Aprendizagem Bayesiana

Prof. Eric A. Antonelo

Programa de Pós-Graduação em Automação e Sistemas - UFSC

5 de setembro de 2014

Exemplo Diagnóstico Médico

Duas hipóteses: (1) paciente tem uma forma particular de câncer, e (2) o paciente não tem. Os dados do laboratório tem 2 possíveis resultados $\oplus$ (positivo) e $\ominus$ (negativo).
Conhecimento a priori de que a população tem a doença é 0.008. Além disso, o teste do laboratório é um indicador imperfeito da doença. O teste retorna um resultado positivo em somente 98% dos casos em que a doença realmente está presente e um resultado negativo correto em somente 97% dos casos em que a doença não está presente. Em outros casos, o teste retorna o resultado oposto

Exemplo Diagnóstico Médico - Probabilidades a Priori

$$P(cancer) = 0.008, P(\neg cancer) = 0.992$$

Exemplo Diagnóstico Médico - Probabilidades a Priori

$P(cancer) = 0.008, P(\neg cancer) = 0.992$
$P(\oplus|cancer) = 0.98, P(\oplus|cancer) = 0.02,$

Exemplo Diagnóstico Médico - Probabilidades a Priori

$P(cancer) = 0.008, P(\neg cancer) = 0.992$
$P(\oplus|cancer) = 0.98, P(\oplus|cancer) = 0.02,$
$P(\ominus|\neg cancer) = 0.03, P(\ominus|\neg cancer) = 0.97$

Exemplo Diagnóstico Médico - Probabilidades a Priori

$P(cancer) = 0.008, P(\neg cancer) = 0.992$
$P(\oplus|cancer) = 0.98, P(\oplus|cancer) = 0.02,$
$P(\ominus|\neg cancer) = 0.03, P(\ominus|\neg cancer) = 0.97$

Suponha agora que observamos um novo paciente para o qual o teste de laboratório retorne um resultado positivo. O diagnóstico deve ser de câncer?

Exemplo Diagnóstico Médico - MAP

$$P(\oplus|cancer) * P(cancer) = (0.98)0.008 = 0.0078$$

Exemplo Diagnóstico Médico - MAP

$$P(\oplus|cancer) * P(cancer) = (0.98)0.008 = 0.0078$$
$$P(\oplus|\neg cancer) * P(\neg cancer) = (0.03)0.992 = 0.0298$$

Exemplo Diagnóstico Médico - MAP

$$P(\oplus|cancer) * P(cancer) = (0.98)0.008 = 0.0078$$
$$P(\oplus|\neg cancer) * P(\neg cancer) = (0.03)0.992 = 0.0298$$
$$h_{MAP} = \neg cancer$$

Exemplo Diagnóstico Médico - MAP

$P(\oplus|cancer) * P(cancer) = (0.98)0.008 = 0.0078$
$P(\oplus|\neg cancer) * P(\neg cancer) = (0.03)0.992 = 0.0298$
$h_{MAP} = \neg cancer$

Como determinar as probabilidades a posteriori? Ex.: $P(cancer|\oplus)$

Exemplo Diagnóstico Médico - MAP

- Como o exemplo ilustra, a aprendizagem Bayesiana, depende muito das probabilidades a priori.

Exemplo Diagnóstico Médico - MAP

- Como o exemplo ilustra, a aprendizagem Bayesiana, depende muito das probabilidades a priori.
- As hipóteses não são completamente aceitas ou rejeitadas, mas tornam-se mais prováveis ou menos prováveis à medida que mais dados são observados.

Exemplo Diagnóstico Médico - MAP

- Como o exemplo ilustra, a aprendizagem Bayesiana, depende muito das probabilidades a priori.
- As hipóteses não são completamente aceitas ou rejeitadas, mas tornam-se mais prováveis ou menos prováveis à medida que mais dados são observados.
- Calcule a $P(cancer|\oplus_2)$, onde observamos pela segunda vez o teste positivo. Qual a h_{MAP} hipótese mais provável nesse caso?

Fórmulas básicas de probabilidades

- Regra do Produto

$$P(A \wedge B) = P(A/B)P(B) = P(B/A)P(A)$$

- Regra da Soma

$$P(A \vee B) = P(A) + P(B) - P(A \wedge B)$$

- Regra de Bayes

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

- Marginalização (ou Teorema da probabilidade total)

$$P(A) = \sum_{i=1}^{n} P(A \wedge B_i) = \sum_{i=1}^{n} P(A/B_i)P(B_i)$$

Aprendizagem Bayes de conceitos por força bruta

Conjunto de treinamento $\langle\langle x_1, d_1\rangle, ..., \langle x_m, d_m\rangle\rangle$,
$d_i = c(x_i)$,
onde $c : X \rightarrow \{0, 1\}$ é a função desejada a ser *aprendida*.

Aprendizagem Bayes de conceitos por força bruta

Conjunto de treinamento $\langle\langle x_1, d_1\rangle, ..., \langle x_m, d_m\rangle\rangle$,
$d_i = c(x_i)$,
onde $c : X \rightarrow \{0, 1\}$ é a função desejada a ser *aprendida*.
Algoritmo:

1. Para cada hipótese h em H, calcule a probabilidade posterior

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

Aprendizagem Bayes de conceitos por força bruta

Conjunto de treinamento $\langle\langle x_1, d_1\rangle, ..., \langle x_m, d_m\rangle\rangle$,
$d_i = c(x_i)$,
onde $c : X \to \{0, 1\}$ é a função desejada a ser *aprendida*.
Algoritmo:

1. Para cada hipótese h em H, calcule a probabilidade posterior

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

2. Retorne a hipótese h_{MAP} em H com a probabilidade posterior mais alta

$$h_{MAP} = \arg\max_{h \in H} P(h|D)$$

Aprendizagem Bayes de conceitos por força bruta

Conjunto de treinamento $\langle\langle x_1, d_1\rangle, ..., \langle x_m, d_m\rangle\rangle$,
$d_i = c(x_i)$,
onde $c : X \to \{0, 1\}$ é a função desejada a ser *aprendida*.
Algoritmo:

1. Para cada hipótese h em H, calcule a probabilidade posterior

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

2. Retorne a hipótese h_{MAP} em H com a probabilidade posterior mais alta

$$h_{MAP} = \arg\max_{h \in H} P(h|D)$$

O que é D ?

