Universidade Federal do Paraná (UFPR)
Especialização em Engenharia Industrial 4.0

Aprendizado por Instâncias
Janelas de Parzen & Knn

David Menotti
www.inf.ufpr.br/menotti/ci171-182

Hoje

- Aprendizado por Instância
  - Janelas Parzen
  - k-NN

Aprendizado por Instâncias

- Introduzir Aprendizado por Instâncias e Métodos não paramétricos para aprendizagem supervisionada.
  - Histograma
  - Janelas de Parzen
  - kNN

Aprendizado por Instâncias

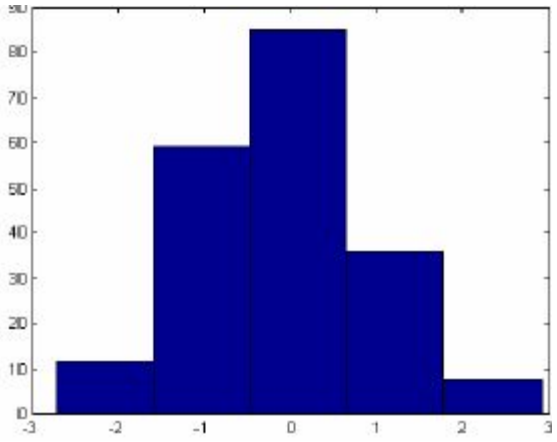
- A teoria de decisão Bayesiana assume que a distribuição do problema em questão é conhecida
  - Distribuição normal
- A grande maioria da distribuições conhecidas são unimodais.
- Em problemas reais a forma da função de densidade de probabilidade (fdp) é desconhecida
- Tudo que temos são os dados rotulados
  - Estimar a distribuição de probabilidades a partir dos dados rotulados.

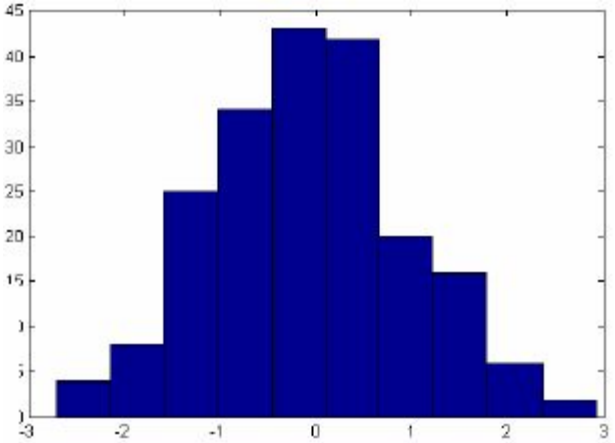
Aprendizado por Instâncias

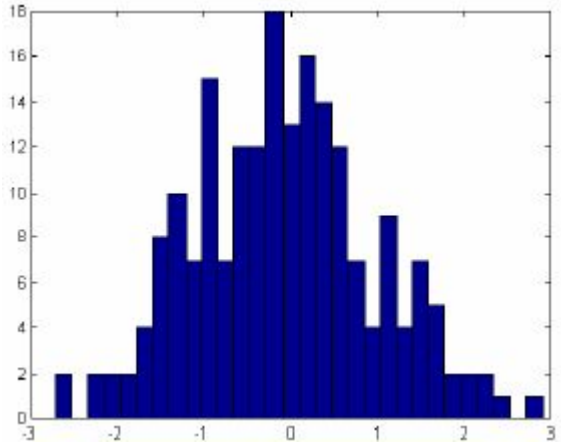
- Métodos não paramétricos podem ser usados com qualquer distribuição.
  - Histogramas
  - Janelas de Parzen
  - Vizinhos mais próximos.

Histogramas

- Método mais antigo e mais simples para estimação de densidade.
  - Depende da origem e da largura (***h***) usada para os intervalos.
  - ***h*** controla a granularidade.

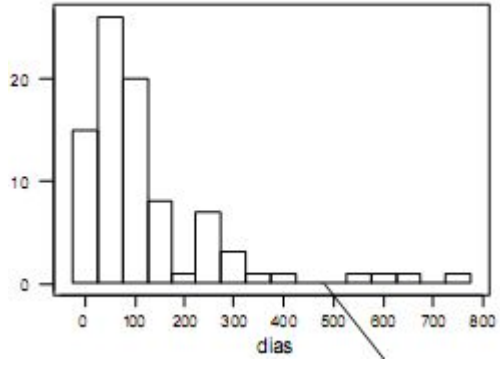






Histogramas

- Se ***h*** é largo
  - A probabilidade no intervalo é estimada com maior confiabilidade, uma vez que é baseada em um número maior de amostras.
  - Por outro lado, a densidade estimada é plana numa região muito larga e a estrutura fina da distribuição é perdida.
- Se ***h*** é estreito
  - Preserva-se a estrutura fina da distribuição, mas a confiabilidade diminui.
  - Pode haver intervalos sem amostra.



Histogramas

- Raramente usados em espaços multi-dimensionais.
  - Em uma dimensão requer **N** intervalos
  - Em duas dimensões $\mathbf{N^2}$ intervalos
  - Em **p** dimensões, $\mathbf{N^p}$ intervalos
- Necessita de grande quantidade de exemplos para gerar intervalos com boa confiabilidade.
  - Evitar descontinuidades.

Estimação de Densidade

- Histogramas nos dão uma boa ideia de como estimar densidade.

- Introduz-se o formalismo geral para estimar **densidades** p(x).

- Ou seja, a probabilidade de que um vetor ***x***, retirado de uma função de densidade desconhecida p(***x***), cairá dentro de uma região ***R*** é

$$\hat{P} = \int_R p(\mathbf{x}')d\mathbf{x}'$$

Estimação de Densidade

- Considerando que ***R*** seja contínua e pequena de forma que *p*(***x***) não varia, teremos

$$\hat{P} = \int_R p(\mathbf{x}')d\mathbf{x}' = p(\mathbf{x}) \times V$$

onde ***V*** é o volume de ***R***.

- Se retirarmos ***n*** pontos de maneira independente de *p*(***x***), então a probabilidade que ***k*** deles caiam na região ***R*** é dada pela lei **binomial**

$$P_k = \binom{n}{k} P^k (1-P)^{n-k}$$

Estimação de Densidade

- O número **médio** de pontos ***k*** caindo em ***R*** é dado pela Esperança Matemática de ***k***,

$$E[k] = n.P$$

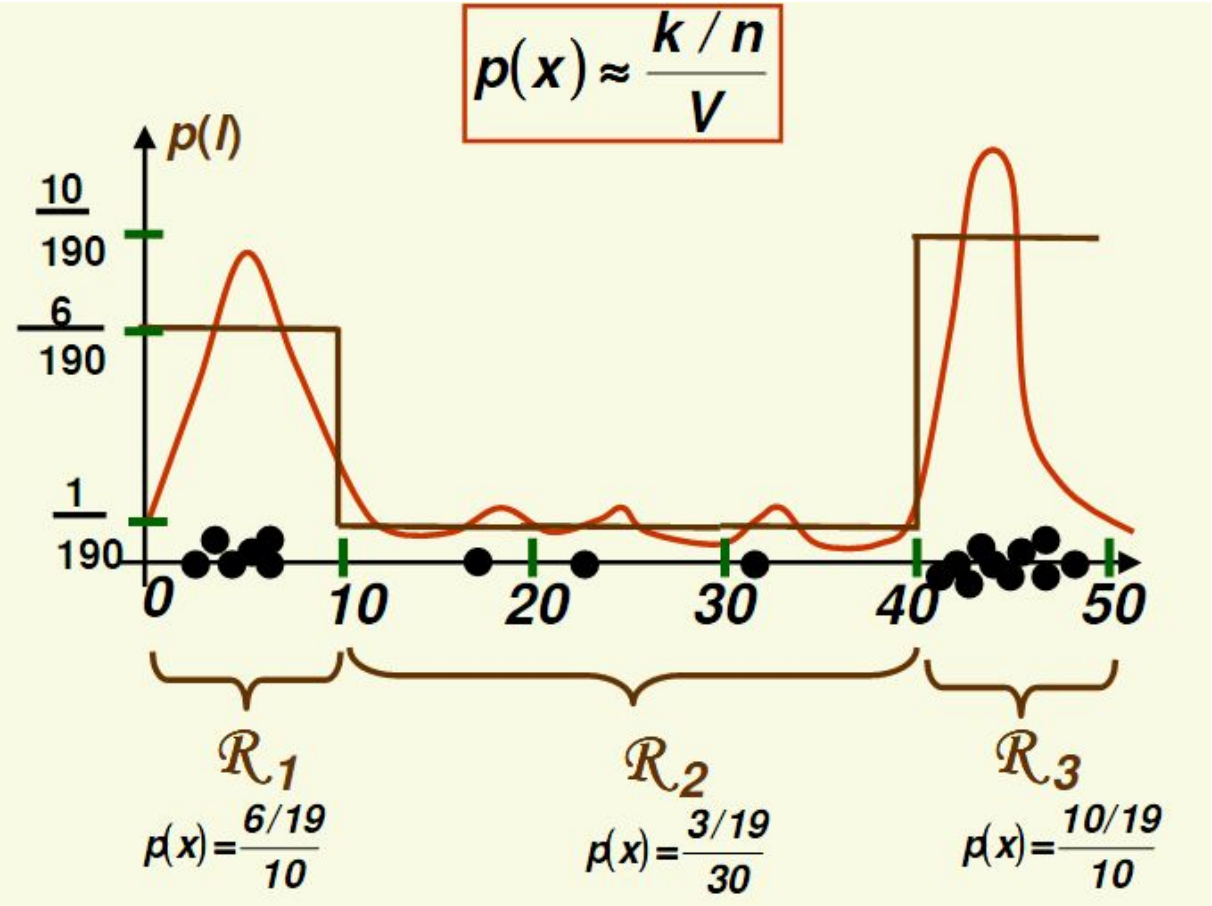
- Considerando ***n*** grande

$$\hat{P} = p(\mathbf{x}) \times V \qquad \hat{P} = \frac{k}{n} \qquad \hat{p}(\mathbf{x}) \times V = \frac{k}{n}$$

