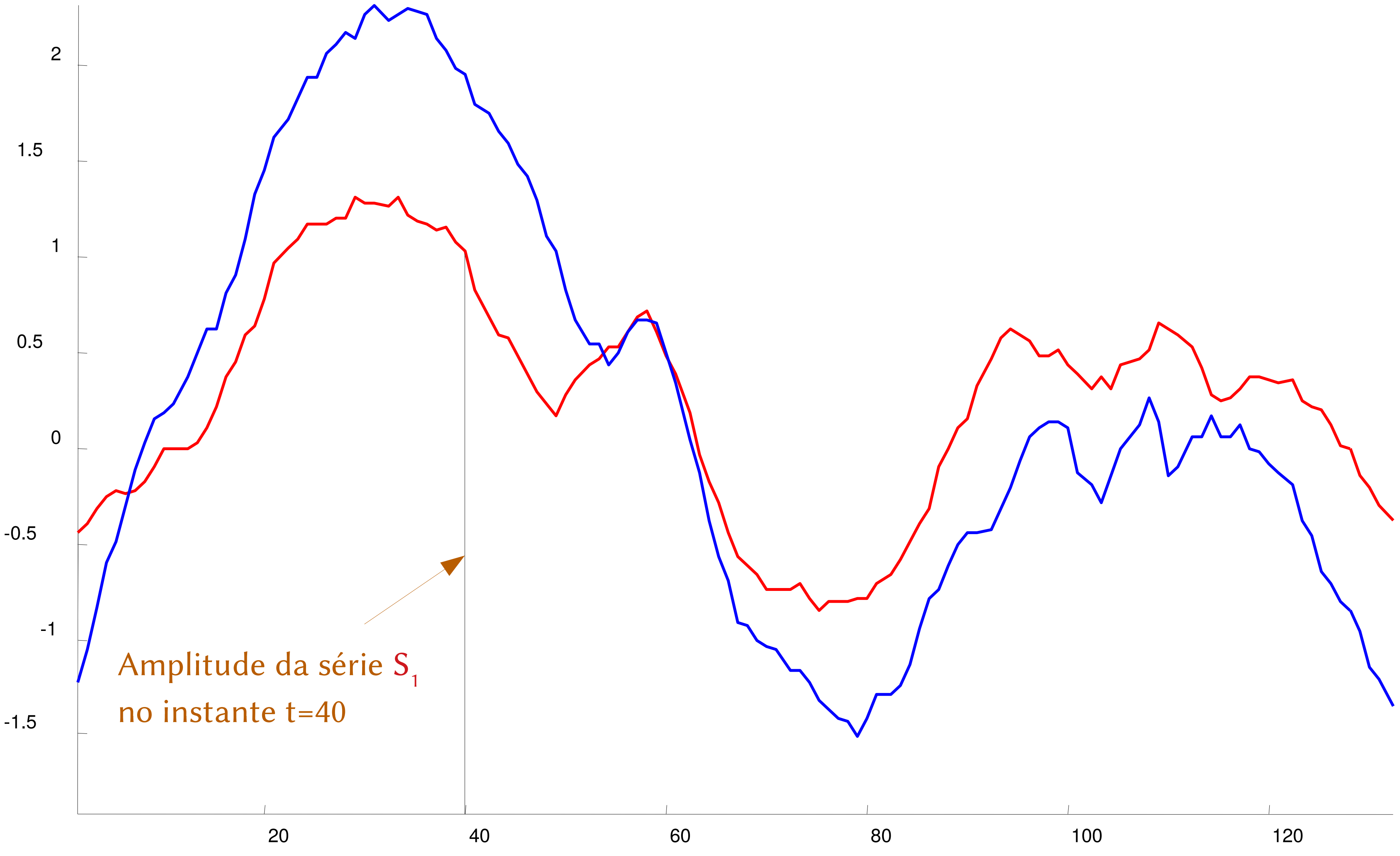
- Uma série temporal S = (s_1, s_2, ..., s_n) é uma coleção de observações tomadas nos instantes 1, 2, ..., n
  - Exemplo: um sinal de áudio



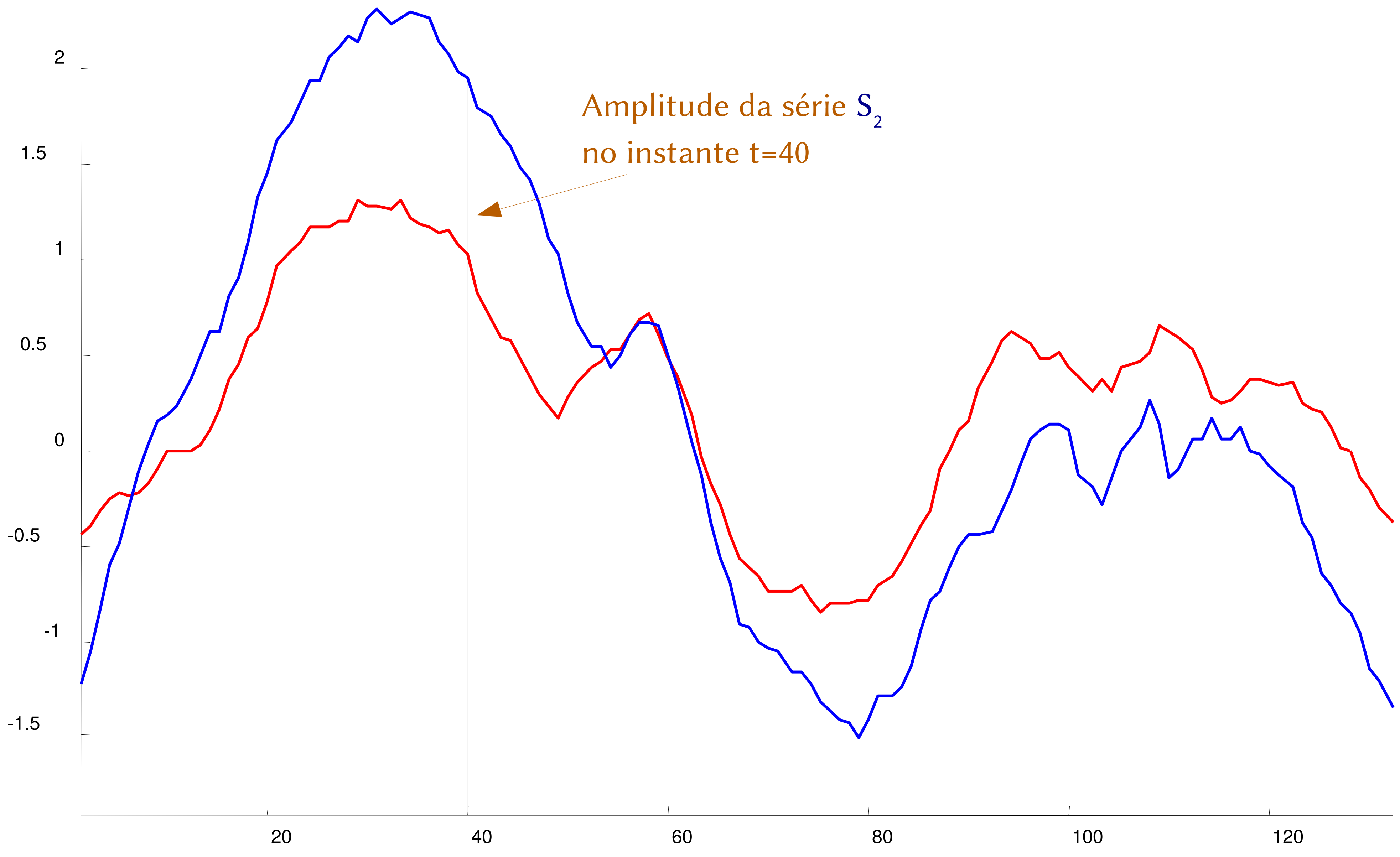
Séries temporais

- O classificador 1-NN é surpeendemente eficiente na classificação de séries temporais
  - Ainda mais eficiente quando são empregadas medidas de distância específicas para sequências ordenadas, como *dynamic time warping* (DTW)
  - Ou quando extraímos características, como atributos baseados em frequências

