Inteligência Artificial

Luís A. Alexandre

UBI

Ano lectivo 2019-20

Conteúdo

Introdução

- Por vezes os agentes têm de tomar decisões sem terem a certeza dos valores de todas as variáveis que podem influenciar essas decisões.
- Ex.: no mundo wumpus, o agente terá por vezes de tomar a decisão de avançar para uma sala sem ter a certeza de que ela não contém um poço.
- Os ambientes reais contêm muita incerteza pois são normalmente muito complexos e isso torna difícil a observação de todas as variáveis relevantes para a tomada de decisões.
- Os agentes podem lidar com essa incerteza usando **teorias probabilísticas** do funcionamento do mundo.
- Nesta aula vamos ver alguns métodos de aprendizagem que os agentes podem usar para obterem esse tipo de teorias.

Conteúdo

Aprendizagem Bayesiana

- A Aprendizagem Bayesiana (AB) permite a tomada de decisões sem ser necessário escolhermos qual é a teoria "certa" que explica o mundo.
- Conforme vamos recebendo informação, a AB vai decidindo entre as teorias que estão disponíveis, qual é a adequada.
- Exemplo: um rebuçado pode ter um de dois sabores: cereja ou limão.

- O fabricante embala os rebuçados sempre no mesmo papel, sendo impossível de distinguir o sabor olhando para o rebuçado embrulhado.

Aprendizagem Bayesiana

- Os rebuçados são vendidos em sacos grandes. Existem 5 variedades de sacos, mais uma vez indistinguíveis:
  - h_1: 100% cereja;
  - h_2: 75% cereja + 25% limão;
  - h_3: 50% cereja + 50% limão;
  - h_4: 25% cereja + 75% limão;
  - h_5: 100% limão;
- A variável aleatória (v.a.) H (de hipótese) indica o tipo de saco, assumindo um dos valores $\{h_1, h_2, \ldots, h_5\}$.
- H não é diretamente observável: conforme são abertos os rebuçados que um saco contém, ficamos a saber o seu sabor (os dados): $D_1, D_2, \ldots, D_n$, onde cada D_i é uma v.a. com valor d_i (cereja ou limão).

Aprendizagem Bayesiana

- Com esta notação podemos escrever a probabilidade de um dado rebuçado ser de cereja, se estivermos perante uma dada hipótese (um tipo de saco), $P(d_j|h_i)$, assim:
  - $P(d_j = cereja|h_1) = 1$;
  - $P(d_j = cereja|h_2) = 0.75$;
  - $P(d_j = cereja|h_3) = 0.5$;
  - $P(d_j = cereja|h_4) = 0.25$;
  - $P(d_j = cereja|h_5) = 0$;
- **Tarefa do agente**: prever qual é o sabor do próximo rebuçado.

Aprendizagem Bayesiana

- A AB **calcula a probabilidade de cada uma das hipóteses, face aos dados disponíveis**, e efetua a previsão de acordo com essas probabilidades.
- Seja **d** uma sequência de dados $\{d_1, d_2, \ldots, d_n\}$ e m o número de hipóteses.
- A probabilidade de cada hipótese h_i, face aos dados recebidos, é dada pela **regra de Bayes** (prob. a posteriori):

$$P(h_i|\mathbf{d}) = \frac{P(\mathbf{d}, h_i)}{P(\mathbf{d})} = \frac{P(\mathbf{d}|h_i)P(h_i)}{\sum_{j=1}^{m} P(\mathbf{d}|h_j)P(h_j)} \quad (1)$$

- Para o exemplo dos rebuçados vamos assumir por enquanto que a distribuição das **probabilidades a priori** $P(h_i)$, $i = 1, \ldots, 5$ é (0.1,0.2,0.4,0.2,0.1).

Aprendizagem Bayesiana

- Na AB as previsões são feitas usando as previsões de cada uma das hipóteses pesadas pelas suas probabilidades (que vêm da eq. (1)), em vez de se escolher apenas uma hipótese (a "melhor").
- Queremos achar a probabilidade do próximo elemento da sequência ser d_{n+1}, quando já observámos **d**. Temos então:

$$P(d_{n+1}|\mathbf{d}) = \sum_{i=1}^{m} P(d_{n+1}|h_i)P(h_i|\mathbf{d}) \quad (2)$$

Aprendizagem Bayesiana

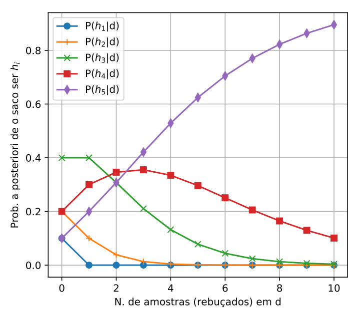
- Vamos assumir que as observações são i.i.d. tal que a **verosimilhança** dos dados em face de cada hipótese é dada por

