Classificação:
1R e Naïve Bayes

Eduardo Raul Hruschka

Agenda:

- Conceitos de Classificação
- Técnicas de Classificação
  - *One Rule* (1R)
  - Naive Bayes (com seleção de atributos)
  - Árvores de Decisão
  - K-Vizinhos Mais Próximos (K-NN)
- Super-ajuste e validação cruzada
- Combinação de Modelos

Baseado nos slides de Gregory Piatetsky-Shapiro, disponíveis no sítio http://www.kdnuggets.com

Visão Geral:

- Aquecimento: 1R;
- Naïve Bayes.

Inferindo regras rudimentares:

- 1R: aprende uma árvore de decisão de um nível.
  - Todas as regras usam somente um atributo.
- Versão Básica:
  - Um ramo para cada valor do atributo;
  - Para cada ramo, atribuir a classe mais freqüente;
  - Taxa de erro de classificação: proporção de exemplos que não pertencem à classe majoritária do ramo correspondente;
  - Escolher o atributo com a menor taxa de erro de classificação;

* Atributos nominais/categóricos;

- Há vários algoritmos de discretização para definir estratégias de corte nos valores dos atributos (<=, <, >, >=).

Algoritmo 1R em pseudo-código:

Para cada atributo:

Para cada valor do atributo gerar uma regra como segue:

Contar a freqüência de cada classe;

Encontrar a classe mais freqüente;

Formar uma regra que atribui à classe mais freqüente este atributo-valor;

Calcular a taxa de erro de classificação das regras;

Escolher as regras com a menor taxa de erro de classificação.

1R para o problema *weather* :

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

Attribute	Rules	Errors	Total errors
Outlook	Sunny → No	2/5	4/14
	Overcast → Yes	0/4	
	Rainy → Yes	2/5	
Temp	Hot → No*	2/4	5/14
	Mild → Yes	2/6	
	Cool → Yes	1/4	
Humidity	High → No	3/7	4/14
	Normal → Yes	1/7	
Windy	False → Yes	2/8	5/14
	True → No*	3/6	

* empate

Qual seria a capacidade de generalização do modelo?

Discussão para o 1R:

- 1R foi descrito por Holte (1993):
  - Contém uma avaliação experimental em 16 bases de dados;
  - Em muitos *benchmarks*, regras simples não são muito piores do que árvores de decisão mais complexas...
  - Complexidade de tempo?
- Implementado no Weka;
- Atualmente usado para análise exploratória de dados;
- Árvores de Decisão estendem essa ideia;
- Mas antes de abordá-las, abordaremos um algoritmo muito eficaz e (computacionalmente) eficiente: NB.

Holte, Robert C., Very Simple Classification Rules Perform Well on Most Commonly Used Datasets, *Machine Learning* 11 (1), pp. 63-90, 1993.

Modelagem Estatística (Bayesiana):

- Contrariamente ao 1R, Naive Bayes (NB) usa todos os atributos. Baseado em duas premissas:
  - Atributos igualmente importantes e condicionalmente independentes
    - o valor de um atributo não influencia no valor de outro atributo, dada a informação da classe;
- Na prática, tais premissas são frequentemente violadas, mas ainda assim o NB é muito competitivo:
  - Probabilidades estimadas não precisam necessariamente ser corretas, o que importa são as avaliações relativas.
- Parece ser consenso entre os mineradores de dados que, na prática, deve ser o primeiro algoritmo a usar.

Noção Intuitiva (base de dados "weather"):

Outlook	Temp	Humidity	Windy	Play
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

Desejamos estimar:

$$P(C_k \mid \mathbf{x}) = \frac{P(\mathbf{x} \mid C_k)P(C_k)}{P(\mathbf{x})}$$

- $P(C_k)$ pode ser estimada a partir da frequência relativa de classes;
- $P(\mathbf{x})$ é a *constante de normalização*:

$$P(\mathbf{x}) = \sum_k P(\mathbf{x} \mid C_k)P(C_k)$$

$\rightarrow$ Como estimar $P(\mathbf{x}/C_k)$?

$\rightarrow$ Assumindo independência condicional temos:

Frequências relativas:

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								
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								

Para um novo exemplo:

Outlook	Temp.	Humidity	Windy	Play
Sunny	Cool	High	True	?