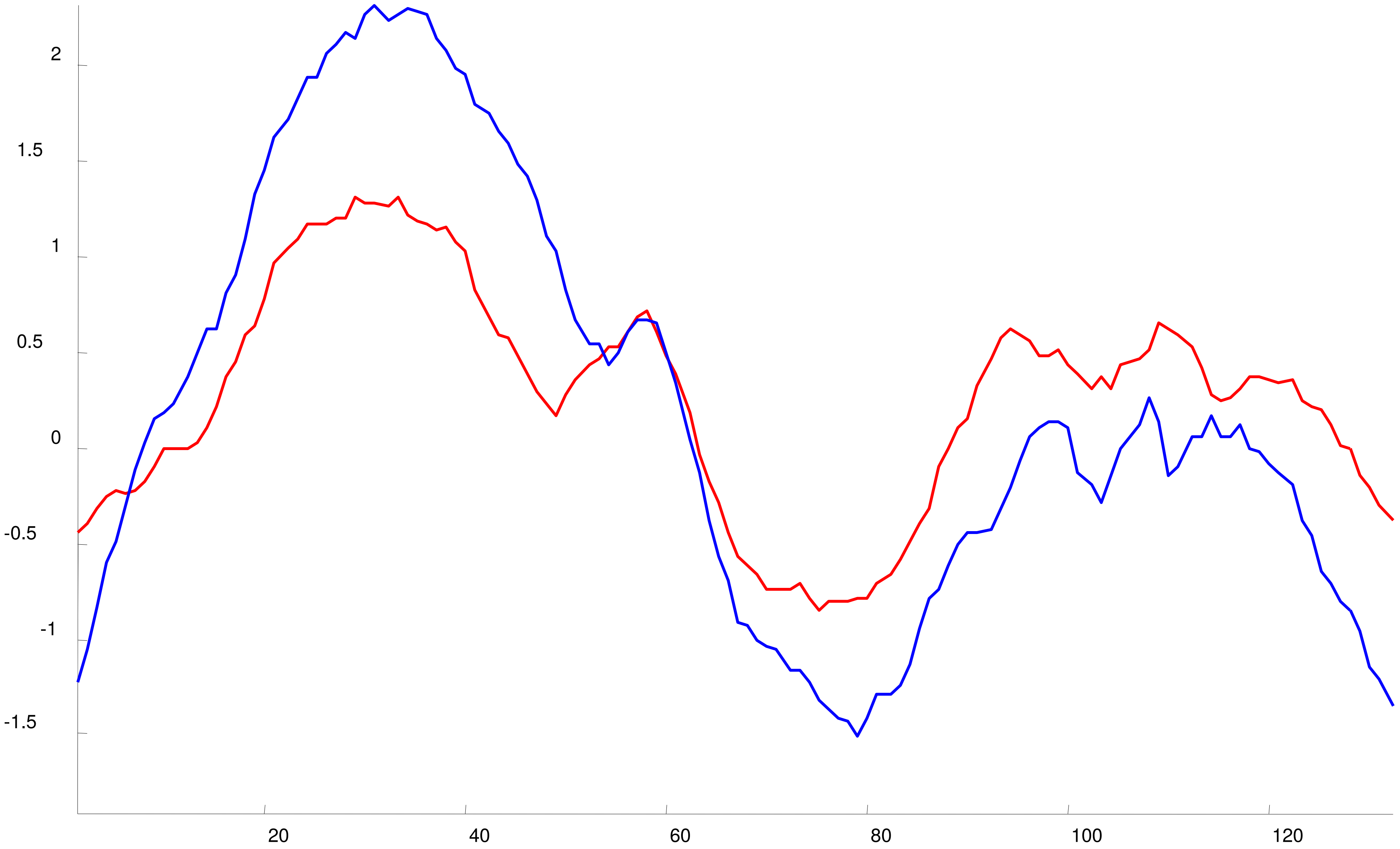
Séries temporais



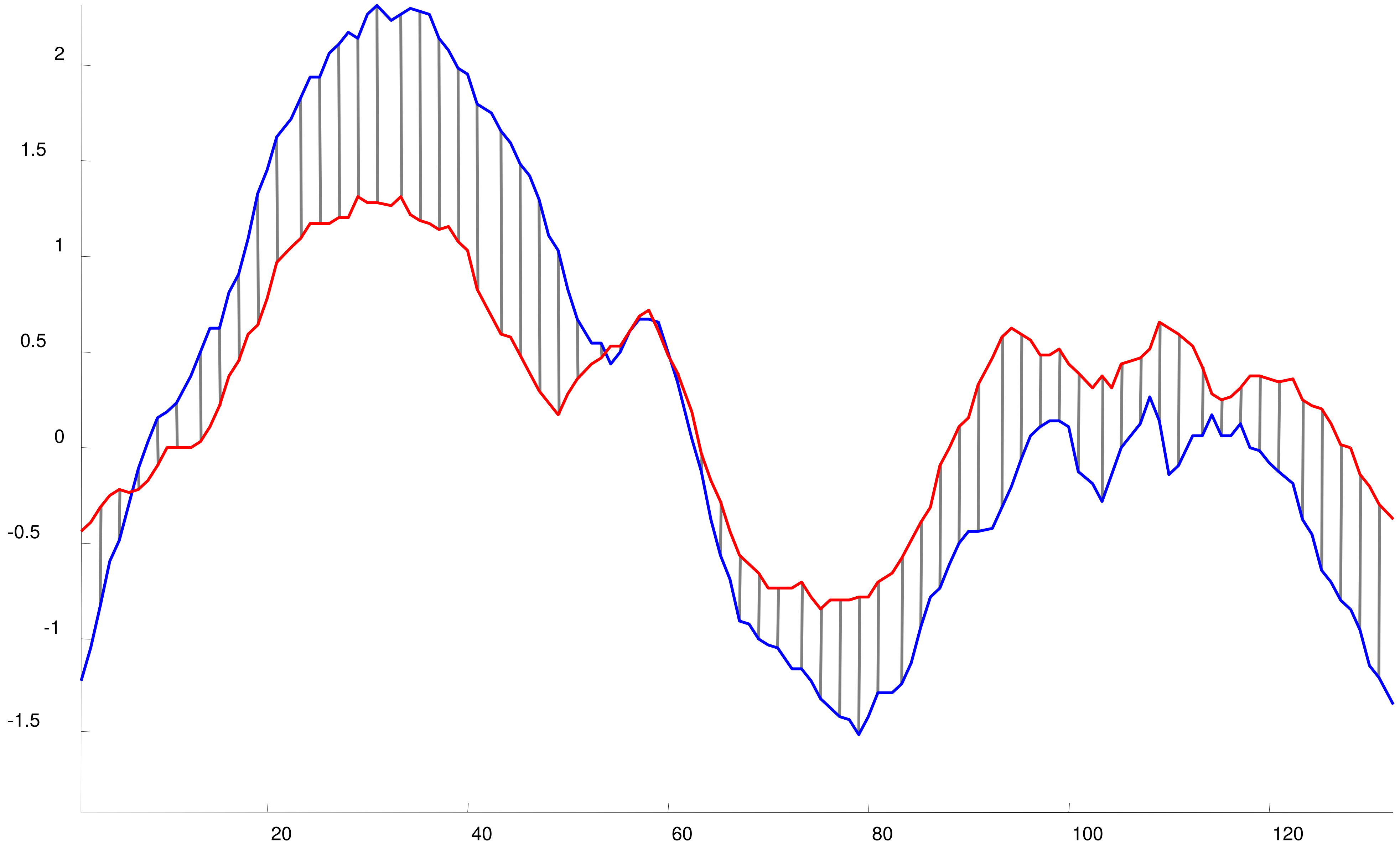
Séries temporais



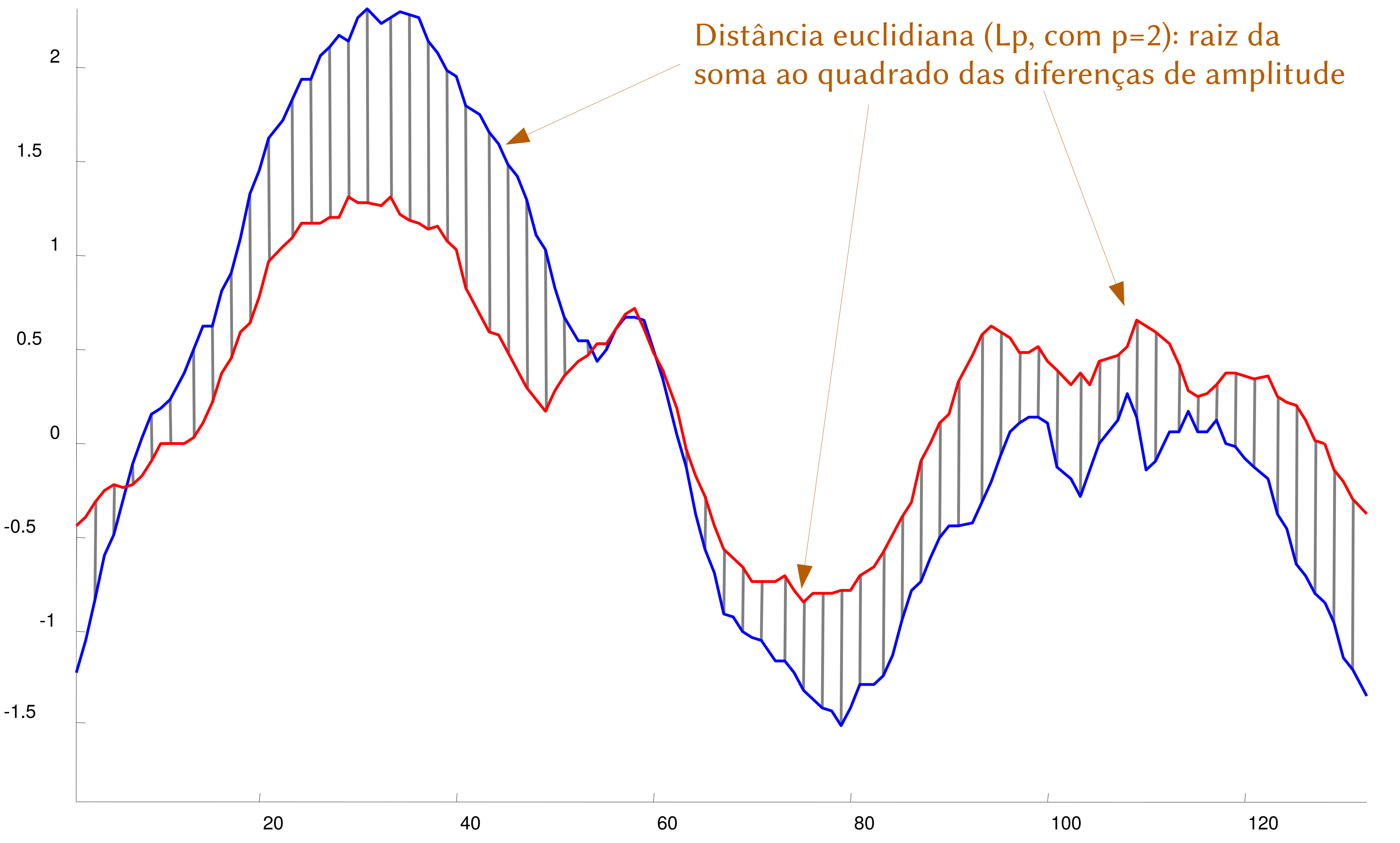
Séries temporais



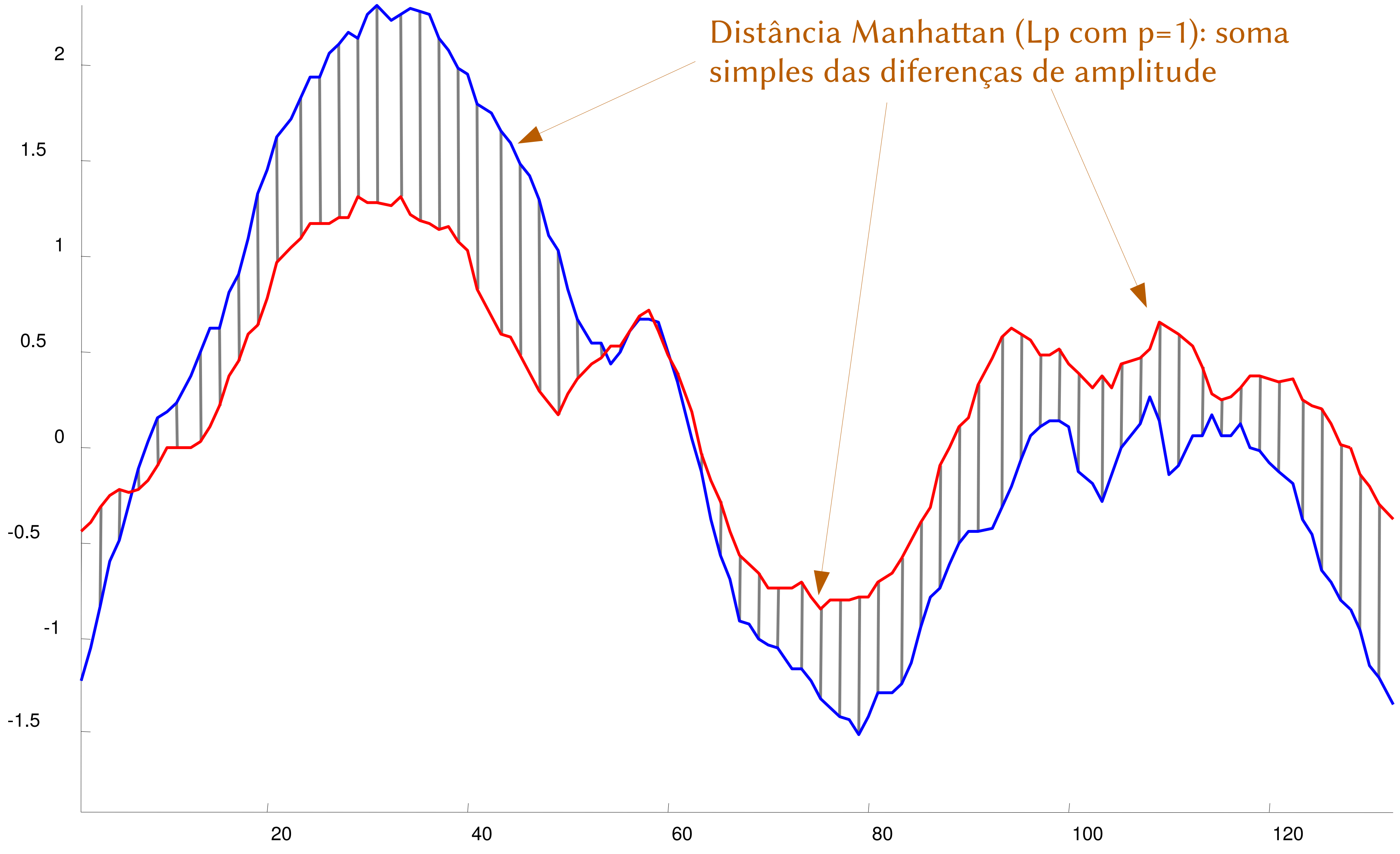
Séries temporais



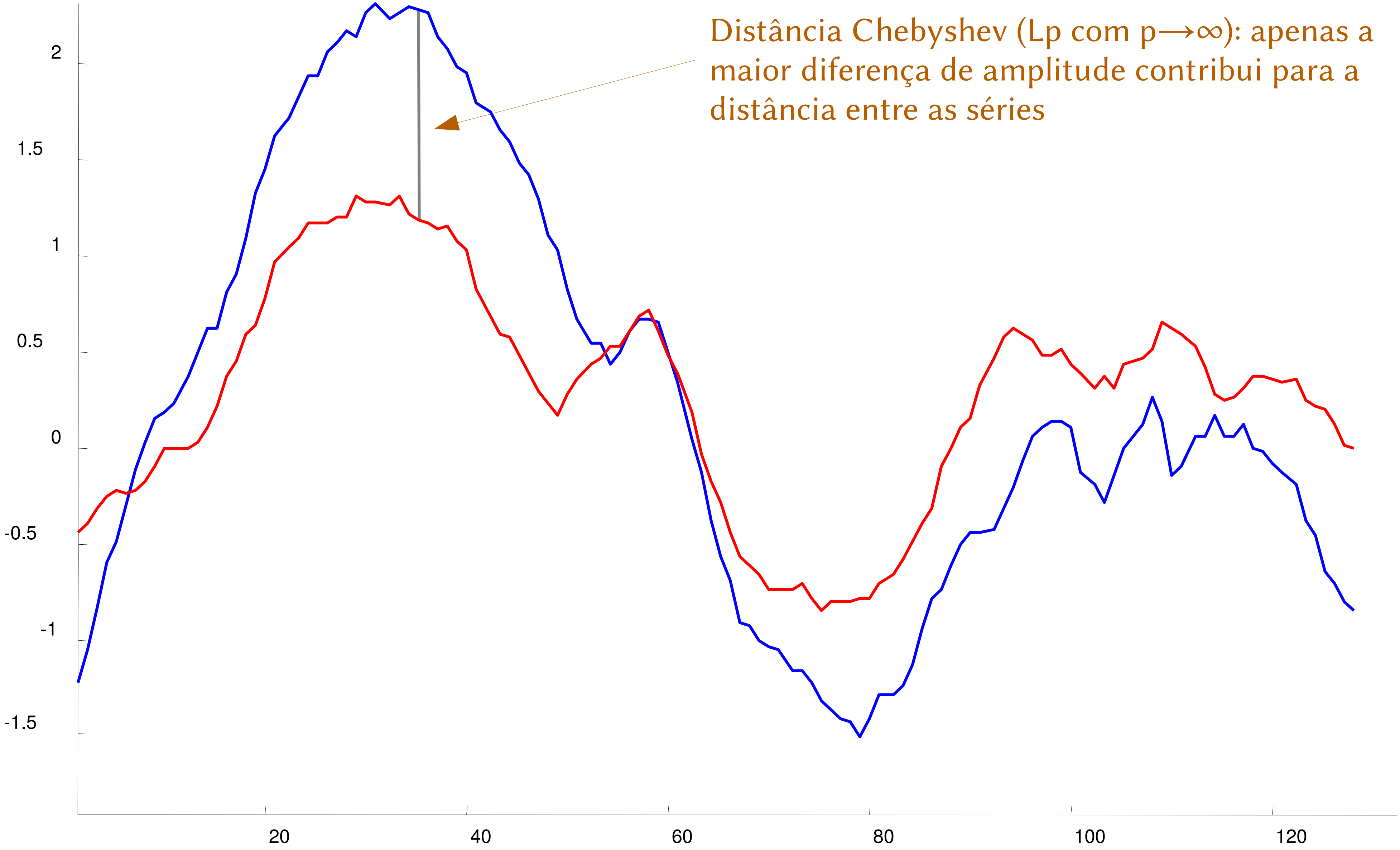
Séries temporais



Séries temporais



Séries temporais



Distância Chebyshev (Lp com p⟶∞): apenas a maior diferença de amplitude contribui para a distância entre as séries

Agenda

- Definições
- Teoria das probabilidades
- Aprendizado Bayesiano e modelos probabilísticos
- Modelos baseados em árvores
- Modelos baseados em regras
- Classificação preguiçosa: k-NN
- Máquina de vetores de suporte

Máquina de Vetores de Suporte

- SVM (*Support Vector Machines*)
  - Modelo de classificação baseado em instâncias
  - Utiliza-se de algumas instâncias denominadas **vetores de suporte**
  - Considerado um modelo de alto desempenho
  - Baseado na teoria de **aprendizado estatístico** e em **otimização matemática**

Máquina de Vetores de Suporte

- Vladimir Vapnik
  - Um dos "pais" da **teoria Vapnik-Chernovenkis**
  - "Pai" da teoria do apren-dizado estatístico
  - E co-inventor do modelo SVM