Aprendizagem Bayes de conceitos por força bruta

Conjunto de treinamento $\langle\langle x_1, d_1\rangle, ..., \langle x_m, d_m\rangle\rangle$,
$d_i = c(x_i)$,
onde $c : X \to \{0, 1\}$ é a função desejada a ser *aprendida*.
Algoritmo:

1. Para cada hipótese h em H, calcule a probabilidade posterior

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

2. Retorne a hipótese h_{MAP} em H com a probabilidade posterior mais alta

$$h_{MAP} = \arg\max_{h \in H} P(h|D)$$

O que é D ?
$D = \langle d_1, ..., d_m\rangle$ (considerando $\langle x_1, ..., x_m\rangle$ fixo, para simplificar)

Aprendizagem Bayes de conceitos por força bruta

Algoritmo Bayes:

1. Passo 1

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

2. Passo 2

$$h_{MAP} = \arg\max_{h \in H} P(h|D)$$

Aprendizagem Bayes de conceitos por força bruta

Algoritmo Bayes:

1. Passo 1

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

2. Passo 2

$$h_{MAP} = \arg\max_{h \in H} P(h|D)$$

Questões:

- Se o espaço de hipóteses H é grande?
- Temos de determinar $P(D|h), P(h)$ e $P(D)$

Relação com Aprendizagem de Conceitos

Assuma um conjunto fixo de instâncias $\langle x_1, \ldots, x_m \rangle$
Assuma D é o conjunto de classificações $D = \langle c(x_1), \ldots, c(x_m) \rangle$
Escolha $P(D|h)$

- $P(D|h) = 1$ se h é consistente com D
- $P(D|h) = 0$ caso contrário

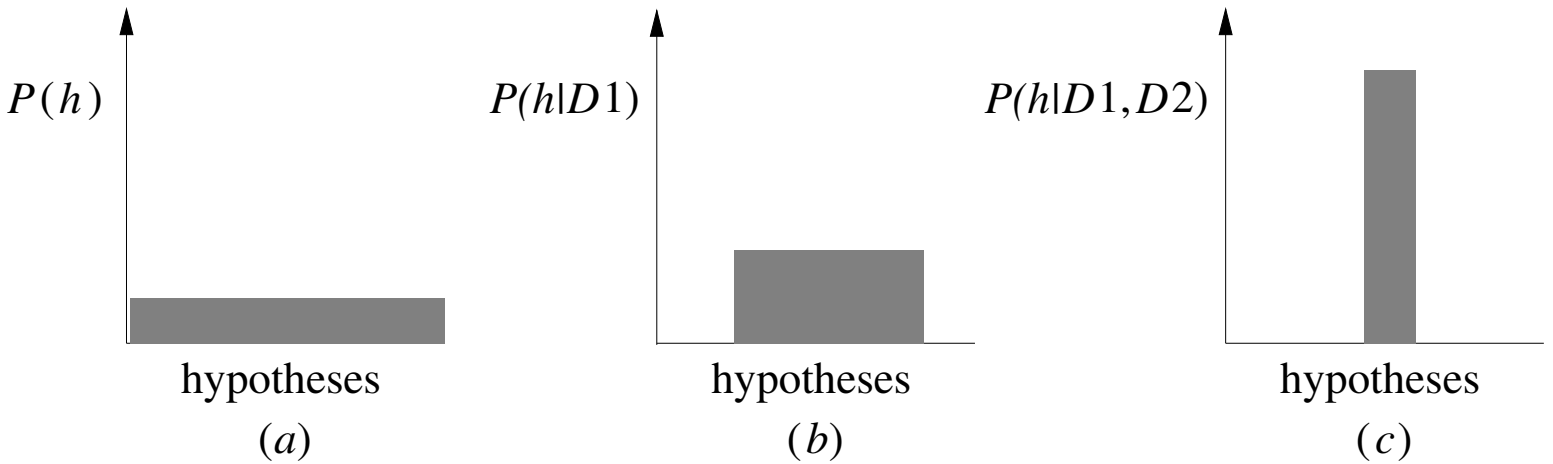
Escolha para $P(h)$ a distribuição *uniforme*

- $P(h) = \frac{1}{|H|}$ para todo h em H

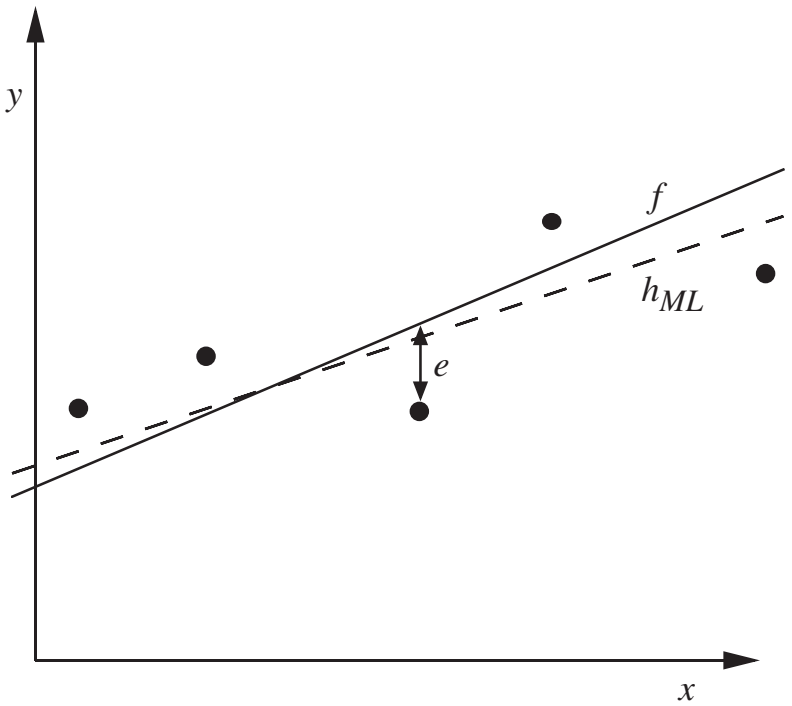
Então,

$$P(h|D) = \begin{cases} \frac{1}{|VS_{H,D}|} & \text{se } h \text{ é consistente com } D \\ 0 & \text{caso contrário} \end{cases}$$

