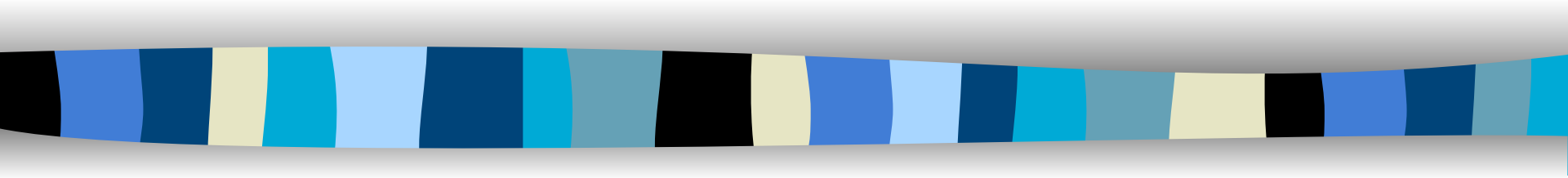
Aprendizagem Bayesiana



Roteiro

- Teorema de Bayes
- Hipóteses MAP e ML
- Aprendizagem de conceitos, redes neurais...
- Classificador bayesiano ótimo
- Algoritmo bayesiano ingênuo
- Exemplo: classificação de textos
- Redes Bayesianas
- Algoritmo Expectation-Maximization

Métodos Bayesianos

Fornece algoritmos práticos de aprendizagem

- Aprendizagem Bayesiana ingênua
- Aprendizagem de Redes Bayesianas
- Combina conhecimento a priori (probabilidade a priori ou incondicional) com dados de observação
- Requer probabilidades à priori

Teorema de Bayes

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

- P(h): probabilidade a priori da hipótese h
- P(D): probabilidade a priori dos dados de treinamento D
- P(h/D): probabilidade de h dado D (probabilidade à posteriori ou condicional)
- P(D/h): probabilidade de D dado h

Escolha de hipóteses

- Geralmente deseja-se a hipótese mais provável observados os dados de treinamento

- Hipótese de *máxima a posteriori* h_{MAP}

$$
\begin{aligned}
h_{MAP} &= \underset{h \in H}{\mathbf{argmax}}\, P(h/D) \\
&= \underset{h \in H}{\mathbf{argmax}}\, \frac{P(D/h)P(h)}{P(D)} \\
&= \underset{h \in H}{\mathbf{argmax}}\, P(D/h)P(h)
\end{aligned}
$$

- Hipótese de *máxima verossimilhança* h_{ML} (supondo que $P(h_i)=P(h_j)$)

$$
h_{ML} = \underset{h_i \in H}{\mathbf{argmax}}\, P(D/h_i)
$$

Aplicação do Teorema de Bayes: Diagnóstico Médico

- Seja

M=doença meningite

S= dor no pescoço

- Um Médico sabe:

P(S/M)=0.5

P(M)=1/50000

P(S)=1/20

$$P(M/S)=\frac{P(S/M)P(M)}{P(S)}$$

$$=\frac{0,5*(1/50000)}{1/20}=0,002$$

- A probabilidade de uma pessoa ter meningite dado que ela está com dor no pescoço é 0,02% ou ainda 1 em 5000.

Fórmulas Básicas de Probabilidade

- *Regra do Produto*: Probabilidade de uma conjunção de dois eventos A e B

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

- *Regra da Soma*: Probabilidade de uma disjunção de dois eventos A e B

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

- *Teorema da Probabilidade Total:* Se os eventos $A_1, \ldots, A_n$ são mutuamente exclusivos e formam uma partição do evento certo

$$P(B) = \sum_{1=1}^{n} P(B/A_i)P(A_i)$$

Algoritmo de aprendizagem da força bruta para hipóteses MAP

1. Para cada hipótese $h \in H$, calcule a probabilidade a posteriori

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

2. Escolha a hipótese h_{MAP} de maior probabilidade à posteriori

$$h_{MAP} = \underset{h \in H}{\mathbf{argmax}} P(h/D)$$

Relação com a aprendizagem de Conceitos

- Suponha um conjunto fixo de instâncias $\{x_1, \ldots, x_m\}$
- Suponha que D é o conjunto de classificações $D = \{f(x_1), \ldots, f(x_m)\}$
- Escolha P(D/h)
  - P(D/h) = 1, se h é consistente com D
  - P(D/h) = 0, senão

Relação com a aprendizagem de Conceitos

- Escolha P(h) sob a hipótese de distribuição uniforme

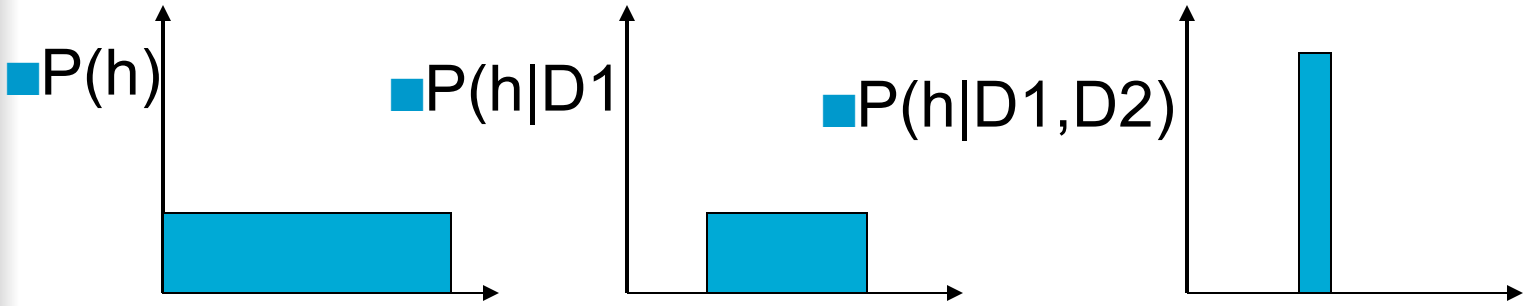
$$P(h) = \frac{1}{|H|} \qquad \forall\, h \in H$$