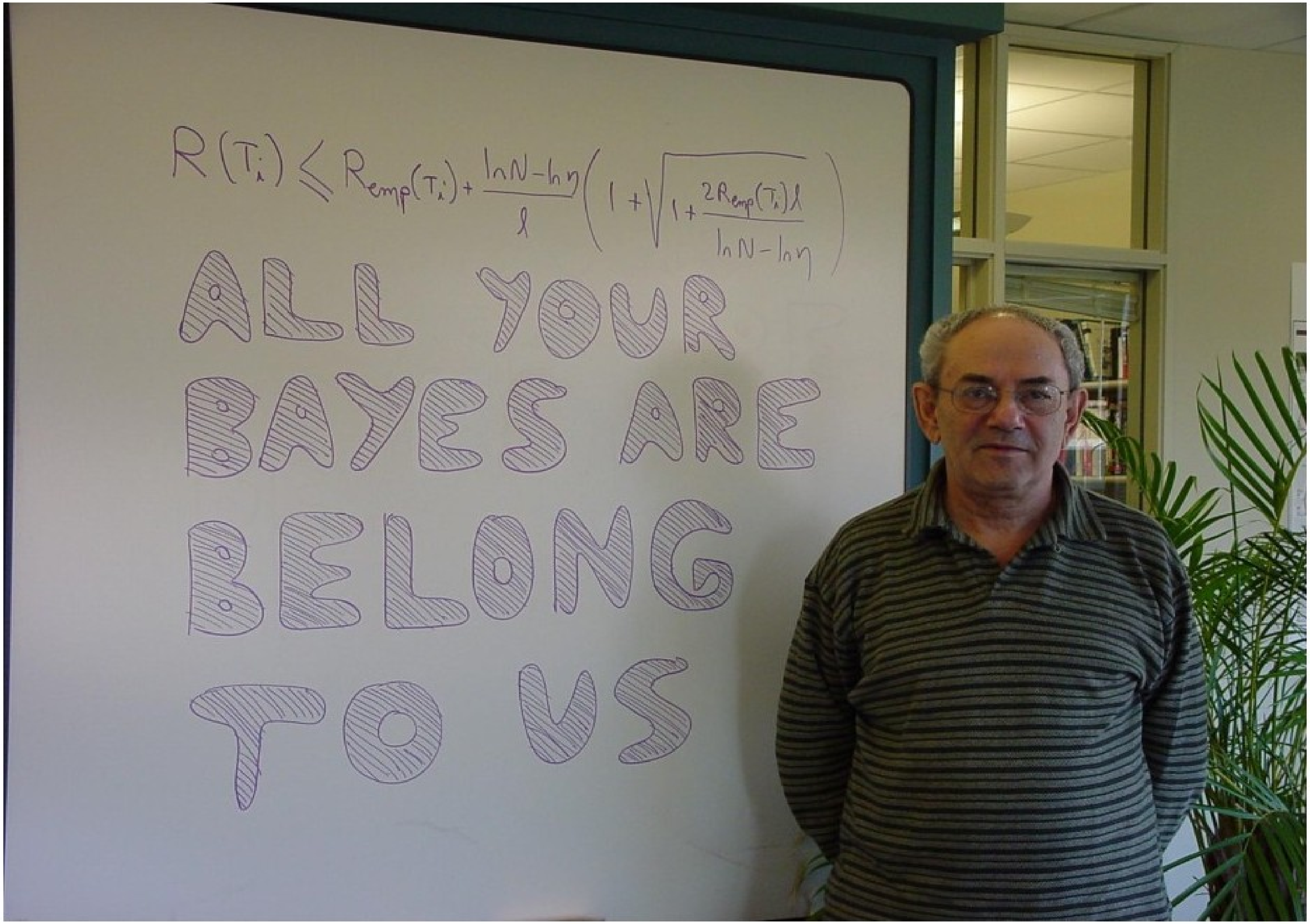
https://thedatascientist.com/vladimir-vapnik-new-model-learning/

Má

Facebook's AI team hires Vladimir Vapnik, father of the popular support vector machine algorithm

JORDAN NOVET   NOVEMBER 25, 2014 1:23 PM

TAGS: ARTIFICIAL INTELLIGENCE, DEEP LEARNING, FACEBOOK, VLADIMIR VAPNIK, YANN LECUN



Above: Vladimir Vapnik
Image Credit: Yann LeCun

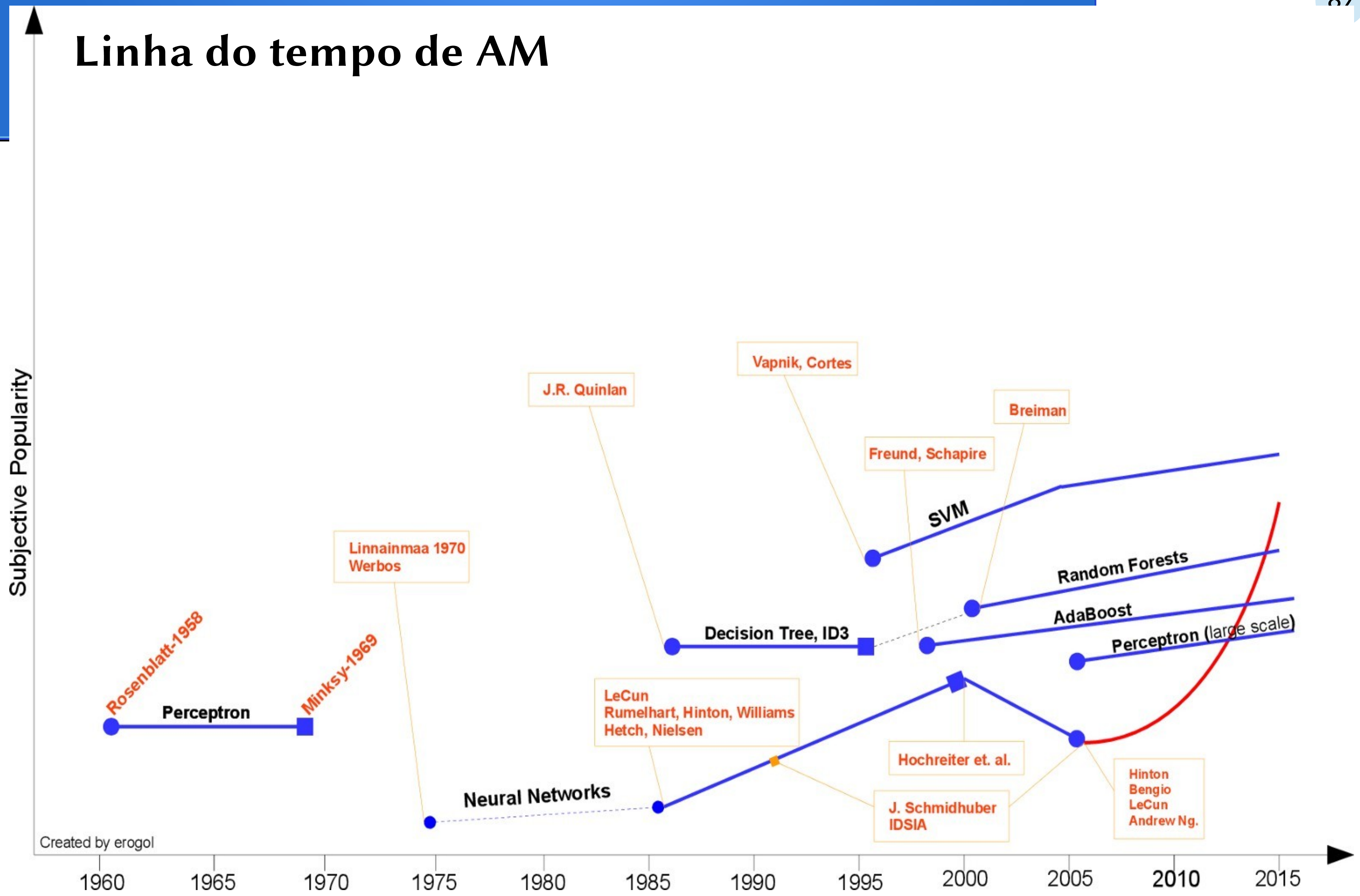
https://venturebeat.com/2014/11/25/facebooks-ai-team-hires-vladimir-vapnik-father-of-the-popular-support-vector-machine-algorithm/

Máquina de Vetores de Suporte

- O modelo SVM foi inventado em 1963
  - Classificador de separação linear
- Em 1992, Boser et al. propuseram uma forma de utilizar o chamado truque de kernel para criar classificadores não lineares

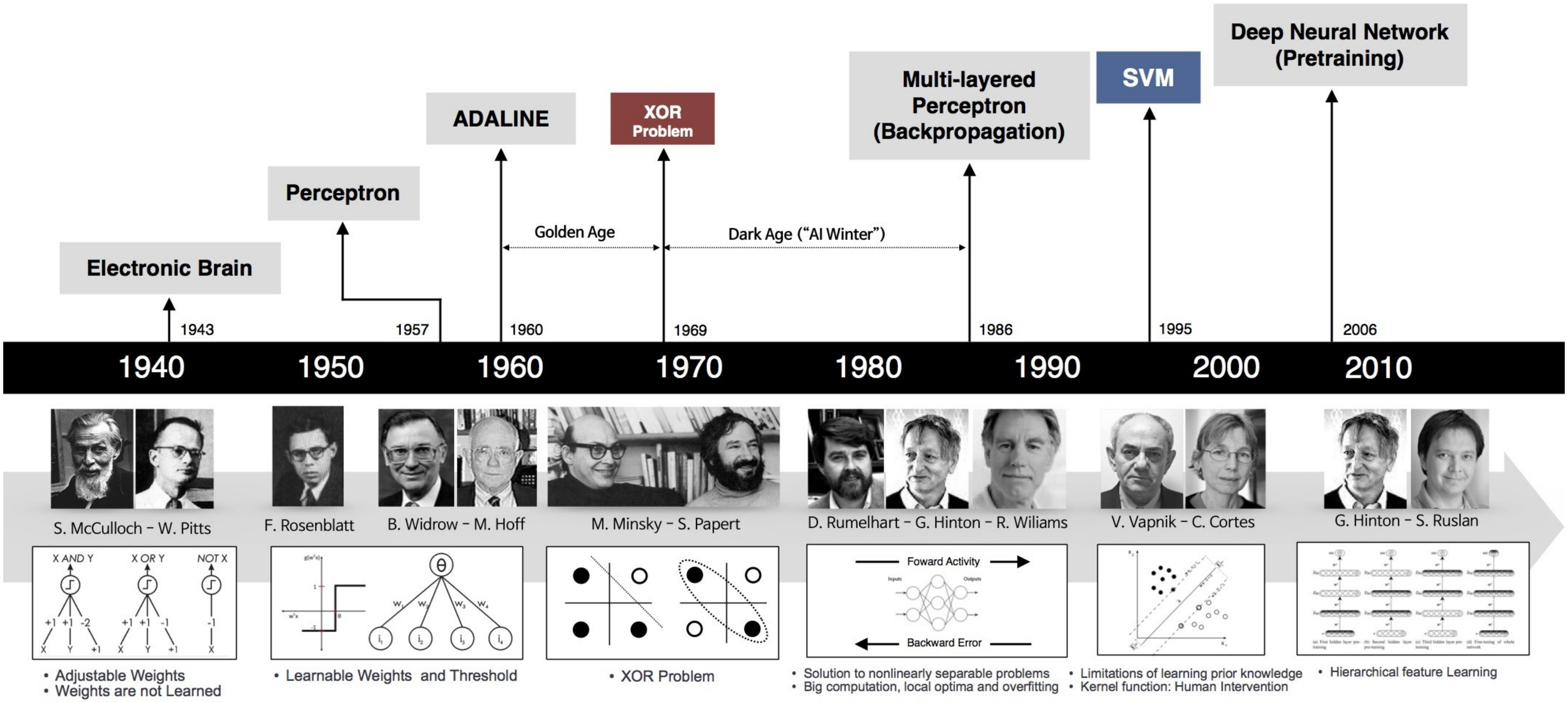
Bernhard E. Boser, Isabelle M. Guyon, Vladimir N. Vapnik (1992). "*A training algorithm for optimal margin classifiers*". *Proceedings of the Fifth Annual Workshop on Computational Learning Theory* – COLT '92. p. 144.

Linha do tempo de AM



https://chatbotnewsdaily.com/since-the-initial-standpoint-of-science-technology-and-ai-scientists-following-blaise-pascal-and-804ac13d8151/

Linha do tempo de AM (2)



https://beamandrew.github.io/deeplearning/2017/02/23/deep_learning_101_part1.html

Máquinas de vetores de suporte

- O que torna o SVM um modelo tão popupar?
  - Simplicidade
  - Teoria estatística

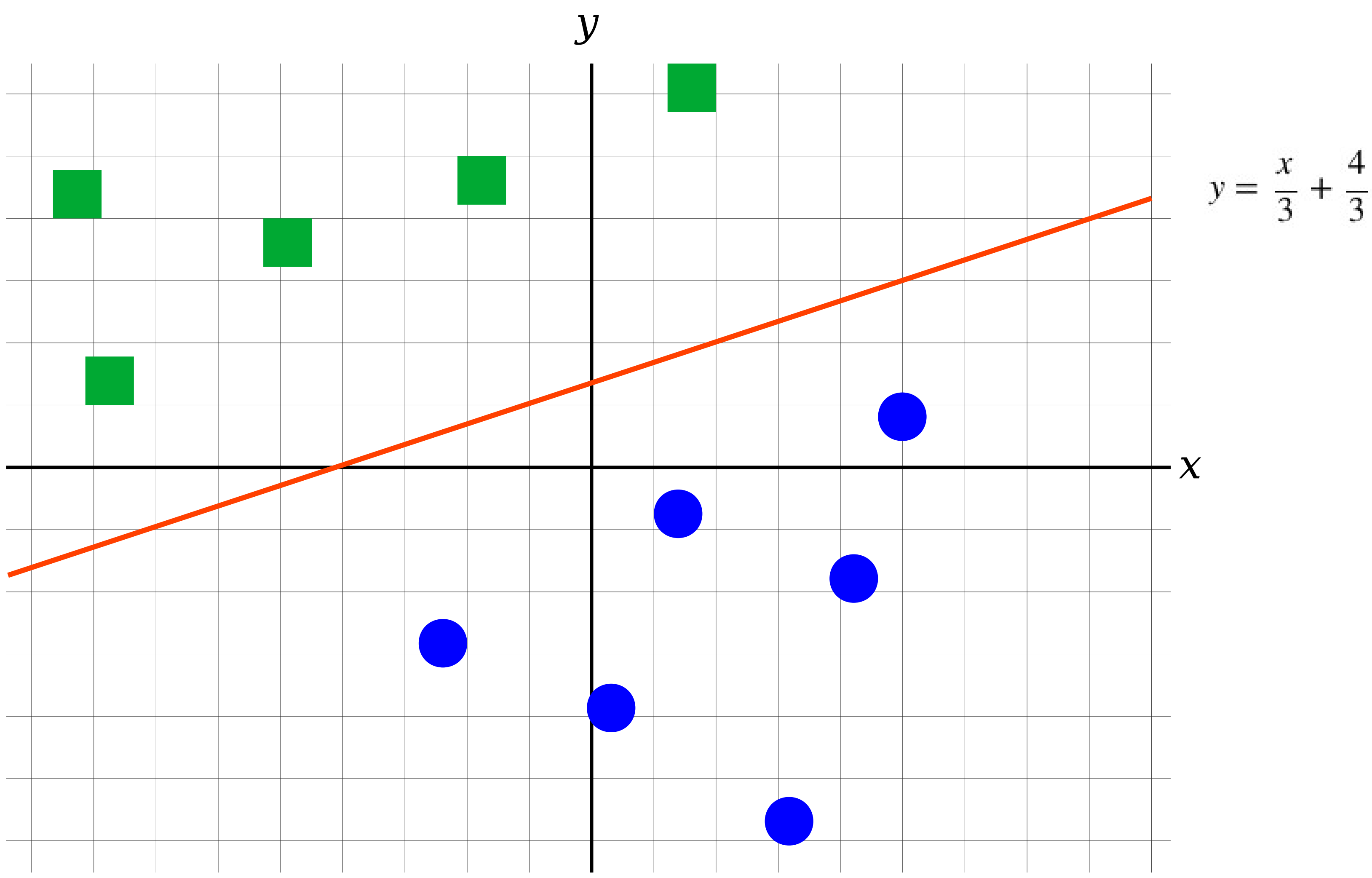
Separação linear

- A separação é regida por uma equação linear
  - $y(\mathbf{x}) = \mathbf{x} \cdot \mathbf{w} + w_0$
  - O vetor $\mathbf{w}$ é a norma da reta de separação
  - O coeficiente w_0, chamado viés, é proporcional à distância entre a reta e a origem