- Logo, a estimação de densidade $p(\boldsymbol{x})$ é

$$\boldsymbol{p(x) \approx \frac{k/n}{V}}$$

Estimação de Densidade



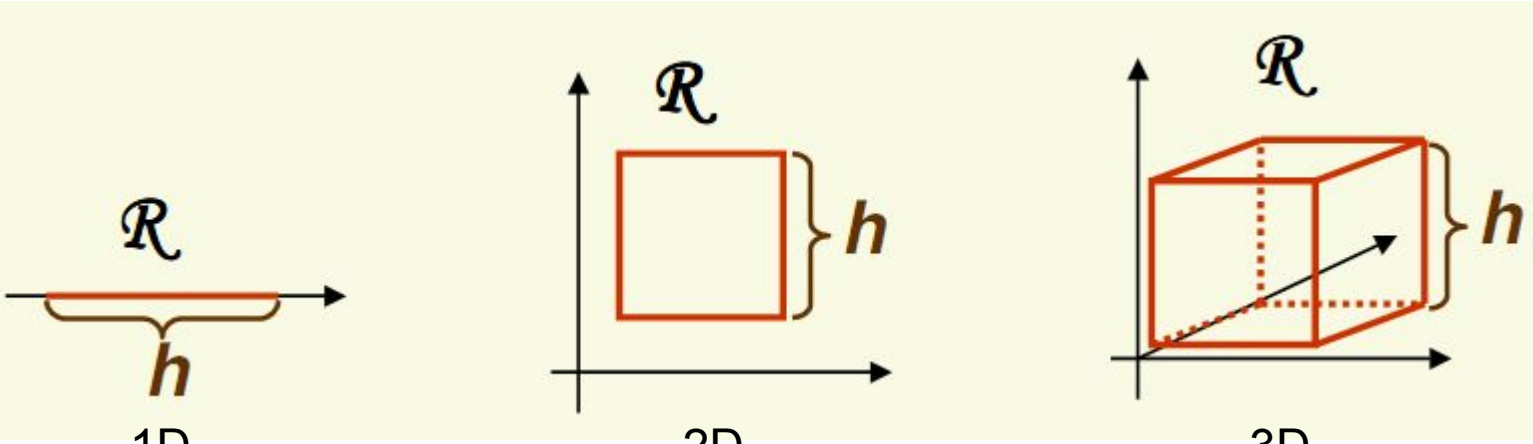
Se as regiões R_i não tem interseção, então temos um histograma.

Estimação de Densidade

- Em problemas reais, existem duas alternativas para estimação de densidade
  - Escolher um valor fixo para ***k*** e determinar o volume ***V*** a partir dos dados
    - Vizinho mais próximo (k-NN)
  - Também podemos fixar o volume ***V*** e determinar ***k*** a partir dos dados
    - Janela de Parzen

Janelas de Parzen

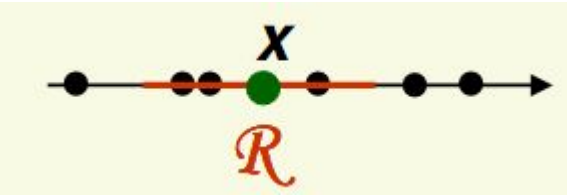
- Nessa abordagem fixamos o tamanho da região ***R*** para estimar a densidade.
- Fixamos o volume ***V*** e determinamos o correspondente ***k*** a partir dos dados de aprendizagem.
- Assumindo que a região ***R*** é um hipercubo de tamanho ***h***, seu volume é h^d



Janelas de Parzen

- Como estimar a densidade no ponto ***x***
  - Centra-se ***R*** em ***x***
  - Conta-se o número de exemplos em ***R***
  - Aplica-se em

$$p(x) \approx \frac{k/n}{V}$$

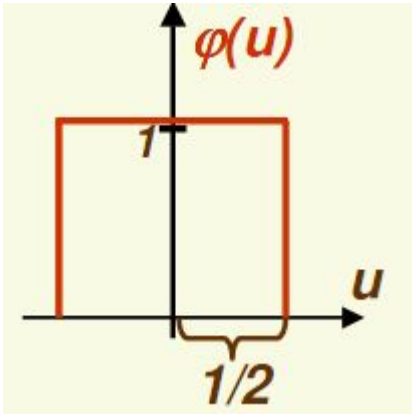


$$p(x) \approx \frac{3/6}{10}$$

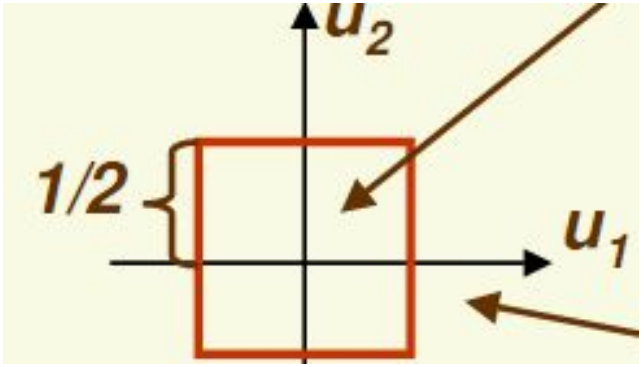
Janelas de Parzen

- Função de *Kernel* ou Parzen *window*
  - # de pontos que caem em ***R***

$$\varphi(u) = \begin{cases} 1 & |u_j| \le \frac{1}{2} \quad j = 1, \ldots, d \\ 0 & \text{otherwise} \end{cases}$$



1D



1 - Dentro

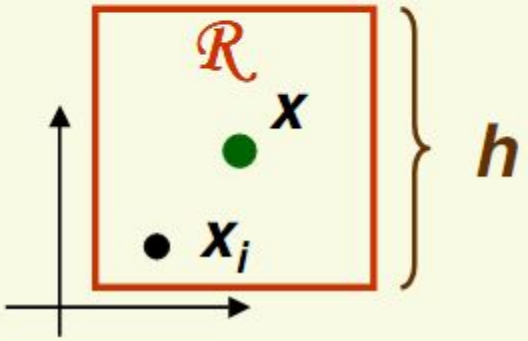
0 - Fora

2D

Janelas de Parzen

- Considerando que temos os exemplos x_1 , x_2 , … , x_n. Temos,

$$\varphi\left(\frac{x - x_i}{h}\right) = \begin{cases} 1 & |x - x_i| \le \frac{h}{2} \quad j = 1,\dots,d \\ 0 & \text{otherwise} \end{cases}$$

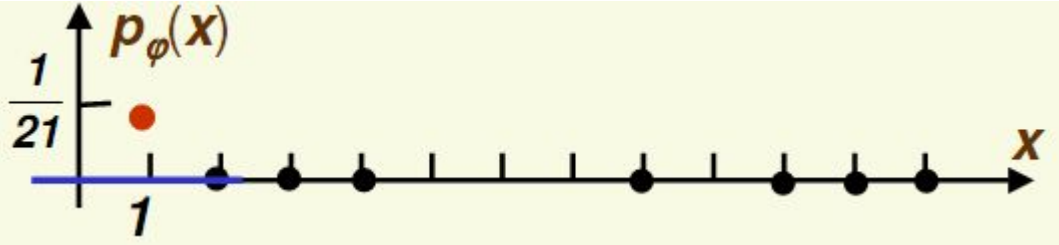


$$\varphi\left(\frac{x - x_i}{h}\right) = \begin{cases} 1 \\ 0 \end{cases}$$

Se x_i estiver dentro do hipercubo com largura h e centrado em x

Caso contrário

Janelas de Parzen: Exemplo em 1D



- Suponha que temos 7 exemplos ***D*** = {2,3,4,8,10,11,12}, e o tamanho da janela ***h*** = 3.
  - Estimar a densidade em ***x*** = 1.

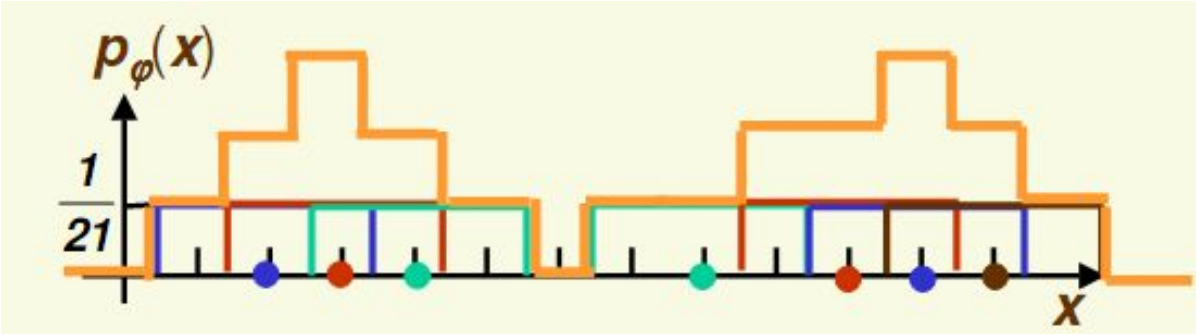
$$p_\varphi(1)=\frac{1}{7}\sum_{i=1}^{i=7}\frac{1}{3}\varphi\left(\frac{1-x_i}{3}\right)=\frac{1}{21}\left[\varphi\left(\frac{1-2}{3}\right)+\varphi\left(\frac{1-3}{3}\right)+\varphi\left(\frac{1-4}{3}\right)+\ldots+\varphi\left(\frac{1-12}{3}\right)\right]$$

$$\left|-\frac{1}{3}\right|\le 1/2 \qquad \left|-\frac{2}{3}\right|>1/2 \qquad |-1|>1/2 \qquad \left|-\frac{11}{3}\right|>1/2$$

$$p_\varphi(1)=\frac{1}{7}\sum_{i=1}^{i=7}\frac{1}{3}\varphi\left(\frac{1-x_i}{3}\right)=\frac{1}{21}[1+0+0+\ldots+0]=\frac{1}{21}$$

Janelas de Parzen: Exemplo em 1D

- Para ver o formato da função, podemos estimar todas as densidades.
- Na realidade, a janela é usada para interpolação.
  - Cada exemplo x_i contribui para o resultado da densidade em x, se x está perto bastante de x_i

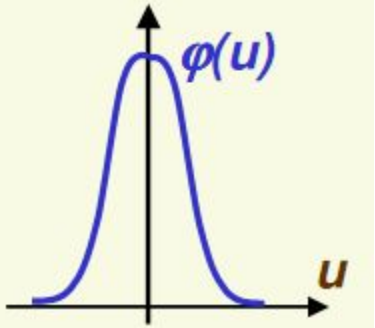


Janelas de Parzen: *Kernel* Gaussiano

- Uma alternativa a janela quadrada usada até então é
  - a janela Gaussiana.
- Nesse caso, os pontos que estão próximos a x_i recebem um peso maior
- A estimação de densidade é então suavizada.