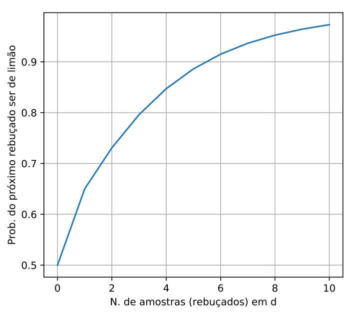
$$P(\mathbf{d}|h_i) = \prod_{j=1}^{n} P(d_j|h_i) \quad (3)$$

- Ex.: se o saco só tiver rebuçados de limão (h_5) e os primeiros 10 forem de limão então $P(\mathbf{d}|h_3) = 0.5^{10} \approx 0.001$, visto metade dos rebuçados de h_3 serem de limão ($P(d_j|h_3) = 0.5,\ j = 1, \ldots, 10$).
- Isto significa que se obervar 10 rebuçados de limão seguidos é muito pouco provável que estejamos perante a hipótese h_3.

Aprendizagem Bayesiana

- As figuras abaixo mostram: à esquerda o resultado da equação (1) e à direita da equação (2), para este exemplo (primeiros 10 rebuçados de limão).





Aprendizagem Bayesiana

- Conclusão da fig. da esquerda do slide anterior: com o aumento do número de observações a verdadeira hipótese acaba por dominar a previsão Bayesiana.
- A **previsão Bayesiana é ótima** no sentido em que, dado o mesmo vetor de probabilidades a priori, qualquer outro método de previsão vai acertar menos vezes.
- O problema é que frequentemente a soma na equação (2) não é calculável analiticamente e nestes casos devemos usar **aproximações**.

Conteúdo

Máximo à posteriori: MAP

- A aproximação mais frequente é fazer a previsão com base apenas na **hipótese mais provável** em vez de fazermos a soma pesada de (2).
- Passos:
  - achar as probabilidades $P(h_i|\mathbf{d})$ usando a eq. (1): podemos ignorar o denominador pois todas vão ter o mesmo valor e vamos no passo seguinte querer saber o máximo
  - escolher a hipótese h_i que tiver maior valor de $P(h_i|\mathbf{d})$: h_i^*.
  - fazemos a previsão com $P(d_{n+1}|\mathbf{d}) = P(d_{n+1}|h_i^*)$
- Esta abordagem é chamada **MAP** (máximo a posteriori).

Máximo à posteriori: MAP

- Exemplo: achar a prob. do próximo rebuçado ser limão após termos visto 2 rebuçados de limão.
- Temos então: $\mathbf{d} = (\textit{limão}, \textit{limão})$.
- Queremos $P(\textit{limão}|\mathbf{d})$.
- Começamos por achar o valor da eq. (1) para as 5 hipóteses do problema (podemos ignorar o denominador):

$$P(h_1|(\textit{limão}, \textit{limão})) = P((\textit{limão}, \textit{limão})|h_1)P(h_1)$$

- O primeiro termo do lado direito obtém-se com a eq. (3):

$$P((\textit{limão}, \textit{limão})|h_1) = P(\textit{limão}|h_1)P(\textit{limão}|h_1) = 0$$

logo $P(h_1|(\textit{limão}, \textit{limão})) = 0$.

Máximo à posteriori: MAP

- Repetindo o processo para as 4 restantes hipóteses obtemos:

$$P(h_2|(\textit{limão}, \textit{limão})) = 0.25^2 \times 0.2 = 0.0125$$

$$P(h_3|(\textit{limão}, \textit{limão})) = 0.5^2 \times 0.4 = 0.1$$
$$P(h_4|(\textit{limão}, \textit{limão})) = 0.75^2 \times 0.2 = 0.1125$$
$$P(h_5|(\textit{limão}, \textit{limão})) = 1 \times 0.1 = 0.1$$

- Escolhemos a hipótese com maior valor, logo $h_i^* = h_4$.
- Vamos fazer a previsão:

$$P(\textit{limão}|\mathbf{d}) = P(\textit{limão}|h_4) = 0.75$$

Máximo à posteriori: MAP

- No exemplo que acabámos de ver, e considerando que se vão obtendo sempre rebuçados de limão, a prob. do próximo rebuçado ser limão varia da seguinte forma:

N. de dados	AB	MAP
1	0.65	0.50
2	0.73	0.75
3	0.80	1.00
4	0.85	1.00
5	0.89	1.00
10	0.97	1.00

- Conforme obtemos mais dados as probabilidades obtidas pelos 2 métodos aproximam-se porque na AB as hipóteses competidoras com a mais provável vão-se tornando menos prováveis.

Conteúdo

Máxima verosimilhança

- Podemos simplificar ainda mais a nossa previsão assumindo que **todas as hipóteses são igualmente prováveis a priori**.
- Isto é o mesmo que dizer que $P(h_i) = 1/m$, onde m é o número de hipóteses.
- Neste caso a solução MAP (achar o h_i que maximiza a eq.(1)) reduz-se a escolher o h_i que maximiza $P(\mathbf{d}|h_i)$ (a eq. (3)).
- Passos:
  - achar as probabilidades $P(\mathbf{d}|h_i)$ usando a eq. (3)
  - escolher a hipótese h_i que tiver maior valor de $P(\mathbf{d}|h_i)$: h_i^*
  - fazemos a previsão com $P(d_{n+1}|\mathbf{d}) = P(d_{n+1}|h_i^*)$
- Esta escolha diz-se que é a escolha de **máxima verosimilhança**, pois a eq. (3) dá-nos a verosimilhança.