Então,

$$P(h/D) = \begin{cases} \dfrac{1}{|VS_{H,D}|}, \text{ se h é consistente com D} \\ 0, \text{ senão} \end{cases}$$

$|VS_{H,D}|$: subconjunto de hipóteses de H consistentes com D

Evolução das probabilidades à posteriori



- Uma análise bayesiana pode ser algumas vezes usada para mostrar que algoritmos particulares produzem hipóteses MAP, mesmo não usando explicitamente o teorema de bayes ou calcular probabilidade (e.g., FIND-S e CANDITATE-ELIMINATION)

Aprendizagem de uma função com valores reais (1/3)

- Suponha uma função f qualquer com valores reais
- Exemplos de treinamento $\langle x_i, d_i \rangle$, onde d_i :
  - **$d_i = f(x_i) + e_i$**
  - **e_i é uma variável aleatória (ruído) - distribuição normal com média 0**
- Então a hipótese de máxima verossimilhança h_{ML} é aquela que minimiza a soma dos erros quadráticos:

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

Aprendizagem de uma função com valores reais (2/3)

$$h_{ML} = \arg\max_{h_i \in H} p(D / h_i)$$

$$= \arg\max_{h_i \in H} \prod_{i=1}^{m} p(d_i / h)$$

$$= \arg\max_{h_i \in H} \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} \exp^{-\frac{1}{2}\left(\frac{d_i - h(x_i)}{\sigma^2}\right)^2}$$

- ≡ Maximizando o log natural:

Aprendizagem de uma função com valores reais (3/3)

$$
\begin{aligned}
h_{ML} &= \arg\max_{h_i \in H} \sum_{i=1}^{m} \ln \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2}\left(\frac{d_i - h(x_i)}{\sigma^2}\right)^2 \\
&= \arg\max_{h_i \in H} \sum_{i=1}^{m} -\frac{1}{2}\left(\frac{d_i - h(x_i)}{\sigma^2}\right)^2 \\
&= \arg\max_{h_i \in H} \sum_{i=1}^{m} -(d_i - h(x_i))^2 \\
&= \arg\min \sum_{i=1}^{m} (d_i - h(x_i))^2
\end{aligned}
$$

H_{ML} para prever probabilidades

- Suponha que queremos aprender uma função probabilística $f: X \rightarrow \{0,1\}$
- O espaço de exemplos X pode representar pacientes em termos de seus sintomas e *f(x)*:
  - o valor 1, se o paciente sobreviver a doença
  - o valor 0, caso contrario
- Porém, em geral, podemos ter o seguinte caso:
  - 92% dos pacientes sobrevivem
  - 8% não sobrevivem

H_{ML} para prever probabilidades/Redes Neurais (1/3)

- Neste contexto, suponha que uma rede neural é treinada de tal forma que sua saída é a probabilidade de *f(x)=1*
  - i.e, aprender a função *f´:X → [0,1], tal que*
  - *f´(x)=P(f(x)=1)*
  - *f´(x=1)=0.92*
- Mas como uma rede neural pode ser usada para aprender f´?
- Que critério deve ser otimizado a fim de que H_{ML} possa ser aprendida neste contexto?

H_{ML} para prever probabilidades/Redes Neurais (2/3)

- Que critério deve ser otimizado a fim de que H_{ML} possa ser aprendida neste contexto?

$$h_{ML} = \underset{h_i \in H}{\operatorname{argmax}} \sum_{i=1}^{m} d_i \ln h(x_i) + (1 - d_i) \ln (1 - h(x_i))$$

- **Cross-entropy**

- A regra de atualização dos pesos para a função sigmoide e´:

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

onde

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

H_{ML} para prever probabilidades/Redes Neurais (1/3)

- Sumário:
  - Tanto a regra anterior quanto a do backpropagation convergem para um H_{ML} em dois cenários diferentes:
  - 1) A regra que minimizar a soma quadrática dos erros procura a H_{ML} supondo que o cj. de treinamento pode ser modelado por ruído normalmente distribuído adicionado a função original
  - 2) A regra que minimiza a "cross entropy" busca pela H_{ML} supondo que o valor booleano observado é uma função probabilística do exemplo de entrada

Classificação mais provável de uma nova instância

- Dada uma nova instância x, qual é a sua *classificação* mais provável?
  - $h_{MAP}(x)$ não é a classificação mais provável

Considere,

- Três hipóteses:
  - $P(h_1/D) = 0.4$, $P(h_2/D) = 0.3$ e $P(h_3/D) = 0.3$

- Dada uma nova instância x,
  - Suponha: $h_1(x) = +$, $h_2(x) = -$ e $h_3(x) = -$
- A classificação mais provável de x?

