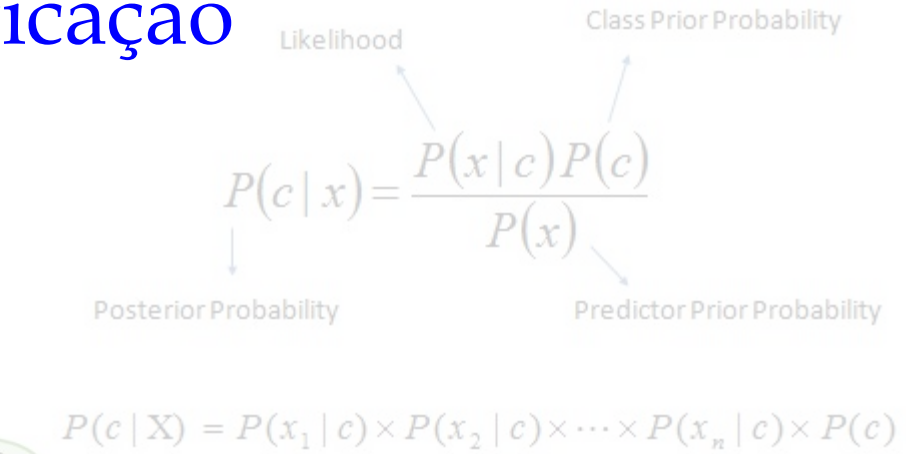
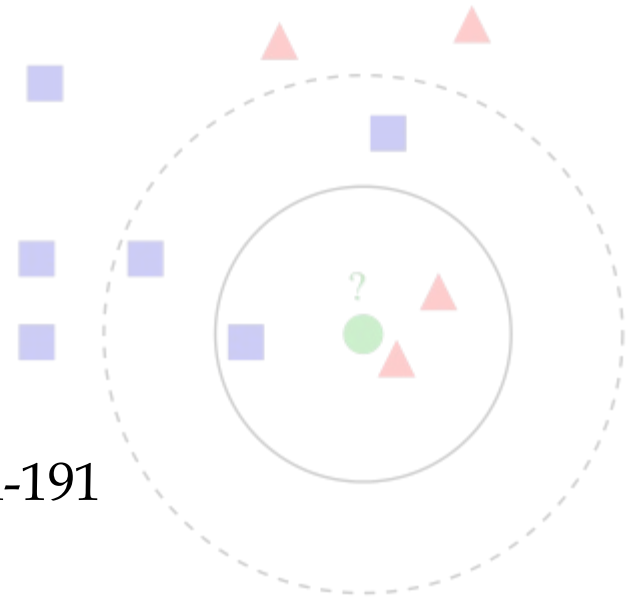
Universidade Federal do Paraná (UFPR)
Especialização em Engenharia Industrial 4.0

Classificação





David Menotti
www.inf.ufpr.br/menotti/am-191

Hoje

- Árvores de Decisão
- Aprendizagem Bayesiana
- Aprendizado por Instância
  - Janelas Parzen
  - k-NN

Árvores de Decisão

Árvores de Decisão

Agenda

- Introdução
- Representação
- Quando Usar
- Algoritmo de Aprendizagem
- Resumo

Árvores de Decisão

- Método prático
- Um dos mais utilizados na aprendizagem indutiva.
- Diferentemente dos métodos de aprendizagem conceitual, são mais robustas a ruídos nos dados
- Aproxima funções alvo de valor discreto, em que a função aprendida é representada por uma árvore de decisão.
- Também pode ser representada por um conjunto de regras (IF-THEN)
- *White model*
  - Diferentemente de uma rede neural que é muitas vezes referenciada como um *Black-box*

Árvores de Decisão

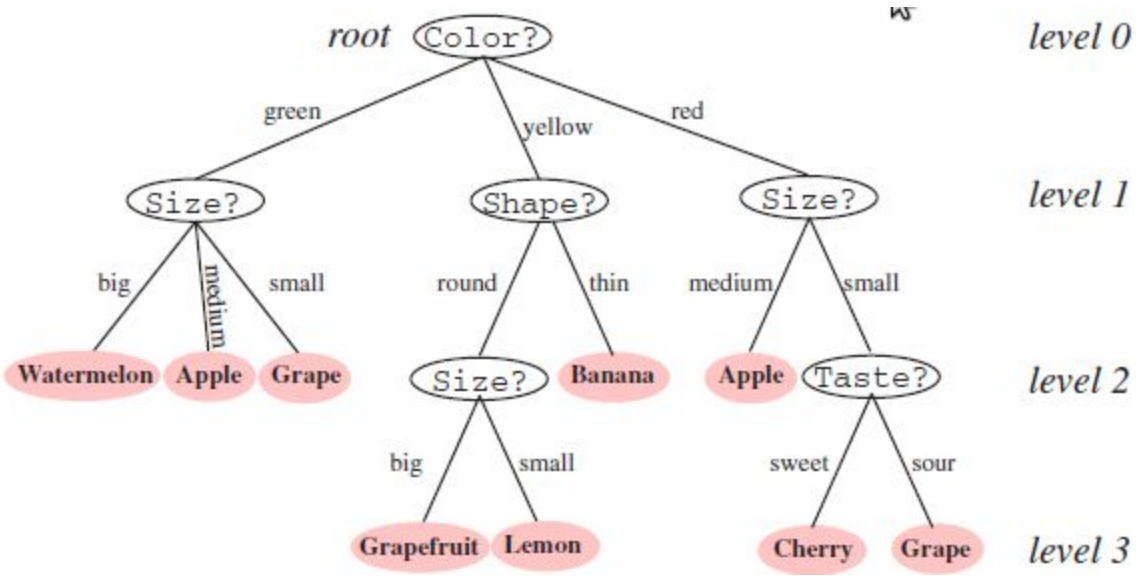
- Um dos métodos de aprendizagem mais conhecidos.
- Não necessita de manipulação de dados, como por exemplo, métodos de normalização
- Pode receber tanto dados numéricos quanto simbólicos.
- Aplicações diversas como auxílio a diagnóstico, análise de risco, etc.

Árvores de Decisão

Entretanto

- Alguns conceitos são de difícil aprendizagem em árvores de decisão, gerando árvores extremamente grandes, por exemplo, XOR.
- Aprendizagem de uma árvore ótima é conhecida como **NP Completo**.
- Utiliza heurísticas.
- Pode não gerar a melhor árvore.

Representação



- Árvores de decisão classificam instâncias ordenando-as sub-árvores acima (ou abaixo), a partir da raiz até alguma folha.
- Cada nó da árvore especifica o teste de algum atributo da instância.
- Cada ramo partindo de um nó corresponde a um dos valores possíveis dos atributos

Representação

- Uma instância é classificada inicialmente pelo nó raiz, testando o atributo especificado por este nó
- Em seguida, movendo-se através do ramo correspondendo ao valor do atributo no exemplo dado
- Este processo é repetido para a sub-árvore originada no novo nó

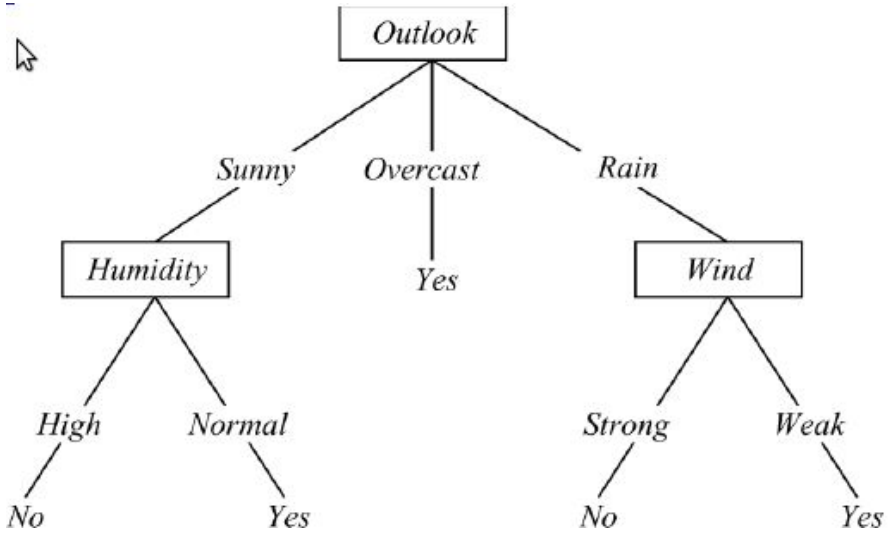
Representação

- Base de dados para o problema/conceito “Play Tennis”

Day	*Outlook*	*Temperature*	*Humidity*	*Wind*	*PlayTennis*
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Representação

Uma árvore de decisão para o conceito *Play Tennis*



- Um exemplo é classificado ordenado-o através da árvore para o nó da folha apropriado.
- Então retorna a classificação associada com esta folha (*Yes* / *No*)

Representação

- Em geral, as árvores de decisão representam uma disjunção de restrições sobre valores dos atributos das instâncias
- Cada caminho entre a raiz da árvore e uma folha corresponde a uma **conjunção** (**E**) de testes de atributos e a própria árvore corresponde a uma **disjunção** (**OU**) destas conjunções.

Exemplo:

```
    ( Outlook = Sunny      E  Humidity = Normal )
OU  ( Outlook = Overcast )
OU  ( Outlook = Rain       E  Wind     = Weak   )
```

Quando Considerar Árvores de Decisão

- Instâncias descritas por pares atributo-valor.
- Instâncias descritas por um conjunto **fixo** de atributos, por exemplo, `Temperatura` com valores definidos (`Quente`, `Frio`).
- **Classe** tem valores discretos de saída, por exemplo, `Sim` e `Não`.
- Dados de treinamento podem conter erros e valores de atributos faltantes.

Casos de aplicações:

- Diagnóstico ou equipamentos médicos
- Análise de Risco
- Modelagem de preferências em agendamento

Algoritmo Básico

- A maioria dos algoritmos de aprendizagem de árvores derivam do algoritmo ID3.
  - C4.5, C5.0 e **J4.8** são mais recentes
  - O ID3 aprende a árvore usando uma estratégia *top-down*

- **Questão inicial?**
  - Qual atributo deve ser testado na raiz da árvore?

- Para cada atributo A da base de dados
  - Avalie A como classifica *Train set*
    - Como avaliar?

Algoritmo Básico

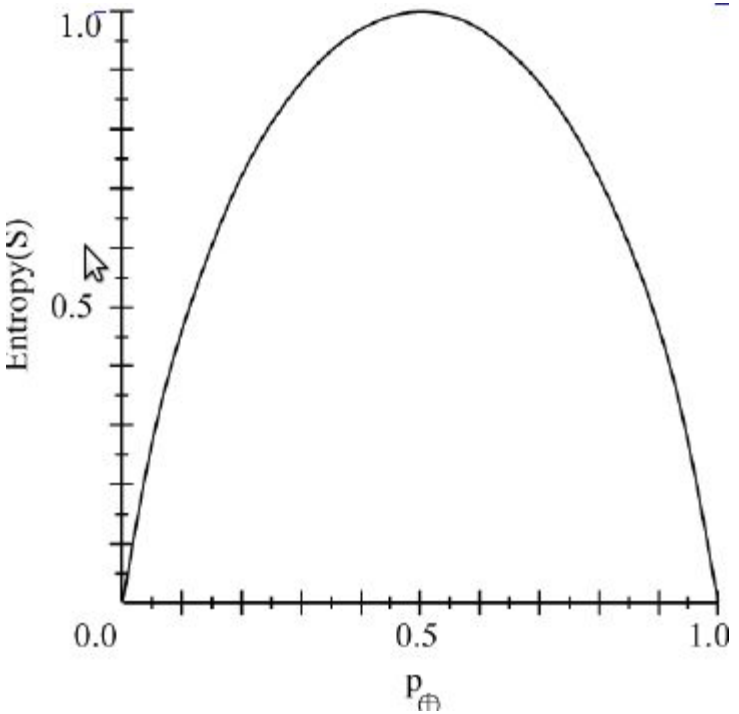
- O melhor atributo é selecionado e usado como **raiz** da árvore.
- Um descendente (**sub-árvore**) do nó raiz é então criado para cada valor possível deste atributo e os exemplos de treinamento são ordenados para o nó descendente apropriado.
- O processo é repetido usando exemplos com cada nó descendente para selecionar o melhor atributo para avaliar naquele ponto da árvore.
- Um algoritmo de busca gulosa (*greed*) é utilizado
  - i.e., não recua para reconsiderar escolhas prévias.
- **Ganho de Informação** é utilizada como medida quantitativa.

Entropia

Medida de aleatoriedade de uma variável

- A entropia de uma variável nominal *X* que pode tomar *i* valores
  - $Entropia(X) = -\sum_i p_i \times \log_2 p_i$
  - Ex: $Entropia([9+, 5-]) = -\left(\frac{9}{14}\log_2\frac{9}{14} + \frac{5}{14}\log_2\frac{5}{14}\right) = 0.94$

- A entropia tem máximo ($log_2\ i$) se $p_i = p_j$ para qualquer i <> j
- A *Entropia*(*X*) = 0 se existe um *i* tal que $p_i = 1$
- É assumido que 0 x log_2 0 = 0



Ganho de Informação

- No contexto das árvores de decisão a entropia é usada para estimar a aleatoriedade da variável a prever: **classe**.
- Dado um conjunto de exemplos, que **atributo** escolher?
  - Os valores de um atributo definem partições do conjunto de exemplos
  - O ganho de informação mede a redução da entropia causada pela partição dos exemplos de acordo com os valores do atributo.

$$Ganho(S, A) = Entropia(S) - \sum_{v \in Valores(A)} \frac{|S_v|}{|S|} Entropia(S_v)$$

- A construção de uma árvore de decisão é guiada pelo objetivo de diminuir a entropia ou seja a aleatoriedade-dificuldade de previsão da variável objetivo

Exemplo

- Base de dados para o problema “Play Tennis”

Day	*Outlook*	*Temperature*	*Humidity*	*Wind*	*PlayTennis*
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Exemplo

- Calcular o Ganho (S , ***Wind***)
  - Valores (***Wind***) = { *Weak* , *Strong* }
  - *S* = [9+ , 5-]
    - S_{Weak} = [6+ , 2-]
    - S_{Strong} = [3+ , 3-]

$$Ganho(S, Wind) = E(S) - \sum_{v \in \{Weak, Strong\}} \frac{|S_v|}{|S|} E(S_v)$$

$$Ganho(S, Wind) = E(S) - \left(\frac{8}{14}\right) E(S_{Weak}) - \left(\frac{6}{14}\right) E(S_{Strong})$$

$$Ganho(S, Wind) = 0,940 - \left(\frac{8}{14}\right) 0,811 - \left(\frac{6}{14}\right) 1,000 = 0,048$$

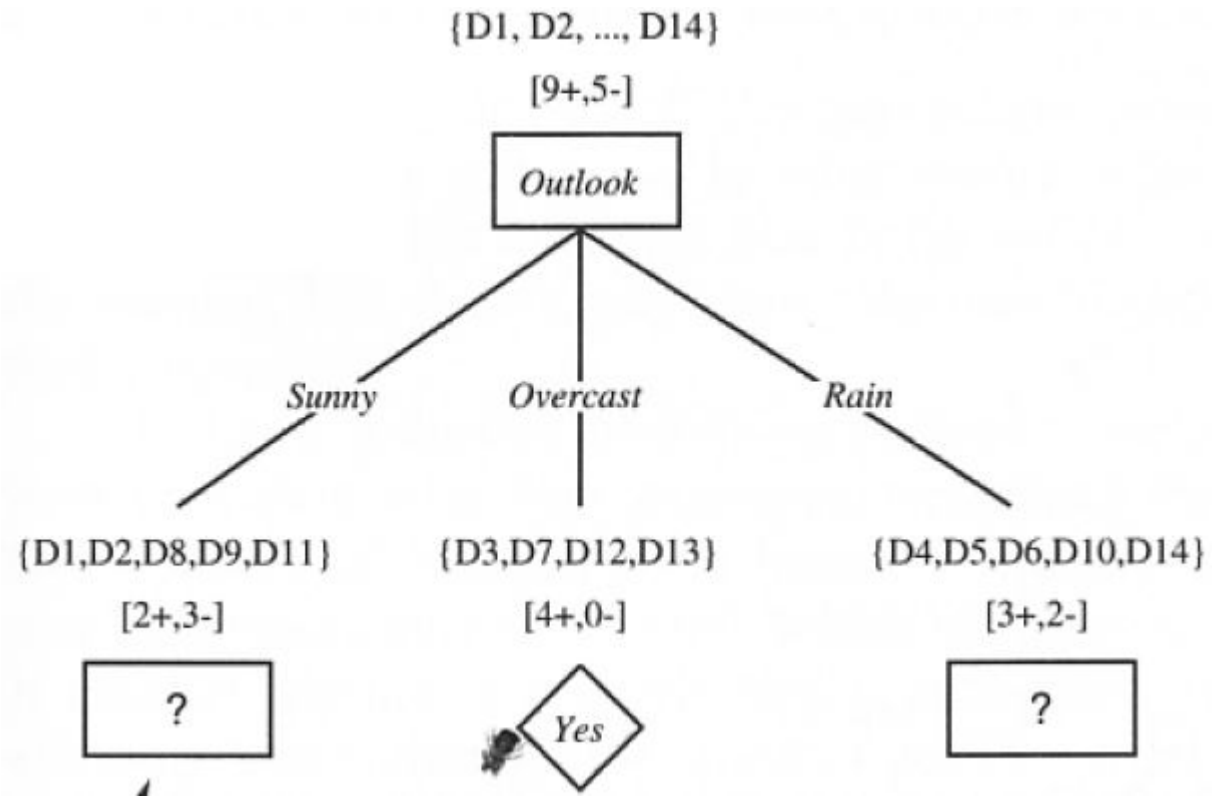
Exemplo

Voltando a pergunta inicial

Qual atributo deve ser testado primeiro na árvore?