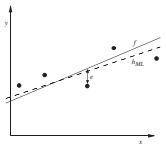
Evolução de probabilidades posteriores



Aprendendo uma função linear real



Aprendendo uma função linear real



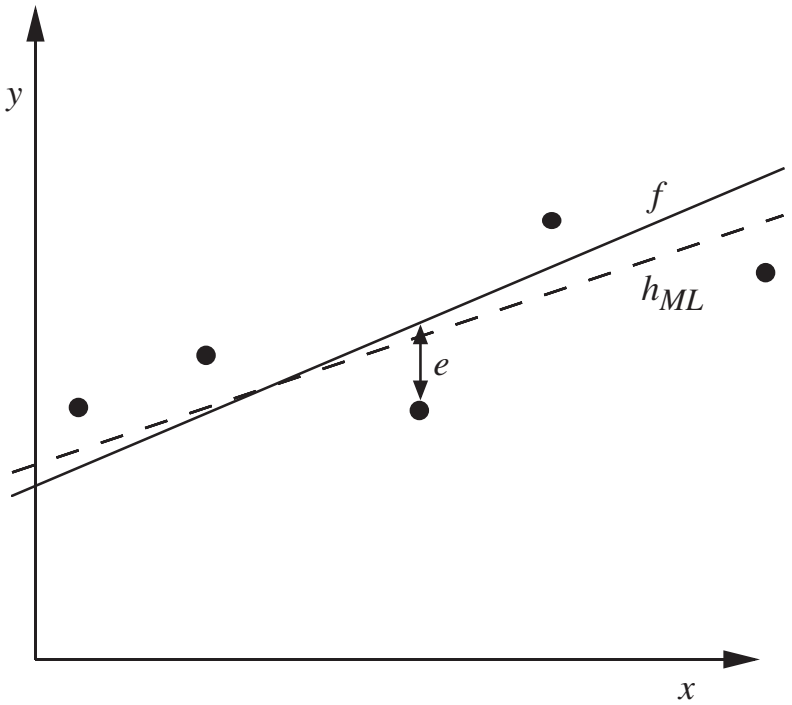
Considere uma função *target* (desejada) real f
Exemplos de treinamento $\langle x_i, d_i \rangle$, onde d_i é um valor de treinamento com ruído

- $d_i = f(x_i) + e_i$
- e_i é uma variável aleatória (ruído) instancianda para cada x_i de modo independente de acordo com uma distribuição Gaussiana com média=0

Então, a hipótese de máximo likelihood h_{ML} é a que minimiza a soma dos erros quadráticos:

$$h_{ML} = \arg\min_{h \in H} \sum_{i=1}^{m} (d_i - h(x_i))^2$$

Aprendendo uma função linear real



Então, a hipótese de máximo likelihood h_{ML} é a que minimiza a soma dos erros quadráticos:

$$h_{ML} = \arg\min_{h \in H} \sum_{i=1}^{m} (d_i - h(x_i))^2$$

Aprendendo a estimar probabilidades

Considere estimar a probabilidade de sobrevivência a partir de dados de pacientes

- Exemplos de treinamento $\langle x_i, d_i \rangle$, onde d_i é 1 ou 0
- Desejamos treinar uma rede neural para ter como saída uma *probabilidade* dado x_i (e não 0 ou 1)

Neste caso, podemos mostrar

$$h_{ML} = \arg\max_{h \in H} \sum_{i=1}^{m} d_i \ln h(x_i) + (1 - d_i) \ln(1 - h(x_i))$$

Regra de atualização para os pesos de uma unidade sigmóide:

$$w_{jk} \leftarrow w_{jk} + \Delta w_{jk}$$

onde

$$\Delta w_{jk} = \eta \sum_{i=1}^{m} (d_i - h(x_i))\ x_{ijk}$$

Princípio do Comprimento Mínimo de Descrição (MDL)

Navalha de Occam: preferir a hipótese mais curta

O **princípio MDL** é motivado interpretando a h_{MAP} sob a luz da teoria da informação.

$$\begin{aligned} h_{MAP} &= \arg\max_{h \in H} P(D|h)P(h) \\ &= \arg\max_{h \in H} \log_2 P(D|h) + \log_2 P(h) \\ &= \arg\min_{h \in H} -\log_2 P(D|h) - \log_2 P(h) \end{aligned}$$

Considere o problema de projetar um código para transmitir mensagens sorteadas aleatoriamente, onde a probabilidade de encontrar a mensagem i é p_i. Estamos interessados no código que minimiza o número esperado de bits para transmitir tal mensagem (códigos mais curtos devem ser atribuídos a mensagens mais prováveis).