Classificador Bayesiano Ótimo 1/2)

- Se a possível classificação do novo exemplo pode ser qualquer $v_j \in V$, a probabilidade de que a classificação correta seja v_j

$$P(v_j / D) = \sum_{h_i \in H} P(v_j / h_i) P(h_i / D)$$

Classificação Bayesiana ótima

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

Classificador Bayesiano Ótimo (1/2)

- Exemplo.
  - $P(h_1/D) = 0.4$, $P(-/h_1) = 0$, $P(+/h_1) = 1$
  - $P(h_2/D) = 0.3$, $P(-/h_1) = 1$, $P(+/h_1) = 0$
  - $P(h_3/D) = 0.3$, $P(-/h_3) = 1$, $P(+/h_1) = 0$

- Portanto $\sum_{h_i \in H} P(+/h_i)P(h_i/D) = 0.4$

$$\sum_{h_i \in H} P(-/h_i)P(h_i/D) = 0.6$$

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

Classificador Bayesiano Ingênuo

- Junto com árvores de decisão, redes neurais, vizinhos mais-próximos, é um dos métodos de aprendizagem mais práticos
- Quando usa-lo
  - Quando se tem disponível um conjunto de treinamento médio ou grande
  - Os atributos que descrevem as instâncias forem condicionalmente independentes dada uma classificação

Aplicações:

- Diagnóstico
- Classificação de documentos (texto)

Classificador Bayesiano Ingênuo

- Suponha uma função de classificação f: X → V, onde cada instância x é descrita pelos atributos $\{a_1, \ldots, a_n\}$
- O valor mais provável de f(x) é

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

- Suposição Bayesiana Ingênua

$$P(a_1, \cdots, a_n / v_j) = \prod_i P(a_i / v_j)$$

- Classificador Bayesiano Ingênuo

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

Algoritmo Bayesiano Ingênuo

- Aprendizagem_Bayesiana_Ingênua(exemplos)

- Para cada v_j
  - $P'(v_j) \leftarrow$ estimativa de $P(v_j)$
  - Para cada valor a_i de cada atributo a
    - $P'(a_i/v_j) \leftarrow$ estimativa de $P(a_i/v_j)$

- Classificador_Novas_Instancias(x)

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

Classificador bayesiano ingênuo: exemplo

Dia	Tempo	Temp.	Humid.	Vento	Jogar
D1	Sol	Quente	Alta	Fraco	Não
D2	Sol	Quente	Alta	Forte	Não
D3	Coberto	Quente	Alta	Fraco	Sim
D4	Chuva	Normal	Alta	Fraco	Sim
D5	Chuva	Frio	Normal	Fraco	Não
D6	Chuva	Frio	Normal	Forte	Não
D7	Coberto	Frio	Normal	Forte	Sim
D8	Sol	Normal	Alta	Fraco	Não
D9	Sol	Frio	Normal	Fraco	Sim
D10	Chuva	Normal	Normal	Fraco	Sim
D11	Sol	Frio	Alta	Forte	?

- P(Sim) = 5/10 = 0.5
- P(Não) = 5/10 = 0.5
- P(Sol/Sim) = 1/5 = 0.2
- P(Sol/Não) = 3/5 = 0.6
- P(Frio/Sim) = 2/5 = 0.4
- P(Frio/Não) = 2/5 = 0.4
- P(Alta/Sim) = 2/5 = 0.4
- P(Alta/Não) = 3/5 = 0.6
- P(Forte/Sim) = 1/5 = 0.2
- P(Forte/Não) = 2/5 = 0.4
- P(Sim)P(Sol/Sim) P(Frio/Sim)
- P(Alta/Sim) P(Forte/Sim) = 0.0032
- P(Não)P(Sol/Não)P(Frio/Não)
- P(Alta/Não) P(Forte/Não) = 0.0288

- ⇒ Jogar_Tenis(D11) = Não

Algoritmo Bayesiano Ingênuo : Dificuldades (1/2)

- Suposição de independência condicional quase sempre violada

$$P(a_1,\cdots,a_n / v_j) = \prod_i P(a_i / v_j)$$

  - Mas funciona surpreendentemente bem

- O que acontece se nenhuma das instancias classificadas como v_j tiver o valor a_i?

$$P'(a_i / v_j) = 0 \Rightarrow P'(v_j)\prod_i P'(a_i / v_j) = 0$$

Algoritmo Bayesiano Ingênuo : Dificuldades (2/2)

- Solução típica

$$P'(a_i / v_j) \leftarrow \frac{n_c + m \times p}{n + m}$$

- n é o número de exemplos para os quais v = v_j
- n_c número de exemplos para os quais v = v_j e a = a_i
- p é a estimativa à priori para $P'(a_i/v_j)$
- m é o peso dado à priori (número de exemplos "virtuais")

Redes Bayesianas

- Interesse
  - Suposição de independência condicional muito restritiva (classificador bayesiano ingênuo)
  - Mas sem esse tipo de suposição em algum nível o problema se torna intratável

- Redes Bayesianas descrevem independência condicional entre subconjuntos de variáveis
  - Permite a combinação do conhecimento a priori sobre a independência entre variáveis com os dados observados

Independência Condicional

- X é *condicionalmente independente* de Y dado Z se a distribuição de probabilidade de X é independente do valor de Y dado o valor de Z