- Determinar o ganho de informação (Gain) para cada atributo candidato.
- Selecionar aquele cujo o ganho de informação é o mais alto.
  - Ganho(S , ***Outlook***) = **0,246**
  - Ganho(S , ***Humidity***) = 0,151
  - Ganho(S , ***Wind***) = 0,048
  - Ganho(S , ***Temperature***) = 0,029

Exemplo

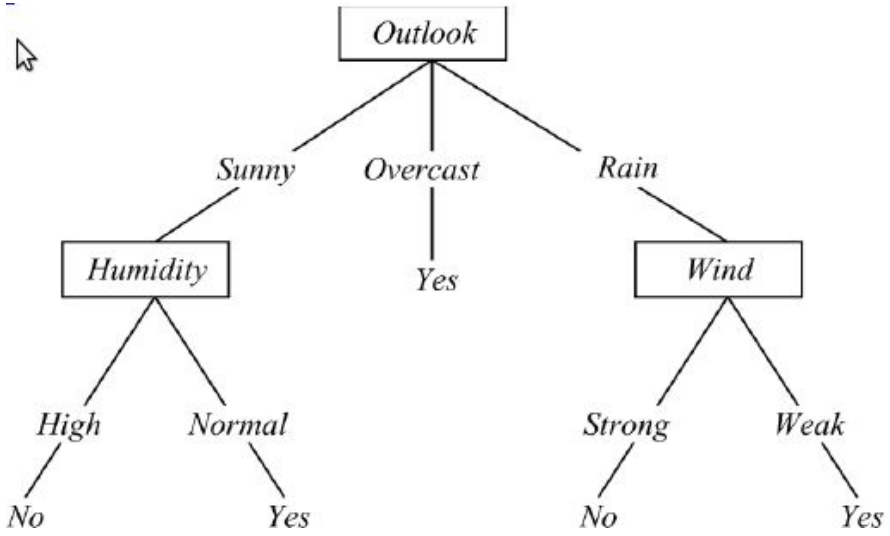

{D1, D2, ..., D14}
[9+,5-]
Outlook
Sunny
Overcast
Rain
{D1,D2,D8,D9,D11}
[2+,3-]
?
{D3,D7,D12,D13}
[4+,0-]
Yes
{D4,D5,D6,D10,D14}
[3+,2-]
?

Exemplo

- O processo para selecionar um novo atributo e particionar os exemplos de treinamento é repetido para cada nó descendente não terminal.
- São utilizados somente os exemplos de treinamento associados com este nó.
- Atributos que foram incorporados anteriormente à árvore são excluídos. Qualquer atributo deve aparecer somente uma vez ao longo de qualquer caminho na árvore.
- Este processo continua até que uma das seguintes condições seja atendida:
  a. Todos os atributos já estejam incluídos ao longo deste caminho da árvore.
  b. Os exemplos de treinamento associados com este nó folha tenham todos o mesmo valor de atributo alvo.

Exemplo

Play Tennis



Atributos de Valor Contínuo (c4.5)

- Em alguns casos, os valores dos atributos podem ser apresentados na forma contínua, por exemplo:

Temperatura:	40	48	60	72	80	90
Play Tennis:	No	No	Yes	Yes	Yes	No

- Escolher um limiar ***c*** que produza o maior ganho de informações.
- Identificar exemplos adjacentes que diferem na classificação do alvo.

- Por exemplo, um limiar poderia ser
  - ***c*** = (48 + 60)/2 = 54

Exemplo

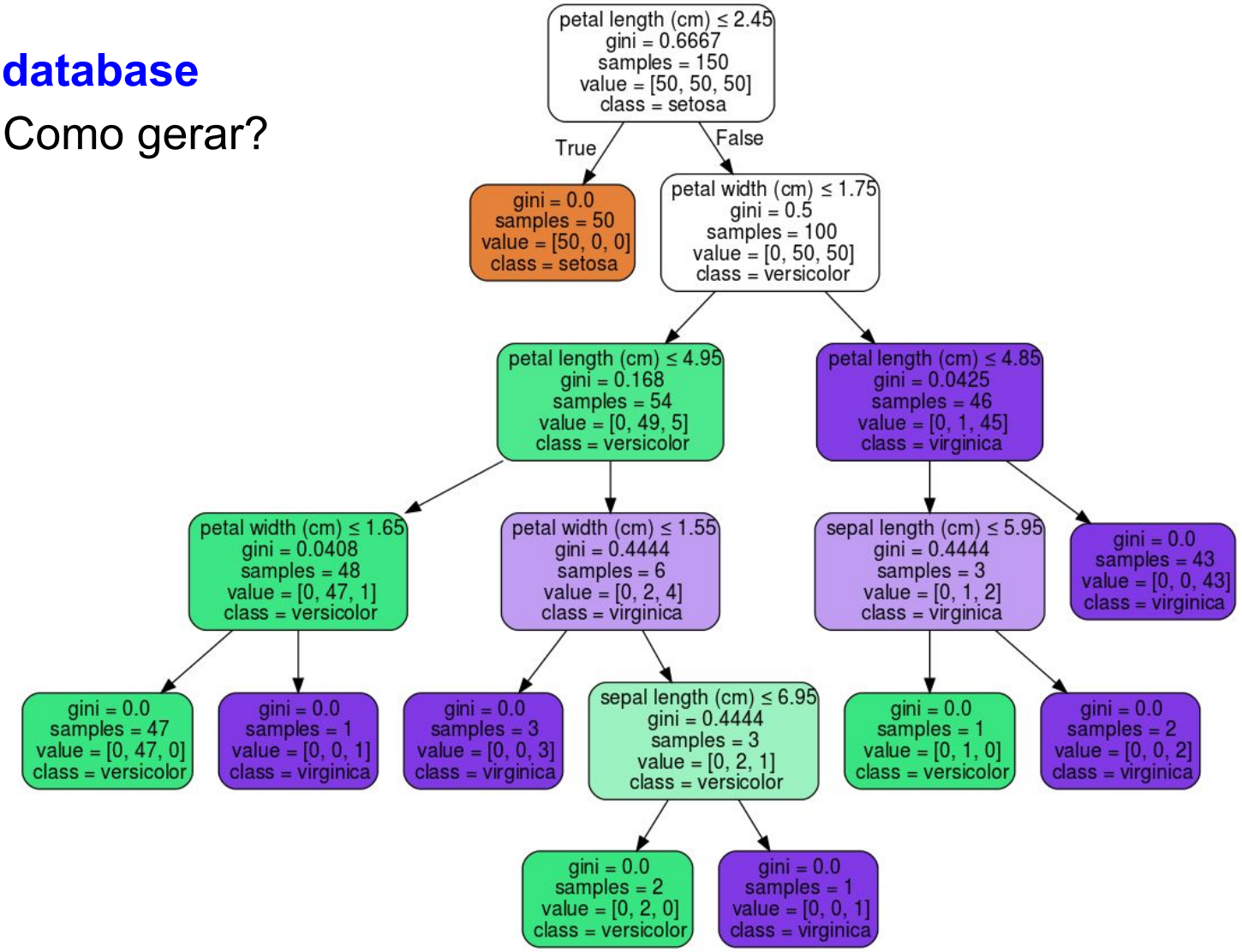
Iris database (UCI Machine Learning Repository)

- 3 Classes:
  - `Versicolor`, `Setosa` & `Virginica`
- 4 Atributos numéricos:
  - ***Sepal Length***, ***Sepal Width***
  - ***Petal Length***, ***Petal Width***
- 150 Instâncias

Exemplo

Iris database

- Como gerar?



Exemplo

Iris database

- Carregar base e criar árvore

```
>>> from sklearn.datasets import load_iris
>>> from sklearn import tree
>>> iris = load_iris()
>>> clf = tree.DecisionTreeClassifier()
>>> clf = clf.fit(iris.data, iris.target)
```

- Como gerar aquela representação gráfica?

```
>>> import graphviz
>>> dot_data = tree.export_graphviz(clf, out_file=None)
>>> graph = graphviz.Source(dot_data)
>>> graph.render("iris")
```

- “Embelezando”

```
>>> dot_data = tree.export_graphviz(clf, out_file=None,
                                    feature_names=iris.feature_names,
                                    class_names=iris.target_names,
                                    filled=True, rounded=True,
                                    special_characters=True)
>>> graph = graphviz.Source(dot_data)
>>> graph
```

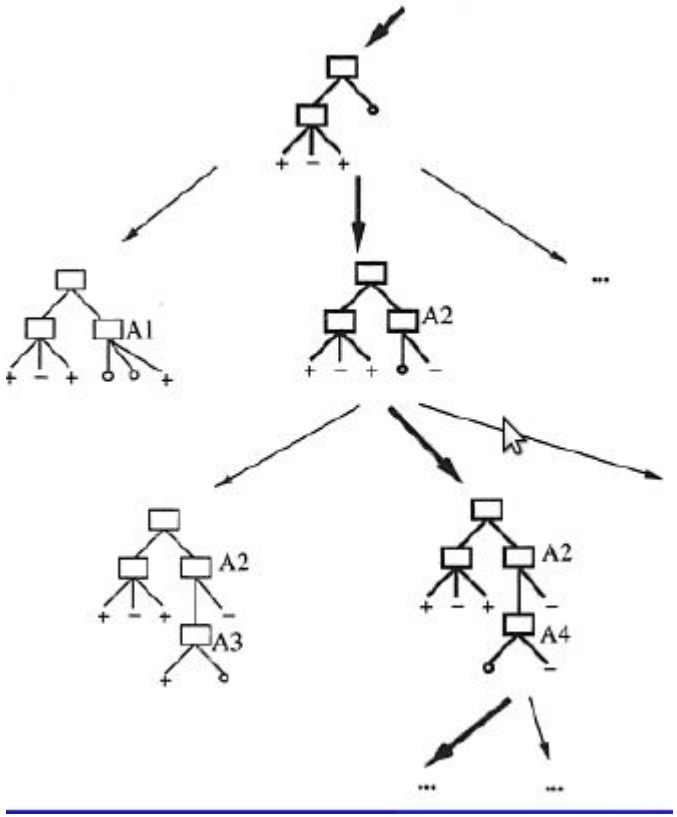
Busca no Espaço de Hipóteses

- O modelo de aprendizagem ID3 pode ser caracterizado como um método de busca em um espaço de hipótese, por uma hipótese que se ajusta aos exemplos de treinamento.

- O espaço de hipóteses buscado pelo ID3 é o conjunto de árvores de decisão possíveis.

- O ID3 realiza uma busca simples
  - *hill climbing* através do espaço de hipótese começando com uma árvore vazia e considerando progressivamente hipóteses mais elaboradas)

Busca no Espaço de Hipóteses



- Espaço de Hipóteses - ID3 procura possíveis árvores a partir da mais simples aumentando a complexidade, usando para isso o ganho de informação

Bias Indutivo

- Dada uma coleção de exemplos de treinamento, geralmente existem várias árvores de decisão consistentes com os exemplos.

- Qual árvore deve ser escolhida?
  - Mecanismo implícito do algoritmo.

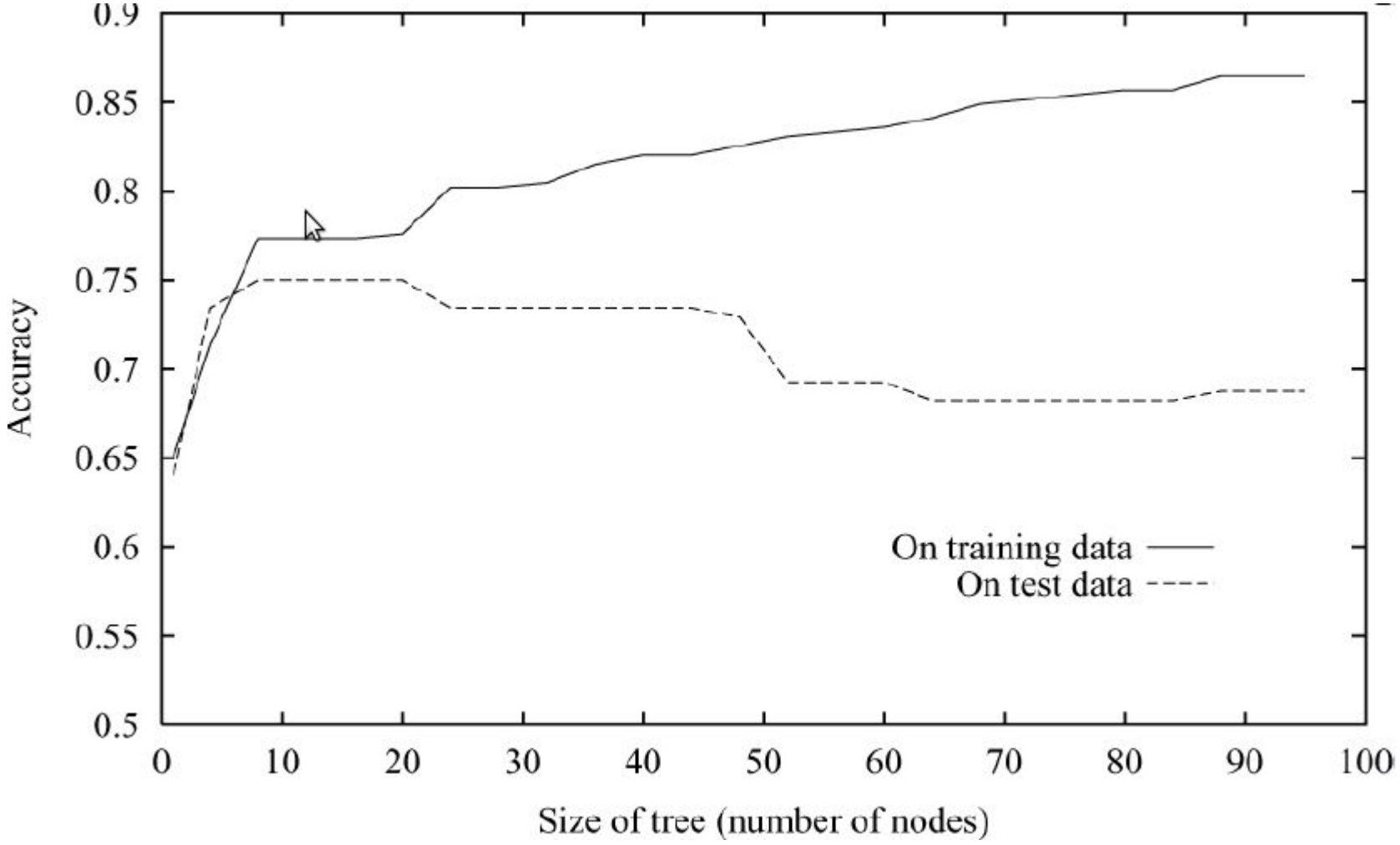
- Preferência é por árvores mais curtas e por aquelas com atributos de alto ganho de informação próximos da raiz.
  - **Bias**: é uma preferência por algumas hipóteses aos invés de uma restrição do espaço de hipóteses.
  - Modelos menos complexos (árvores menores) são preferíveis (menor risco de *overfitting*).