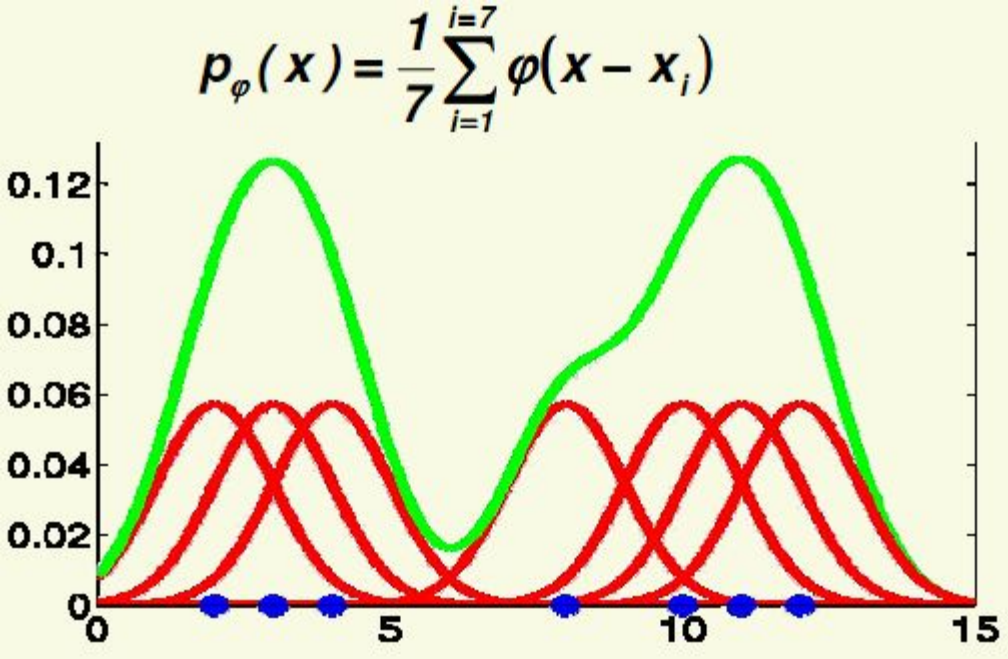
$$p_\varphi(x) = \frac{1}{n}\sum_{i=1}^{i=n}\frac{1}{h^d}\varphi\left(\frac{x - x_i}{h}\right)$$

$$\varphi(u) = \frac{1}{\sqrt{2\pi}}e^{-u^2/2}$$



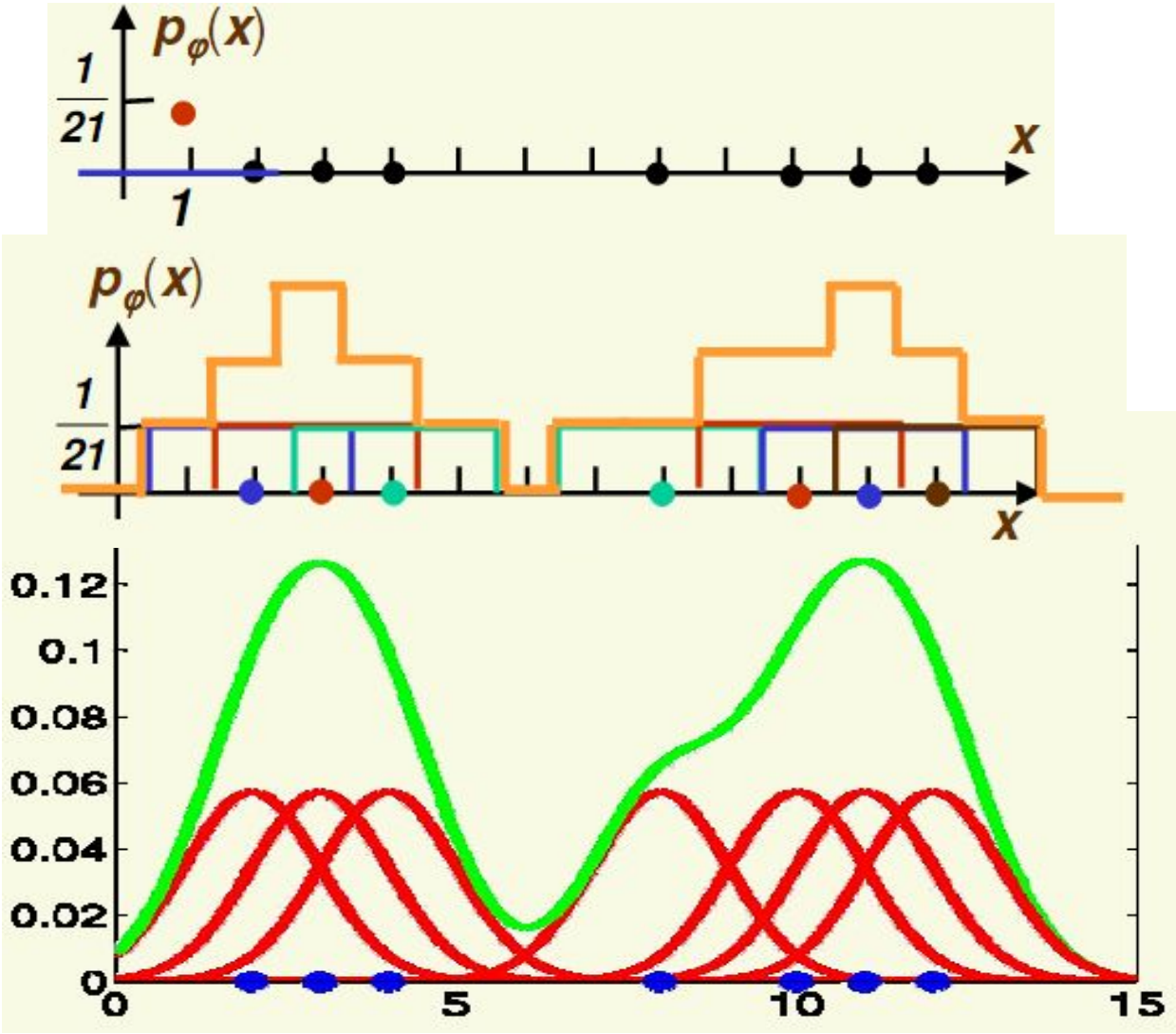
Janelas de Parzen: *Kernel* Gaussiano

- Voltando ao problema anterior
  D = {2,3,4,8,10,11,12}, para ***h*** = 1



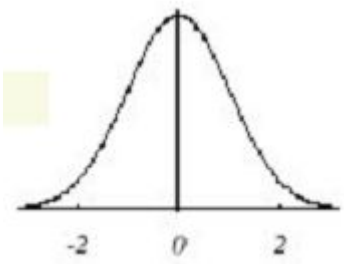
http://www.eee.metu.edu.tr/~alatan/Courses/Demo/AppletParzen.html

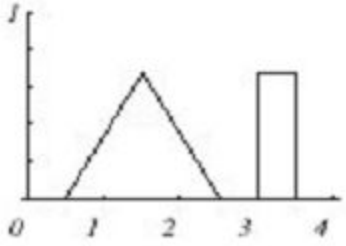
Janelas de Parzen



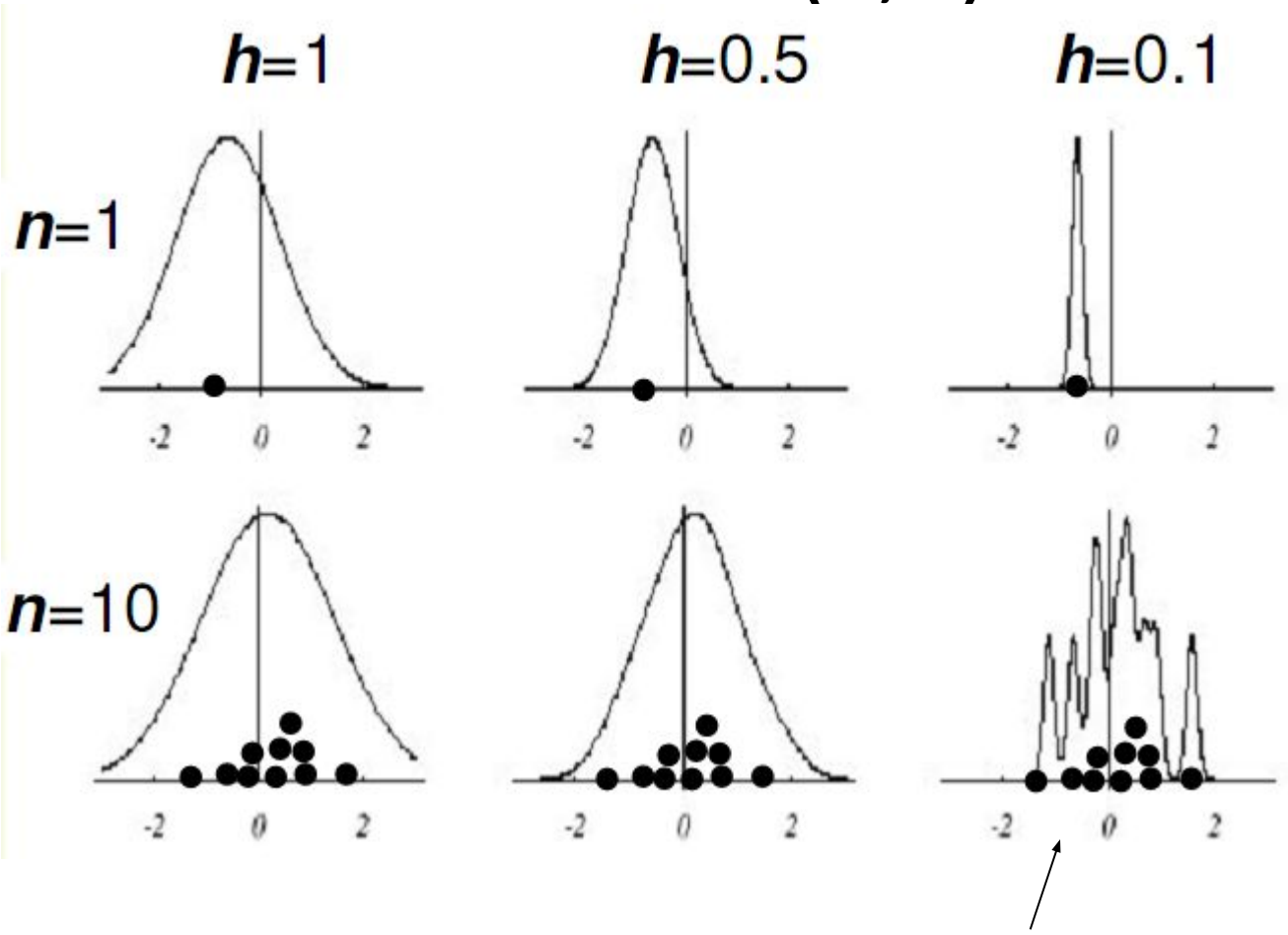
Janelas de Parzen

- Para testar esse método, vamos usar duas distribuições.
  - Normal **N**(0,1) e Mistura de Triângulo/Uniforme.
  - Usar a estimação das densidades e comparar com as verdadeiras densidades.
  - Variar a quantidade de exemplos ***n*** e o tamanho da janela ***h***





Janelas de Parzen: Normal N(0,1)



Poucos exemplos e ***h*** pequeno,
temos um fenômeno similar a um *overfitting*.