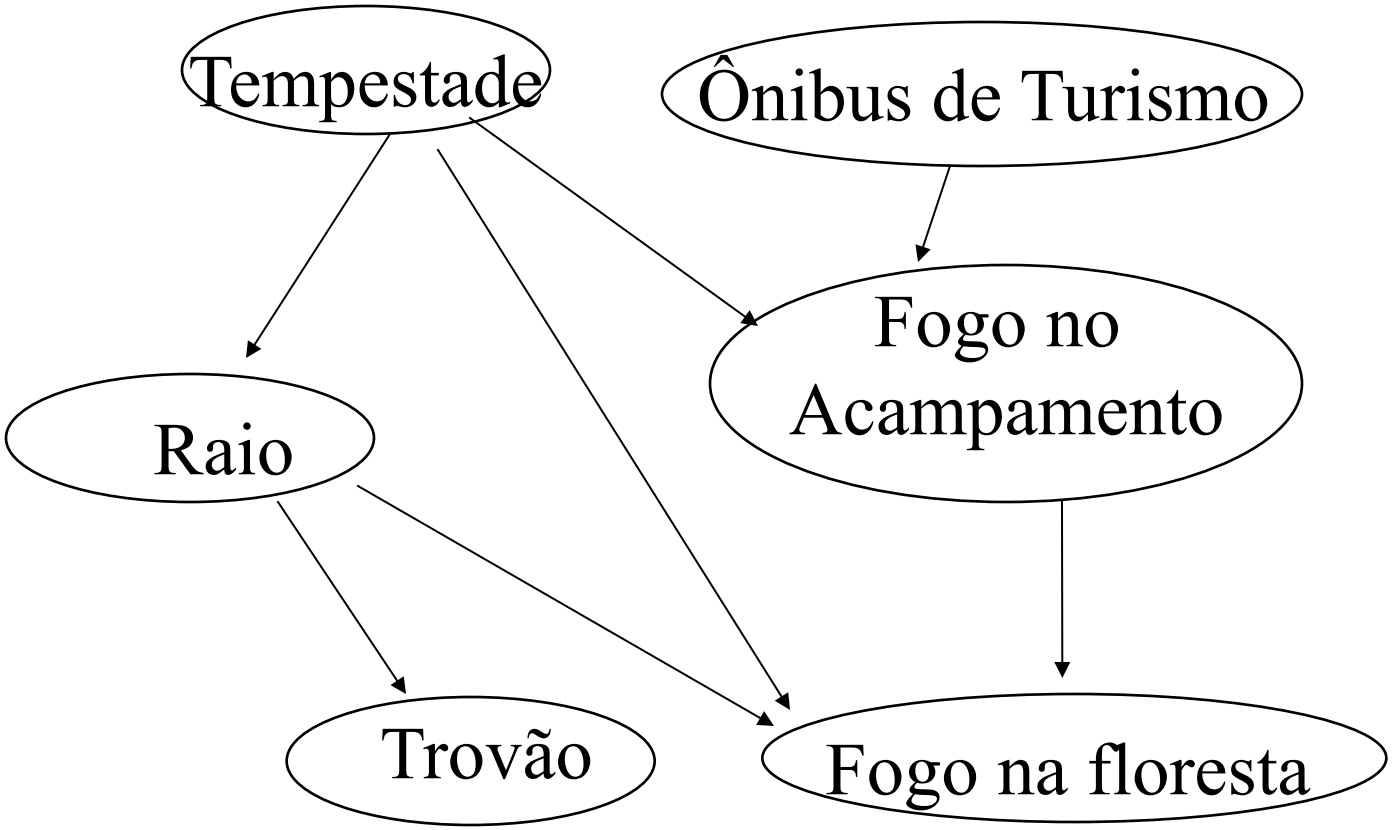
$$(\forall x_i, y_i, z_i) P(X = x_i / Y = y_j, Z = z_k) = P(X = x_i / Z = z_k)$$

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

- Exemplo: Trovão é condicionalmente independente de Chuva, dado Relâmpago
  - P(Trovão/ Chuva, Relâmpago) = P(Trovão/ Relâmpago)
  - Regra do Produto:

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

Redes Bayesianas - Representação



Redes Bayesianas - Representação

	T,O	T,¬O	¬T, O	¬T, ¬O
FC	0.4	0.1	0.8	0.2
¬FC	0.6	0.9	0.2	0.8

Fogo no Acampamento

- A rede representa um conjunto de asserções de independência condicional
  - Cada nó é condicionalmente independente dos seus não descendentes, dados os seus predecessores (pais) imediatos
  - Grafo acíclico direto