Overfitting

- Como detectar *overfitting* em árvores de decisão.
  - Erro da hipótese *h* sobre os dados de treinamento: $err_t(h)$
  - Erro da hipótese *h* sobre os todos os dados de: $err_{all}(h)$

- Uma hipótese $h \in H$ tem *overfitting* sobre os dados de treinamento se existir uma hipótese alternativa $h' \in H$ tal que
  - $err_t(h) < err_t(h')$ E
  - $err_{all}(h) > err_{all}(h')$

Overfitting



Evitando o *Overfitting*

- Como o *overfitting* pode ser evitado?
  - Parar o crescimento quando a partição de dados não for **estatisticamente significante**.
  - Desenvolver uma árvore completa e então fazer uma poda (*Pruning*).
    - Pode ser feita diretamente na árvore ou ainda no conjunto de regras geradas pela árvore.

- Como selecionar a melhor árvore?
  - Medida de desempenho sobre um conjunto de dados de validação

Árvores de Decisão - Resumo

- A aprendizagem de árvores de decisão fornece um método prático para a aprendizagem de conceito e para a aprendizagem de outras funções de valor discreto.
- A família de algoritmos ID3 infere árvores de decisão expandindo-as a partir da raiz e descendo, selecionando o próximo melhor atributo para cada novo ramo de decisão adicionado na árvore.
- O **bias indutivo** implícito no ID3 inclui uma preferência por árvores menores.
- ***Overfitting*** é um aspecto importante na aprendizagem de árvores de decisão.

Aprendizagem Bayesiana

Aprendizagem Bayesiana

Agenda

- Introdução
- Teorema de Bayes
- Classificador Naïve Bayes

Aprendizagem Bayesiana

- O pensamento Bayesiano fornece uma abordagem probabilística para a aprendizagem.
- Está baseado na suposição de que as quantidades de interesse são reguladas por distribuições de probabilidades.
- Quantificar o custo/benefício entre diferentes decisões de classificação usando probabilidades e custos associados à classificação.
- Teorema de Bayes
  - Mostra como alterar as probabilidades *a priori* tendo em conta novas evidências de forma a obter probabilidades *a posteriori*.

Aprendizagem Bayesiana

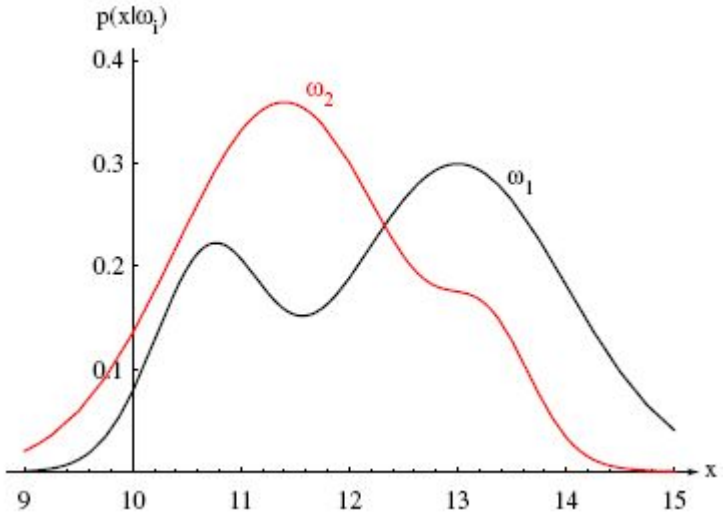
Terminologia

- Classes ω_i (variável aleatória)
- Probabilidades *a priori* $P(\omega_i)$
  - Conhecimento a priori que se tem sobre o problema, ou seja, conhecimento a priori sobre a aparição de exemplos das classes do problema.
- Função de Densidade Probabilidade $P(x)$
  - Frequência com a qual encontramos uma determinada característica
  - Evidências

Aprendizagem Bayesiana

Terminologia

- Densidade de Probabilidade Condicional
  - $P(x|\omega_j)$ (*Likelihood*)
  - Frequência com que encontramos uma determinada característica dado que a mesma pertence a classe ω_j



Densidade de duas classes em que x representa uma característica qualquer

Aprendizagem Bayesiana

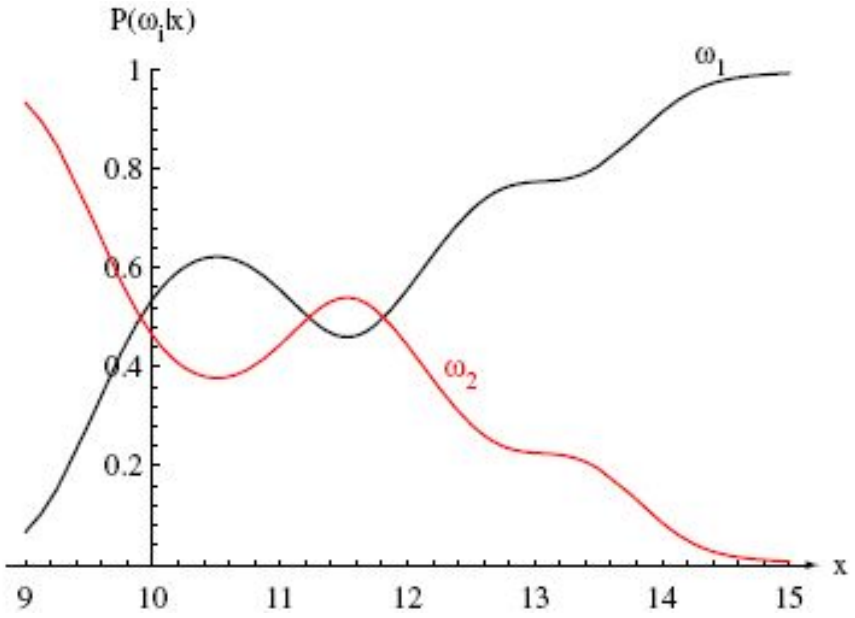
Terminologia

- Probabilidade *a posteriori*
  - $P(\omega_j|x)$ - Probabilidade que o padrão pertença a classe ω_j data a característica x

- Regra de decisão
  - ω_1, se $P(\omega_1) > P(\omega_2)$
  - ω_2, caso contrário

Aprendizagem Bayesiana

Tomando decisão usando Bayes



Probabilidades *a posteriori* calculadas usando $P(\omega_1) = 2/3$ e $P(\omega_2) = 1/3$

Nesse caso, para um valor de $x = 14$, a probabilidade do padrão pertencer a é de 0,08, enquanto que a probabilidade do padrão pertencer a é de 0,92.

Para cada x, as probabilidades a posterior somam 1.

Aprendizagem Bayesiana

Teorema de Bayes

- Basicamente o teorema de Bayes mostra como rever as crenças sempre que novas evidências são coletadas.
- Ou seja, atualizar a probabilidade *a posteriori* utilizando para isso a probabilidade *a priori*, as verossimilhanças e as evidências

$$P(A|B) = \frac{P(A) \times P(B|A)}{P(B)}$$

- $P(A|B)|$ é a probabilidade a posteriori
- $P(A)$ é a probabilidade a priori
- $P(B|A)|$ são as verossimilhanças (likelihood)
- $P(B)|$ são as evidências, dado por $\sum P(A_i) \times P(B|A_i)$

Aprendizagem Bayesiana

Exemplo

- Um médico sabe que a meningite causa torcicolo em 50% dos casos. Porém, o médico sabe que a meningite atinge 1/50.000 e também que a probabilidade de se ter torcicolo é de 1/20.
- Usando Bayes para saber a probabilidade de uma pessoa ter meningite dado que ela está com torcicolo

Temos então

- $P(T \mid M) = 0{,}5$ $P(M) = 1 / 50.000$ $P(T) = 1 / 20$

$$P(M|T) = \frac{P(M) \times P(T|M)}{P(T)} = \frac{\frac{1}{50000} \times 0.5}{1/20}$$

$$P(M|T) = 0,0002 = \frac{2}{10000} = 0,02\%$$

Aprendizagem Bayesiana

Exercício

- Considere o sistema de classificação de peixes. Para essa época do ano, sabe-se que a probabilidade de pescar salmão é maior que pescar robalo, P(salmao) = 0,82 e P(robabo) = 0,18.
- Suponha que a única característica que você pode contar é a intensidade do peixe ou seja, se ele é claro ou escuro. Sabe-se que 49.5\% dos salmões tem intensidade clara e que 85\% dos robalos tem intensidade clara.
- Calcule a probabilidade de ser salmão dado que o peixe pescado tem intensidade clara.

$$P(S|C) = \frac{P(S) \times P(C|S)}{P(C)} = \frac{0{,}82 \times 0{,}495}{0{,}82 \times 0{,}495 + 0{,}18 \times 0{,}85} = 0,726$$

Classificador Naïve Bayes

- Um dos algoritmos de aprendizagem mais práticos e utilizados na literatura.

- Denominado Naive (ingênuo) por assumir que os atributos são condicionalmente independentes, ou seja, a informação de um evento não é informativa sobre nenhum outro.

- Apesar dessa premissa, o classificador reporta bom desempenho em diversas tarefas de classificação onde há dependência.

- Aplicações bem sucedidas:
  - Diagnóstico
  - Classificação de documentos textuais

Classificador Naïve Bayes

- Se aplica a tarefas de aprendizagem onde cada instância x é descrita por uma conjunção de valores de atributos em que a função alvo, $f(x)$ pode assumir qualquer valor de um conjunto V
- Um conjunto de exemplos de treinamento da função alvo é fornecido a uma nova instância é apresentada, descrita pela tupla de valores de atributos $< a_1, a_2, \ldots, a_n >$.
- A tarefa é predizer o valor alvo (ou classificação) para esta nova instância.

Classificador Naïve Bayes

- O classificador é baseado na suposição de que os valores dos atributos são **condicionalmente independentes** dados o valor alvo.
- Se usarmos Bayes para múltiplas evidências, temos

$$P(H|E_1, E_2, \ldots, E_n) = \frac{P(E_1,E_2,\ldots,E_n|H)\times P(H)}{P(E_1,E_2,\ldots,E_n)}$$

- Considerando a hipótese de independência, podemos reescrever o teorema de Bayes da seguinte forma:

$$P(H|E_1, E_2, \ldots, E_n) = \frac{P(E_1|H)\times,P(E_2|H)\times,\ldots,P(E_n|H)\times P(H)}{P(E_1,E_2,\ldots,E_n)}$$

O denominador pode ser ignorado por se tratar de um termo comum.

Classificador Naïve Bayes

Exemplo - Considere o seguinte problema:

	outlook	temperature	humidity	windy	play
1	sunny	hot	high	false	no
2	sunny	hot	high	true	no
3	overcast	hot	high	false	yes
4	rainy	mild	high	false	yes
5	rainy	cool	normal	false	yes
6	rainy	cool	normal	true	no
7	overcast	cool	normal	true	yes
8	sunny	mild	high	false	no
9	sunny	cool	normal	false	yes
10	rainy	mild	normal	false	yes
11	sunny	mild	normal	true	yes
12	overcast	mild	high	true	yes
13	overcast	hot	normal	false	yes
14	rainy	mild	high	true	no

Classificador Naïve Bayes

Construindo o Modelo (NB)

- O primeiro passo consiste em construir o modelo de probabilidades condicionais Naïve Bayes (NB) - **14 dias**

Outlook			Temperature			Humidity			Windy			Play	
	Yes	*No*		*Yes*	*No*		*Yes*	*No*		*Yes*	*No*	*Yes*	*No*
Sunny	2	3	Hot	2	2	High	3	4	False	6	2	9	5
Overcast	4	0	Mild	4	2	Normal	6	1	True	3	3		
Rainy	3	2	Cool	3	1								

- A tabela acima contém a frequência de diferentes evidências.
- Por exemplo, existem duas instâncias mostrando (***Outlook*** = `Sunny`) para (***Play*** = `Yes`)

Classificador Naïve Bayes

Computando probabilidades

- Após definir todas as frequências é necessário calcular todas as probabilidades condicionais e as probabilidades *a priori*.

Outlook			Temperature			Humidity			Windy			Play	
	Yes	*No*		*Yes*	*No*		*Yes*	*No*		*Yes*	*No*	*Yes*	*No*
Sunny	2/9	3/5	Hot	2/9	2/5	High	3/9	4/5	False	6/9	2/5	9/14	5/14
Overcast	4/9	0/5	Mild	4/9	2/5	Normal	6/9	1/5	True	3/9	3/5		
Rainy	3/9	2/5	Cool	3/9	1/5								

- Por exemplo
  - *P*(***Outlook*** = `Sunny` | ***Play*** = `Yes`) = 2/9
  - *P*(***Play*** = `Yes`) = 9/14

Classificador Naïve Bayes

Predição

- De posse do modelo, podemos usá-lo para predizer um evento "***Play***"' com base em um conjunto qualquer de evidências.
- Por exemplo: `[Sunny,Cool,High,True,?]`

P(`Yes` | *E*) = (*P*(***Outlook*** = `Sunny` | `Yes`) x
P(***Temp*** = `Cool` | `Yes`) x
P(***Humidity*** = `High` | `Yes`) x
P(***Windy*** = `True` | `Yes`) x
P(`Yes`)) / *P*(*E*)

P(*E*) pode ser ignorada por se tratar de um denominador comum quando queremos comparar as duas classes. Deste modo, temos

$$P(Yes|E) = \frac{2}{9} \times \frac{3}{9} \times \frac{3}{9} \times \frac{3}{9} \times \frac{9}{14}$$

Classificador Naïve Bayes

Predição

- Calculando a predição para as duas classes
  - Para *Yes* temos $\frac{2}{9} \times \frac{3}{9} \times \frac{3}{9} \times \frac{3}{9} \times \frac{9}{14} = 0,0053$
  - Para *No* temos $\frac{3}{5} \times \frac{1}{5} \times \frac{4}{5} \times \frac{3}{5} \times \frac{5}{14} = 0,0206$

- Convertendo esses valores para probabilidade através da normalização, temos
  - *P*(*Yes* | *E*) = 0,0053 / (0,0053+0.0206) = 0,205 = 20,5%
  - *P*(*No* | *E*) = 0,0206 / (0,0053+0.0206) = 0,795 = 79,5%

Classificador Naïve Bayes

Técnica de Suavização

- Em alguns casos, a frequência pode ser zero, como por exemplo

  P(***Outlook*** = `Overcast` | ***Play*** = *No*) = 0 / 5

- Isso cria um problema para calcular *P*(*No*), a qual será sempre zero quando esta evidência for utilizada.
  - Toda instância que contiver esta evidência terá probabilidade nula.