Janelas de Parzen: Normal N(0,1)

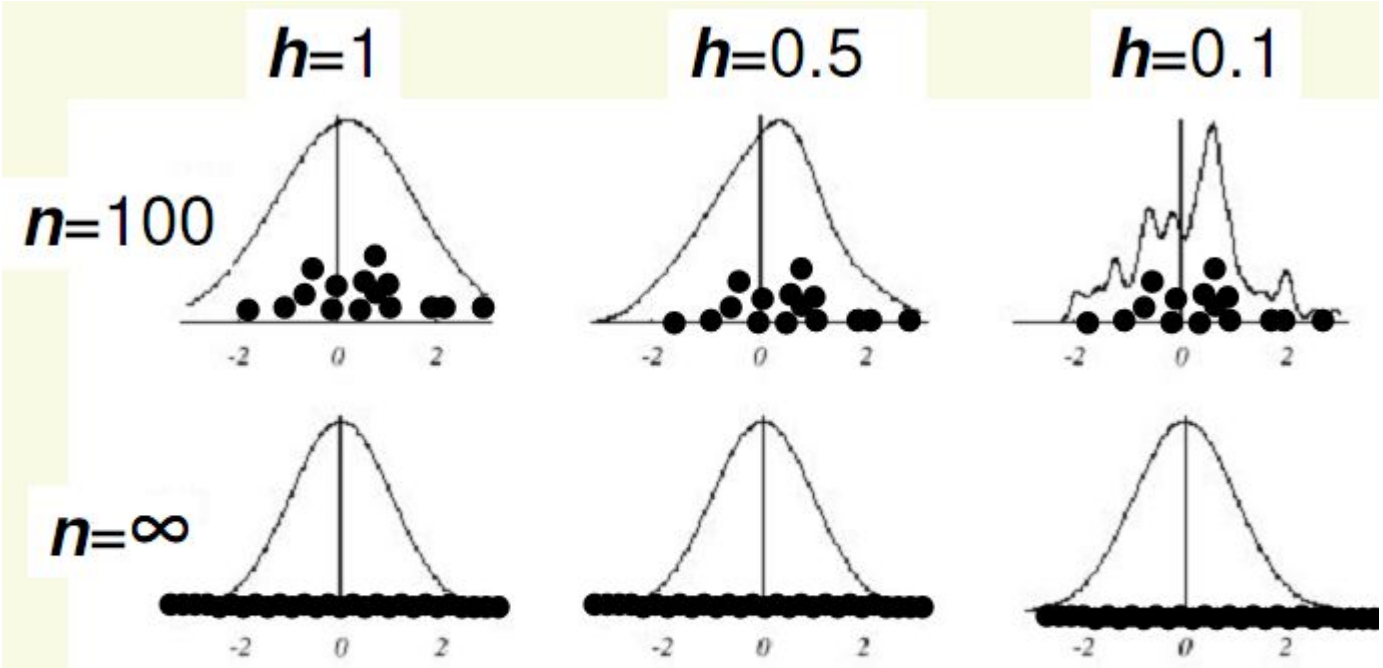
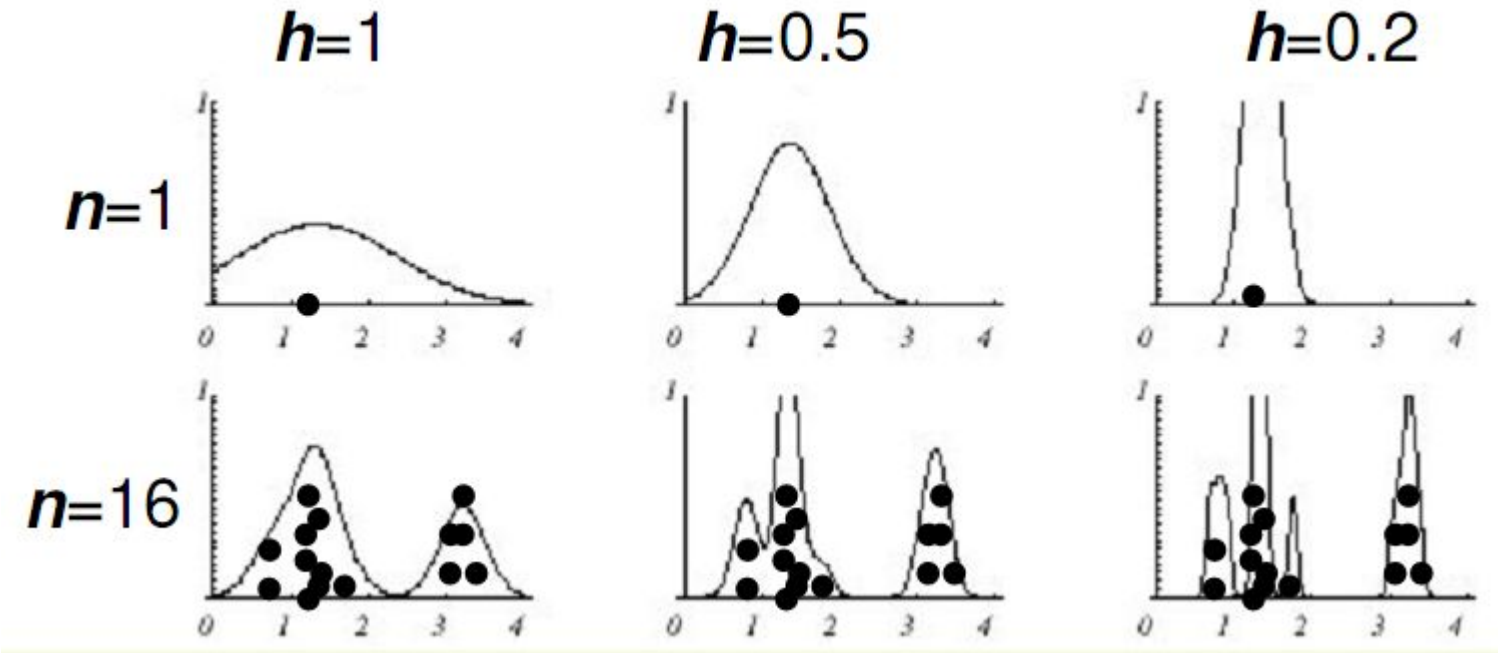


FIGURE 4.5. Parzen-window estimates of a univariate normal density using different window widths and numbers of samples. The vertical axes have been scaled to best show the structure in each graph. Note particularly that the $n = \infty$ estimates are the same (and match the true density function), regardless of window width. From: Richard

Janelas de Parzen:
Mistura de Triângulo e Uniforme



Janelas de Parzen:
Mistura de Triângulo e Uniforme

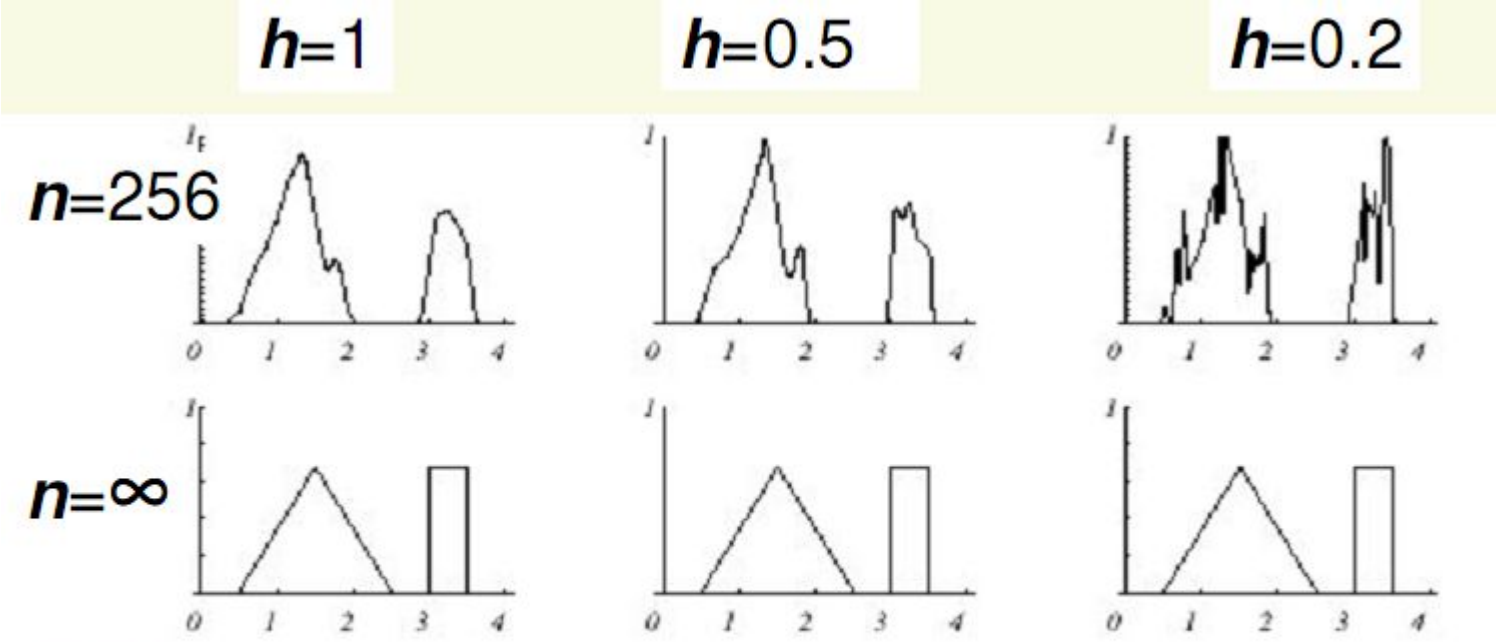
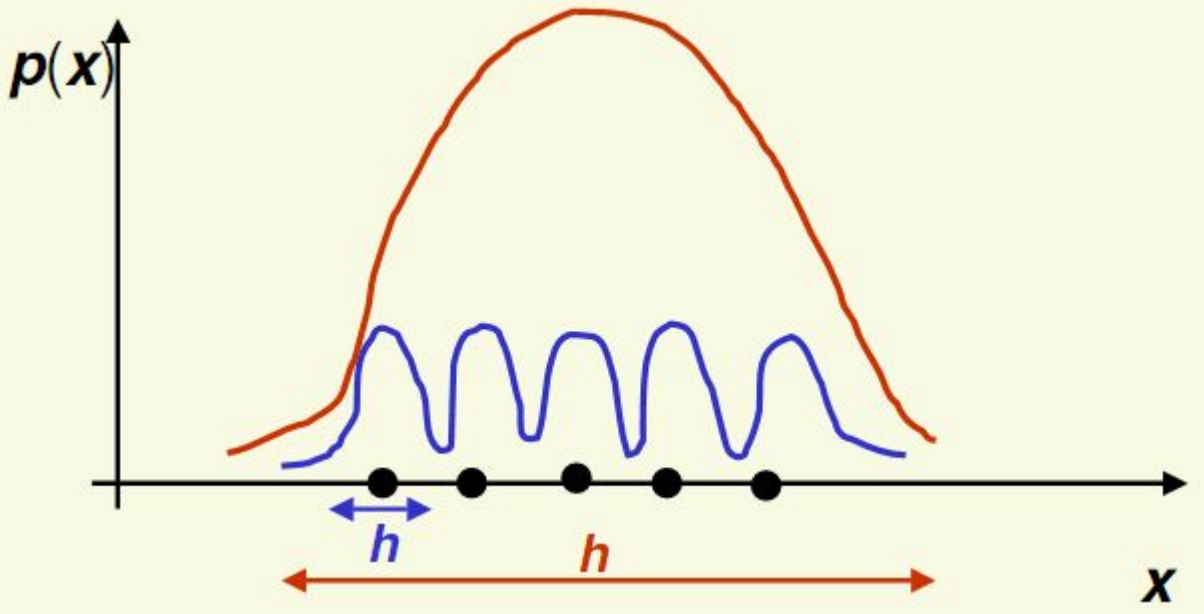