Redes Bayesianas

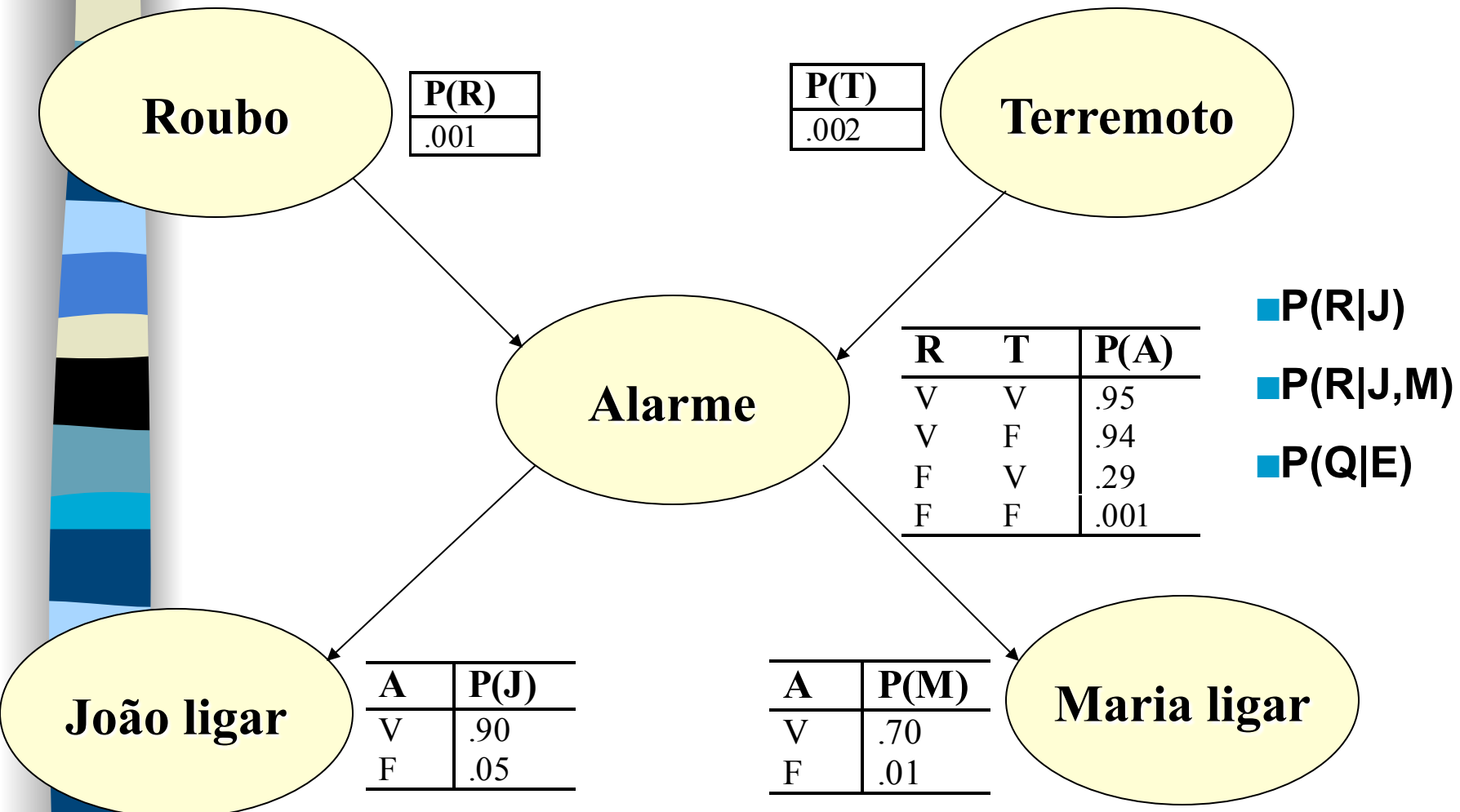
- Representa a distribuição de probabilidade conjunta entre todas as variáveis
  - Exemplo, P(Tempestade, ..., Fogo na Floresta)
  - Em geral

$$P(y_1, \ldots, y_n) = \prod_{i=1}^{n} P(y_i / Pais(Y_i))$$

onde Pais(Y_i) significa predecessores imediatos de Y_i no grafo

- A distribuição conjunta é definida pelo grafo mais os (y_i / Pais(Y_i))

Rede Bayesiana com as probabilidades condicionais



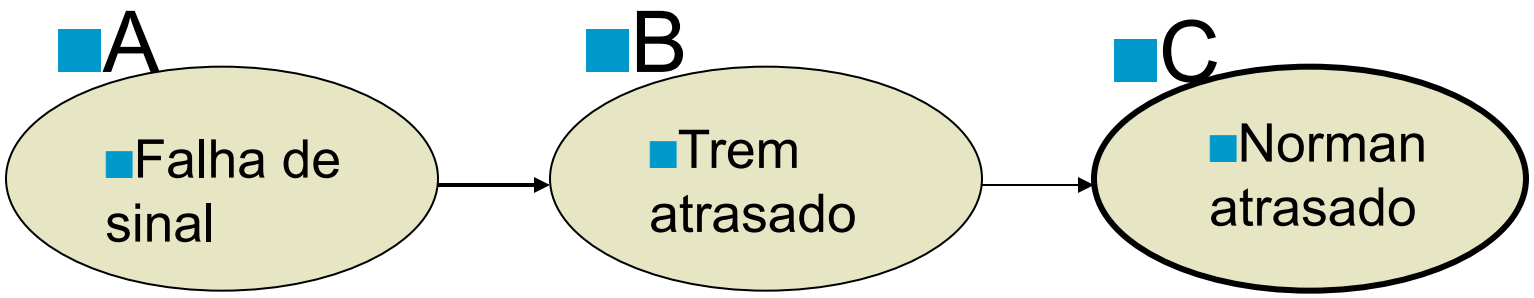
Como Redes bayesianas lidam com evidências

- ***Evidência Forte*** **(instanciação) para um nó X é uma evidência que X está definitivamente em um valor particular**
  - **Suponha que X represente um resultado de uma determinada partida de um time de futebol *{ganha, perdida, empate}*. Uma evidência forte seria o conhecimento que a partida foi definitivamente ganha. X seria instanciado com o valor "ganha"**
- ***Evidência Soft*** **para um nó X é qualquer evidência que permita a atualização da probabilidade a priori do estado de X.**
  - **Se sabemos que o time está ganhando a partida por 4-0 aos 35 minutos do 2o. tempo, então a probabilidade de "*ganha*" seria bastante alta, enquanto a probabilidade de "*perdida*" e "empate" seria baixa**