Reta, norma e viés

- Para ilustrar como a reta pode ser descrita por sua norma e um viés, considere um conjunto de pontos de duas categorias e uma reta separadora

Reta, norma e viés



Reta, norma e viés

- Considere a equação

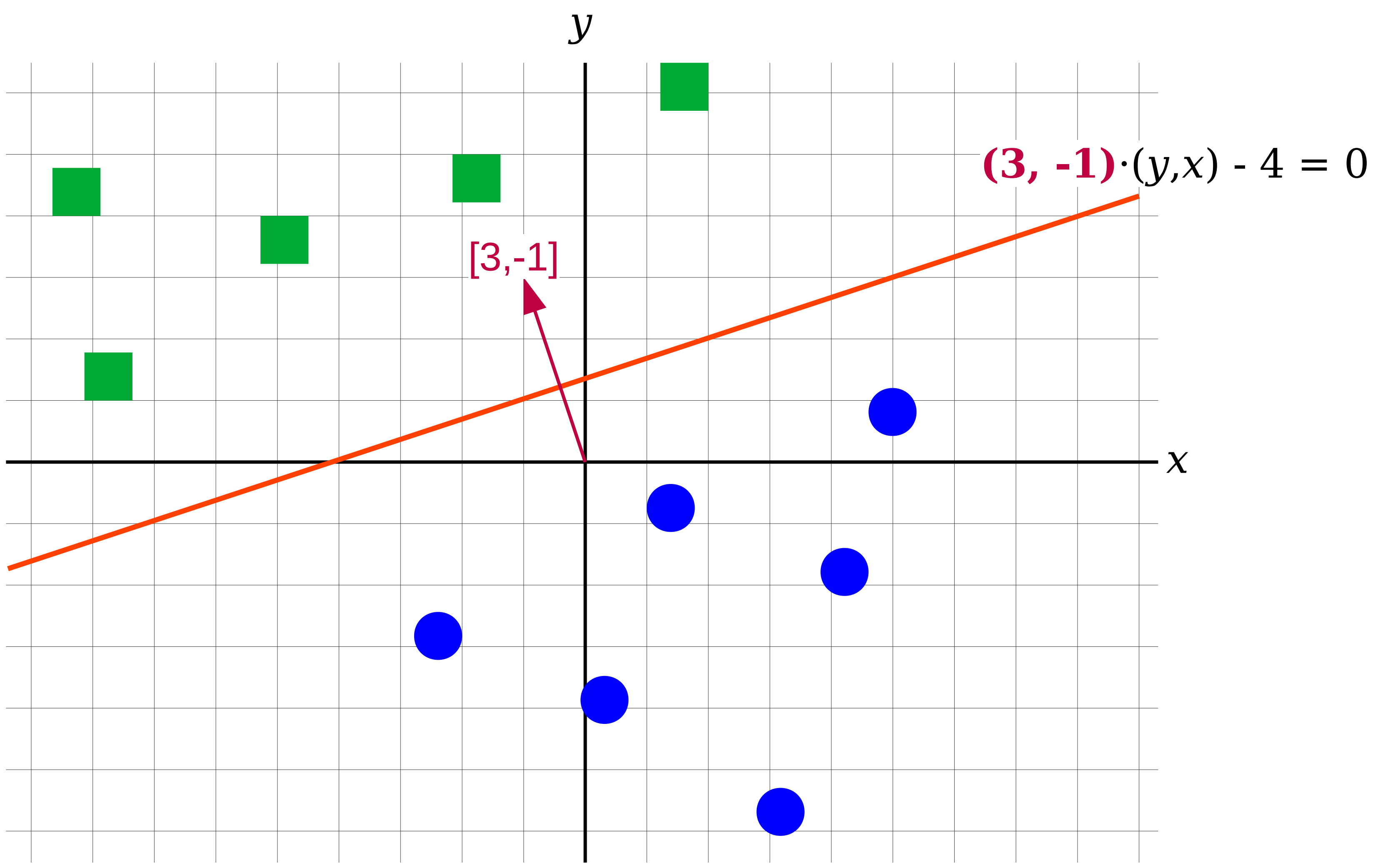
$$y = \frac{x}{3} + \frac{4}{3}$$

- Re-escrita na forma de produto escalar, temos

$$\left(1, \frac{-1}{3}\right) \cdot (y, x) - \frac{4}{3} = 0$$

$$(3, -1) \cdot (y, x) - 4 = 0$$

Reta, norma e viés



Reta, norma e viés

- Note que o vetor [3, -1] está na direção do caminho mais curto entre a origem e a reta
- O tamanho do vetor é $||\mathbf{w}|| = \sqrt{10} \approx 3{,}16$
- A distância entre a reta e a origem é $4/||\mathbf{w}|| \approx 1{,}2$

SVM linear

- Os parâmetros do modelo são os coeficientes do vetor $\mathbf{w}$ e o viés w_0
- O modelo classifica de forma que
  - $y(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + w_0$
  - Se $y(\mathbf{x}) \geq 0$, então os exemplos pertencem à classe positiva
  - Se $y(\mathbf{x}) < 0$, então os exemplos pertencem à classe negativa

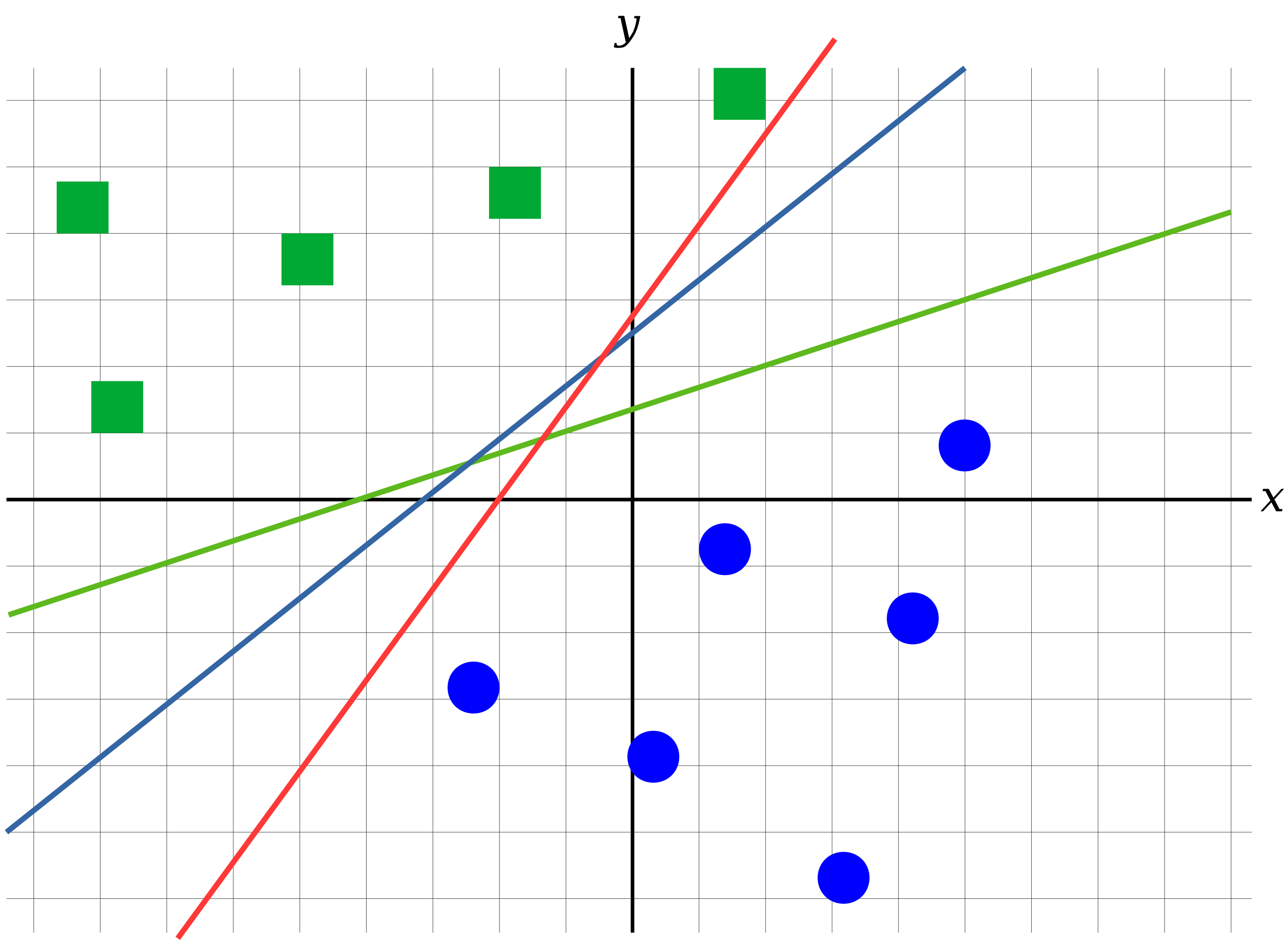
SVM linear

- Em 2 dimensões, o vetor **w** descreve uma reta e o parâmetro w_0 define sua distância à origem
  - Em 3 dimensões, um plano cuja distância à origem é proporcional a w_0
  - Para *n* dimensões, um hiperplano cuja distância à origem é proporcioal a w_0

SVM linear: treinamento

- O indutor SVM precisa encontrar um hiperplano que permita separar adequadamente os exemplos das classes
  - Vamos supor, inicialmente, que se tratam de problemas de classificação binária
  - Em seguida, utilizaremos múltiplos classificadores SVM para problemas multiclasse
    - (Não confundir "multiclasse" e "multirrótulo")

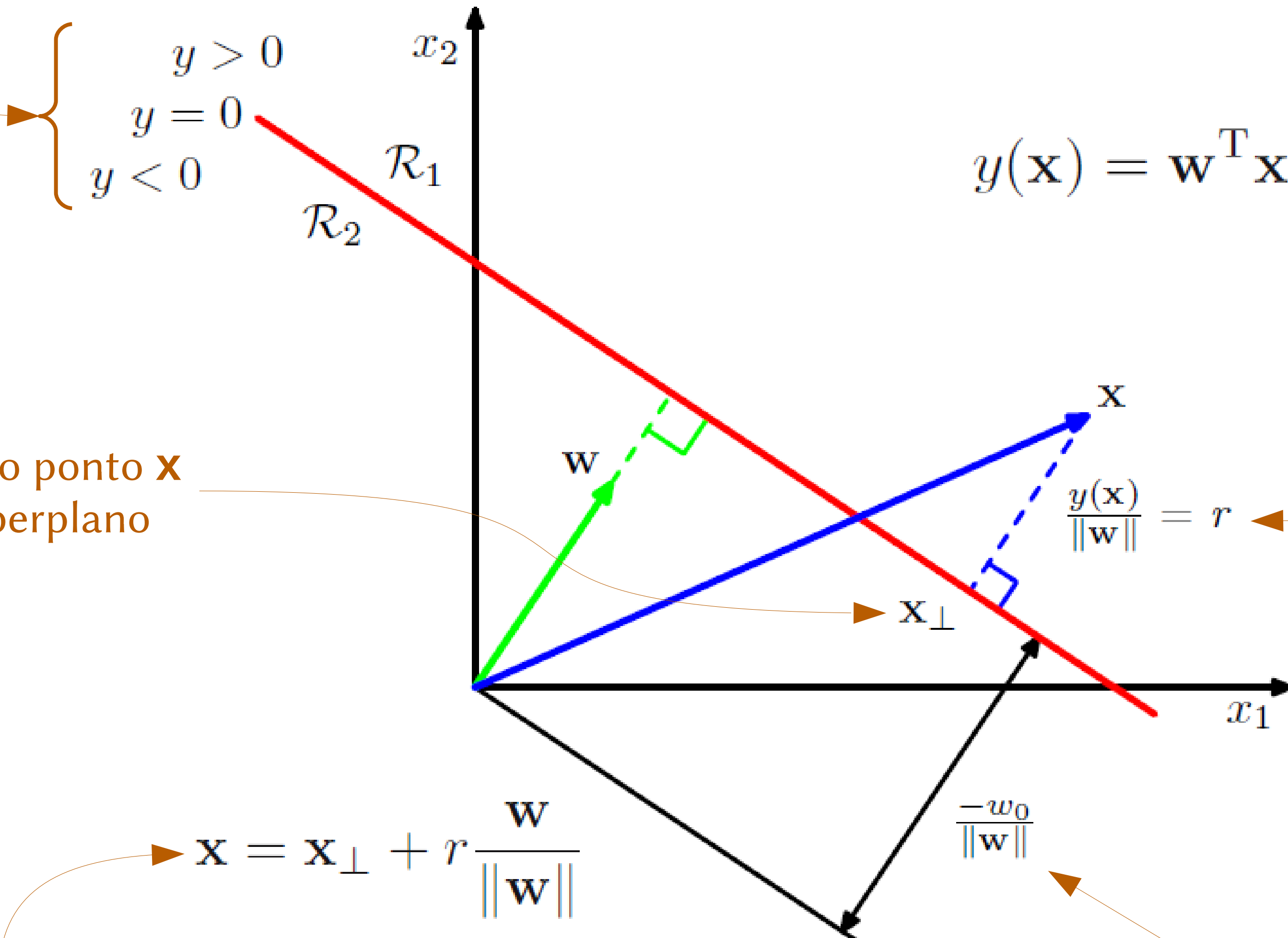
SVM linear: treinamento



Qual hiperplano deve ser escolhido?