FIGURE 4.7. Parzen-window estimates of a bimodal distribution using different window widths and numbers of samples. Note particularly that the $n = \infty$ estimates are the same (and match the true distribution), regardless of window width. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Janelas de Parzen: Tamanho da Janela

- Escolhendo ***h***, “chuta-se” a região na qual a densidade é aproximadamente constante.
- Sem nenhum conhecimento da distribuição é difícil saber onde a densidade é aproximadamente constante.

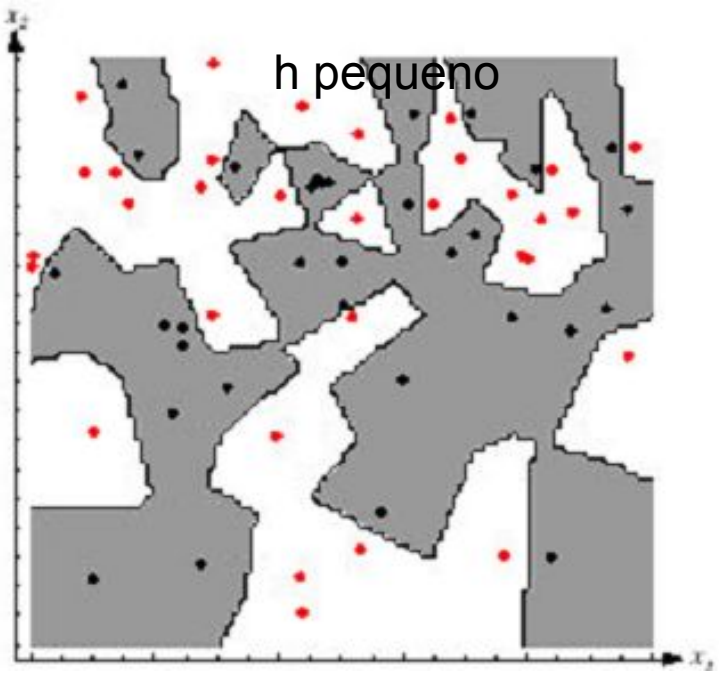


Janelas de Parzen: Tamanho da Janela

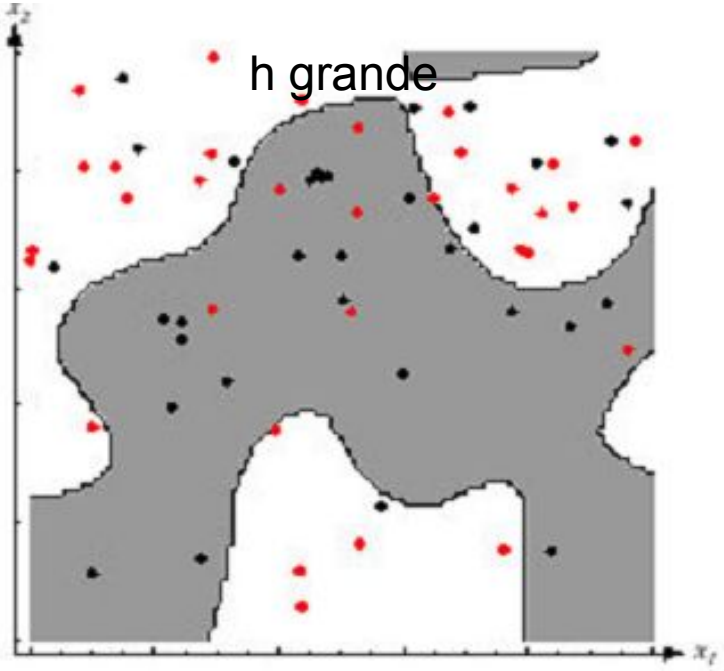
- Se ***h*** for muito pequeno
  - Fronteiras muito especializadas
- Se ***h*** for muito grande
  - Generaliza demais
- Encontrar um valor ideal para ***h*** não é uma tarefa trivial, mas pode ser estabelecido a partir de uma base de validação.
  - Aprender ***h***

Janelas de Parzen: Tamanho da Janela

Qual problema foi melhor resolvido?



h pequeno: Classificação perfeita
Um caso de ***overfitting***



h maior: Melhor generalização

Regra de classificação:
Calcula-se $P(x/c_j)$, j = 1,...,m e associa x a classe onde P é máxima

Os k-Vizinhos Mais Próximos
k-Nearest Neighboor

Vizinho mais Próximo (kNN)

- Relembrando a expressão genérica para estimação da densidade

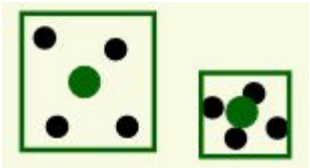
$$p(x) \approx \frac{k/n}{V}$$

- Na Janela de Parzen, fixamos o ***V*** e determinamos ***k*** (número de pontos dentro de ***V***)

- No kNN, fixamos ***k*** e encontramos ***V*** que contém os ***k*** pontos.

kNN

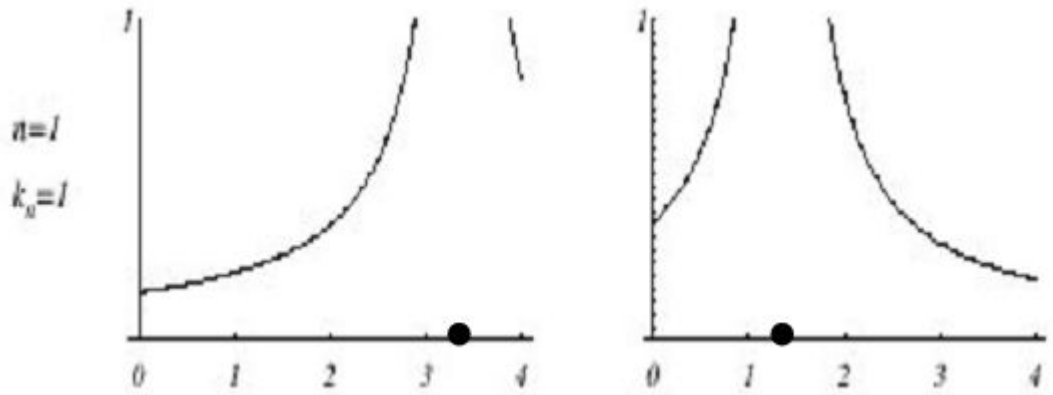
- Um alternativa interessante para o problema da definição da janela ***h***.
  - Nesse caso, o volume é estimado em função dos dados
    - Coloca-se a célula sobre **x**.
    - Cresce até que ***k*** elementos estejam dentro dela.



kNN

- Qual seria o valor de ***k*** ?
  - Uma regra geral seria $k = \sqrt{n}$
  - Não muito usada na prática.

- Porém, kNN não funciona como uma estimador de densidade, a não ser que tenhamos um número infinito de exemplos
  - O que não acontece em casos práticos.



Funções descontínuas

kNN

- Entretanto, podemos usar o kNN para estimar diretamente a probabilidade a *posteriori* P(c_i | ***x***)

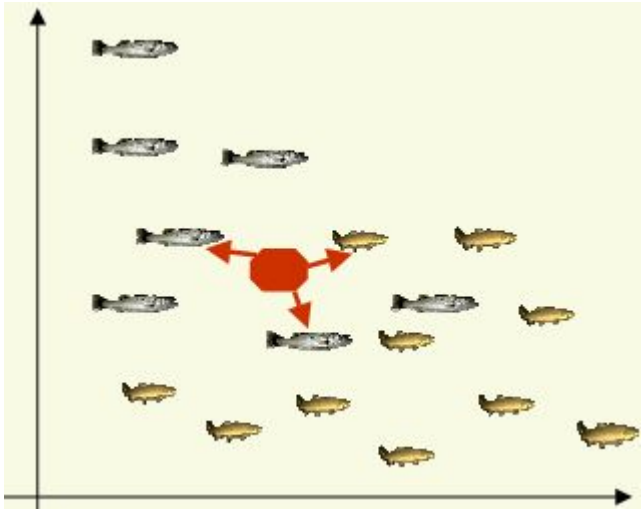
- Sendo assim, não precisamos estimar a densidade p(***x***).

$$p(c_i \mid x) = \frac{p(x,c_i)}{p(x)} = \frac{p(x,c_i)}{\sum_{j=1}^{m} p(x,c_j)} \approx \frac{k_i / n}{V \sum_{j=1}^{m} \frac{k_j / n}{V}} = \frac{k_i}{\sum_{j=1}^{m} k_j} = \frac{k_i}{k}$$

Ou seja, p(c_i|x) e a fração de exemplos que pertencem a classe c_i

kNN

- A interpretação para o kNN seria
  - Para um exemplo não rotulado x, encontre os k mais similares a ele na base rotulada e atribua a classe mais frequente para x.
- Voltando ao exemplo dos peixes



Para k = 3, teriamos 2 robalos e 1 salmão.
Logo, classifica-se ***x*** como robalo.