Tipos de Conexão

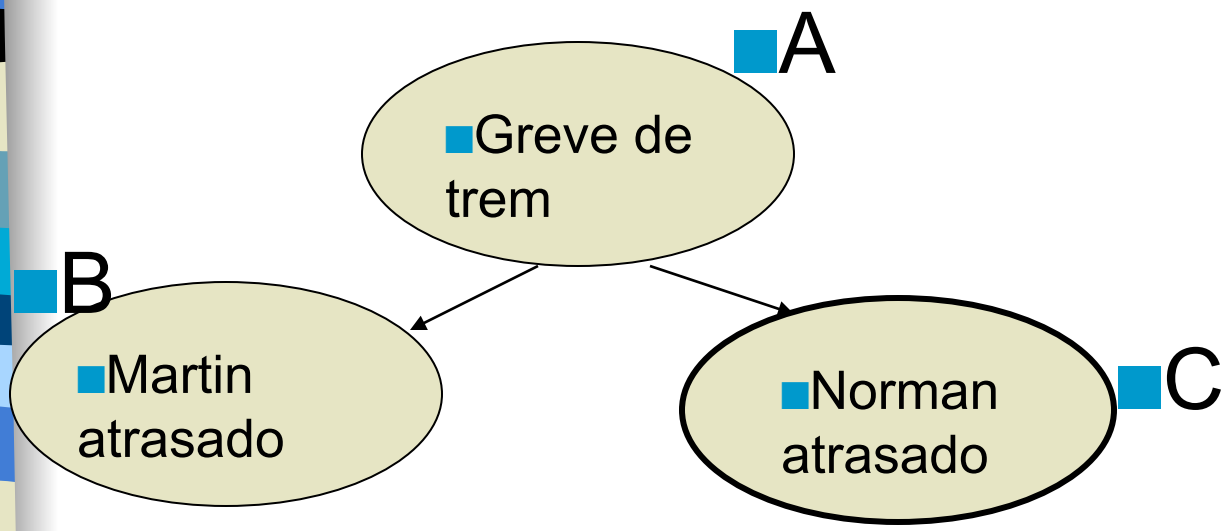
- **Conexão serial**
  - **Qualquer evidência que entra no início da conexão pode ser transmitida ao longo do caminho direcionado, dado que nenhum nó intermediário no caminho é instanciado**
- **Conexão divergente**
  - **Evidências podem ser transmitidas entre dois nós filhos do mesmo pai, dado que o pai não está instanciado**
- **Conexão convergente**
  - **Evidências só podem ser transmitidas entre dois pais, qdo. o nó filho (convergente) recebeu alguma evidência que pode ser soft ou forte.**

Conexão Serial



- Evidências podem ser transmitidas de A para C, a menos que B seja instanciado

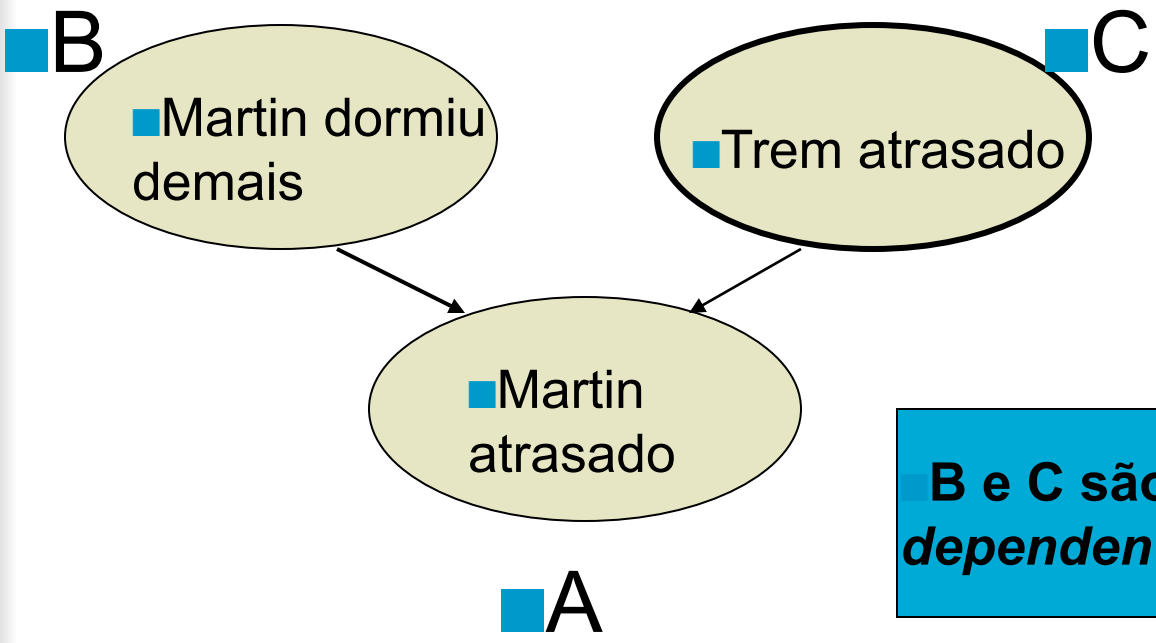
Conexão Divergente



- **B e C são *condicionalmente independentes*, dado A**

- **Evidências podem ser transmitidas de B para C através de uma conexão divergente A, a menos que A esteja instanciado**