Princípio do Comprimento Mínimo de Descrição (MDL)

Fato interessante da **teoria da informação**:

> *O código ótimo (comprimento de código mais curto esperado) para um evento com probabilidade p é $-\log_2 p$* ***bits***.

$$\begin{aligned} h_{MAP} &= \arg\max_{h \in H} P(D|h)P(h) \\ &= \arg\max_{h \in H} \log_2 P(D|h) + \log_2 P(h) \\ &= \arg\min_{h \in H} -\log_2 P(D|h) - \log_2 P(h) \qquad (1) \end{aligned}$$

Então, interpretamos (1):

- $-\log_2 P(h)$ é o comprimento de h sob o código ótimo
- $-\log_2 P(D|h)$ é o comprimento de D dado h sob o código ótimo

$\rightarrow$ preferimos a hipótese que minimiza

$$comprimento(h) + comprimento(\text{classificações erradas})$$

Princípio do Comprimento Mínimo de Descrição (MDL)

Princípio MDL:preferir a hipótese h que minimiza (onde a equação (1) pode ser re-escrita):

$$h_{MDL} = \arg\min_{h \in H} L_{C_1}(h) + L_{C_2}(D|h)$$

onde $L_C(x)$ é o comprimento de descrição de x sob a codificação C (C_1, C_2 são as codificações para H, e $D|h$, resp.)

Exemplo: H = árvores de decisão, D = valores desejados de treinamento

- $L_{C_1}(h)$ é # bits para descrever a árvore h
- $L_{C_2}(D|h)$ é # bits para descrever D dado h
  - Note que $L_{C_2}(D|h) = 0$ se exemplos classificados perfeitamente por h. Precisa somente descrever as exceções
- Assim h_{MDL} faz um compromisso entre tamanho da árvore e erro de treinamento

Classificação mais Provável de Novas Instâncias

Até agora, buscamos a *hipótese* mais provável dado o conjunto de dados D (i.e., h_{MAP})

Dado uma nova instância x, qual é sua *classificação* mais provável?

- $h_{MAP}(x)$ não é a classificação mais provável!

Considere:

- Três possíveis hipóteses:

$$P(h_1|D) = .4,\ P(h_2|D) = .3,\ P(h_3|D) = .3$$

- Dado uma nova instância x,

$$h_1(x) = +,\ h_2(x) = -,\ h_3(x) = -$$

- Qual é a classificação mais provável de x?

Classificador de Bayes Ótimo

Classificação Ótima de Bayes:

$$\arg\max_{v_j \in V} \sum_{h_i \in H} P(v_j|h_i)P(h_i|D)$$

Exemplo:

Classificação Ótima de Bayes:

$$\arg\max_{v_j \in V} \sum_{h_i \in H} P(v_j|h_i)P(h_i|D)$$

Exemplo:

$$\begin{aligned} P(h_1|D) &= .4, \quad P(-|h_1) = 0, \quad P(+|h_1) = 1 \\ P(h_2|D) &= .3, \quad P(-|h_2) = 1, \quad P(+|h_2) = 0 \\ P(h_3|D) &= .3, \quad P(-|h_3) = 1, \quad P(+|h_3) = 0 \end{aligned}$$

portanto

$$\begin{aligned} \sum_{h_i \in H} P(+|h_i)P(h_i|D) &= .4 \\ \sum_{h_i \in H} P(-|h_i)P(h_i|D) &= .6 \end{aligned}$$

Classificação Ótima de Bayes:

Exemplo:

$$P(h_1|D) = .4, \quad P(-|h_1) = 0, \quad P(+|h_1) = 1$$
$$P(h_2|D) = .3, \quad P(-|h_2) = 1, \quad P(+|h_2) = 0$$
$$P(h_3|D) = .3, \quad P(-|h_3) = 1, \quad P(+|h_3) = 0$$

portanto

$$\sum_{h_i \in H} P(+|h_i)P(h_i|D) = .4$$
$$\sum_{h_i \in H} P(-|h_i)P(h_i|D) = .6$$

e

$$\arg\max_{v_j \in V} \sum_{h_i \in H} P(v_j|h_i)P(h_i|D) = -$$

Classificador Gibbs

O classificador ótimo de Bayes fornece o melhor resultado, mas pode ser caro se há muitas hipóteses.
Algoritmo de Gibbs:

1. Escolha uma hipótese aleatoriamente, de acordo com $P(h|D)$
2. Use esta hipótese para classificar a nova instância

Fato surpreendente: Assuma que as classes (conceitos) são sorteadas aleatoriamente de acordo com as probabilidades a priori dadas. Então:

$$E[error_{Gibbs}] \leq 2E[error_{BayesOptimal}]$$

Se temos uma distribuição a priori uniforme correta sobre H, então

- Escolha qualquer hipótese de VS, com probabilidade uniforme
- O erro esperado é no máximo duas vezes o erro do Bayes ótimo

Classificador Naive Bayes

Junto com árvores de decisão, redes neurais, vizinho mais próximo (nearest nbr), um dos métodos de aprendizagem mais práticos.

Quando usar

- Conjunto de treinamento disponível de tamanho moderado a grande
- Atributos que descrevem instâncias são condicionalmente independentes dado a classificação (classe)

Aplicações de sucesso:

- Diagnóstico
- Classificação de documentos de texto

Classificador Naive Bayes

Assuma a função 'target' $f : X \rightarrow V$, onde cada instância x é descrita por atributos $\langle a_1, a_2 \ldots a_n \rangle$.
O valor mais provável de $f(x)$ é:

$$\begin{aligned} v_{MAP} &= \arg\max_{v_j \in V} P(v_j | a_1, a_2 \ldots a_n) \\ v_{MAP} &= \arg\max_{v_j \in V} \frac{P(a_1, a_2 \ldots a_n | v_j) P(v_j)}{P(a_1, a_2 \ldots a_n)} \\ &= \arg\max_{v_j \in V} P(a_1, a_2 \ldots a_n | v_j) P(v_j) \end{aligned}$$

Naive Bayes assume:

$$P(a_1, a_2 \ldots a_n | v_j) = \prod_i P(a_i | v_j)$$

que resulta:

Classificador Naive Bayes: $v_{NB} = \arg\max_{v_j \in V} P(v_j) \prod_i P(a_i | v_j)$

Algoritmo Naive Bayes

Naive_Bayes_Learn(*examples*)

Para cada classe v_j

$\hat{P}(v_j) \leftarrow$ estimativa de $P(v_j)$

Para cada valor de atributo a_i dado v_j

$\hat{P}(a_i|v_j) \leftarrow$ estimativa de $P(a_i|v_j)$

Classify_New_Instance(x)

$$v_{NB} = \arg\max_{v_j \in V} \hat{P}(v_j) \prod_{a_i \in x} \hat{P}(a_i|v_j)$$