Verosssimilhança para as duas classes:

Para "yes" = $2/9 \times 3/9 \times 3/9 \times 3/9 \times 9/14 = 0.0053$

Para "no" = $3/5 \times 1/5 \times 4/5 \times 3/5 \times 5/14 = 0.0206$

Convertendo para probabilidades por meio de normalização:

P("yes") = 0.0053 / (0.0053 + 0.0206) = 0.205

P("no") = 0.0206 / (0.0053 + 0.0206) = 0.795

Regra de Bayes:

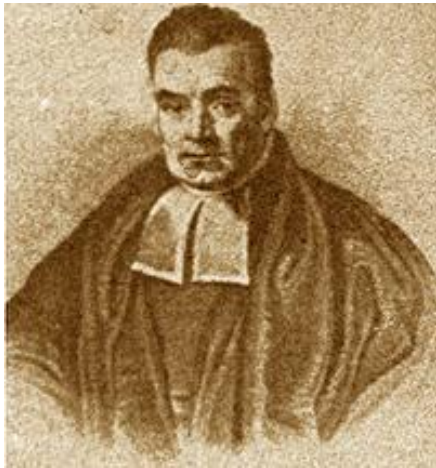
- Probabilidade de um evento *H* dada a evidência *E* :

$$\Pr[H \mid E] = \frac{\Pr[E \mid H]\Pr[H]}{\Pr[E]}$$

- Probabilidade *a priori* para H : $\Pr[H]$
  - Probabilidade de um evento antes de verificar a evidência
- Probabilidade *a posteriori* para H : $\Pr[H \mid E]$
  - Probabilidade de um evento após verificar a evidência

Thomas Bayes

(1702-1761)



Naïve Bayes para classificação:

- Qual é a probabilidade da classe dado um exemplo?
  - Evidência *E* = exemplo (valores dos atributos previsores);
  - Evento *H* = classe para um exemplo;
- Premissa *naïve*: evidência *dividida* em partes (i.e. atributos) independentes.

$$Pr[H \mid E] = \frac{Pr[E_1 \mid H]\, Pr[E_2 \mid H] \ldots Pr[E_n \mid H]\, Pr[H]}{Pr[E]}$$

Para o nosso exemplo:

Outlook	Temp.	Humidity	Windy	Play
Sunny	Cool	High	True	?

← ***Evidência E***

$$\Pr[yes \mid E] = \Pr[Outlook = Sunny \mid yes]$$
$$\times \Pr[Temperature = Cool \mid yes]$$
$$\times \Pr[Humidity = High \mid yes]$$
$$\times \Pr[Windy = True \mid yes]$$
$$\times \frac{\Pr[yes]}{\Pr[E]}$$

Probabilidade da classe "yes"

$$= \frac{\frac{2}{9} \times \frac{3}{9} \times \frac{3}{9} \times \frac{3}{9} \times \frac{9}{14}}{\Pr[E]}$$

Problema da freqüência *zero:*

- O que acontece se um determinado valor de atributo não aparece na base de treinamento, mas aparece no exemplo de teste?
  (e.g. "outlook=overcast" para classe "no")
  - Probabilidade correspondente será zero.
  - *Probabilidade a posteriori* será também zero.

- Possível solução: adicionar 1 ao contador para cada combinação de valor-classe (Estimador de *Laplace*). Como resultado, as probabilidades nunca serão *zero.*

Estimativas das probabilidades modificadas:

- No caso geral, pode-se adicionar uma constante μ diferente de 1;
- Exemplo: atributo *outlook* para a classe *yes:*

$\frac{2+\mu/3}{9+\mu}$	$\frac{4+\mu/3}{9+\mu}$	$\frac{3+\mu/3}{9+\mu}$
Sunny	***Overcast***	***Rainy***

Valores ausentes:

- Treinamento: excluir exemplo da base;
- Classificação: omitir atributo com valor ausente do cálculo;
- Exemplo:

Outlook	Temp.	Humidity	Windy	Play
?	Cool	High	True	?

Verossimilhança para "yes" = $3/9 \times 3/9 \times 3/9 \times 9/14 = 0.0238$

Verossimilhança para "no" = $1/5 \times 4/5 \times 3/5 \times 5/14 = 0.0343$

Chance ("yes") = 0.0238 / (0.0238 + 0.0343) = 41%

Chance ("no") = 0.0343 / (0.0238 + 0.0343) = 59%

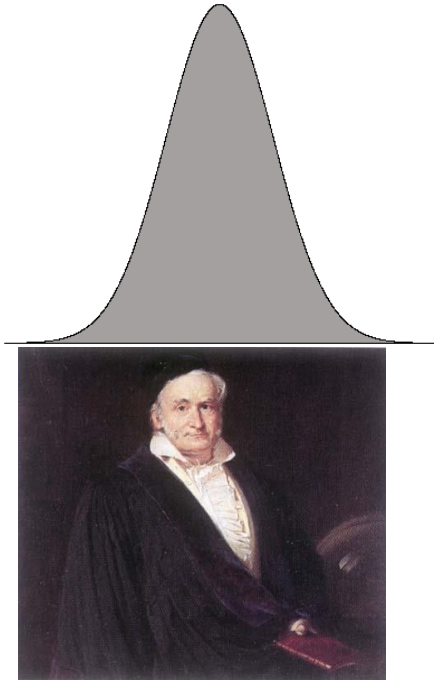
Atributos numéricos:

- Por exemplo, pode-se assumir uma distribuição Gaussiana para estimar as probabilidades:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\sigma = \frac{1}{n-1}\sum_{i=1}^{n} (x_i - \mu)^2$$

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



Karl Gauss, 1777-1855

Estatísticas para "weather":

Outlook			Temperature		Humidity		Windy			Play	
	Yes	*No*	*Yes*	*No*	*Yes*	*No*		*Yes*	*No*	*Yes*	*No*
Sunny	2	3	64, 68,	65, 71,	65, 70,	70, 85,	False	6	2	9	5
Overcast	4	0	69, 70,	72, 80,	70, 75,	90, 91,	True	3	3		
Rainy	3	2	72, …	85, …	80, …	95, …					
Sunny	2/9	3/5	μ =73	μ =75	μ =79	μ =86	False	6/9	2/5	9/14	5/14
Overcast	4/9	0/5	σ =6.2	σ =7.9	σ =10.2	σ =9.7	True	3/9	3/5		
Rainy	3/9	2/5									

- Valor de densidade:

$$f(temperature = 66 \mid yes) = \frac{1}{\sqrt{2\pi}\,6.2} e^{-\frac{(66-73)^2}{2*6.2^2}} = 0.0340$$

Discussão para Naïve Bayes:

- Naïve Bayes funciona bem mesmo quando suas premissas são violadas. Classificação não requer estimativas precisas da probabilidade, desde que a probabilidade máxima seja atribuída à classe correta (Domingos & Pazzani, On the Optimality of the Simple Bayesian Classifier under Zero-One Loss, Machine Learning 29, 103-130, 1997).
- Entretanto, a existência de muitos atributos redundantes pode causar problemas;
- Muitos atributos numéricos não seguem uma distribuição *Guassiana* ($\rightarrow$ *kernel density estimators*).

Extensões para o Naïve Bayes:

- Selecionar melhores atributos;
- Redes Bayesianas;

→ Abordaremos formas simples e eficazes de selecionar atributos:

  - Abordagens do tipo filtro;
  - Abordagens do tipo *wrapper*;
  - Abordagens inerentes ao classificador (e.g., 1R);
  - Abordagens híbridas.

Naive Bayes Wrapper (NBW):

- Atributos irrelevantes e redundantes podem comprometer acurácia de classificação;
- Selecionar atributos com base no desempenho do classificador NB. Informalmente pode-se sumarizar o NBW como segue:

1) Construir um classificador NB para cada atributo X_i ($i = 1,\ldots, n$). Escolher X_i para o qual o NB apresenta a melhor acurácia e inseri-lo em A_S = {atributos selecionados};
2) Para todo $X_i \notin A_S$ construir um NB formado por $\{X_i\} \cup A_S$. Escolher o melhor classificador dentre os disponíveis e verificar se é melhor do que o obtido anteriormente:
   a) SE sim, ENTÃO atualizar A_S, inserindo o atributo adicional e repetindo o passo 2);
   b) SE não, ENTÃO parar e usar o classificador obtido anteriormente.

Complexidade Computacional:

- NB possui complexidade de tempo linear com o número de exemplos e de atributos;
- Constante de tempo do NB também é baixa (computar frequências relativas e/ou densidades);
- Algoritmo NB é facilmente paralelizável;
- O que dizer sobre o NBW?
  - Teoria: $O(2^n)$, onde n é o número de atributos;
  - Busca gulosa *poda* o espaço de busca do problema de otimização combinatória: $O(n + (n-1) + ... + 1) = O(n^2)$
  - Por exemplo, para n=100 temos: 1.2×10^{30} versus 10^4 avaliações de classificadores diferentes para escolher o melhor.

Agenda:

- Conceitos de Classificação
- Técnicas de Classificação
  - *One Rule* (1R)
  - Naive Bayes (com seleção de atributos)
  - Árvores de Decisão
  - K-Vizinhos Mais Próximos (K-NN)
- Super-ajuste e validação cruzada
- Combinação de Modelos