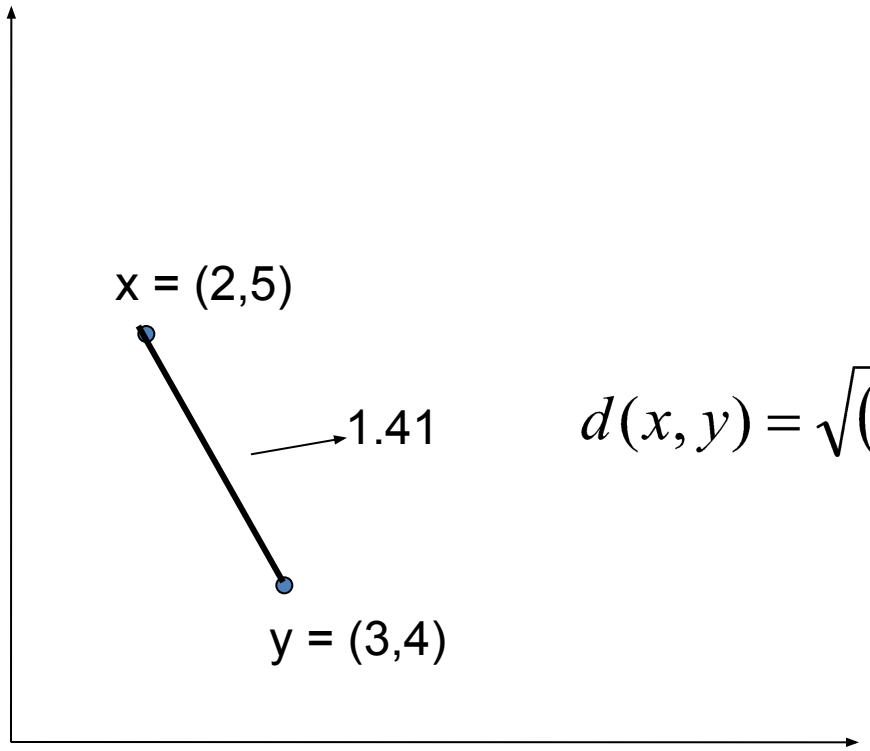
kNN

- Significado de ***k***:
  - Classificar ***x*** atribuindo a ele o rótulo representado mais frequentemente dentre as ***k*** amostras mais próximas.
  - Contagem de votos.

- Uma medida de proximidade bastante utilizada é a distância Euclidiana:

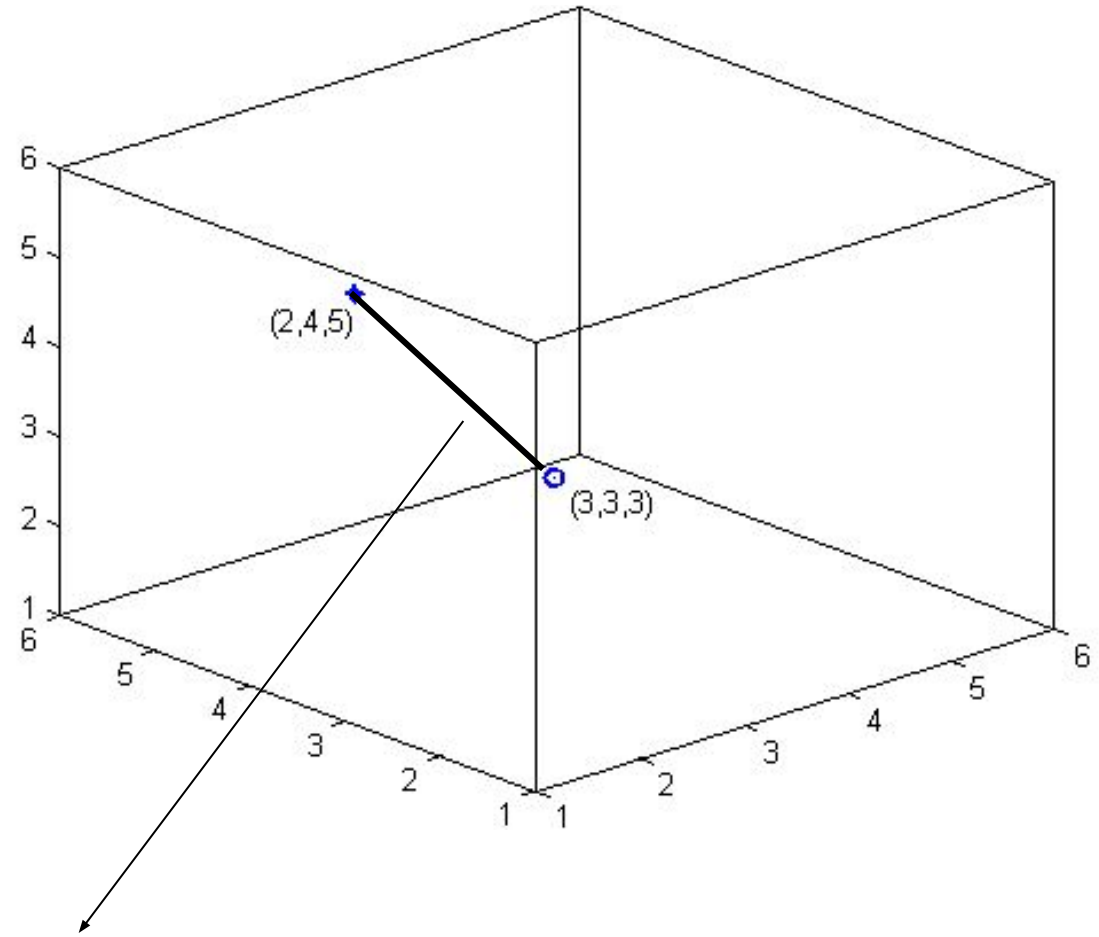
$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Distância Euclidiana



$$d(x,y) = \sqrt{(2-3)^2 + (5-4)^2} = \sqrt{2} = 1.41$$

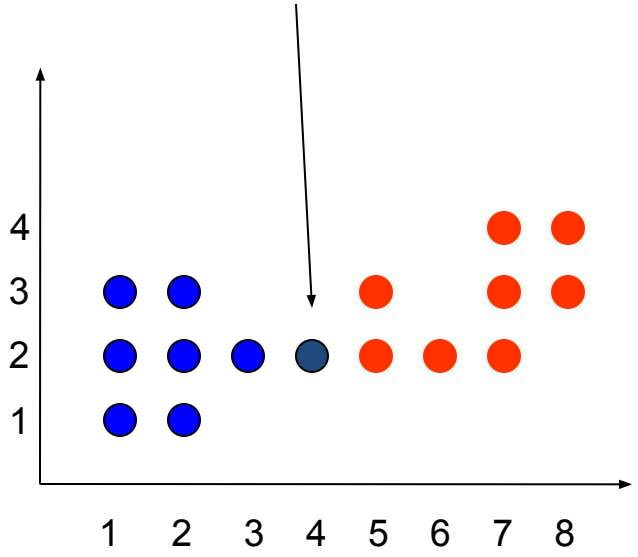
Distância Euclidiana



$$d(x,y) = \sqrt{(2-3)^2 + (4-3)^2 + (5-3)^2} = \sqrt{6} = 2.44$$

k-NN: Um Exemplo

A qual classe pertence este ponto?
Azul ou vermelho?



Calcule para os seguintes valores de k:

k=1 não se pode afirmar

k=3 vermelho – 5,2 - 5,3

k=5 vermelho – 5,2 - 5,3 - 6,2

k=7 azul – 3,2 - 2,3 - 2,2 - 2,1

A classificação pode mudar de acordo com a escolha de *k*.

Matriz de Confusão

- Matriz que permite visualizar as principais confusões do sistema.
- Considere um sistema com 3 classes, 100 exemplos por classe.

100% de classificação

	c1	c2	c3
c1	100		
c2		100	
c3			100

Erros de classificação

	c1	c2	c3
c1	90	10	
c2		100	
c3	5		95

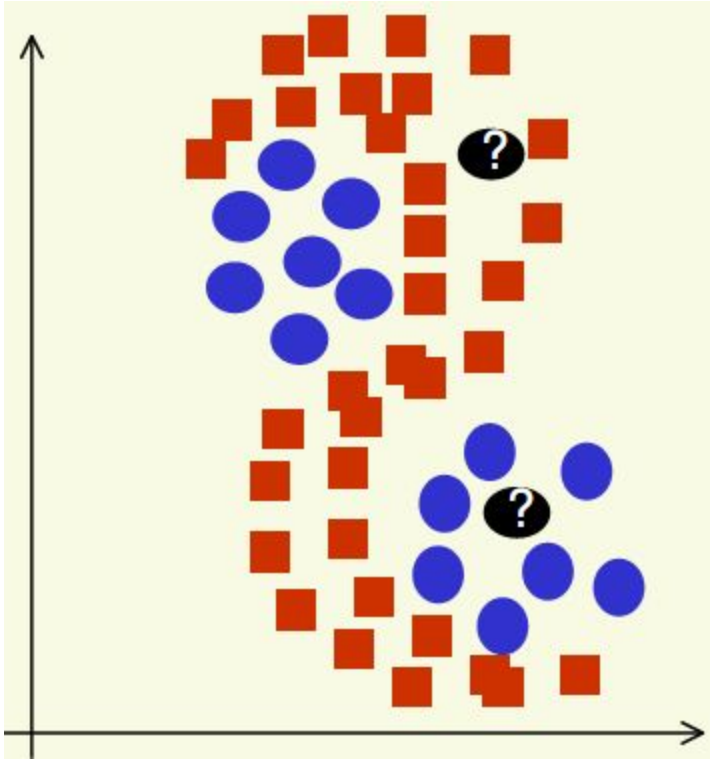
10 exemplos de C1 foram classificados como C2

kNN: Funciona bem?