Classificador Naïve Bayes

Técnica de Suavização

- A técnica de suavização mais utilizada é a estimação de Laplace

$$P'(H|E) = \frac{n_c + \mu p}{n + \mu}$$

- n_c é o número de hipóteses existentes para a classe
  (Ex: Zero para ***Outlook*** = `Overcast` e *Play* = *No*)
- n número de exemplos totais para o treinamento
- Considerado que as evidências são igualmente distribuídas,
  tempo $p = \frac{1}{3}$ (`Sunny` , `Overcast` , `Rainy`)
- μ número de exemplos virtuais (distribuição uniforme?)

Classificador Naïve Bayes

Técnica de Suavização

- Reestimando os valores usando Laplace, teríamos
  - *P*(***Outlook*** = `Sunny` | ***Play*** = *No*) = (3 + 3x1/3) / (5+3) = 4 / 8
  - *P*(***Outlook*** = `Overcast` | ***Play*** = *No*) = (0 + 3x1/3) / (5+3) = 1 / 8
  - *P*(***Outlook*** = `Rainy` | ***Play*** = *No*) = (2 + 3x1/3) / (5+3) = 3 / 8
- Desta forma, todos os valores foram redistribuídos mantendo uma proporção similar

Classificador Naïve Bayes

Calculando as probabilidades para atributos contínuos

Existem duas maneiras

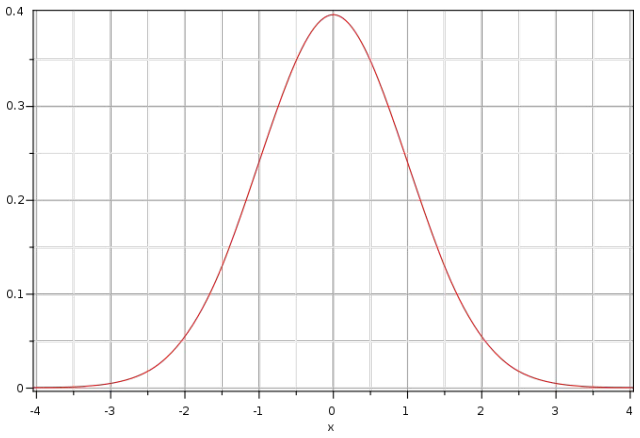
- Discretizar os atributos contínuos em algumas categorias, usando conhecimento sobre o domínio / aplicação.
  - Por exemplo
    - Temperatura acima de 80F pode ser considerada alta.
    - Idade acima de 18 anos, entre 18 e 65 anos, acima dos 65 anos
    - Salário anual: 50k, 100k, 200k, 500k, 1M
    - Gordura total: <100, [100,200], >200

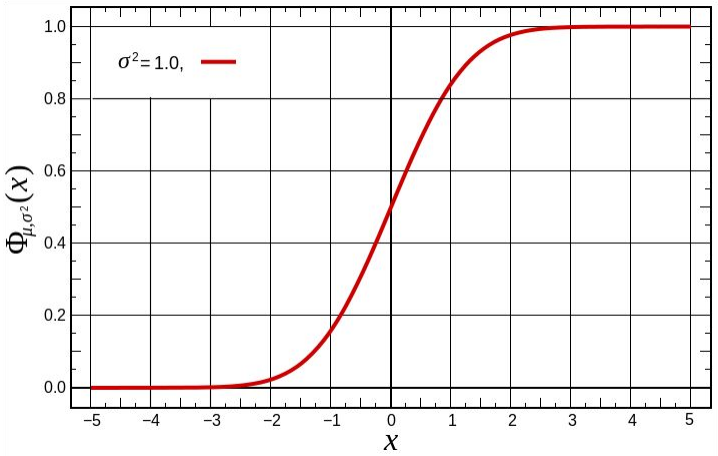
Classificador Naïve Bayes

Calculando as probabilidades para atributos contínuos

- Outra forma consiste em usar uma função de densidade de probabilidade (PDF) e desta forma preservar os valores contínuos.
  - Nesse caso assumimos que as variáveis contínuas seguem uma **distribuição normal**
  - Com isso em mente, podemos calcular a **média** e **desvio padrão** de cada variável usando a base de aprendizagem.
  - De posse da média e desvio padrão, basta aplicar a fórmula da normal para estimar a probabilidade

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

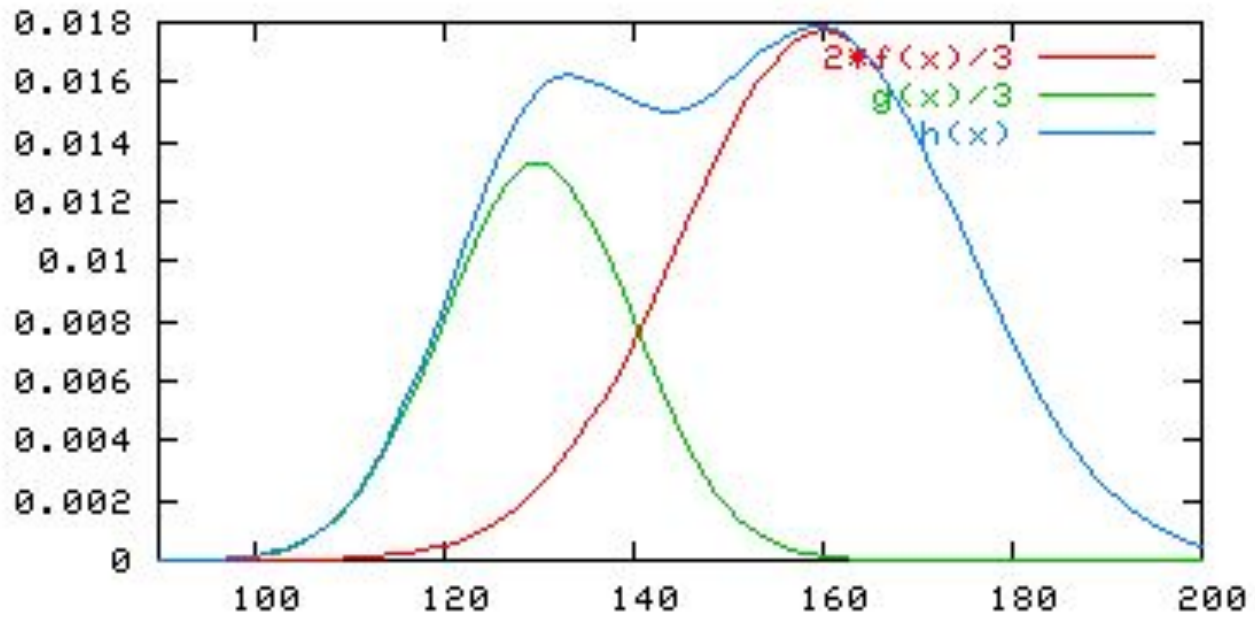




Classificador Naïve Bayes

Calculando as probabilidades para atributos contínuos

- Mistura de Gaussianas (GMM)
  - É possível ajustar várias Gaussianas aos dados



Classificador Naïve Bayes

Exemplo

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Normal distribution

		Humidity	*Mean*	*StDev*
Play Golf	yes	86 96 80 65 70 80 70 90 75	79.1	10.2
	no	85 90 70 95 91	86.2	9.7

$$P(\text{humidity} = 74 \mid \text{play} = \text{yes}) = \frac{1}{\sqrt{2\pi}(10.2)} e^{-\frac{(74-79.1)^2}{2(10.2)^2}} = 0.0344$$

$$P(\text{humidity} = 74 \mid \text{play} = \text{no}) = \frac{1}{\sqrt{2\pi}(9.7)} e^{-\frac{(74-86.2)^2}{2(9.7)^2}} = 0.0187$$

Classificador Naïve Bayes

Exemplo

Attribute	Class yes (0.63)	no (0.38)
outlook		
sunny	3.0	4.0
overcast	5.0	1.0
rainy	4.0	3.0
[total]	12.0	8.0
temperature		
mean	72.9697	74.8364
std. dev.	5.2304	7.384
weight sum	9	5
precision	1.9091	1.9091
humidity		
mean	78.8395	86.1111
std. dev.	9.8023	9.2424
weight sum	9	5
precision	3.4444	3.4444
windy		
TRUE	4.0	4.0
FALSE	7.0	3.0
[total]	11.0	7.0

Calcular a probabilidade para

E = [***Outlook*** = Rainy, ***Temp*** = 65, ***Humid*** = 70, ***Wind*** = True]

P(*Yes* | E) = 4 / 12 x 0,023 x 0,0271 x 4/11 x 0,63 ⟹ 0,75 = 75%

P(*No* | E) = 3 / 8 x 0,022 x 0,0094 x 4/7 x 0,38 ⟹ 0,25 = 25%

Classificador Naïve Bayes

Exemplo de Aplicação

- Classificação de texto, identificação de autoria, etc
  - Considere por exemplo que seja necessário construir um classificador que discrimine entre documentos relevantes e não relevantes
  - Base de treinamento deve conter documentos das duas classes
  - Quais características devem ser usadas?
    - *Bag of Words*: Conjunto pré-definido de palavras
    - Frequência de aparição dessas palavras pode ser utilizada como características

Aprendizado por Instâncias

Aprendizado por Instâncias

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

Aprendizado por Instâncias

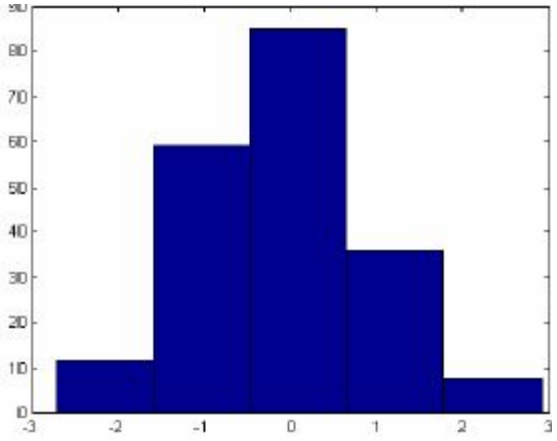
- A teoria de decisão Bayesiana assume que a distribuição do problema em questão é conhecida
  - Distribuição normal
- A grande maioria da distribuições conhecidas são unimodais.
- Em problemas reais a forma da função de densidade de probabilidade (fdp) é desconhecida
- Tudo que temos são os dados rotulados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

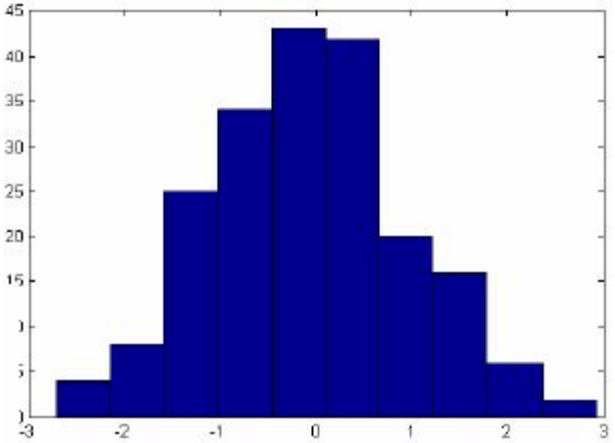
Aprendizado por Instâncias

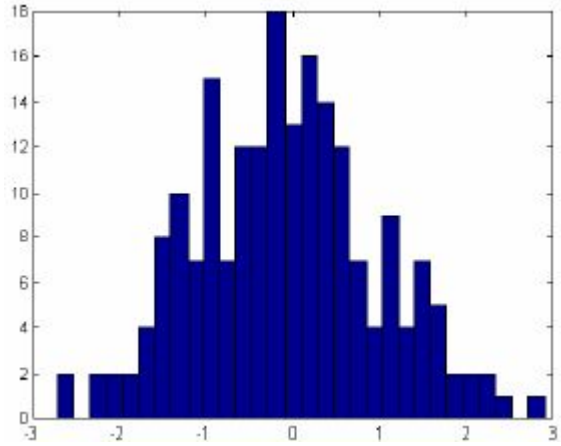
- Métodos não paramétricos podem ser usados com qualquer distribuição.
  - Histogramas
  - Janelas de Parzen
  - Vizinhos mais próximos.

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.