Naive Bayes: Exemplo

Considere o problema *PlayTennis* novamente, e uma nova instância

$$\langle Outlk = sun, Temp = cool, Humid = high, Wind = strong \rangle$$

Desejamos calcular:

$$v_{NB} = \arg\max_{v_j \in V} P(v_j) \prod_i P(a_i|v_j)$$

Devemos estimar:

$$P(y)\ P(sun|y)\ P(cool|y)\ P(high|y)\ P(strong|y)$$
$$P(n)\ P(sun|n)\ P(cool|n)\ P(high|n)\ P(strong|n)$$

Tabela: Exemplos de treinamento

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

Naive Bayes: Exemplo

Nova instância:

$$\langle Outlk = sun, Temp = cool, Humid = high, Wind = strong \rangle$$

Desejamos calcular:

$$v_{NB} = \arg\max_{v_j \in V} P(v_j) \prod_i P(a_i|v_j)$$

$$P(y)\ P(sun|y)\ P(cool|y)\ P(high|y)\ P(strong|y) = .005$$
$$P(n)\ P(sun|n)\ P(cool|n)\ P(high|n)\ P(strong|n) = .021$$

Naive Bayes: Exemplo

Nova instância:

$$\langle Outlk = sun, Temp = cool, Humid = high, Wind = strong \rangle$$

Desejamos calcular:

$$v_{NB} = \arg\max_{v_j \in V} P(v_j) \prod_i P(a_i|v_j)$$

$$P(y)\ P(sun|y)\ P(cool|y)\ P(high|y)\ P(strong|y) = .005$$
$$P(n)\ P(sun|n)\ P(cool|n)\ P(high|n)\ P(strong|n) = .021$$

$$\rightarrow v_{NB} = n$$

Naive Bayes: Sutilezas

1. A suposição de independência condicional é frequentemente violada:

$$P(a_1, a_2 \ldots a_n | v_j) = \prod_i P(a_i | v_j)$$

   - ...mas funciona surpreendentemente bem de qualquer modo. Note que não precisamos estimar as probabilidades posteriores $\hat{P}(v_j|x)$; precisamos somente

$$\arg\max_{v_j \in V} \hat{P}(v_j) \prod_i \hat{P}(a_i|v_j) = \arg\max_{v_j \in V} P(v_j) P(a_1 \ldots, a_n | v_j)$$

   - veja [Domingos & Pazzani, 1996] para uma análise
   - as prob. posteriores do Naive Bayes tem valores muitas vezes irrealisticamente perto de 1 ou 0

2. E se nenhuma das instâncias de treinamento de classe v_j tiverem o valor de atributo a_i? Então

$$\hat{P}(a_i|v_j) = 0, \text{ e...}$$

$$\hat{P}(v_j) \prod_i \hat{P}(a_i|v_j) = 0$$

Solução típica é uma estimativa Bayesiana para $\hat{P}(a_i|v_j)$

$$\hat{P}(a_i|v_j) \leftarrow \frac{n_c + mp}{n + m}$$

onde

- n é o número de exemplos de treinamento para o qual $v = v_j$,
- n_c número de exemplos para o qual $v = v_j$ e $a = a_i$
- p é a estimativa a priori para $\hat{P}(a_i|v_j)$
- m é o peso dado para o 'prior' (i.e. número de exemplos "virtuais")

Aprendendo a classificar texto

Por quê?

- Aprender que notícias são interessantes
- Aprender a classificar páginas web por tópico

Naive Bayes está entre os algoritmos mais efetivos

Quais atributos devemos usar para representar documentos de texto??

Aprendendo a classificar texto

Conceito desejado (Classe) *Interesting*? : *Document* $\rightarrow \{+, -\}$

1. Representar cada documento por um vetor de palavras
   - um atributo por posição de palavra no documento
2. Aprendizagem: Usar os exemplos de treinamento para estimar
   - $P(+)$
   - $P(-)$
   - $P(doc|+)$
   - $P(doc|-)$

Suposição de independência condicional Naive Bayes

Aprendendo a classificar texto

Conceito desejado (Classe) *Interesting*? : $Document \rightarrow \{+, -\}$

1. Representar cada documento por um vetor de palavras
   - um atributo por posição de palavra no documento
2. Aprendizagem: Usar os exemplos de treinamento para estimar
   - $P(+)$
   - $P(-)$
   - $P(doc|+)$
   - $P(doc|-)$

Suposição de independência condicional Naive Bayes

$$P(doc|v_j) = \prod_{i=1}^{length(doc)} P(a_i = w_k|v_j)$$

onde $P(a_i = w_k|v_j)$ é a probabilidade que a palavra na posição i é w_k, dado v_j
uma suposição a mais: $P(a_i = w_k|v_j) = P(a_m = w_k|v_j), \forall i, m$

Aprendendo a classificar texto

Learn_naive_Bayes_text(*Examples*, V)

1. *Colete todas palavras e outros tokens que ocorrem em Examples*

- *Vocabulary* $\leftarrow$ todas palavras e tokens distintos em *Examples*

2. *Calcular as probabilidades requeridas* $P(v_j)$ *e* $P(w_k|v_j)$

- Para cada classe v_j em V fazer
  - $docs_j \leftarrow$ subconjunto de *Examples* com classe v_j
  - $P(v_j) \leftarrow \frac{|docs_j|}{|Examples|}$
  - $Text_j \leftarrow$ um documento único criado concatenando todos membros de $docs_j$
  - $n \leftarrow$ número total de palavras em $Text_j$ (contando palavras duplicadas múltiplas vezes)
  - para cada palavra w_k no *Vocabulary*
    - $n_k \leftarrow$ número de vezes a palavra w_k ocorre em $Text_j$
    - $P(w_k|v_j) \leftarrow \frac{n_k+1}{n+|Vocabulary|}$