- Certamente o kNN é uma regra simples e intuitiva.
- Considerando que temos um número ilimitado de exemplos
  - O melhor que podemos obter é o erro Bayesiano (E*)
  - Para ***n*** tendendo ao infinito, pode-se demonstrar que o erro do kNN é menor que 2E*
- Ou seja, se tivermos bastante exemplos, o kNN vai funcionar bem.

kNN: Distribuições Multi-Modais

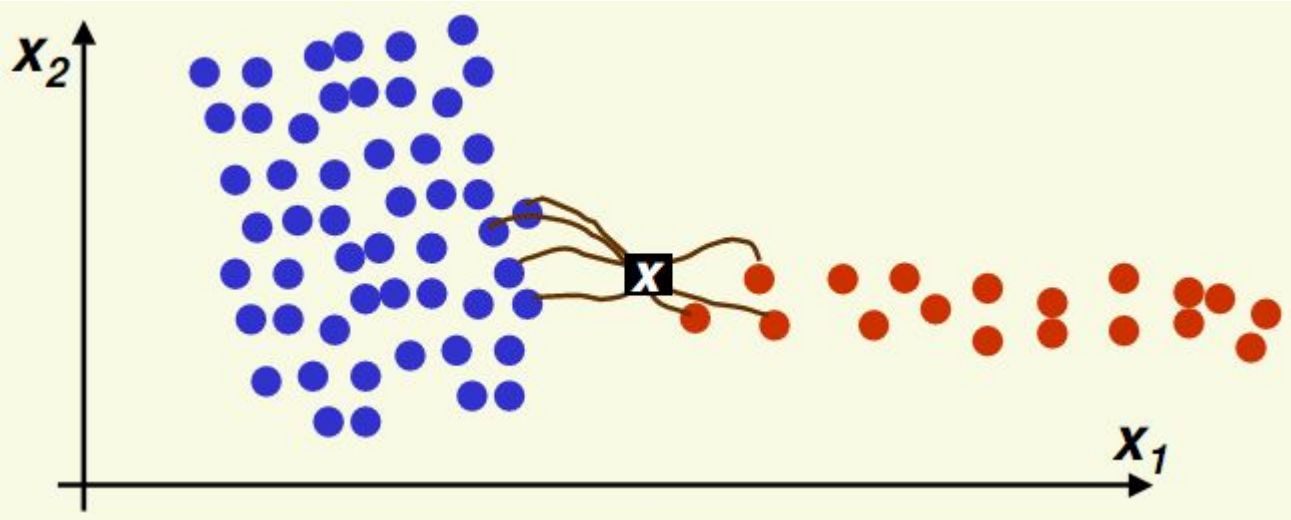
- Um caso complexo de classificação no qual o kNN tem sucesso.



kNN: Como escolher *k*

- Não é um problema trivial.
  - ***k*** deve ser grande para minimizar o erro.
    - ***k*** muito pequeno leva a fronteiras ruidosas.
  - ***k*** deve ser pequeno para que somente exemplos próximos sejam incluídos.
- Encontrar o balanço não é uma coisa trivial.
  - Base de validação

kNN: Como escolher k



- Para ***k*** = 1,...,7
  - o ponto ***x*** é corretamente classificado (**vermelho**)
- Para ***k*** > 7,
  - a classificação passa para a classe azul (**erro**)

kNN: Complexidade

- O algoritmo básico do kNN armazena todos os exemplos. Suponha que tenhamos ***n*** exemplos
  - O(***n***) é a complexidade para encontrar o vizinho mais próximo.
  - O(***nk***) complexidade para encontrar ***k*** exemplos mais próximos
- Considerando que precisamos de um ***n*** grande para o kNN funcionar bem, a complexidade torna-se problema.

kNN: Diagrama de Voronoi

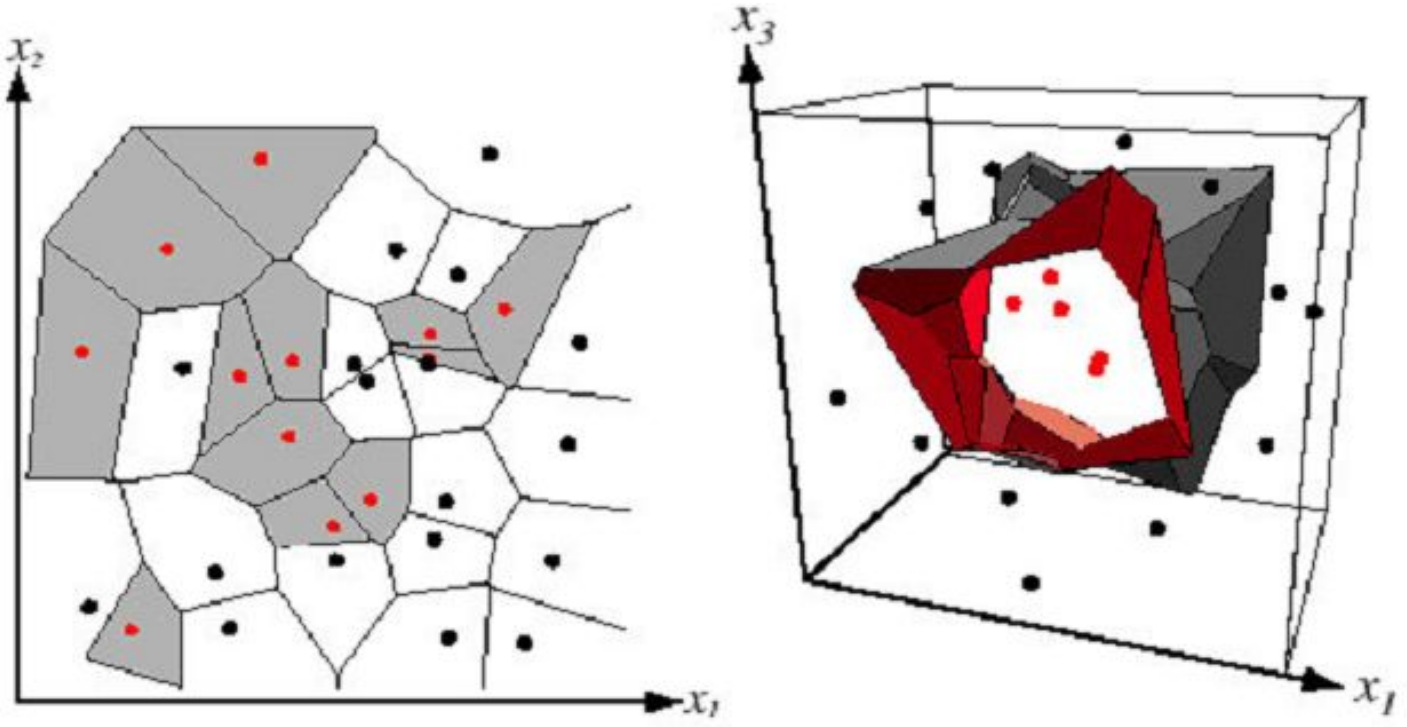
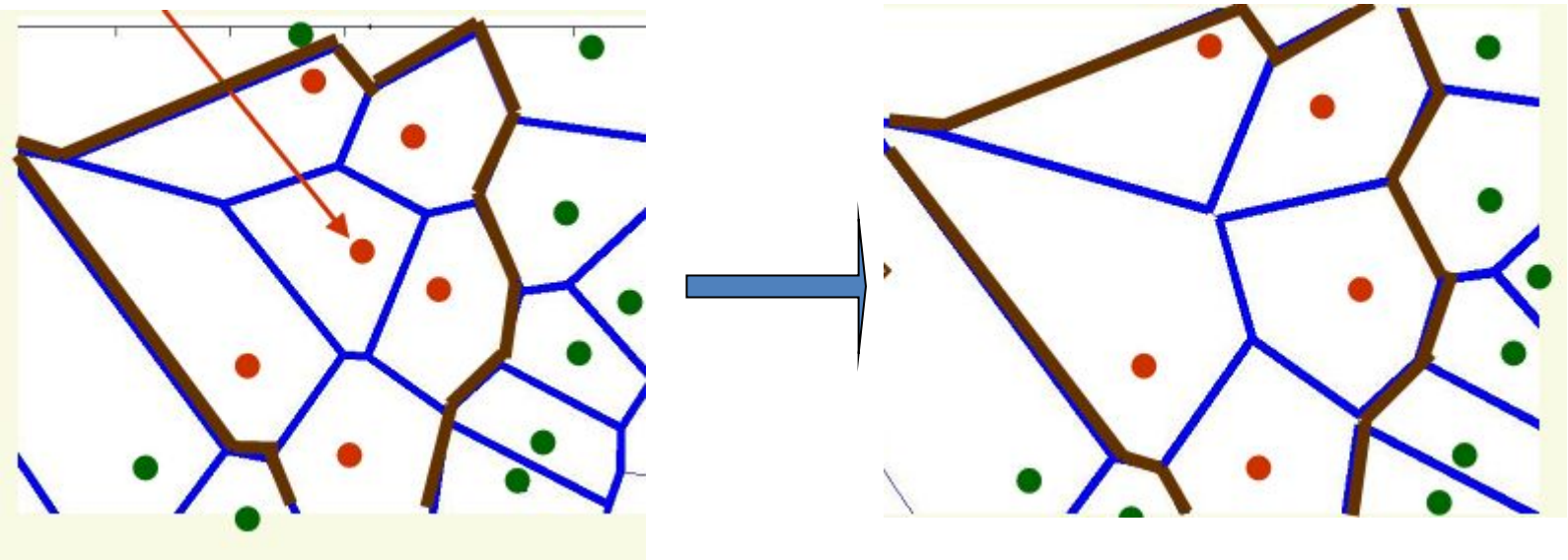


FIGURE 4.13. In two dimensions, the nearest-neighbor algorithm leads to a partitioning of the input space into Voronoi cells, each labeled by the category of the training point it contains. In three dimensions, the cells are three-dimensional, and the decision boundary resembles the surface of a crystal. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

kNN: Reduzindo complexidade

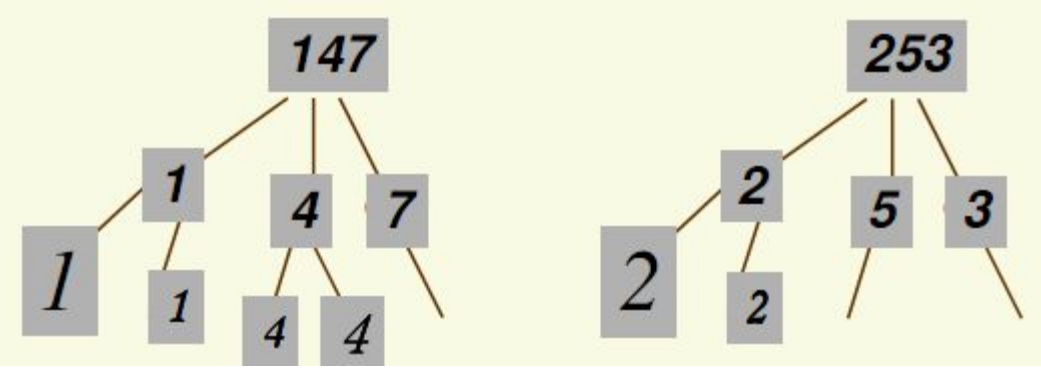
- Se uma **célula** dentro do diagrama de Voronoi possui os mesmos vizinhos, ela pode ser removida.