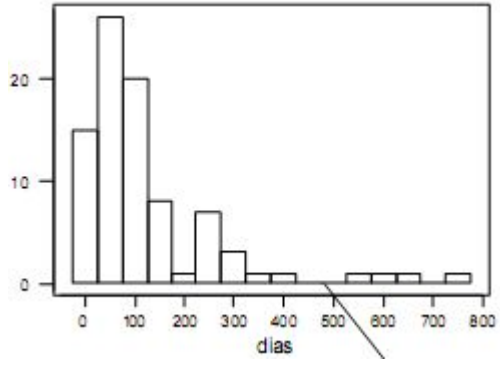






Histogramas

- 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***,

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

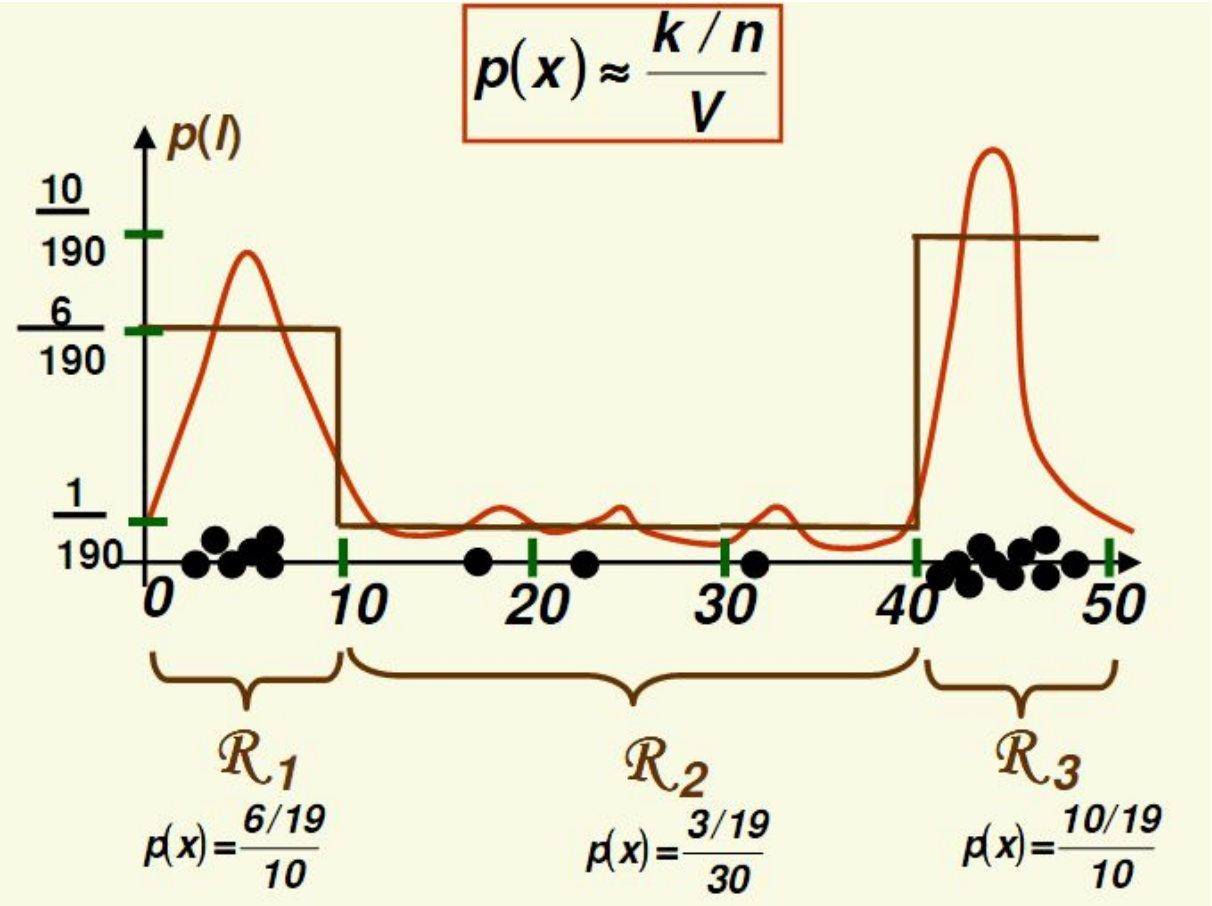
- Considerando ***n*** grande

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

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

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

Estimação de Densidade



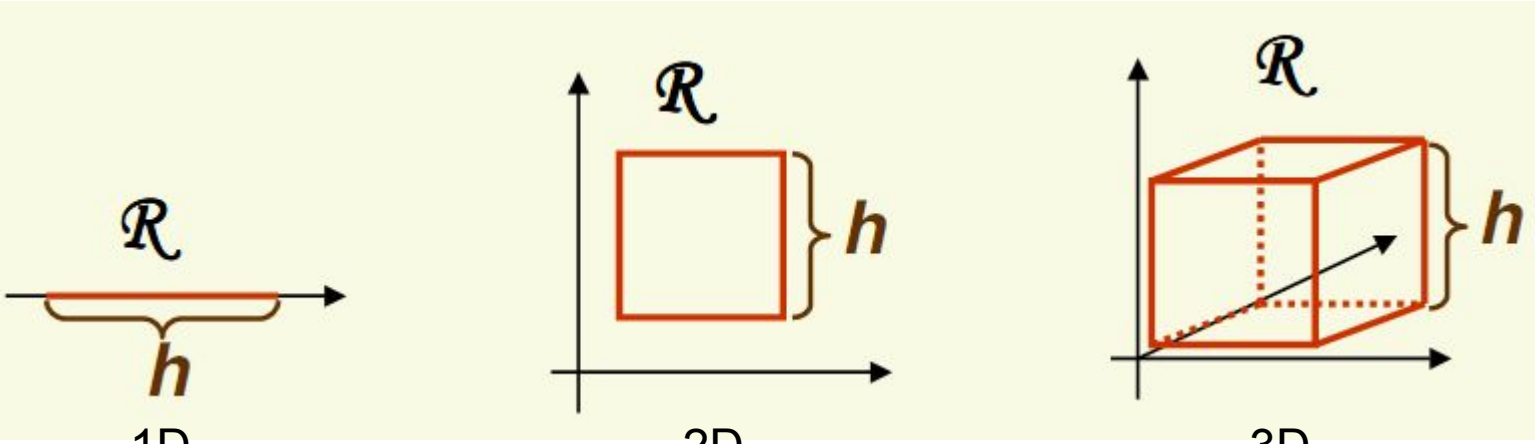
Se as regiões R_i não tem interseção, então temos um histograma.

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

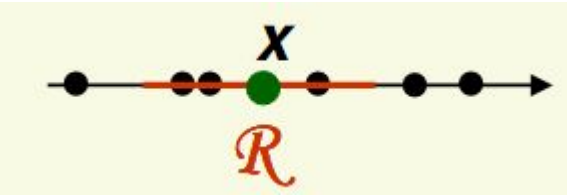
- Nessa abordagem fixamos o tamanho da região ***R*** para estimar a densidade.
- Fixamos o volume ***V*** e determinamos o correspondente ***k*** a partir dos dados de aprendizagem.
- Assumindo que a região ***R*** é um hipercubo de tamanho ***h***, seu volume é h^d



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

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

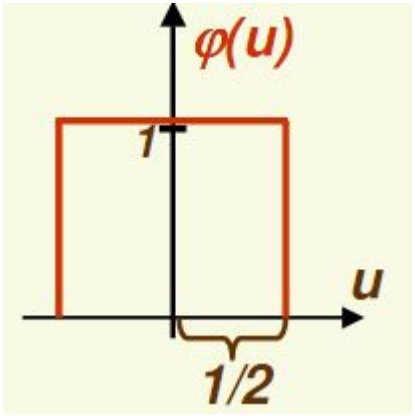


$$p(x) \approx \frac{3/6}{10}$$

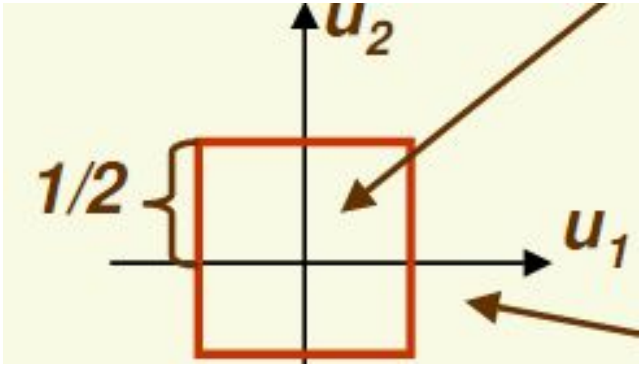
Janelas de Parzen

- 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

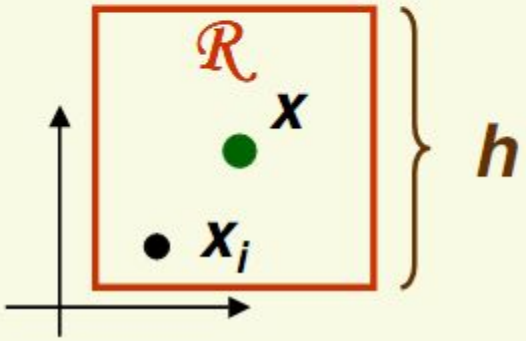
0 - Fora

2D

Janelas de Parzen

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

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

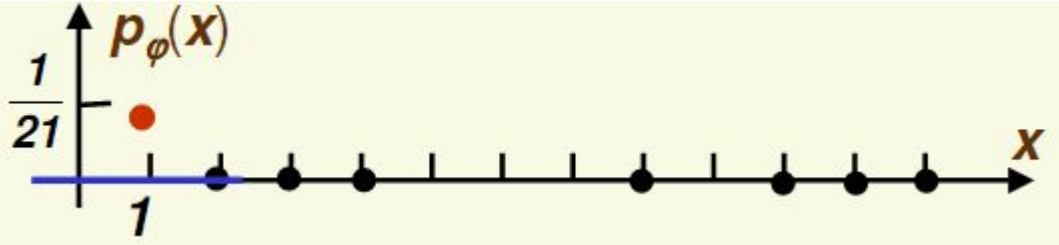


$$\varphi\left(\frac{x - x_i}{h}\right) = \begin{cases} 1 \\ 0 \end{cases}$$

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.

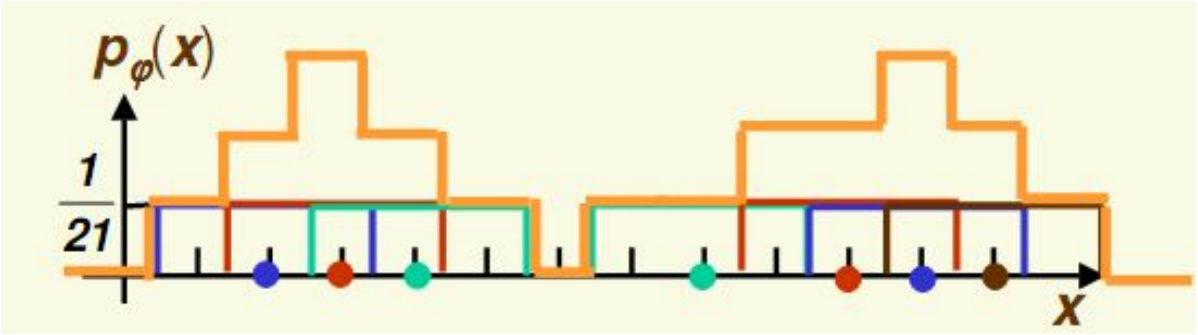
$$p_\varphi(1) = \frac{1}{7}\sum_{i=1}^{i=7}\frac{1}{3}\varphi\left(\frac{1-x_i}{3}\right) = \frac{1}{21}\left[\varphi\left(\frac{1-2}{3}\right)+\varphi\left(\frac{1-3}{3}\right)+\varphi\left(\frac{1-4}{3}\right)+\ldots+\varphi\left(\frac{1-12}{3}\right)\right]$$

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

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

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

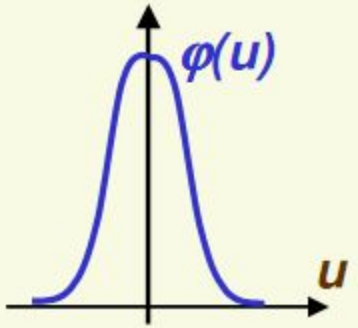


Janelas de Parzen: *Kernel* Gaussiano

- 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.

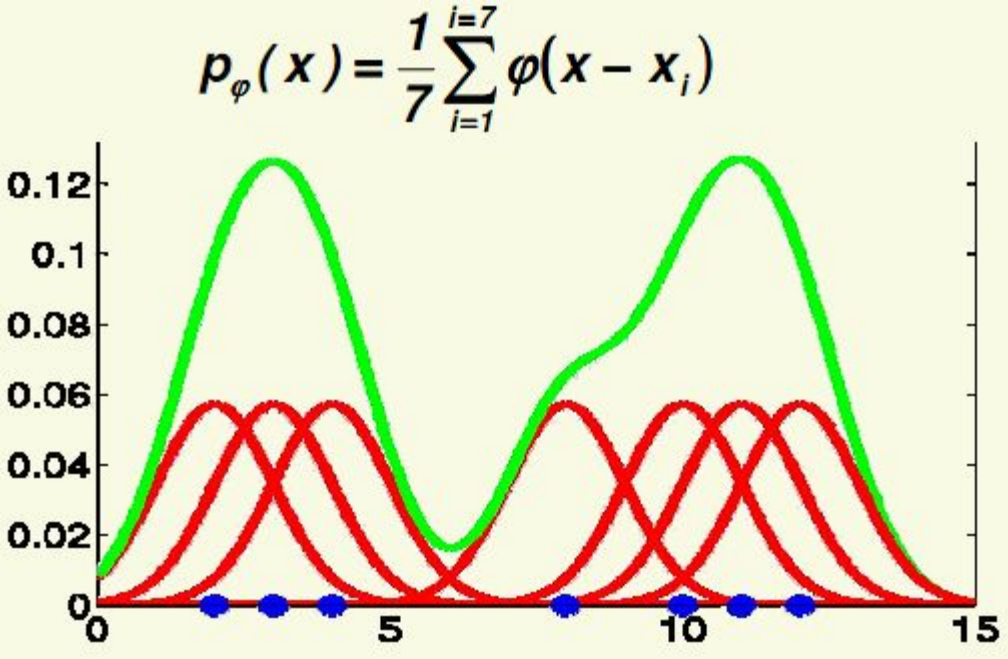
$$p_\varphi(x) = \frac{1}{n}\sum_{i=1}^{i=n}\frac{1}{h^d}\varphi\left(\frac{x - x_i}{h}\right)$$

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



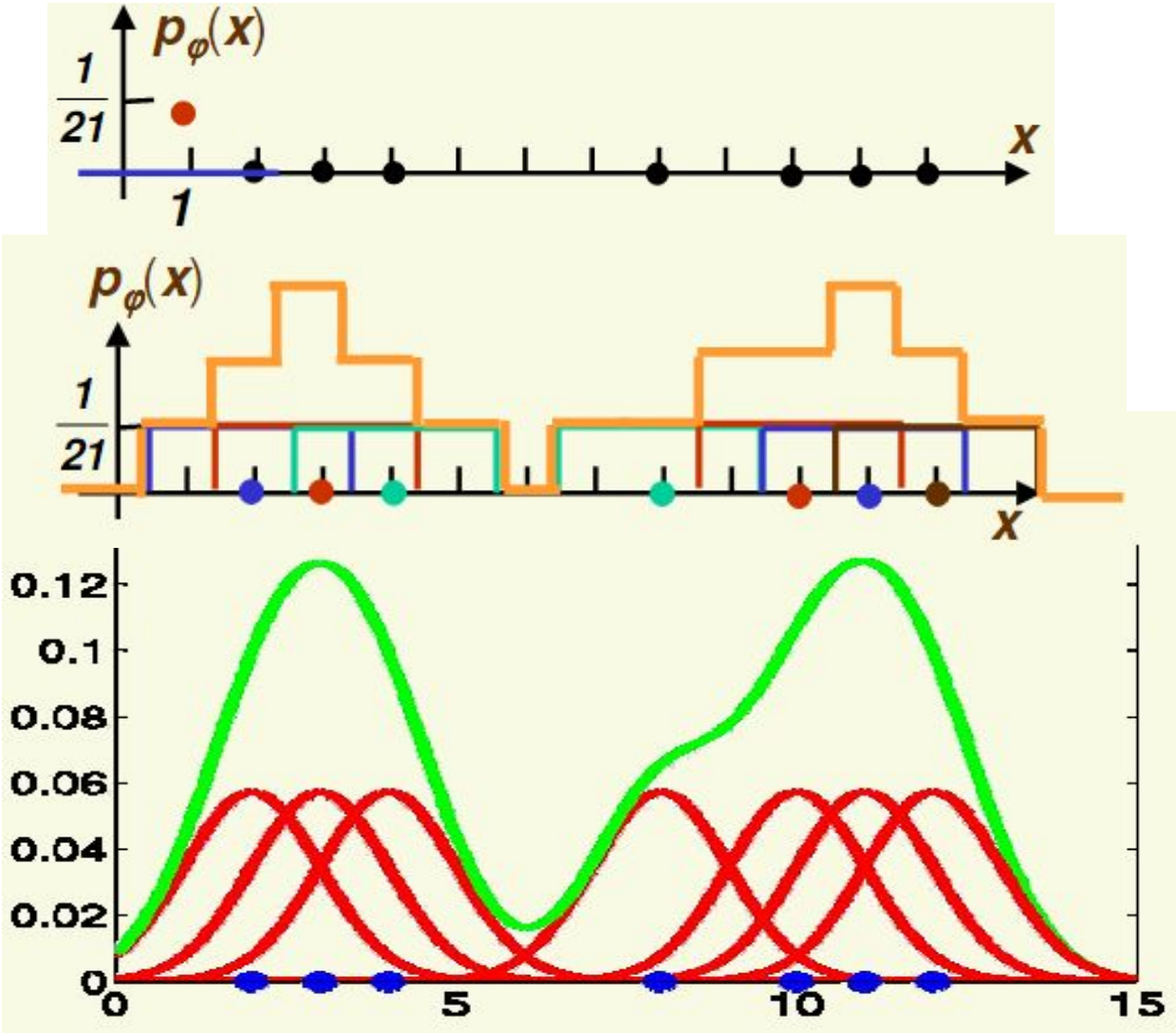
Janelas de Parzen: *Kernel* Gaussiano

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



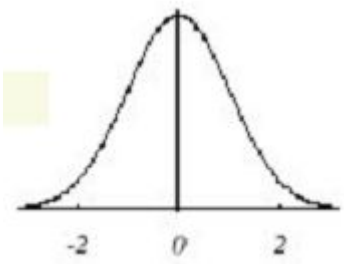
http://www.eee.metu.edu.tr/~alatan/Courses/Demo/AppletParzen.html