Aprendendo a classificar texto

Classify_naive_Bayes_text(*Doc*)

- *positions* $\leftarrow$ todas posições de palavras em *Doc* que contém tokens encontrados em *Vocabulary*
- Retornar v_{NB}, onde

$$v_{NB} = \arg\max_{v_j \in V} P(v_j) \prod_{i \in positions} P(a_i|v_j)$$

Vinte NewsGroups

Dados 1000 documentos de treinamento para cada grupo
Aprender a classificar novos documentos de acordo a qual newsgroup eles pertecem

comp.graphics	misc.forsale
comp.os.ms-windows.misc	rec.autos
comp.sys.ibm.pc.hardware	rec.motorcycles
comp.sys.mac.hardware	rec.sport.baseball
comp.windows.x	rec.sport.hockey
alt.atheism	sci.space
soc.religion.christian	sci.crypt
talk.religion.misc	sci.electronics
talk.politics.mideast	sci.med
talk.politics.misc	
talk.politics.guns	

Naive Bayes: 89% acurácia de classificação

Artigo de rec.sport.hockey

Path: cantaloupe.srv.cs.cmu.edu!das-news.harvard.edu!ogicse!uwm.edu From: xxx@yyy.zzz.edu (John Doe) Subject: Re: This year's biggest and worst (opinion)... Date: 5 Apr 93 09:53:39 GMT

I can only comment on the Kings, but the most obvious candidate for pleasant surprise is Alex Zhitnik. He came highly touted as a defensive defenseman, but he's clearly much more than that. Great skater and hard shot (though wish he were more accurate). In fact, he pretty much allowed the Kings to trade away that huge defensive liability Paul Coffey. Kelly Hrudey is only the biggest disappointment if you thought he was any good to begin with. But, at best, he's only a mediocre goaltender. A better choice would be Tomas Sandstrom, though not through any fault of his own, but because some thugs in Toronto decided

Redes Bayesianas (de Crença)

Interessante porque:

- A suposição de Naive Bayes de independência condicional muito restritiva
- Mas sem tais suposições, o cálculo ou inferência são intratáveis
- Redes Bayesianas descrevem independências condicionais entre *subconjuntos* de variáveis

$\rightarrow$ permite combinar conhecimento a priori sobre (in)dependências entre variáveis com dados de treinamento observados

Independência Condicional

Definição: *X é* condicionalmente independente *de Y dado Z se a distribuição de probabilidade que governa X é independente do valor de Y dado o valor de Z; isto é, se*

$$(\forall x_i, y_j, z_k)\ P(X = x_i | Y = y_j, Z = z_k) = P(X = x_i | Z = z_k)$$

mais compactamente, escrevemos:

$$P(X|Y,Z) = P(X|Z)$$

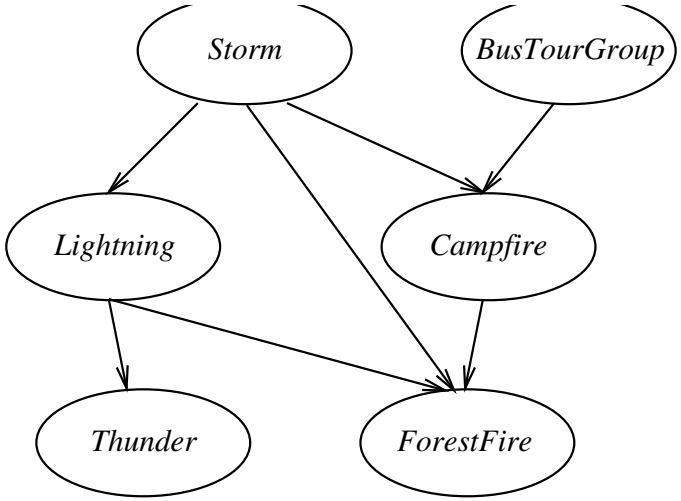
Exemplo: *Thunder* é condicionalmente independente de *Rain*, dado *Lightning*

$$P(Thunder|Rain, Lightning) = P(Thunder|Lightning)$$

Naive Bayes usa independência condicional para justificar

$$\begin{aligned} P(X, Y|Z) &= P(X|Y,Z)P(Y|Z) \\ &= P(X|Z)P(Y|Z) \end{aligned}$$

Redes Bayesianas

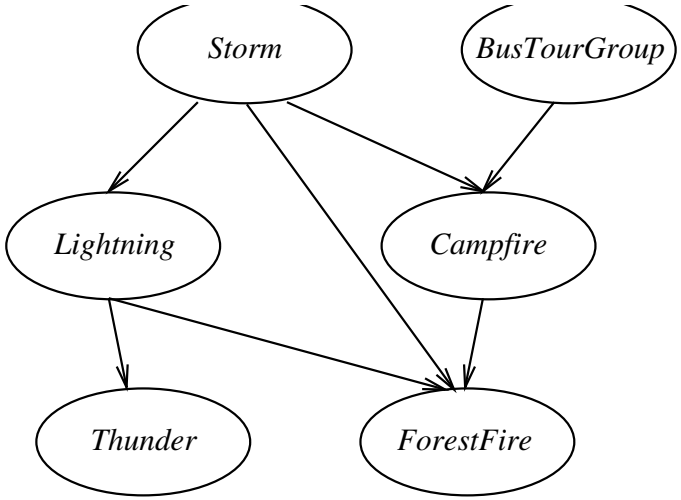


	S,B	$S,\neg B$	$\neg S,B$	$\neg S,\neg$
C	0.4	0.1	0.8	0.2
$\neg C$	0.6	0.9	0.2	0.8

Campfire

Rede representa um conjunto de asserções de independência condicionais:

- Cada nó é dito ser condicionalmente independente de seus não-descendentes, dado seus antecessores imediatos.
- Grafo acíclico direcionado



	S,B	$S,\neg B$	$\neg S,B$	$\neg S,\neg$
C	0.4	0.1	0.8	0.2
$\neg C$	0.6	0.9	0.2	0.8

Campfire

Representa a distribuição de probabilidade conjunta sobre todas variáveis.