Mantemos a mesma fronteira e diminuímos a quantidade de exemplos

kNN: Reduzindo complexidade

- kNN protótipos
  - Consiste em construir protótipos (centróides) para representar a base
  - Diminui a complexidade, mas não garante as mesmas fronteiras



kNN: Distância

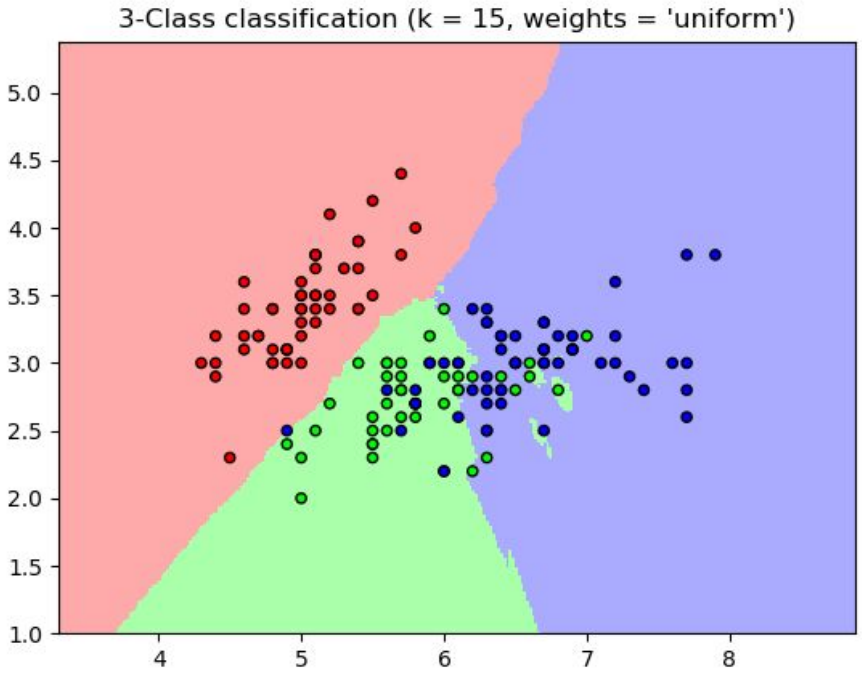
- Ponderar a contribuição de cada um dos ***k*** vizinhos de acordo com suas distâncias até o ponto x_t que queremos classificar, dando maior peso aos vizinhos mais próximos.
- Podemos ponderar o voto de cada vizinho, de acordo com o quadrado do inverso de sua distância de x_t.

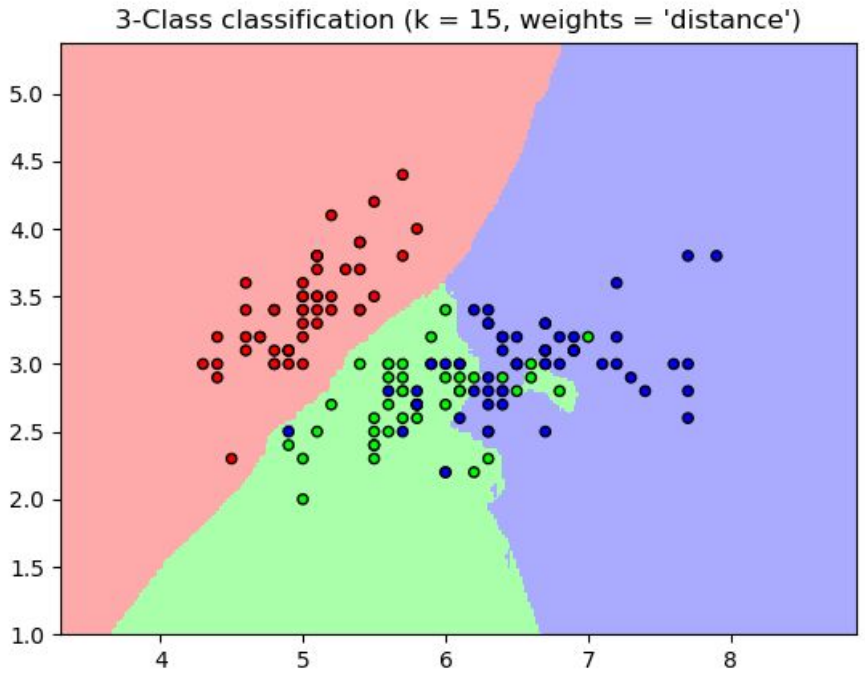
$$f(x_t) = \underset{c \in C}{argmax} \sum_i \omega_i \delta(c, f(x_i)) \qquad \omega_i = \frac{1}{d(x_t, x_i)^2}$$

- Porém, se $x_t = x_t$, o denominador $d(x_t, x_i)^2$ torna–se zero. Neste caso fazemos $f(x_t) = f(x_i)$.

kNN: Distância

- A distância é capaz de modelar fronteiras mais suaves





http://scikit-learn.org/stable/auto_examples/neighbors/plot_classification.html#sphx-glr-auto-examples-neighbors-plot-classification-py

kNN: Seleção da Distância

- Até então assumimos a distância Euclidiana para encontrar o vizinho mais próximo.

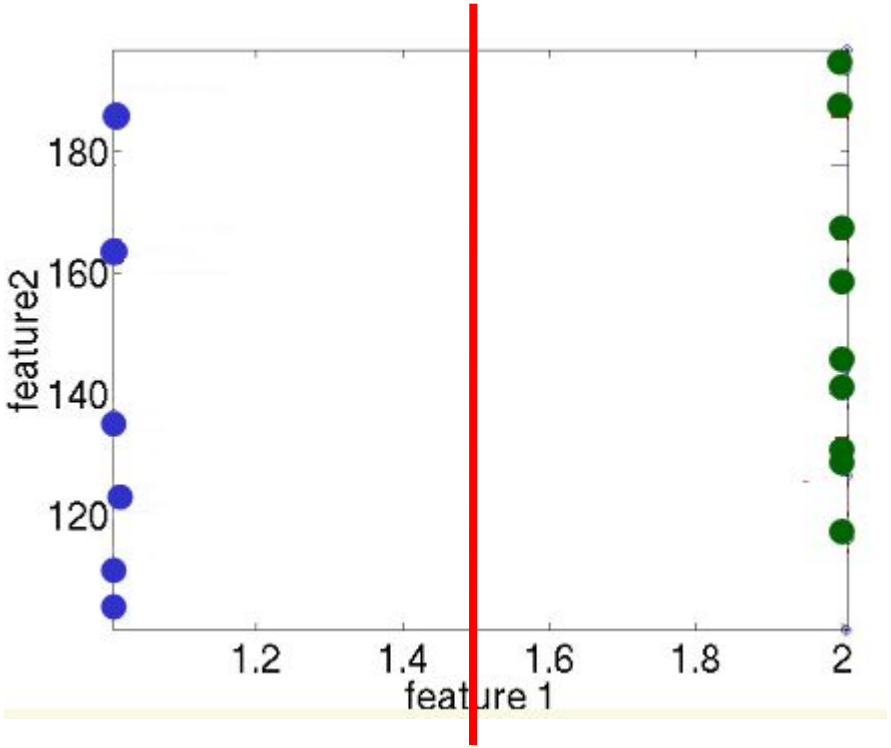
$$D(a,b) = \sqrt{\sum_k (a_k - b_k)^2}$$

- Entretanto algumas características (dimensões) podem ser mais discriminantes que outras.

- Distância Euclidiana dá a mesma importância a todas as características

kNN: Seleção da Distância

- Considere as seguintes características
  - Qual delas discrimina a classe verde da azul?



kNN: Seleção da Distância

- Agora considere que um exemplo Y = [1, 100] deva ser classificado.
- Considere que tenhamos dois vizinhos X_1 = [1,150] e X_2 = [2,110]

$$D\left(\begin{bmatrix}1\\100\end{bmatrix},\begin{bmatrix}1\\150\end{bmatrix}\right)=\sqrt{(1-1)^2+(100-150)^2}=50 \qquad D\left(\begin{bmatrix}1\\100\end{bmatrix},\begin{bmatrix}2\\110\end{bmatrix}\right)=\sqrt{(1-2)^2+(100-110)^2}=10.5$$

- Y não será classificado corretamente.

kNN: Normalização

- Note que as duas características estão em escalas diferentes.
  - Característica 1 varia entre 1 e 2
  - Característica 2 varia entre 100 e 200
- Uma forma de resolver esse tipo de problema é a **normalização**.
- A forma mais simples de normalização consiste em dividir cada característica pelo somatório de todas as características

kNN:Normalização

	Antes da Normalização		Após a Normalização	
	$Feat_1$	$Feat_2$	$Feat_1$	$Feat_2$
A	1	100	0,0099	0,9900
B	1	150	0,00662	0,9933
C	2	110	0,0178	0,9821

Distâncias

A - B = 0,0046
A - C = 0,01125

kNN: Normalização

- Outra maneira eficiente de normalizar consiste em deixar cada característica centrada na média 0 e desvio padrão 1.

- Se X é uma variável aleatória com média μ e desvio padrão σ,

$$X' = (X - \mu)/ \sigma$$

tem média 0 e desvio padrão 1.

kNN: Seleção da Distância

- Entretanto, em altas dimensões, se existirem várias características irrelevantes, a normalização não irá ajudar.

$$D(a,b) = \sqrt{\sum_k (a_k - b_k)^2} = \sqrt{\sum_i (a_i - b_i)^2 + \sum_j (a_j - b_j)^2}$$

Discriminante Ruídos

- Se o número de características discriminantes for menor do que as características irrelevantes, a distância Euclidiana será dominada pelos ruídos.

Referências

- x