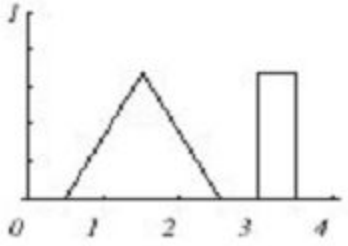
Janelas de Parzen



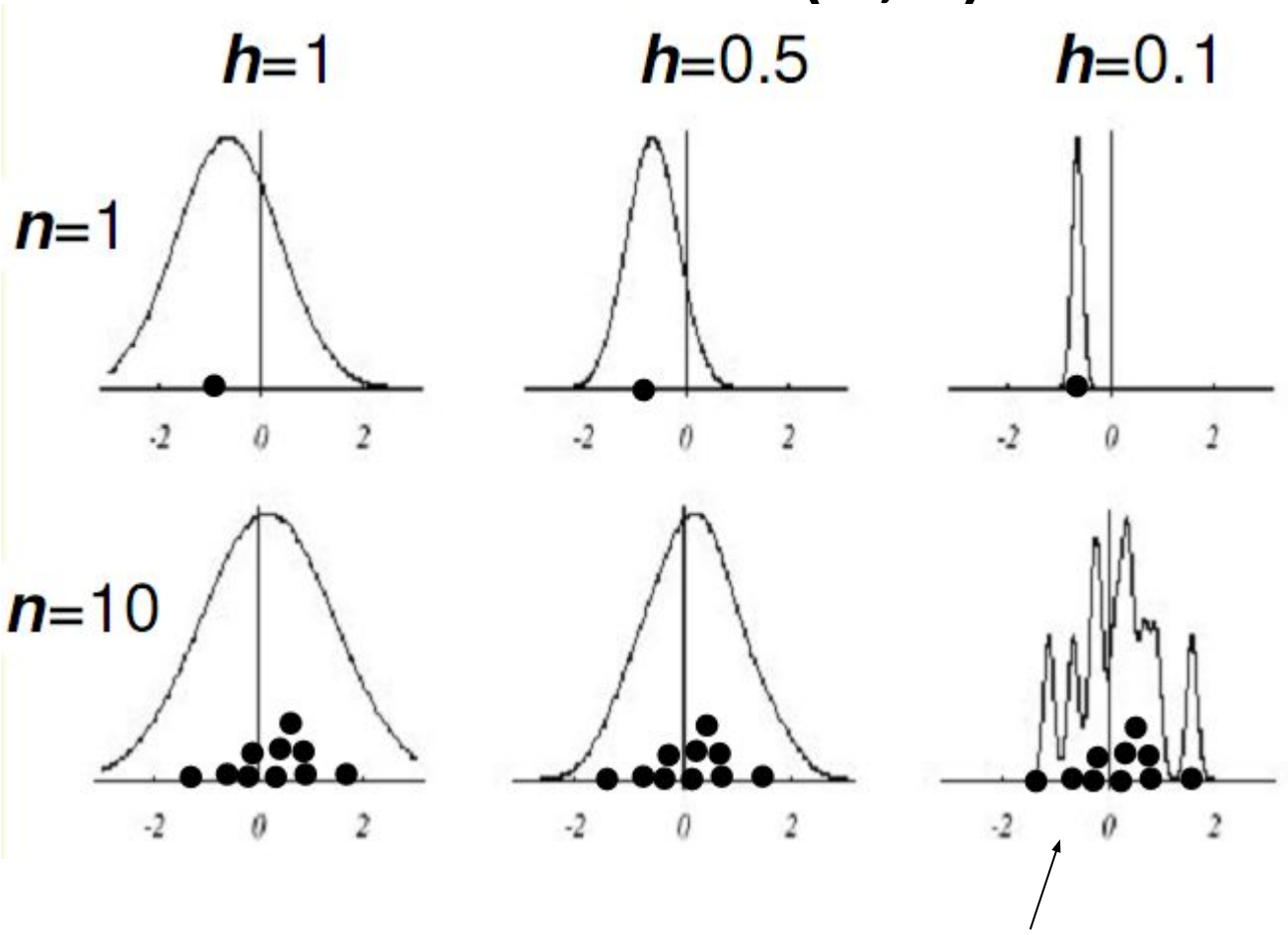
Janelas de Parzen

- 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)

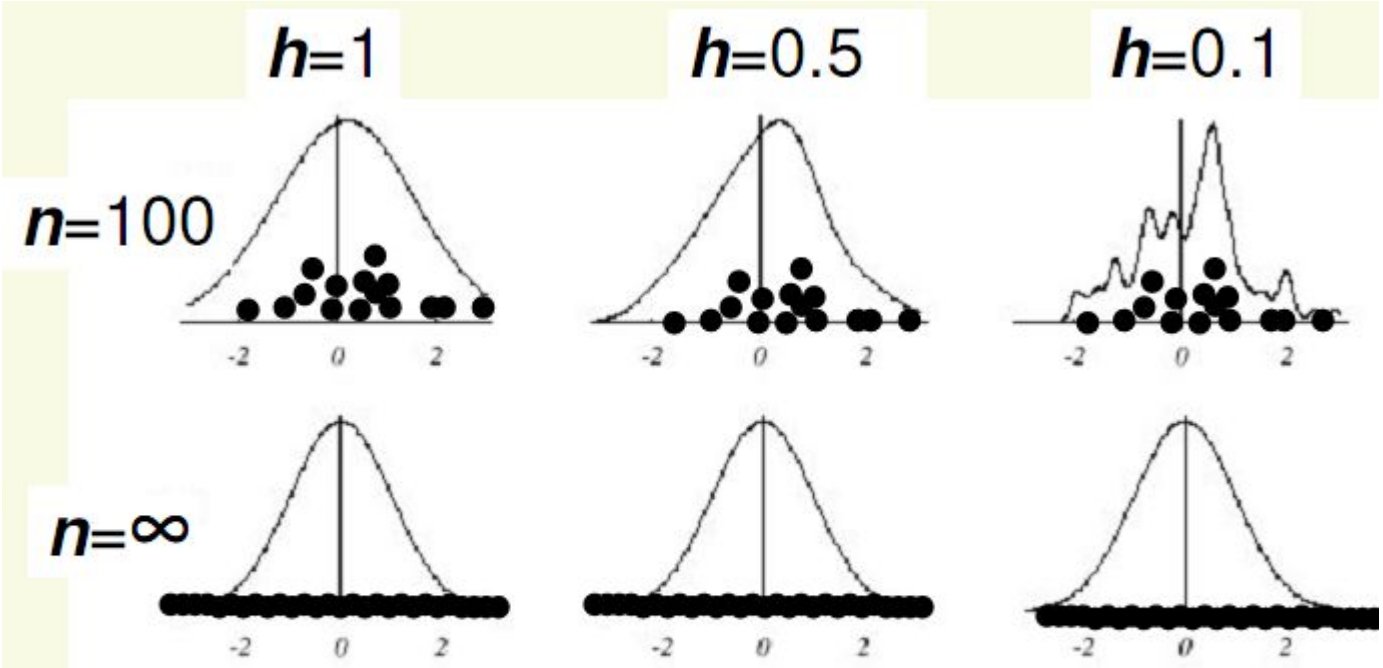
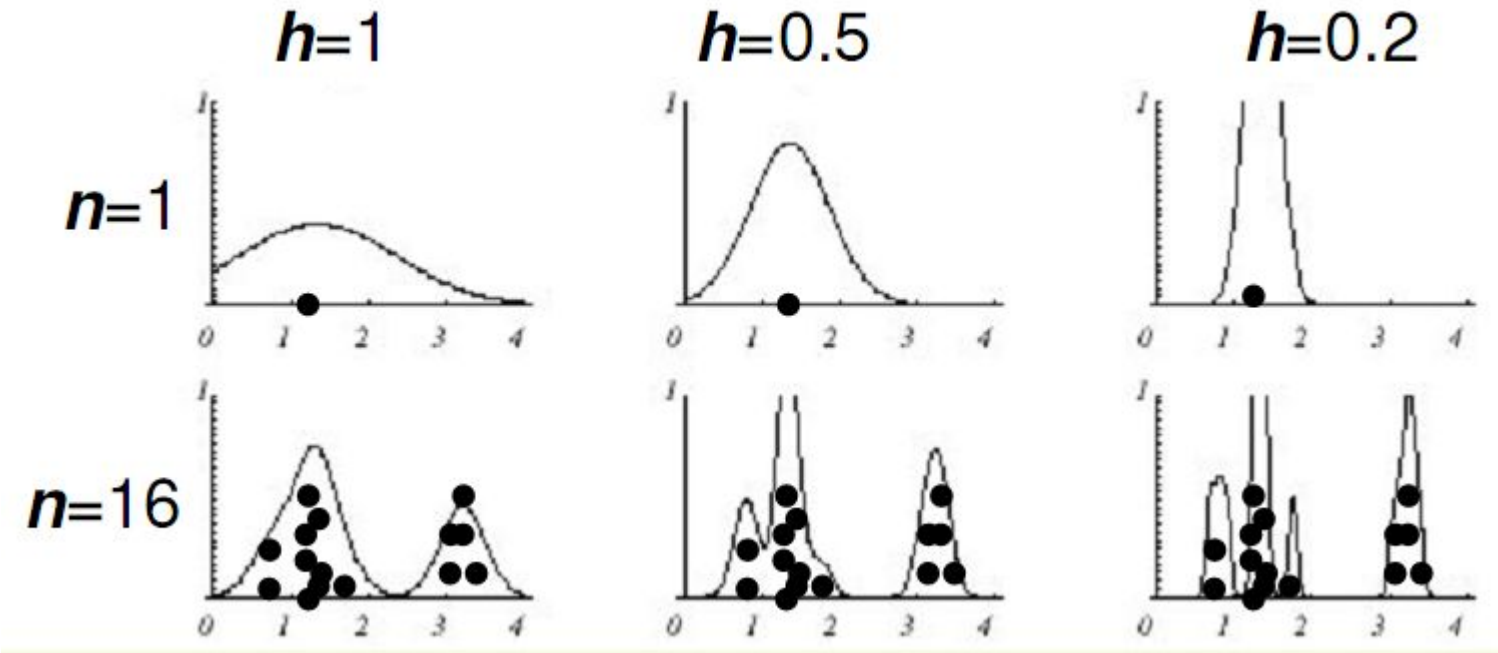


FIGURE 4.5. Parzen-window estimates of a univariate normal density using different window widths and numbers of samples. The vertical axes have been scaled to best show the structure in each graph. Note particularly that the $n = \infty$ estimates are the same (and match the true density function), regardless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

Janelas de Parzen:
Mistura de Triângulo e Uniforme



Janelas de Parzen:
Mistura de Triângulo e Uniforme

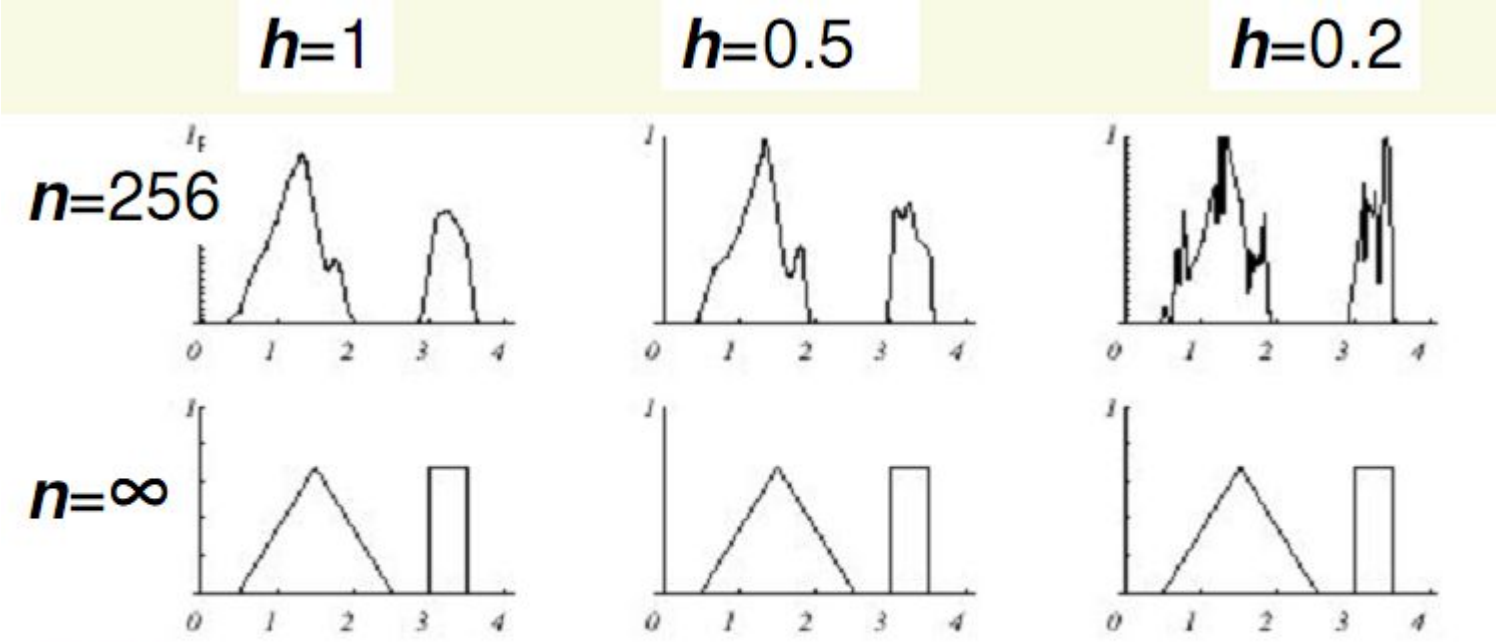
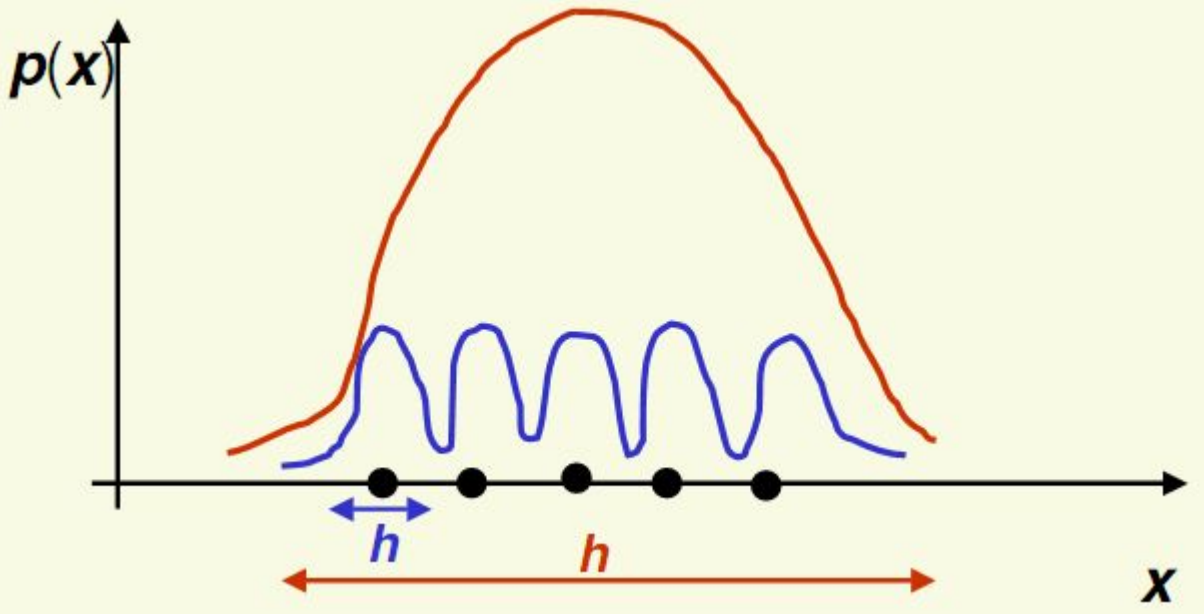


FIGURE 4.7. Parzen-window estimates of a bimodal distribution using different window widths and numbers of samples. Note particularly that the $n = \infty$ estimates are the same (and match the true distribution), regardless of window width. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern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