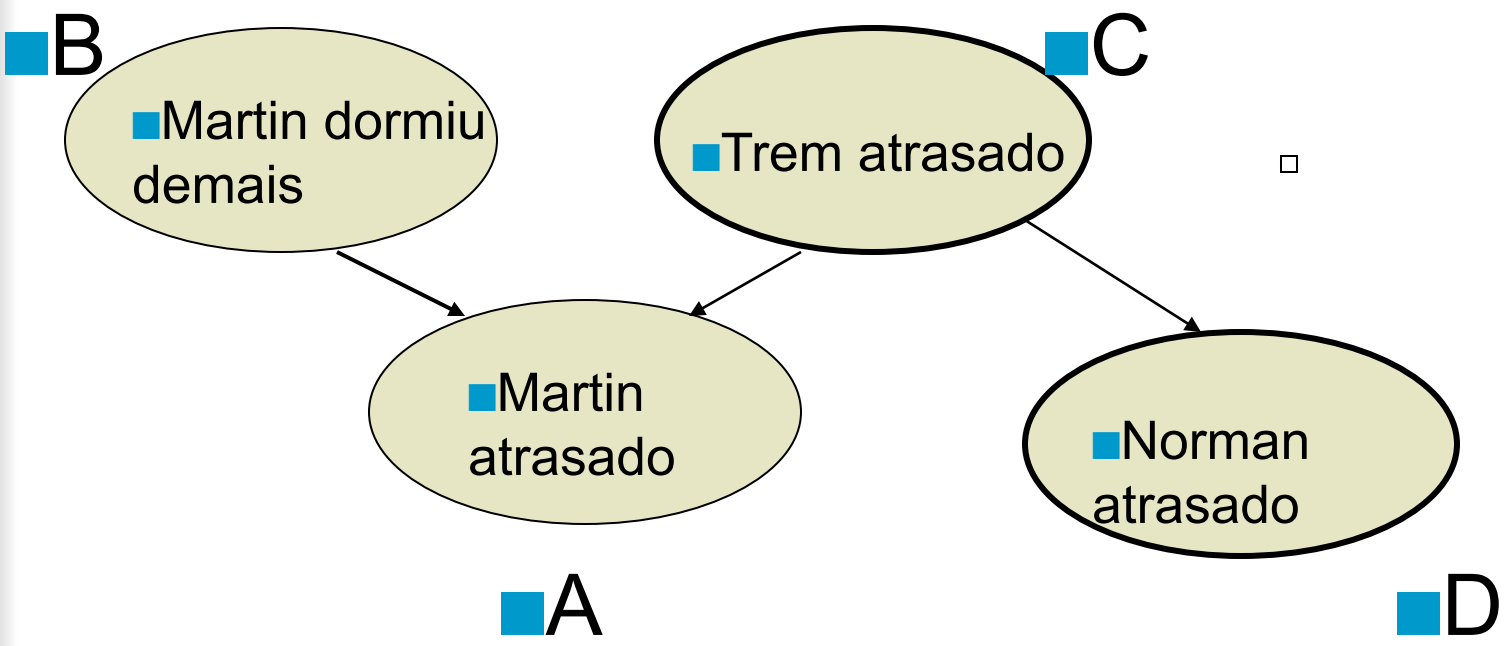
Conexão Divergente



B e C são *condicionalmente dependente* de A

Em uma conexão convergente uma evidência só pode ser transmitida entre pais B e C , qdo. um nó convergente A recebeu alguma evidência (forte ou soft).

Explaining away



Inferência em Redes Bayesianas

- Como inferir as probabilidades dos valores de uma ou mais variáveis na rede, à partir das probabilidades dos valores das outras variáveis
  - A rede Bayesiana contém toda a informação necessária para essa inferência
  - Quando se trata de apenas uma variável, a inferência é trivial
  - No caso geral, o problema é NP hard
- Na prática, pode-se alcançar-la de várias formas
  - Métodos exatos de inferência funcionam bem para algumas estruturas de rede
  - Métodos de tipo Monte Carlo "simulam" a rede aleatoriamente para obter soluções aproximadas

Aprendizagem de Redes Bayesianas (1/2)

- Variantes da tarefa de aprendizagem
  - A estrutura da rede pode ser conhecida ou desconhecida
  - O conjunto de treinamento pode fornecer valores para todas as variáveis da rede ou para somente algumas
- Se a estrutura é conhecida e todas as variáveis observadas
  - Então é tão fácil como treinar um classificador Bayesiano ingênuo

Aprendizagem de Redes Bayesianas (2/2)

- Suponha a estrutura conhecida e variáveis parcialmente observáveis
  - Exemplo, observa-se *fogo na Floresta, Tempestade, Ônibus de turismo*, mas não *Raio, Fogo no Acampamento*
  - Problema similar ao treinamento de uma rede neural com neurônios ocultos
  - Aprende-se a tabela de probabilidades condicionais de cada nó usando o algoritmo do gradiente ascendente
  - O sistema converge para a rede h que maximiza localmente P(D/h)