Hiperplano de separação máxima

Relação do sinal da equação y(x) com sua posição no espaço



$$y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0$$

Distância entre um ponto **x** e o hierplano

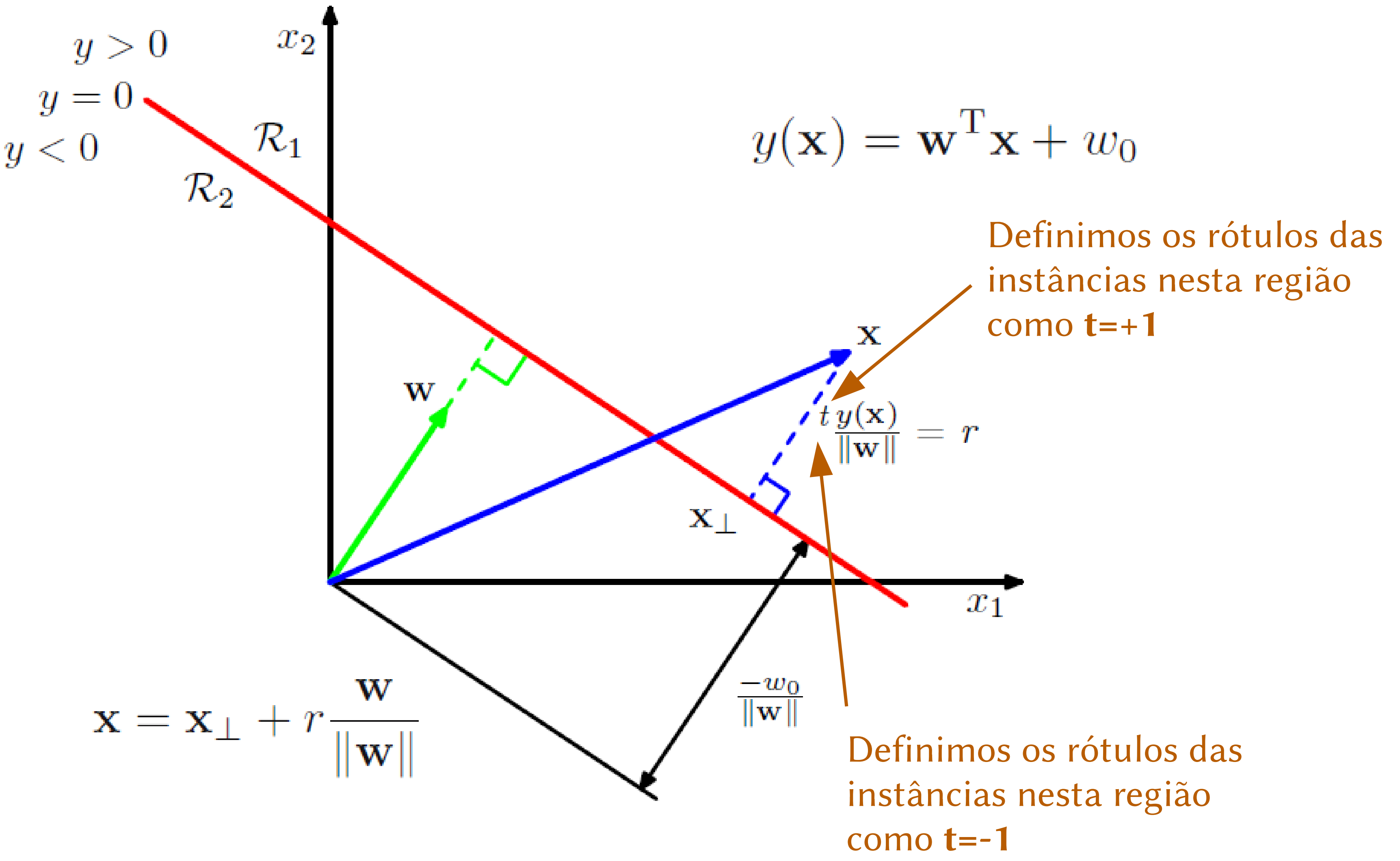
Projeção do ponto **x** sobre o hiperplano

$$\mathbf{x} = \mathbf{x}_\perp + r\frac{\mathbf{w}}{\|\mathbf{w}\|}$$

"Saindo" da projeção do ponto **x**, na direção do vetor w, com distância r, "chegamos" ao ponto **x**

Distância entre o hiperplano e a origem

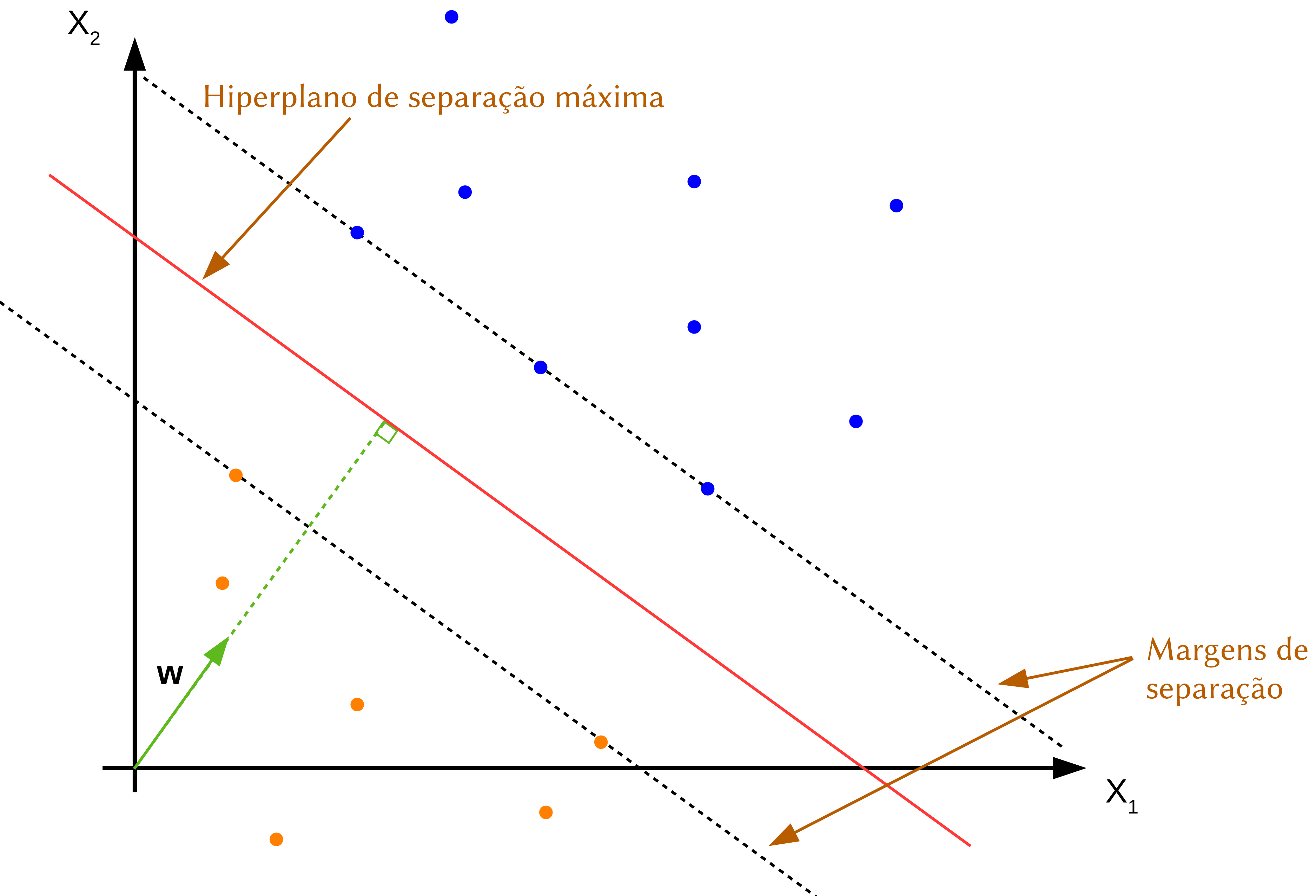
Hiperplano de separação máxima



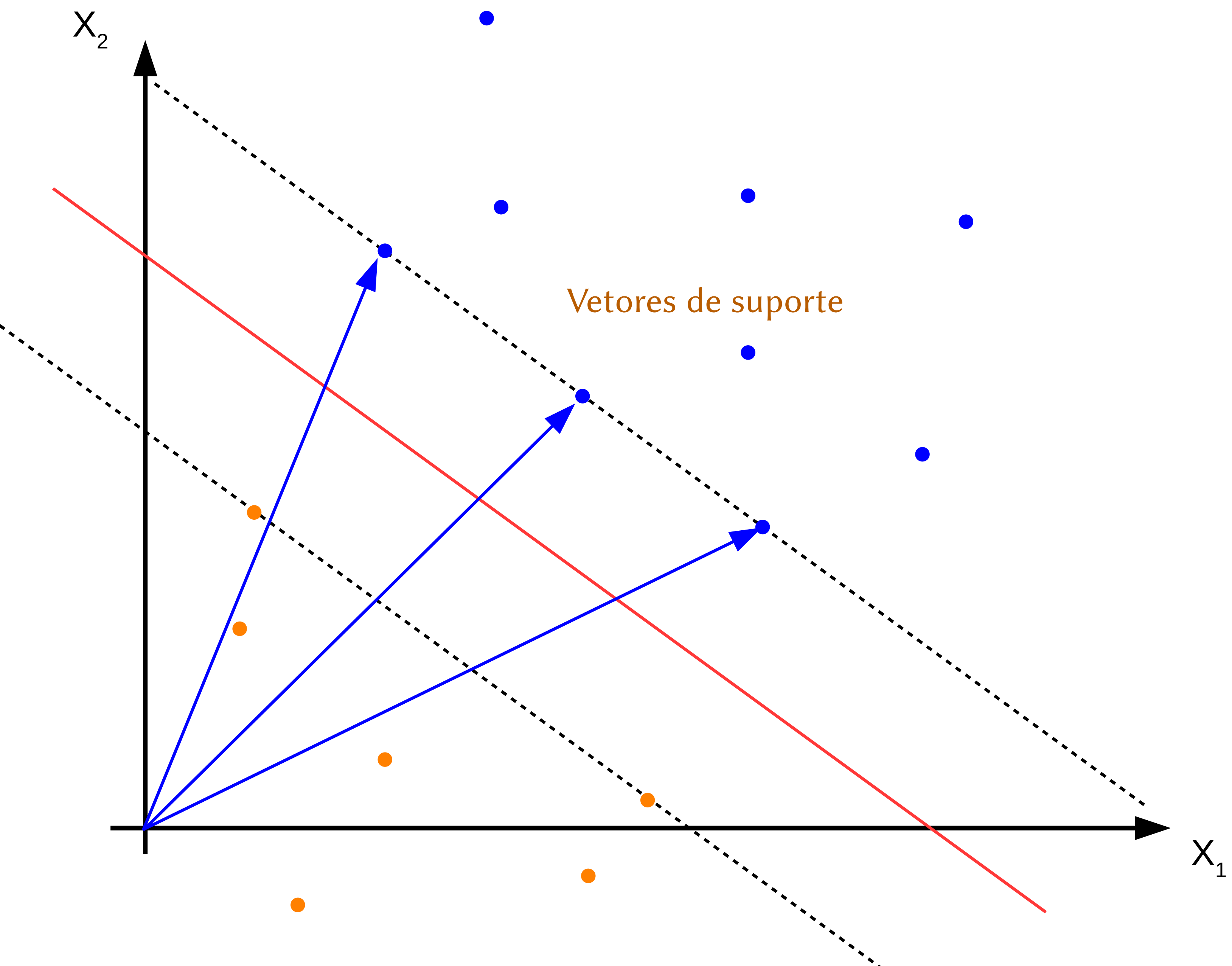
Hiperplano de separação máxima

- Vamos encontrar o hiperplano cuja distância para os pontos seja a maior possível
  - Os pontos mais próximos do hiperplano são denominados vetores de suporte
  - Definiremos **w** de modo que, para os vetores de suporte da classe positiva, teremos $y(\mathbf{x}_i) = 1$ e $y(\mathbf{x}_i) = -1$ para vetores da classe negativa
  - Assim, $t_i\mathbf{x}_i = 1$ para todos os vetores de suporte