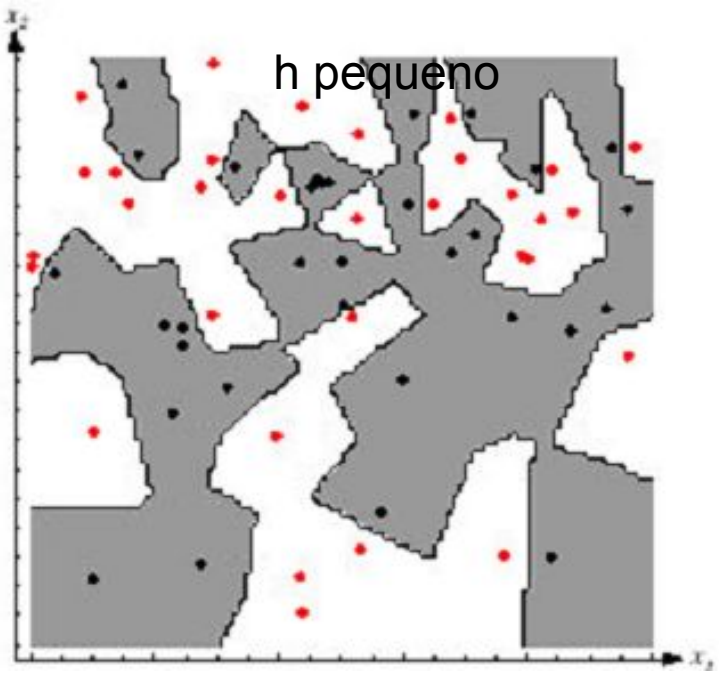


Janelas de Parzen: Tamanho da Janela

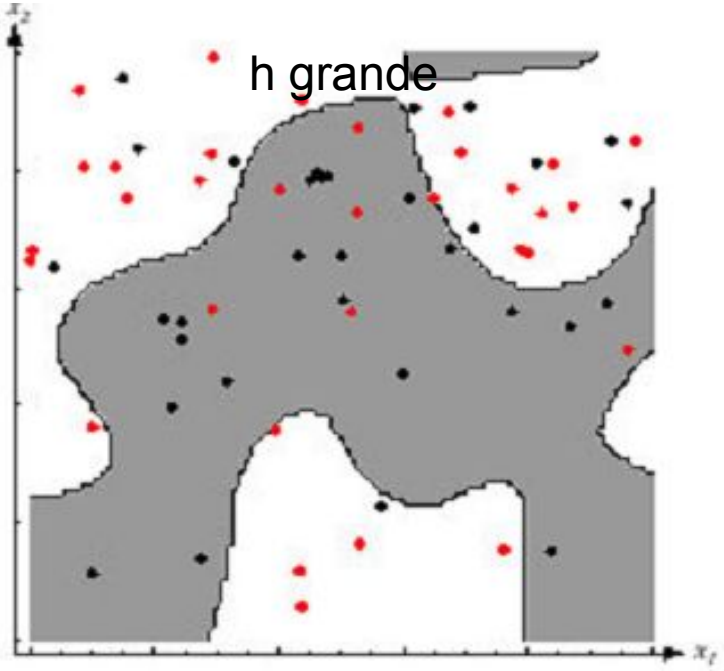
- Se ***h*** for muito pequeno
  - Fronteiras muito especializadas
- Se ***h*** for muito grande
  - Generaliza demais
- Encontrar um valor ideal para ***h*** não é uma tarefa trivial, mas pode ser estabelecido a partir de uma base de validação.
  - Aprender ***h***

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

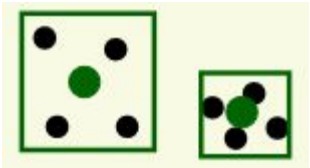
$$p(x) \approx \frac{k/n}{V}$$

- 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

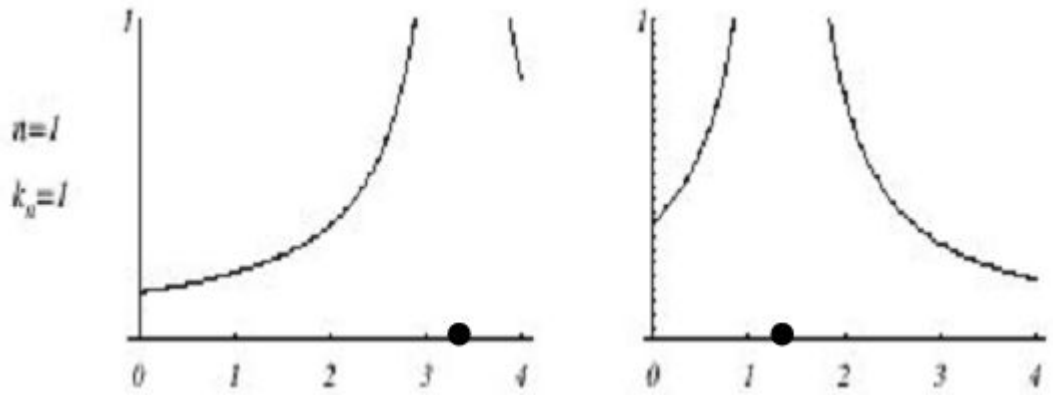
- Um alternativa interessante para o problema da definição da janela ***h***.
  - Nesse caso, o volume é estimado em função dos dados
    - Coloca-se a célula sobre **x**.
    - Cresce até que ***k*** elementos estejam dentro dela.



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***)

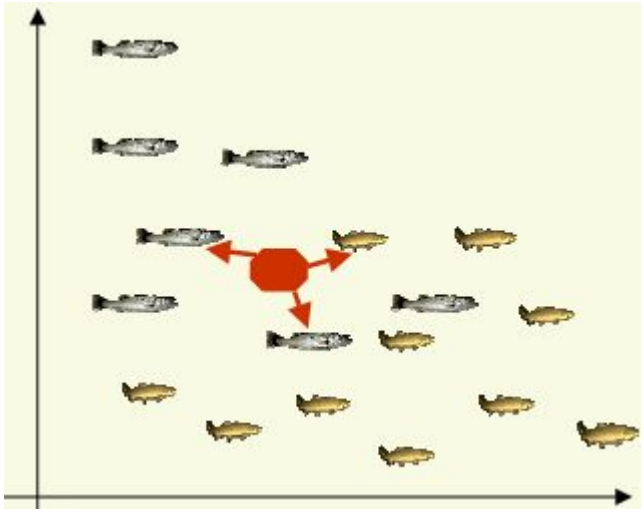
- Sendo assim, não precisamos estimar a densidade p(***x***).

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

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.

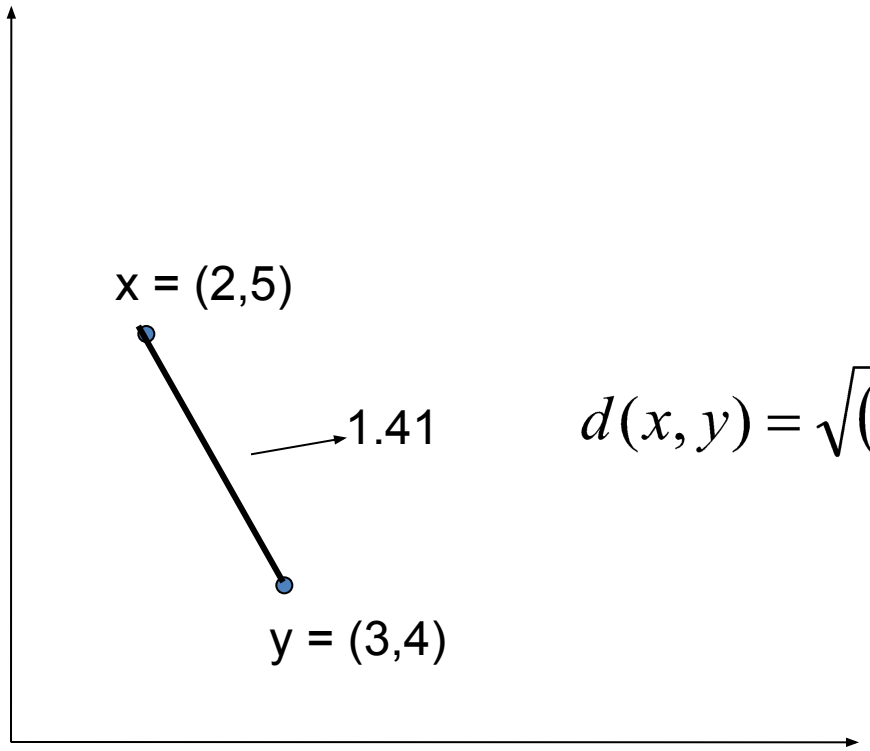
kNN

- 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:

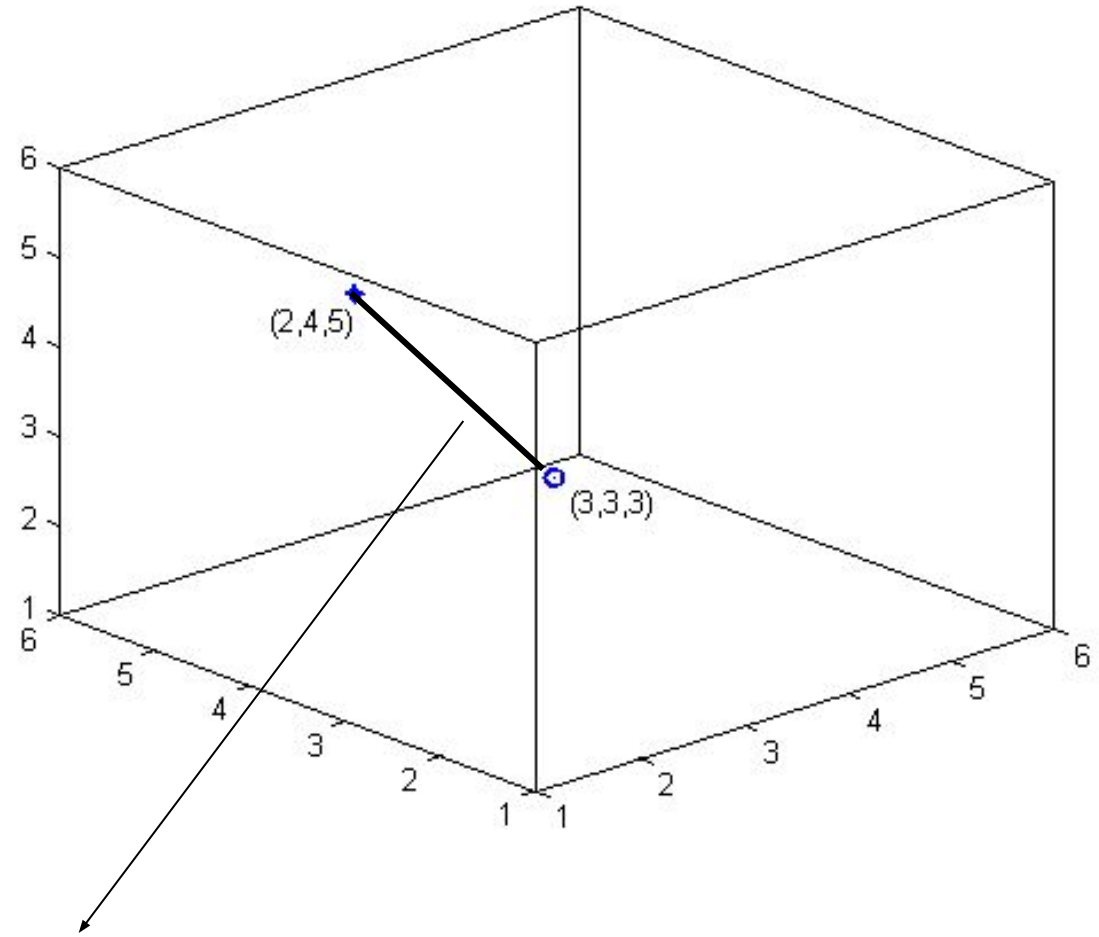
$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Distância Euclidiana



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

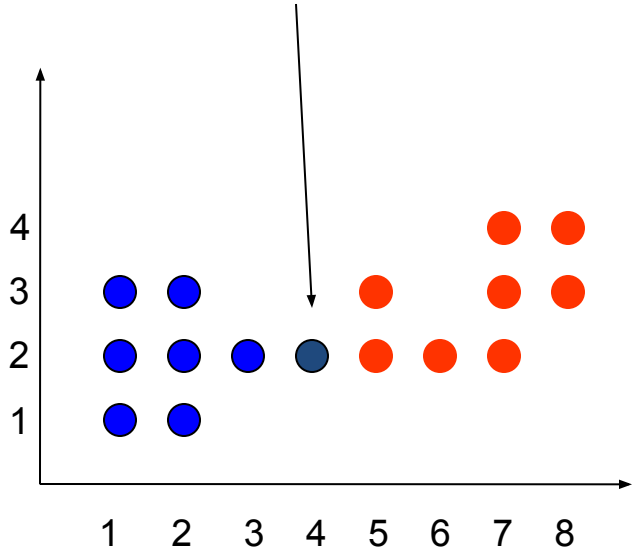
Distância Euclidiana



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

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

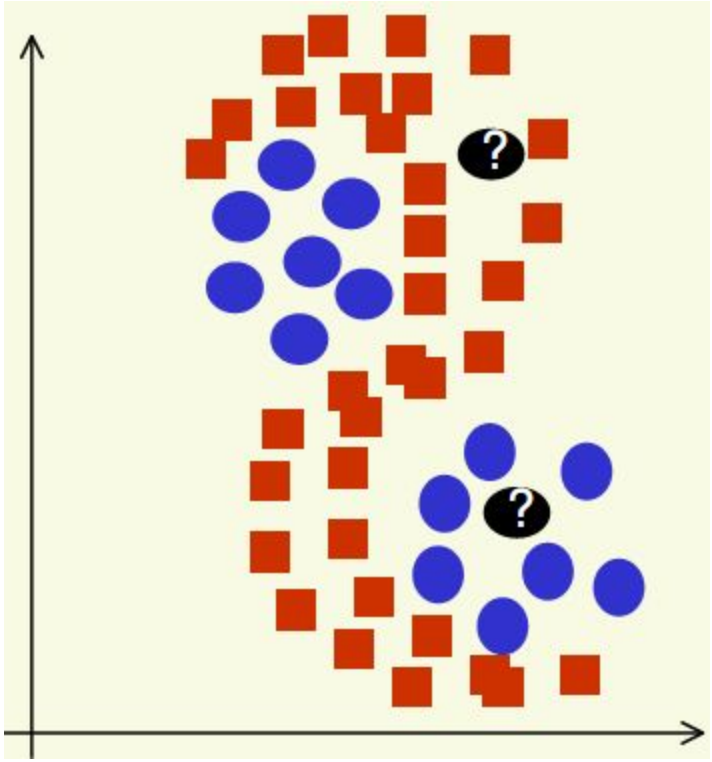
10 exemplos de C1 foram classificados como C2

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

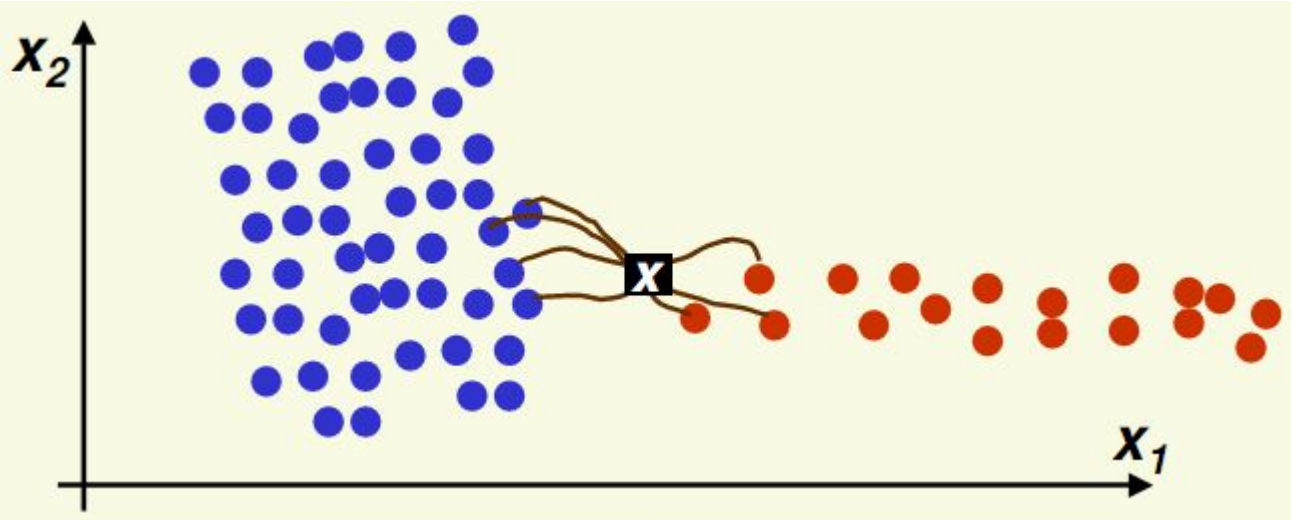
- Um caso complexo de classificação no qual o kNN tem sucesso.