Gradiente Ascendente p/ Redes Bayesianas

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

$w_{ijk} = P(Y_i = y_{ij} / Pais(Y_i) = \text{lista } u_{ik} \text{ de valores})$

Exemplo, se Y_i = Fogo no Acampamento, então u_{ik} pode ser {*Tempestade = T, Ônibus de Turismo = F*}

Aplicar o gradiente ascendente repetidamente

1. Atualizar todos os 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. Normalizar os w_{ijk} para assegurar

$$\sum_j w_{ijk} = 1$$

$$0 \leq w_{ijk} \leq 1$$

Aprendizagem em Redes Bayesianas

- O algoritmo EM também pode ser usado. Repetidamente:
  1. Calcule as probabilidades das variáveis não observadas, supondo h verdadeira
  2. Calcule novo w_{ijk} que maximize E[ln P(D/h)], onde D agora inclui tanto as variáveis observadas como as probabilidades calculadas das não observadas
- Quando a estrutura é desconhecida
  - Tópico de pesquisa ativo ...

Sumário de Redes Bayesianas

- Combina conhecimento a priori com dados observados
- O impacto do conhecimento a priori (quando correto) é a redução da amostra de dados necessários
- Área de pesquisa ativa
  - Passar de variáveis Booleanas para variáveis numéricas
  - Distribuições em vez de tabelas
  - Lógica de primeira ordem no lugar de proposicional
  - Métodos de inferência mais efetivos

Redes Bayesianas X Redes Neurais (1/3)

- **Como sistemas de representação**
  - **ambos são representações baseadas em atributos**
  - **ambos manipulam entradas e saídas contínuas**
  - **Nós em redes bayesianas representam proposições com semântica bem definida e relacionamentos probabilísticos com outras proposições; os nós de uma rede neural não representam proposições específicas**

Redes Bayesianas X Redes Neurais (1/3)

- **Como mecanismos de inferência ambos**
  - **Redes neurais feedforward podem executar em tempo linear, enquanto inferência para redes bayesianas gerais é um problema NP-difícil**
  - **Flexibilidade das redes bayesians. A qualquer momento, qualquer subconjunto de variáveis podem ser tratados como entrada, e qualquer outro subconjunto pode ser a saída; enquanto que em redes neurais feedforward têm entradas e saídas fixas.**

Redes Bayesianas X Redes Neurais (1/3)

- **Como sistemas de aprendizagem**
  - **Difícil de comparar ambos os sistemas - redes probabilísticas adaptativas é um tópico recente de pesquisa**
  - **Redes bayesianas são mais lentas em geral, mais às vezes podem ser mais rápida devido ao conhecimento a priori**
  - **Redes bayesianas representam proposições localmente. Isto faz com que estas redes convirjam mais rápido para uma representação de domínio que tem uma estrutura local**

Expectation Maximization (EM)

- Quando usar:
  - Os dados são observáveis apenas parcialmente
  - Aprendizagem não supervisionada (Clustering, os grupos são desconhecidos)
  - Aprendizagem Supervisionada (alguns valores de algumas variáveis não são observados
- Alguns usos
  - Treinamento das redes Bayesianas
  - Clustering
  - etc

Geração de Dados a partir de k Gaussianas

- Cada instancia x é gerada
  1. Escolhendo uma das k Gaussianas com probabilidade uniforme
  - Gerando uma instancia aleatoriamente de acordo com essa Gaussiana

EM para a estimação de k médias

Dado

- Instancias de X geradas pela mistura de k distribuições Gaussianas
- Médias desconhecidas $\langle \mu_1, \ldots, \mu_k \rangle$ das k Gaussianas
- Não se sabe que instancia x_i foi gerada por qual Gaussiana

Determine

- Estimativas de Máxima Verossimilhança de $\langle \mu_1, \ldots, \mu_k \rangle$

Considere a descrição completa de cada instancia como $y_i = \langle x_i, z_{i1}, z_{i2} \rangle$, onde

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

EM para a estimação de k médias

EM algoritmo: Selecione aleatoriamente uma hipótese inicial h = < μ_1, μ_2>

Passo E: Calcule o valor esperado E[z_{ij}] de cada variável oculta z_{ij}, supondo válida a hipótese atual h = < μ_1, μ_2>

$$E[z_{ij}] = \frac{p(x = x_i / \mu = \mu_j)}{\sum_{n=1}^{2} p(x = x_i / \mu = \mu_n)} = \frac{\exp\left[-\frac{1}{2\sigma^2}(x_i - \mu_j)^2\right]}{\sum_{i=1}^{n} \exp\left[-\frac{1}{2\sigma^2}(x_i - \mu_n)^2\right]}$$

EM para a estimação de k médias

Passo M: Calcule a nova hipótese de Máxima Verossimilhança $h' = \langle \mu'_1, \mu'_2 \rangle$, supondo que o valor assumido por cada variável oculta z_{ij} é o valor esperado $E[z_{ij}]$ já calculado. Troque $h = \langle \mu_1, \mu_2 \rangle$ por $h' = \langle \mu'_1, \mu'_2 \rangle$.

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

Converge para um máximo de verossimilhança h local e fornece estimativas para as variáveis ocultas z_{ij}

Bibliografia

- Russel, S, & Norvig, P. (1995). Artificial Intelligence: a Modern Approach (AIMA) Prentice-Hall. Pages 436-458, 588-593
- Mitchell, T. & (1997): Machine Learning, McGraw-Hill. Cap.6