- e.g., $P(Storm, BusTourGroup, \ldots, ForestFire)$
- em geral, $$P(y_1, \ldots, y_n) = \prod_{i=1}^{n} P(y_i | Parents(Y_i))$$

  onde $Parents(Y_i)$ denota os antecessores imediatos de Y_i no grafo
- então, a distribuição conjunta é definida completamente pelo grafo, mais os valores $P(y_i | Parents(Y_i))$

Inferência em Redes Bayesianas

Como podemos inferir valores (de probabilidades) de uma ou mais variáveis da rede, dados os valores observados de outras variáveis?

- A rede Bayesiana contém todas informação necessária para essa inferência
- Se somente uma variável tem valor desconhecido, é fácil inferir
- No caso geral, o problema é NP-difícil

Na prática,

- Métodos de inferência exatos funcionam bem para algumas estruturas de rede
- Métodos de Monte Carlo 'simulam' a rede aleatoriamente para calcular soluções aproximadas

Aprendizagem de Redes Bayesianas

Diversas variantes desta tarefa de aprendizagem

- Estrutura da rede pode ser *conhecida* ou *desconhecida*
- Exemplos de treinamento podem conter valores de *todas* variáveis da rede, ou somente de *algumas*

Se a estrutura é conhecida e todas variáveis são observadas

- Então é tão fácil quanto treinar o classificador Naive Bayes

Suponha uma estrutura conhecida, e variáveis parcialmente observáveis

e.g., observe *ForestFire, Storm, BusTourGroup, Thunder*, mas não *Lightning, Campfire*...

- Similar a treinar uma rede neural com unidades ocultas
- Na verdade, podemos aprender tabelas de probabilidade condicional da rede usando ascenso do gradiente (*gradient ascent*)
- Converge para a rede h que maximiza (localmente) $P(D|h)$

Ascenso do Gradiente para Redes Bayesianas

Seja w_{ijk} uma entrada na tabela de probabilidade condicional para a variável Y_i na rede

$$w_{ijk} = P(Y_i = y_{ij} | Parents(Y_i) = \text{a lista de valores } u_{ik})$$

e.g., se $Y_i = Campfire$, então u_{ik} poderia ser
$\langle Storm = T, BusTourGroup = F \rangle$

Execute ascenso do gradiente repetindo os procedimentos

1. atualize todos w_{ijk} usando os dados de treinamento D

$$w_{ijk} \leftarrow w_{ijk} + \eta \sum_{d \in D} \frac{P_h(y_{ij}, u_{ik} | d)}{w_{ijk}}$$

2. então, re-normalize w_{ijk} para garantir
   - $\sum_j w_{ijk} = 1$
   - $0 \leq w_{ijk} \leq 1$

Mais sobre Aprendizagem de Redes Bayesianas

O algoritmo EM também pode ser usado. Repetindo os passos:

1. Calcule as probabilidades das variáveis não observadas, assumindo e usando a hipótese atual h (w_{ijk})
2. Calcule novos w_{ijk} para maximizar $E[\ln P(D|h)]$ onde D inclui, agora, ambas as variáveis observadas e as (probabilidades calculadas das) variáveis não observadas

Quando a estrutura é desconhecida...

- Algoritmos que usam busca gulosa para adicionar/remover arestas e nós
- Tópico de pesquisa

Resumo: Redes Bayesianas

- Combinam conhecimento a priori com dados observados
- Impacto do conhecimento a priori (quando correto) é reduzir a complexidade da inferência/amostras
- Áreas ativas de pesquisa
  - Estender de variáveis booleanas para reais
  - Distribuições parametrizadas ao invés de tabelas
  - Estender para primeira-ordem ao invés de sistemas proposicionais
  - Métodos mais efetivos de inferência
  - ...

Expectation Maximization (EM)

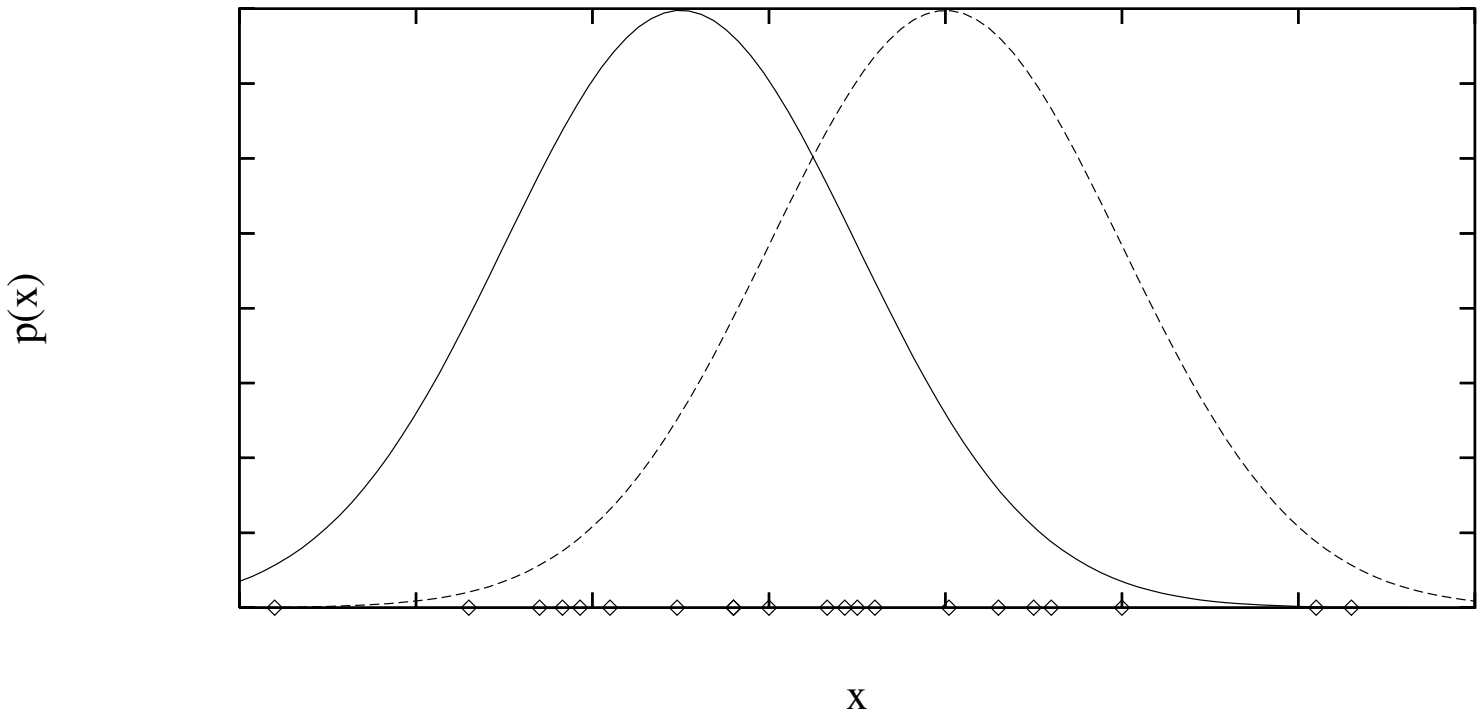
Maximização da Estimativa
Quando usar:

- Dados são somente parcialmente observáveis
- Agrupamento não-supervisionado (clustering) (Classe não-observável)
- Aprendizagem supervisionada (atributos de instâncias não-observáveis)

Algumas aplicações:

- Treinamento de Redes Bayesianas
- Agrupamento não-supervisionado
- Treinando Modelos Ocultos de Markov (MOM, HMMs)

Gerando Dados a Partir de uma Mistura de k Gaussianas



Cada instância x gerada

1. Escolhando uma das k Gaussianas com probabilidade uniforme
2. Gerando uma instância aleatoriamente de acordo com tal Gaussiana

EM para Estimar k Médias

Dado:

- Instâncias de X geradas por uma mistura de k Distribuições Gaussianas
- Médias $\langle \mu_1, \ldots, \mu_k \rangle$ desconhecidas das k Gaussianas
- Não sabemos que instância x_i foi gerada por qual Gaussiana

Determinar:

- As estimativas de Máximo likelihood $\langle \mu_1, \ldots, \mu_k \rangle$

Pense na descrição completa de cada instância como $y_i = \langle x_i, z_{i1}, z_{i2} \rangle$, onde

- z_{ij} é 1 se x_i é gerado pela j-ésima Gaussiana
- x_i observável
- z_{ij} não-observável

EM para Estimar k Médias

Algoritmo EM: Escolha um valor inicial para $h = \langle \mu_1, \mu_2 \rangle$, então repita:

Passo E: Calcular o valor esperado $E[z_{ij}]$ de cada variável oculta z_{ij}, assumindo que a hipótese atual $h = \langle \mu_1, \mu_2 \rangle$ é verdadeira.

$$E[z_{ij}] = \frac{p(x = x_i | \mu = \mu_j)}{\sum_{n=1}^{2} p(x = x_i | \mu = \mu_n)}$$

$$= \frac{e^{-\frac{1}{2\sigma^2}(x_i - \mu_j)^2}}{\sum_{n=1}^{2} e^{-\frac{1}{2\sigma^2}(x_i - \mu_n)^2}}$$

Passo M: Calcular uma nova hipótese de Máximo likelihood $h' = \langle \mu'_1, \mu'_2 \rangle$, assumindo que o valor tomado por cada variável oculta z_{ij} é seu valor esperado $E[z_{ij}]$ calculatado acima. Substituir $h = \langle \mu_1, \mu_2 \rangle$ por $h' = \langle \mu'_1, \mu'_2 \rangle$.

$$\mu_j \leftarrow \frac{\sum_{i=1}^{m} E[z_{ij}] \; x_i}{\sum_{i=1}^{m} E[z_{ij}]}$$

Algoritmo EM

Converge para o máximo likelihood h
e fornece estimativas das variáveis ocultas z_{ij}

Na verdade, o máximo local em $E[\ln P(Y|h)]$

- Y são os dados completos (variáveis observáveis + não-observáveis)
- O valor esperado é tomado sobre os possíveis valores das variáveis não-observadas em Y

Problema Geral EM

Dado:

- Dados observados $X = \{x_1, \ldots, x_m\}$
- Dados não-observados $Z = \{z_1, \ldots, z_m\}$
- Distribuição de probabilidade parametrizada $P(Y|h)$, onde
  - $Y = \{y_1, \ldots, y_m\}$ são os dados completos $y_i = x_i \cup z_i$
  - h são os parâmetros

Determinar:

- h que maximiza (localmente) $E[\ln P(Y|h)]$

Muitas aplicações:

- Treinamento de Redes Bayesianas
- Agrupamento não-supervisionado (e.g., k means)
- Modelos Ocultos de Markov (Hidden Markov Models)

Método EM Geral

Defina a função de likelihood $Q(h'|h)$ que calcula $Y = X \cup Z$ usando o X observado e os parâmetros atuais h para estimar Z

$$Q(h'|h) \leftarrow E[\ln P(Y|h')|h, X]$$

Algoritmo EM:

Passo de Estimação (E): Calcule $Q(h'|h)$ usando a hipótese atual h e os dados observados X para estimar a distribuição de probabilidade sobre Y.

$$Q(h'|h) \leftarrow E[\ln P(Y|h')|h, X]$$

Passo de Maximização (M): Substitua a hipótese h pela hipótese h' que maximiza esta função Q.

$$h \leftarrow \arg\max_{h'} Q(h'|h)$$

Bibliografia

Capítulo 6 - 'Bayesian learning' do livro
Machine Learning, T. Mitchell, McGraw Hill, 1997.