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

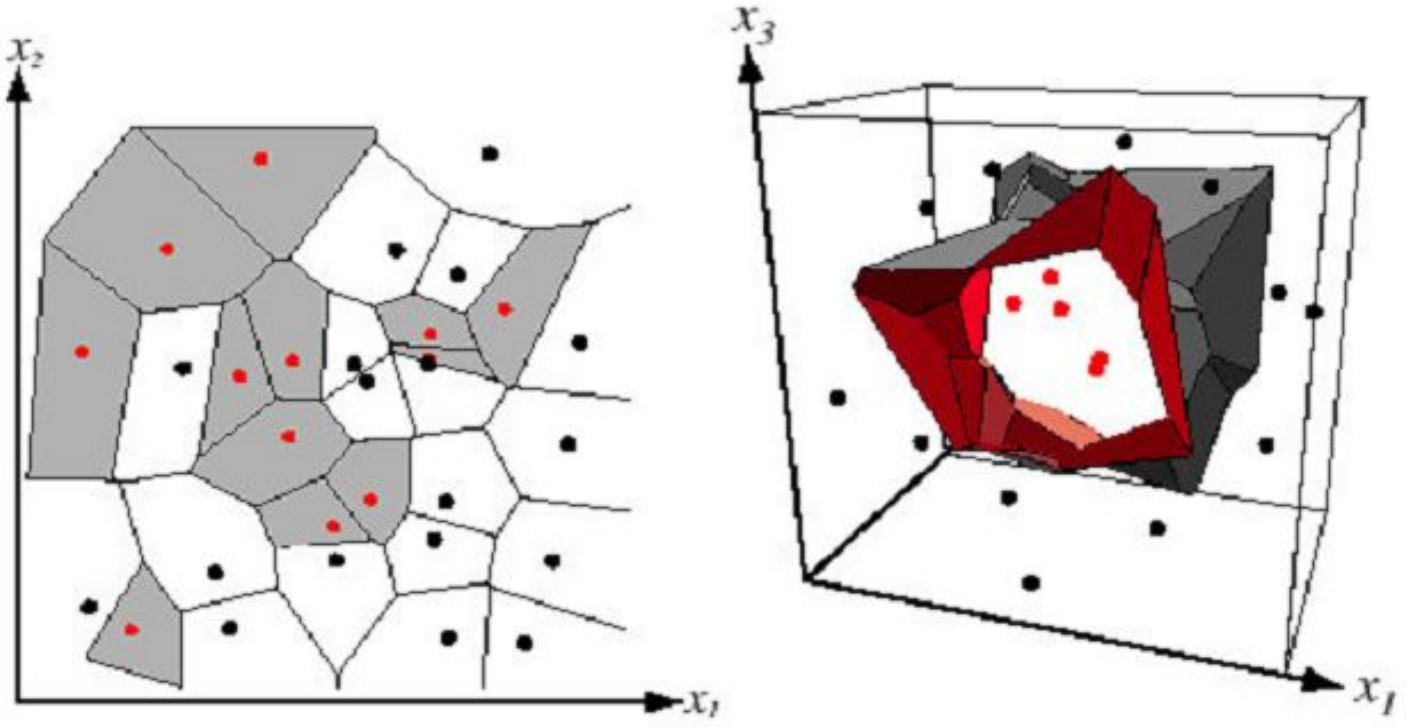
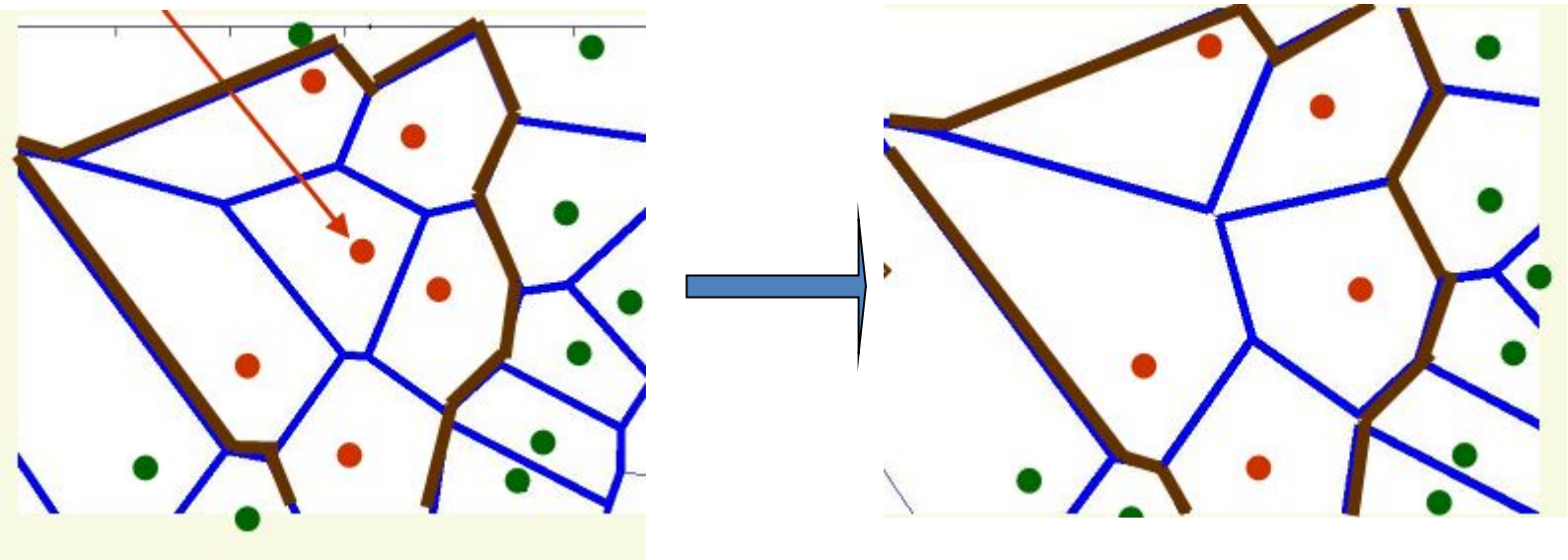


FIGURE 4.13. In two dimensions, the nearest-neighbor algorithm leads to a partitioning of the input space into Voronoi cells, each labeled by the category of the training point it contains. In three dimensions, the cells are three-dimensional, and the decision boundary resembles the surface of a crystal.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

kNN: Reduzindo complexidade

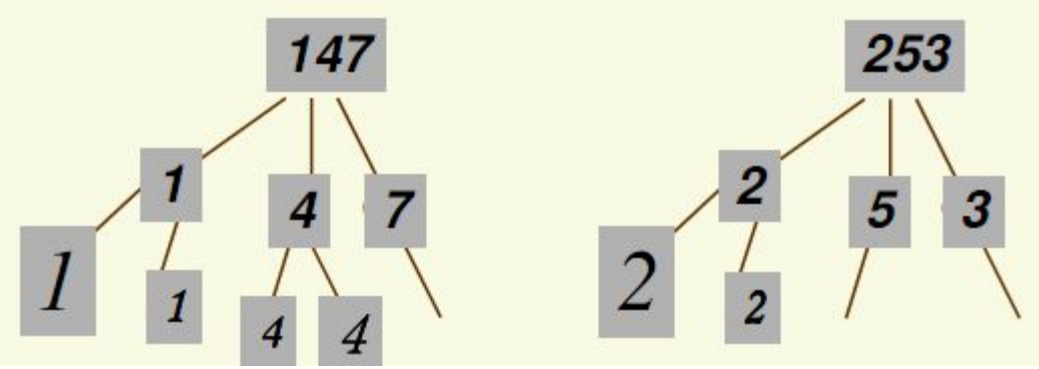
- Se uma **célula** dentro do diagrama de Voronoi possui os mesmos vizinhos, ela pode ser removida.



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

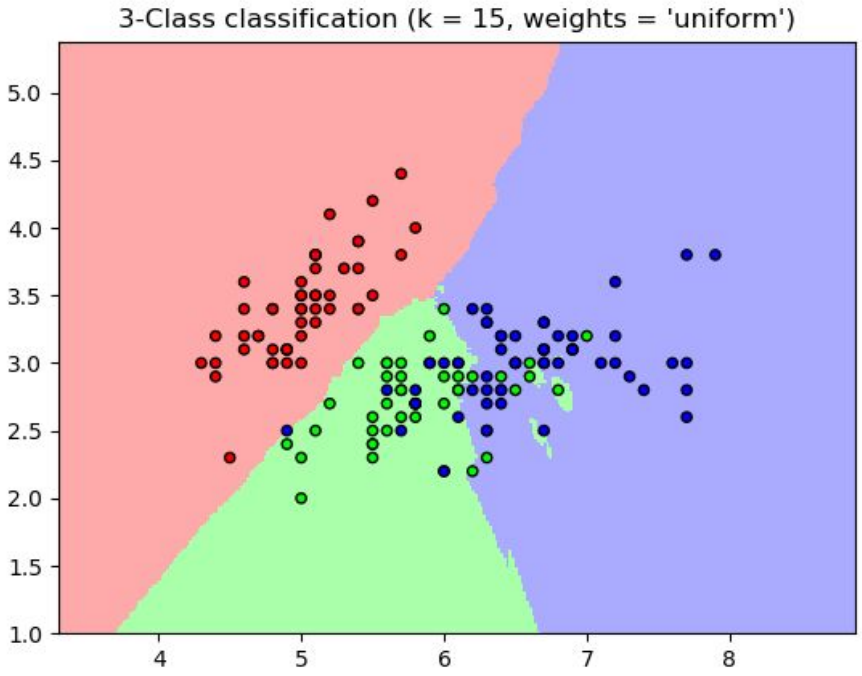
- Ponderar a contribuição de cada um dos ***k*** vizinhos de acordo com suas distâncias até o ponto x_t que queremos classificar, dando maior peso aos vizinhos mais próximos.
- Podemos ponderar o voto de cada vizinho, de acordo com o quadrado do inverso de sua distância de x_t.

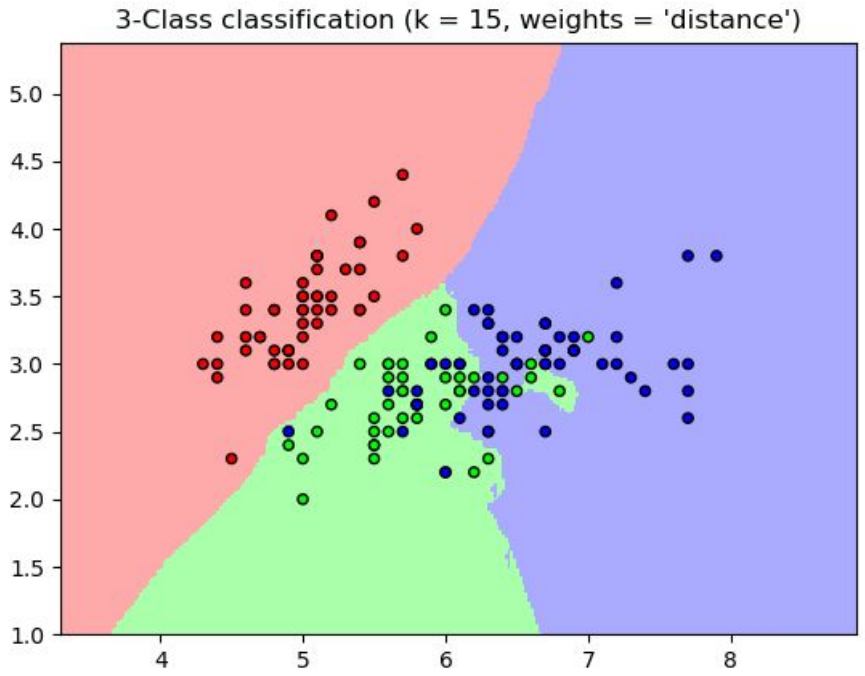
$$f(x_t) = \underset{c \in C}{argmax} \sum_i \omega_i \delta(c, f(x_i)) \qquad \omega_i = \frac{1}{d(x_t, x_i)^2}$$

- 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.

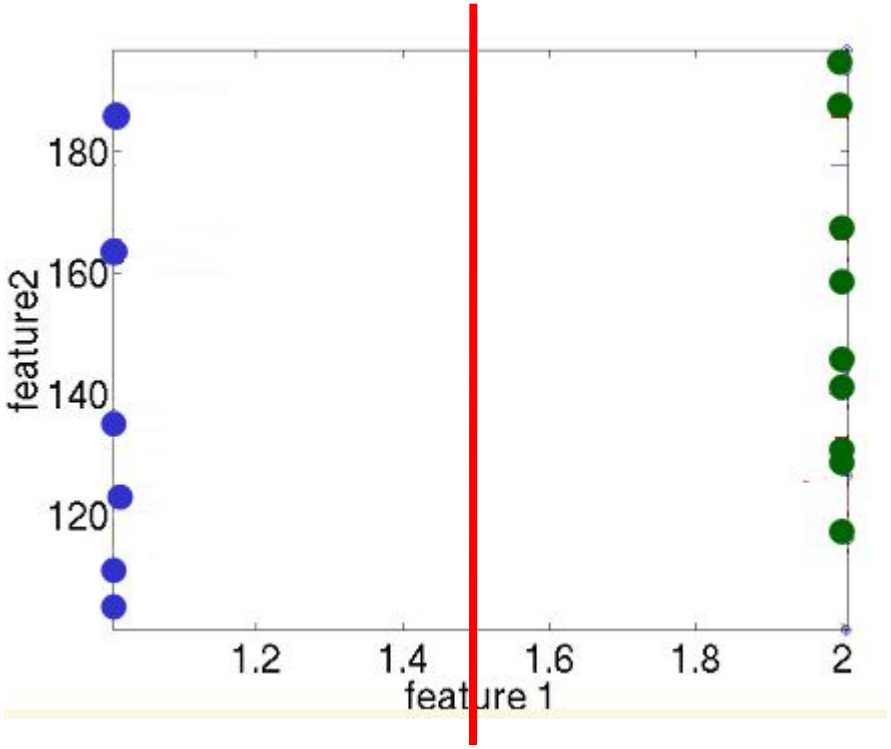
$$D(a,b) = \sqrt{\sum_k (a_k - b_k)^2}$$

- 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

- Luiz E. S Oliviera, **Árvores de Decisão**, DInf / UFPR, 2017.

- Luiz E. S Oliviera, **Aprendizagem Bayesiana**, DInf / UFPR, 2017.

- João Gama, **Árvores de Decisão**, notas de aula jgama@ncc.up.pt, 2012.