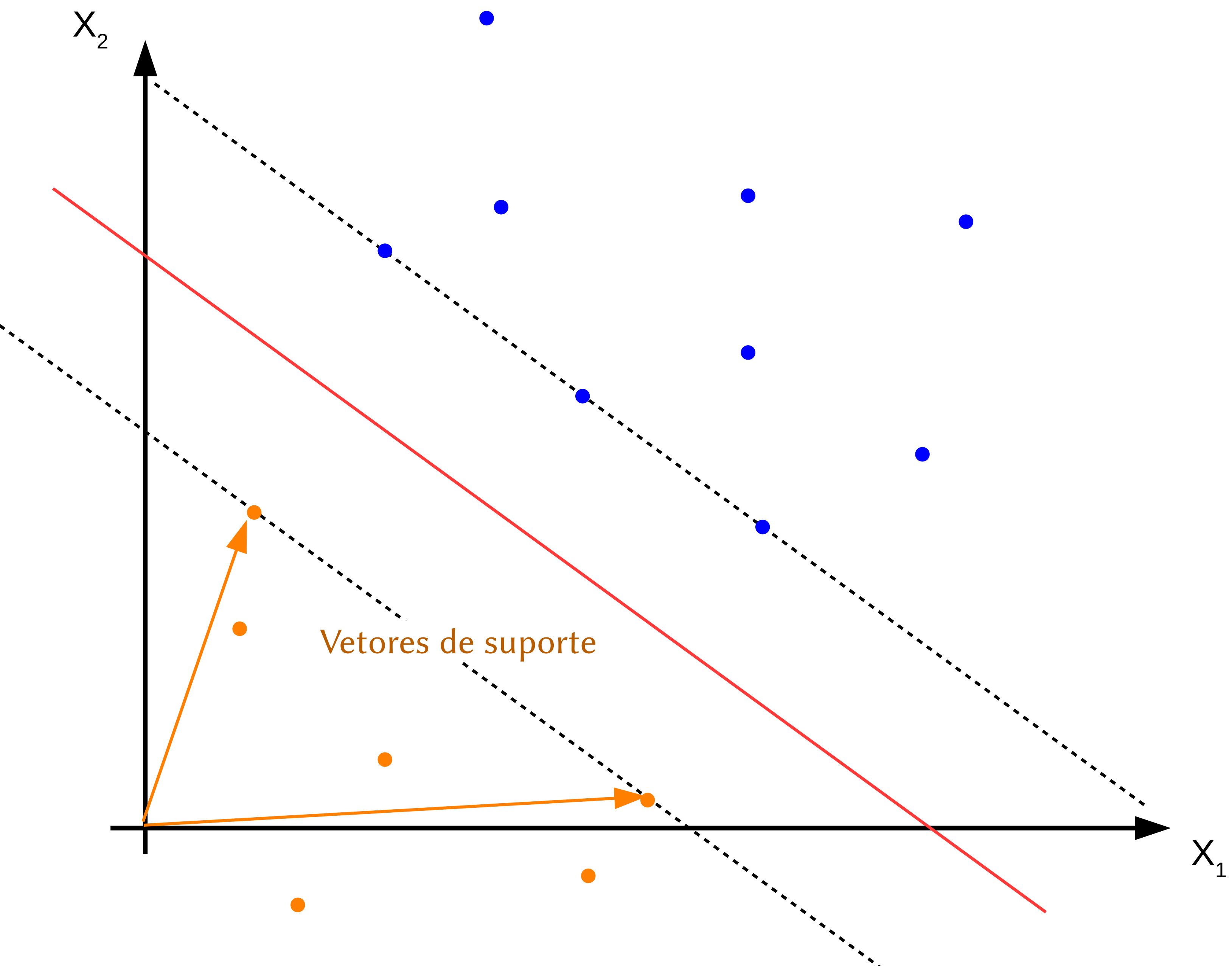
Hiperplano de separação máxima



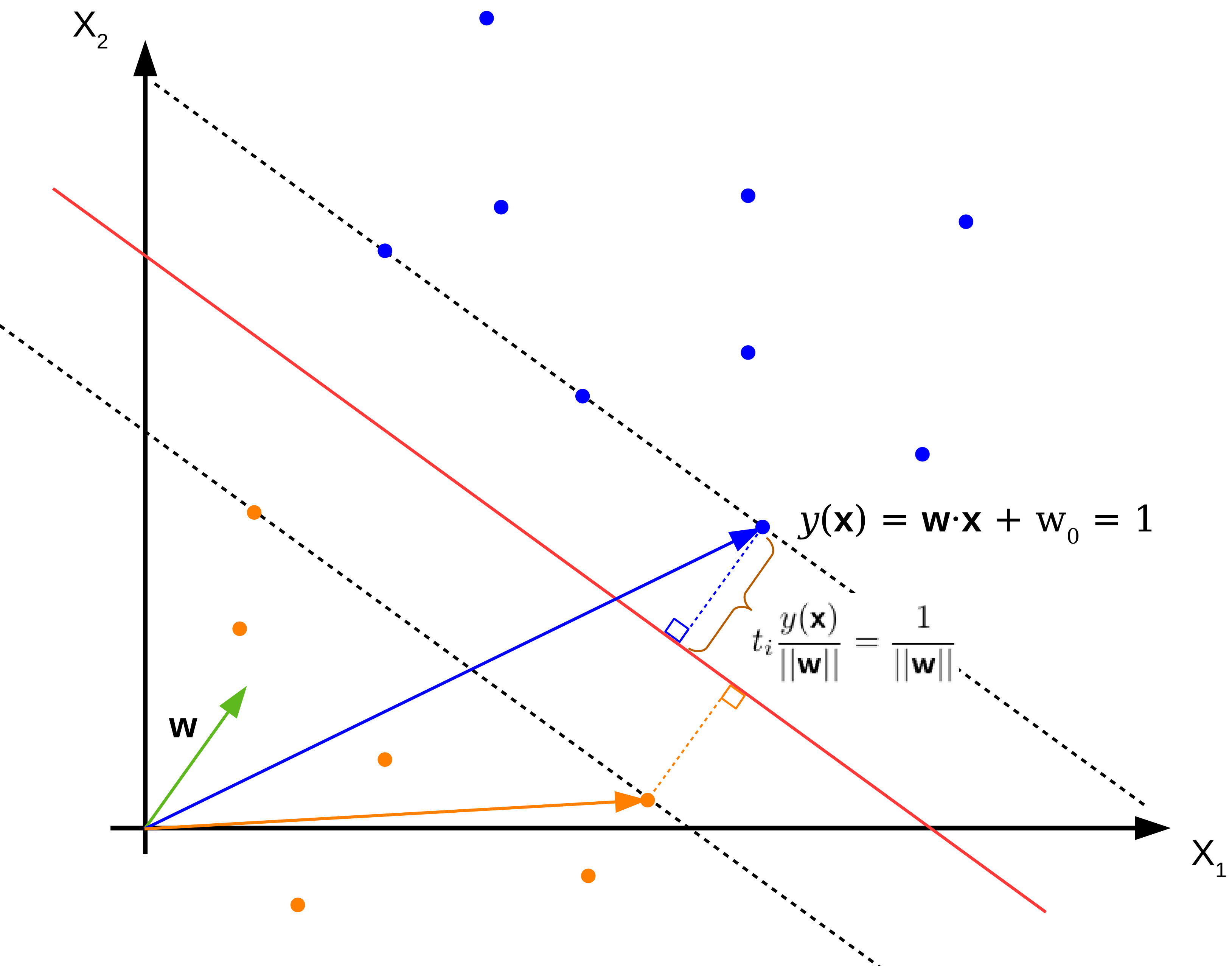
Hiperplano de separação máxima



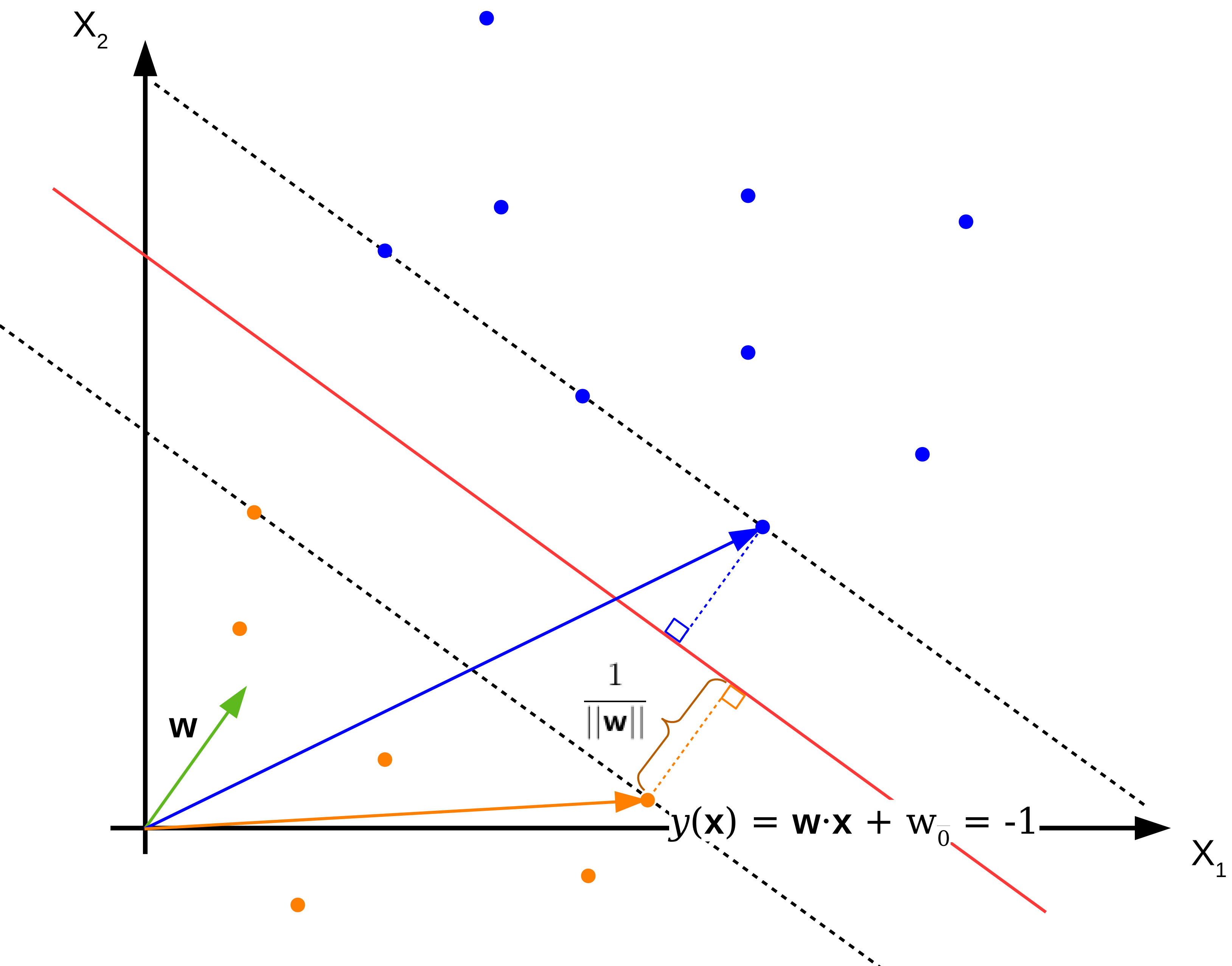
Hiperplano de separação máxima



Hiperplano de separação máxima



Hiperplano de separação máxima



Hiperplano de separação máxima

- O hiperplano define a seguinte restrição para todos os exemplos de treinamento
  - $t_i y(\mathbf{x}_i) \geq 1$
  - $t_i(\mathbf{w} \cdot \mathbf{x}_i + w_0) \geq 1$
- Isso pode ser traduzido em um problema de otimização
  - Minimize $\mathbf{w}$
  - Sujeito a $t_i(\mathbf{w} \cdot \mathbf{x}_i + w_0) \geq 1$

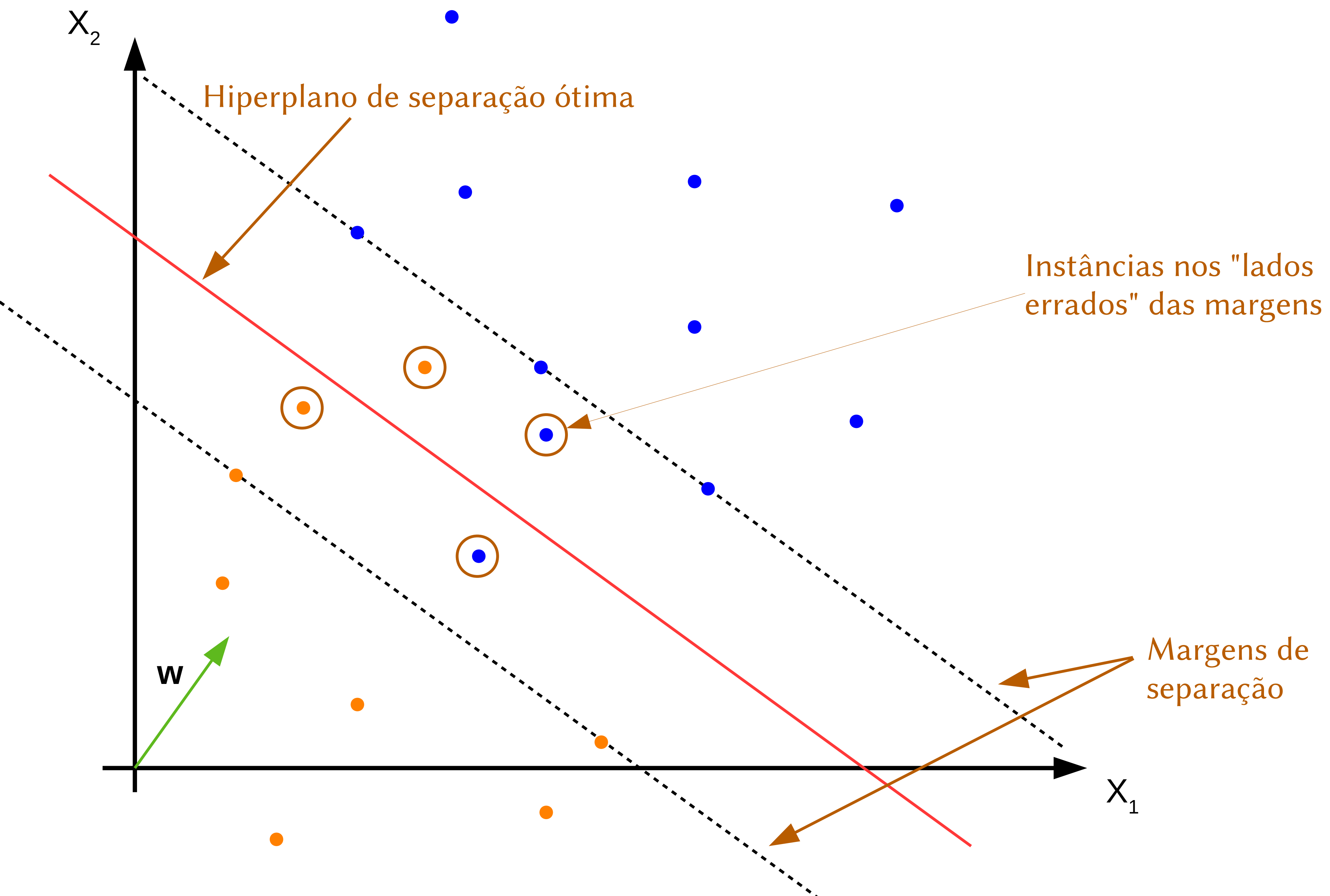
Hiperplano de separação máxima

- É um problema de otimização quadrática
  - Existe um algoritmo $\mathcal{O}(n^3)$ que encontra os parâmetros que minimizam **w** com as restrições dadas
  - Mas vamos verificar a margem flexível

Hiperplano de margem flexível

- O classificador proposto só pode ser empregado em problemas que são linearmente separáveis
  - Em alguns casos, essa separação pode ser impossível
  - Nossa restrição é muito forte
- Podemos flexibilizar essa restrição admitindo que alguns pontos fiquem no lado errado da margem
  - Função de perda

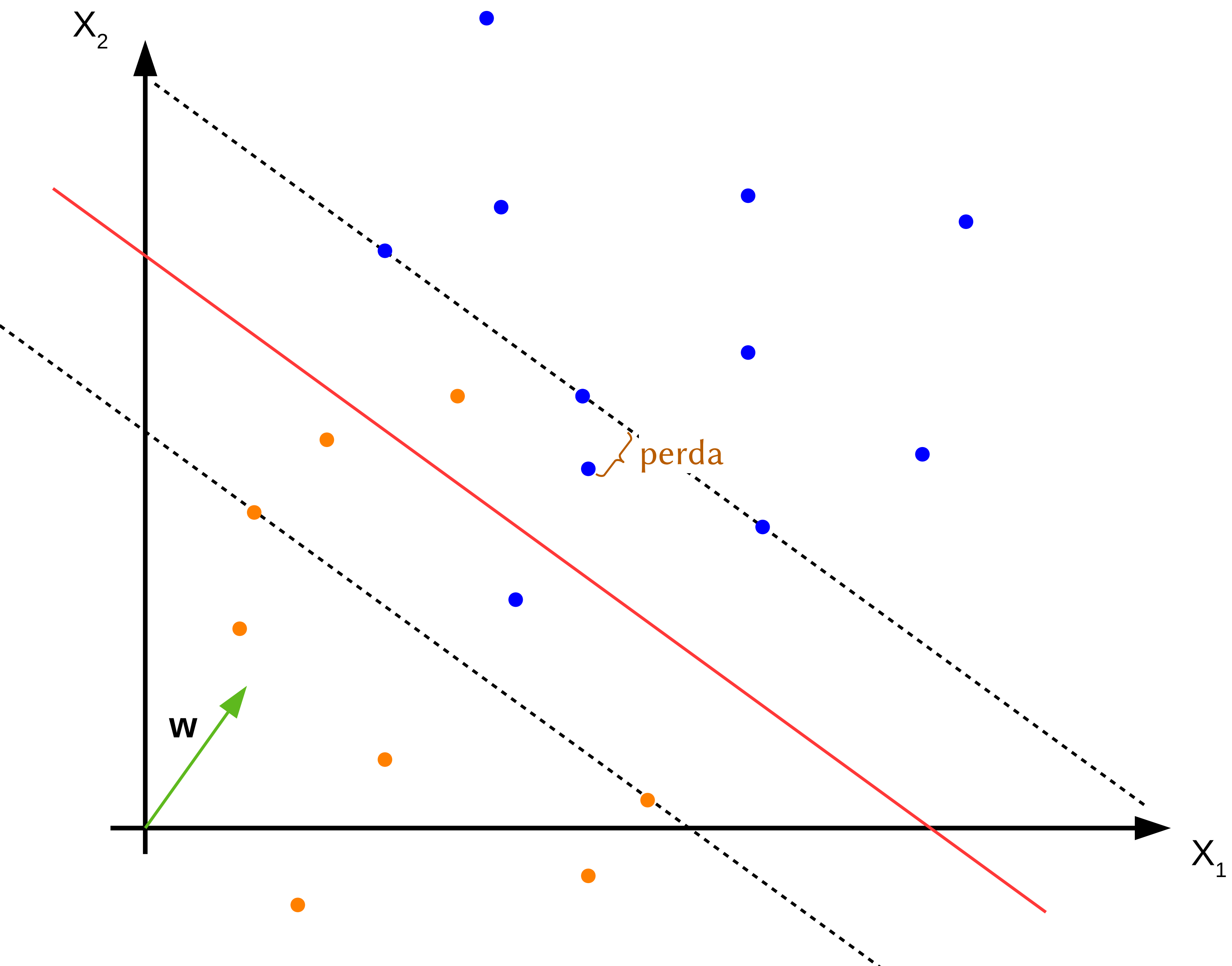
Hiperplano de margem flexível



Hiperplano de margem flexível

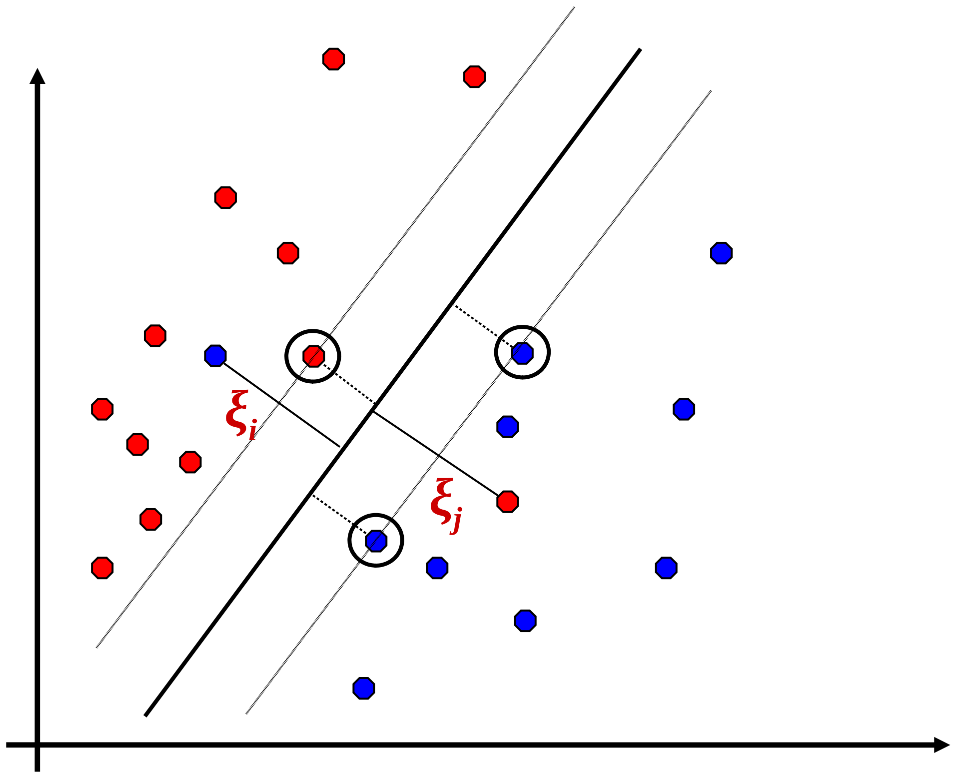
- Empregamos uma função de perda
  - *Hinge loss*
  - $L(\mathbf{x}_i) = \max[0,\ 1 - t_i(\mathbf{w} \cdot \mathbf{x}_i + w_0)]$
    - Para um ponto na margem correta, $t_i(\mathbf{w} \cdot \mathbf{x}_i + w_0) = 1$, portanto $L(\mathbf{x}_i) = 0$
    - Para pontos "dentro" da margem correta, $t_i(\mathbf{w} \cdot \mathbf{x}_i + w_0) > 1$, portanto $L(\mathbf{x}_i) = 0$
    - Para pontos "do lado errado", $t_i(\mathbf{w} \cdot \mathbf{x}_i + w_0) < 1$, portanto $L(\mathbf{x}_i) > 1$

Hiperplano de margem flexível



Hiperplano de margem flexível

- Também podemos pensar na margem flexível através da introdução de variáveis auxiliares ξ_i
- Otimize o mesmo problema original, utilizando as variáveis auxiliares para compensar as classificações incorretas



Hiperplano de margem flexível

- Podemos encontrar w minimizando

$$\left[\frac{1}{n}\sum_{i=1}^{n}\max(0, 1 - t_i(\mathbf{w}\cdot\mathbf{x}_i + w_0)\right] + \lambda||\mathbf{w}||^2$$

- Ou transformando para o seguinte problema de otimização

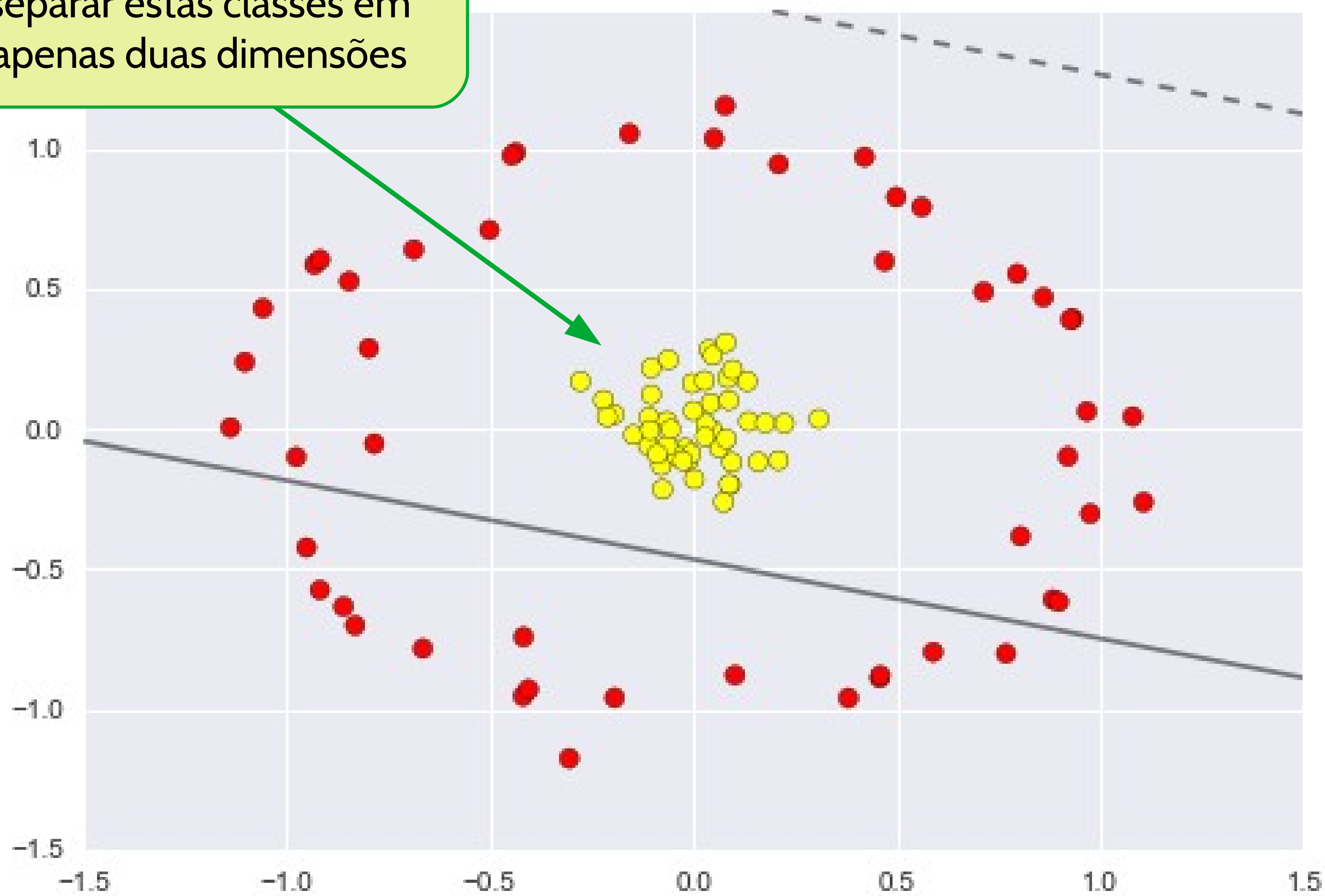
$$\begin{aligned} &(\min) \quad \frac{1}{n}\sum_{i=1}^{n}\xi_i + C||\mathbf{w}||^2 \\ &\text{s/a} \quad t_i(\mathbf{w}\cdot\mathbf{x}_i + w_0) \geq 1 \\ &\qquad\quad \xi_i \geq 0 \end{aligned}$$

Truque de *kernel*

- Note que, mesmo com a margem flexível, existe um limite além do qual o SVM não consegue mais separar os exemplos das diferentes classes
  - Alguns conjuntos não podem ser separados por um hiperplano
  - Nesse caso, o SVM faz uma transformação do espaço de atributos

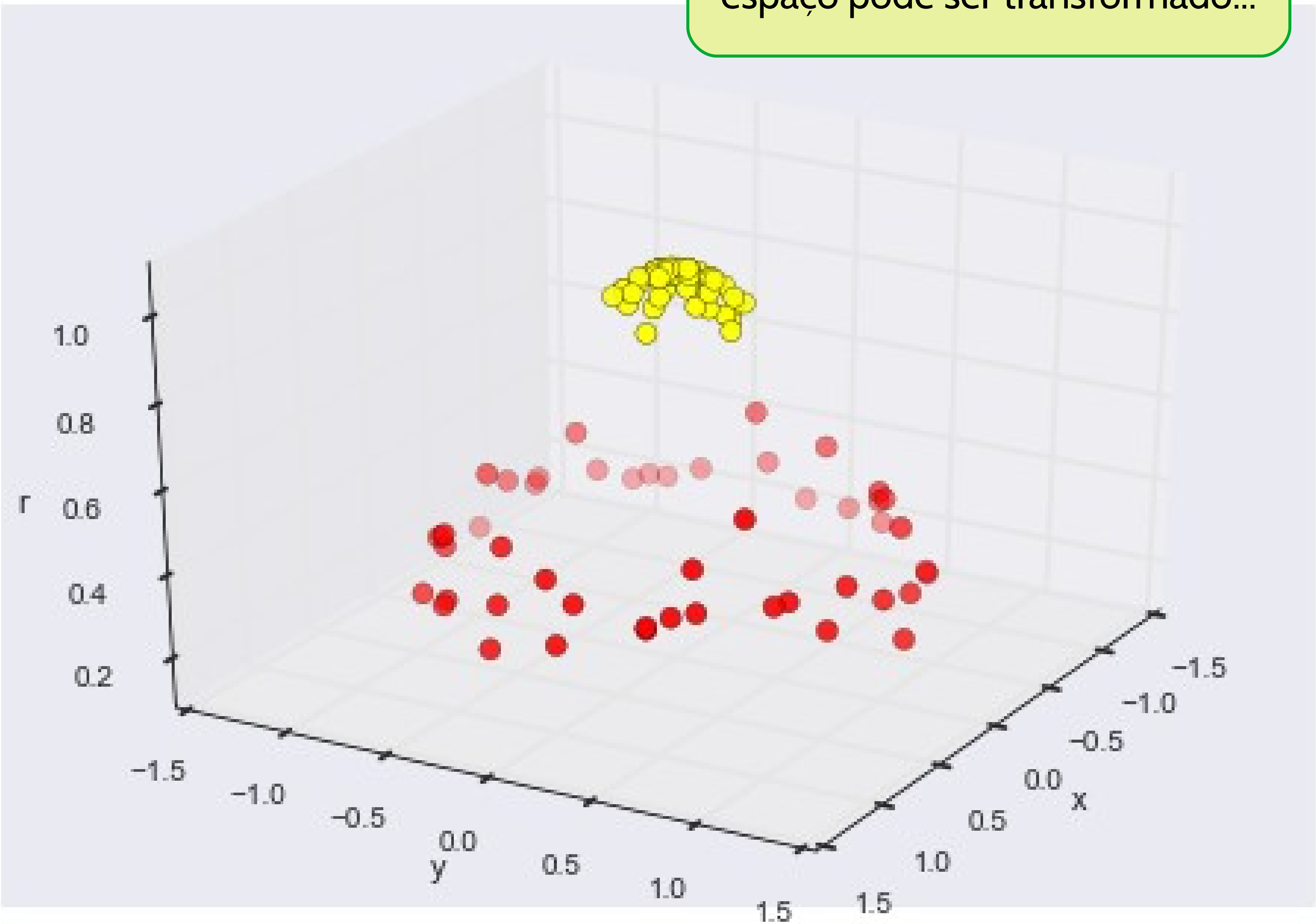
Truque de *kernel*

Nenhum hiperplano pode separar estas classes em apenas duas dimensões



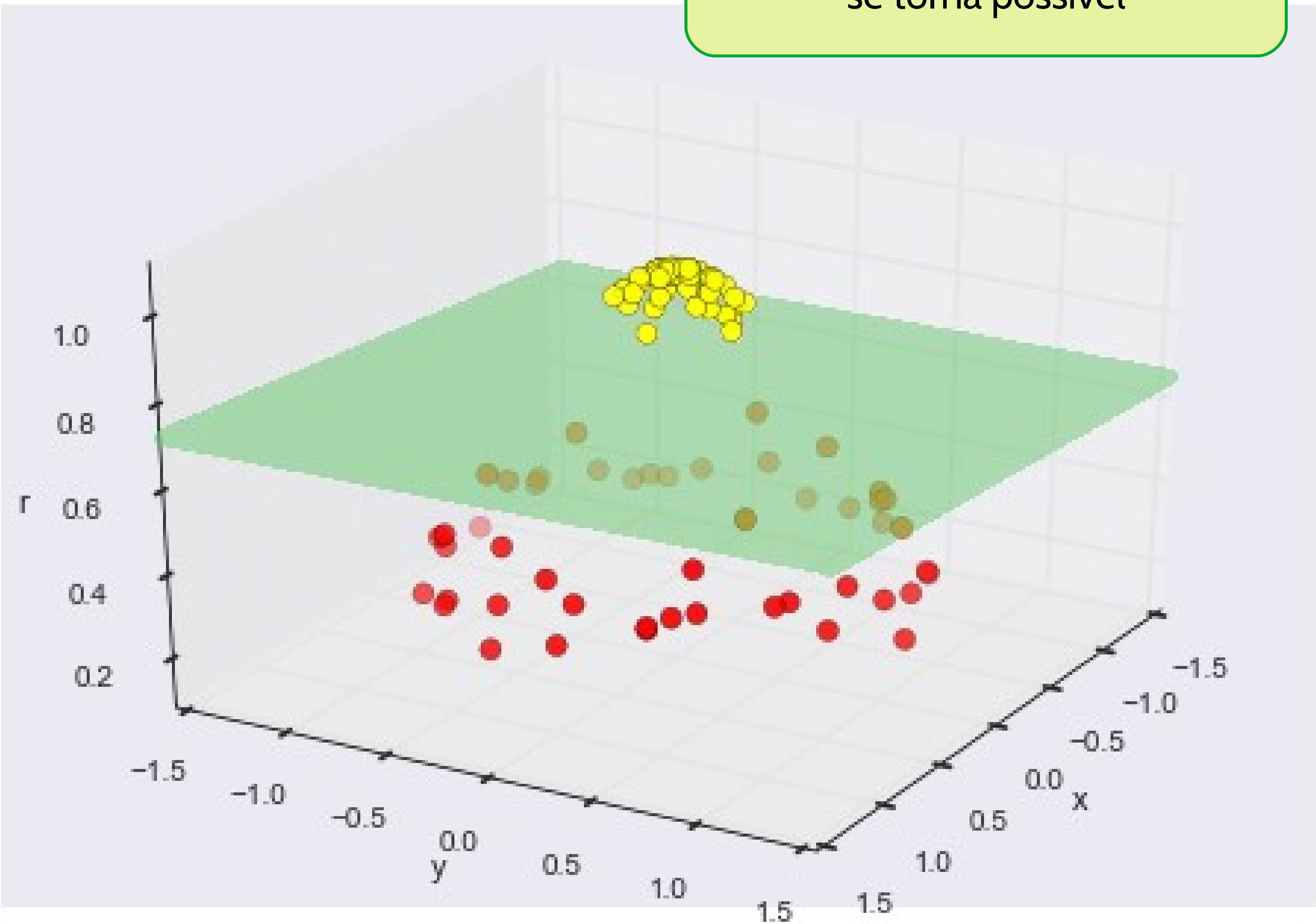
Truque de *kernel*

Mas em três dimensões o espaço pode ser transformado...



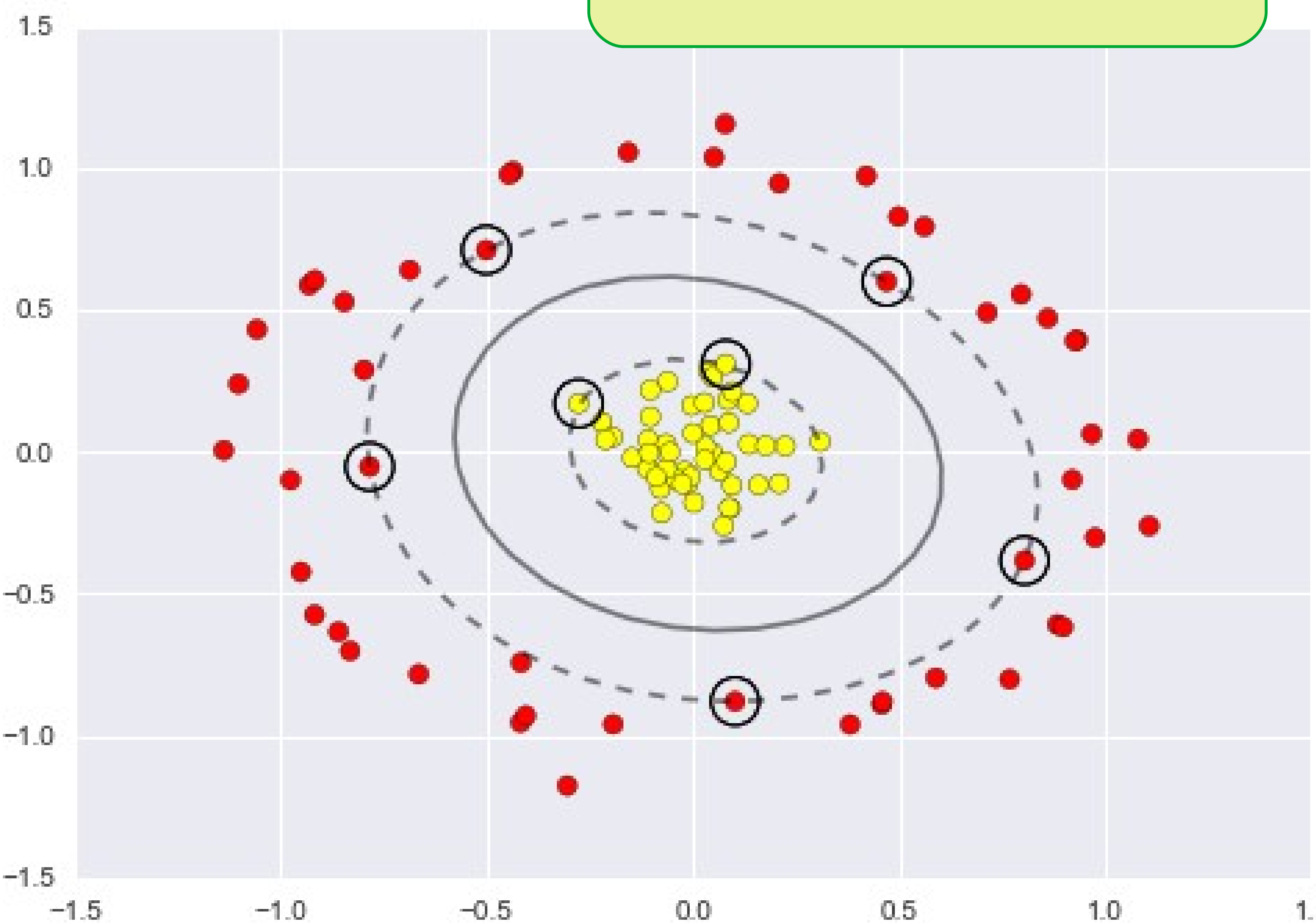
Truque de *kernel*

...e um hiperplano de separação se torna possível



Truque de *kernel*

Este é o mesmo hiperplano, visualizado no espaço original



Truque de *kernel*

- Essa projeção é realizada com uma função *kernel*
  - De forma simplificada, a função *kernel* fornece a distância entre os pontos no espaço de alta dimensão sem calcular explicitamente a projeção dos pontos
  - Substitui o produto escalar na função de perda

Truque de *kernel*

- Teste o SVM com truque de kernel:
- https://cs.stanford.edu/~karpathy/svmjs/demo/

Truque de *kernel*

- A motivação para o truque de *kernel* pode ser feita observando o classificador SVM **sem** *kernel*
- Para transportar os pontos do espaço original para um **espaço de características**, utilizamos uma função $\phi(\cdot)$ que faz a projeção dos pontos
  - Exemplo:
    - $\phi\colon R^2 \rightarrow R^3,\ \phi([x_1, x_2]) = [x_1{}^2, \sqrt{2}x_1x_2, x_2{}^2]$

Truque de *kernel*

- Desenvolvendo a distância entre o plano e a origem, encontramos

$$\frac{t_n y(x_n)}{\| \mathrm{w} \|} = \frac{t_n(\mathrm{w}^T \Phi(x_n) + b)}{\| \mathrm{w} \|}$$

(b é equivalente a w_0)

- E o problema de maximizar a margem se torna

$$\arg\max_{\mathrm{w},b} \left\{ \frac{1}{\| \mathrm{w} \|} \min_n \left[t_n(\mathrm{w}^T \Phi(x_n) + b) \right] \right\}$$

Truque de *kernel*

- Note que a margem será a mesma se multiplicarmos **w** e b por uma constante
  - Portanto podemos definir que as margens de separação serão tais que
    - Para pontos sobre as margens:

$$t_n(\mathbf{w}^T \Phi(x_n) + b) = 1$$

    - Para todos os demais:

$$t_n(\mathbf{w}^T \Phi(x_n) + b) \geq 1$$

Truque de *kernel*

- Agora destacamos que maximizar 1/||**w**|| equivale a minimizar ||**w**||^2, portanto as restrições sobre $t_n y(\mathbf{x}_n)$ se tornam restrições e minimizamos ||w||

$$\arg\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2$$

$$\text{t.q.}\quad t_n\left(\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n)+b\right) \geqslant 1,$$

$$\text{pour}\quad n=1,\ldots,N$$

Truque de *kernel*

- É possível remover as restrições empregando multiplicadores de Lagrange[1]

$$L(\mathbf{w}, b, \mathbf{a}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{n=1}^{N} a_n \left\{t_n(\mathbf{w}^{\mathrm{T}}\phi(\mathbf{x}_n) + b) - 1\right\}$$

$$a_n \geq 0$$

[1] C.M. Bishop. *Pattern Recognition and Machine Learning*. Apêndice E.

Truque de *kernel*

- Calculando as derivadas de L em zero com respeito a **w** e b encontramos as restrições

$$\mathbf{w} = \sum_{n=1}^{N} a_n t_n \phi(\mathbf{x}_n) \qquad 0 = \sum_{n=1}^{N} a_n t_n$$

Representação dual

- Eliminando **w** e b de L utilizando as restrições, encontramos a **representação dual** do problema de otimização do classificador de margem rígida

$$\tilde{L}(\mathbf{a}) = \sum_{n=1}^{N} a_n - 1/2 \sum_{n=1}^{N} \sum_{m=1}^{N} a_n a_m t_n t_m (x_n^{\mathsf{T}} x_m)$$

sujeito a $\sum_{n=1}^{N} a_n t_n = 0$ e $a_n \geq 0$

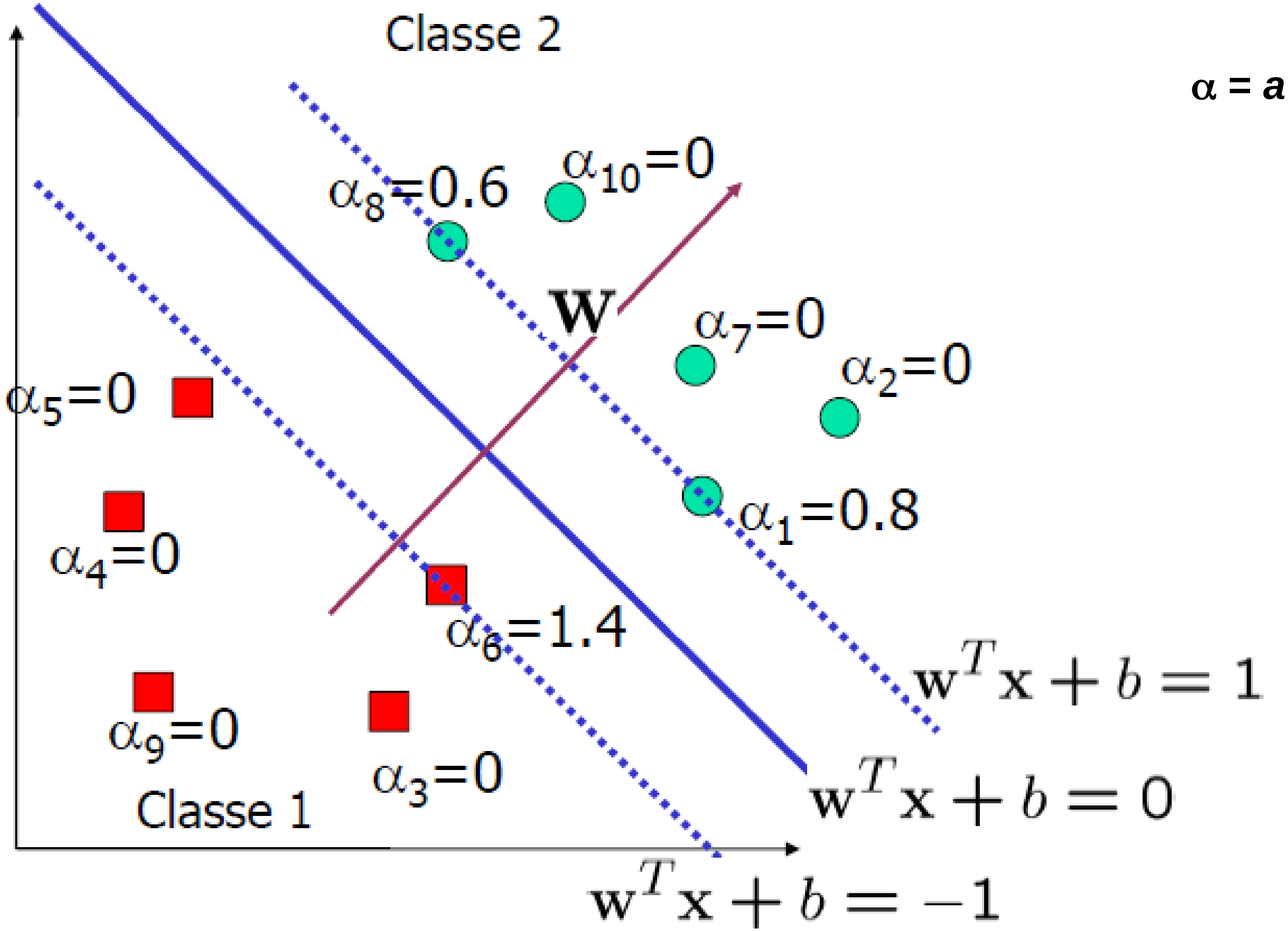
Representação dual

- Pode-se demonstrar que a solução obedece

$$\begin{aligned} a_n &\geqslant 0 \\ t_n y(\mathbf{x}_n) - 1 &\geqslant 0 \\ a_n \{t_n y(\mathbf{x}_n) - 1\} &= 0 \end{aligned}$$

- Os pontos para os quais $a_n > 0$ são chamados **vetores de suporte**

Representação dual



α = *a*

Representação dual

- Para a margem flexível, temos

$$(\max) \quad \tilde{L}(\mathrm{a}) = \sum_{n=1}^{N} a_i - 1/2 \sum_{n,m=1}^{N} a_n a_m t_n t_m (x_n^{\top} x_m)$$

$$\text{s/a} \qquad \sum_{n,m=1}^{N} a_n t_n = 0$$

$$C \geq a_n \geq 0$$

Função kernel

- Note que, em ambos os casos, o problema foi simplificado de modo que necessitamos calcular apenas o produto escalar dos pontos
- Em vez de utilizar a função de transformação, empregamos uma função de *kernel*
  - K(x, y) retorna o produto escalar de x e y em algum espaço de produto escalar

Função kernel

- Para a margem flexível, temos

$$(\max) \quad \sum_{n=1}^{N} a_i - 1/2 \sum_{n,m=1}^{N} a_n a_m t_n t_m K(x_n, x_m)$$

Função Kernel

$$\text{s/a} \quad \sum_{n,m=1}^{N} a_n t_n = 0$$

$$C \geq a_n \geq 0$$

Função kernel

- Kernel polinomial

$$K(x, x') = (x \cdot x')^d$$

Grau do polinômio

- Kernel RBF

$$k(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right)$$

Parâmetro:

$$\gamma = \frac{1}{\sigma^2}$$

SVM: recomendações

- Recomenda-se que o parâmetro C seja definido por meio de validação cruzada, começando em 10^{-6} e variando até 10^{6}
- Quanto maior o valor de σ, menor é a capacidade do modelo
  - Valores muito elevados de σ causam *underfitting*
  - Valores muito baixo tendem a *overfitting*