Máxima verosimilhança

- Façamos a previsão com este critério de qual será a probabilidade do próximo rebuçado ser limão se já tivermos visto 2 de limão.
- Vamos achar o valor da eq. (3) para todas as hipóteses e escolhemos a que tiver maior valor:

$$P((\textit{limão}, \textit{limão})|h_1) = 0$$

$$P((\textit{limão}, \textit{limão})|h_2) = 0.25^2 = 0.0625$$
$$P((\textit{limão}, \textit{limão})|h_3) = 0.5^2 = 0.25$$
$$P((\textit{limão}, \textit{limão})|h_4) = 0.75^2 = 0.5625$$
$$P((\textit{limão}, \textit{limão})|h_5) = 1$$

- $P(\textit{limão}|(\textit{limão}, \textit{limão})) = P(\textit{limão}|h_5) = 1$

Comparação

- A probabilidade do próximo rebuçado ser limão se já tivermos visto 2 de limão, de acordo com os 3 métodos que vimos é

N. de dados	AB	MAP	MV
2	0.73	0.75	1.00

- Será que maior valor significa melhor previsor?

Comparação

Queremos $P(d_{n+1}|\mathbf{d})$:

- AB: $\sum_{i=1}^{m} P(d_{n+1}|h_i)P(h_i|\mathbf{d})$
- MAP: $P(d_{n+1}|h_i^*)$, sendo que $h_i^* = \arg\max_{h_i} P(h_i|\mathbf{d})$. Simplificação: só interessa a hipótese mais provável.
- MV: $P(d_{n+1}|h_i^*)$, sendo que $h_i^* = \arg\max_{h_i} P(\mathbf{d}|h_i)$. Simplificações: só interessa a hipótese mais provável e são todas igualmente prováveis a priori.

Conteúdo

Naive Bayes

- O NB é um método que usa muito do que vimos atrás para construir um **classificador**.
- Partimos do princípio que os atributos são condicionalmente independentes uns dos outros: é daqui que vem o "Naive".
- Exemplo: para classificarmos uma laranja podemos dizer que é redonda, cor de laranja e com cerca de 10cm de diâmetro. O NB considera que estes 3 atributos contribuem de forma independente para a probabilidade de um dado fruto ser uma laranja.
- A probabilidade de uma observação (um ponto do conjunto de teste) pertencer à classe C é dada por

$$P(C|x_1, \ldots, x_n) = \alpha P(C) \prod_i P(x_i|C) \quad (4)$$

onde os x_i são os atributos medidos e α é uma constante.

Naive Bayes

- Para classificarmos um novo ponto, escolhemos a classe que tiver maior probabilidade de ser a correta, dadas as observações:

$$\hat{y} = \arg\max_C P(C) \prod_i P(x_i|C) \tag{5}$$

- A probabilidade a priori de cada classe, $P(C)$, é facilmente achada contando o número de pontos que pertence a cada classe e dividindo pelo total de pontos no conjunto de treino.
- Conseguimos obter os valores de $P(x_i|C)$ de uma forma semelhante.

Naive Bayes: exemplo

```
from sklearn.naive_bayes import GaussianNB
from sklearn import datasets

iris = datasets.load_iris()
gnb = GaussianNB()
y_pred = gnb.fit(iris.data, iris.target).predict(iris.data)
print("Número de erros em %d pontos de treino: %d"
        % (iris.data.shape[0],(iris.target != y_pred).sum()))

--------
Output:
--------

Número de erros em 150 pontos de treino: 6
```


Naive Bayes

- Vantagens do NB:
  - consegue lidar com problemas com muitos atributos: o número de parâmetros que usa cresce **linearmente** com o número de atributos do problema
  - computacionalmente eficiente
  - lida bem com o ruído

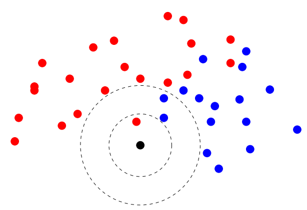
Conteúdo

k-vizinhos mais próximos

- Vejamos um classificador chamado *k*-NN (*k-nearest neighboor* ou *k*-vizinhos mais próximos).
- Consideremos que queremos classificar um ponto *A* do conjunto de teste, fazemos:
  - achar a distância de *A* a todos os pontos do conjunto de treino (pontos relativamente aos quais conhece a verdadeira classe).
  - classificamos *A* na classe mais comum entre os *k* pontos de conjunto de treino mais próximos de *A*.

k-vizinhos mais próximos

- No exemplo abaixo queremos classificar o ponto a preto numa das duas classes possíveis do problema.



- Dependendo do valor de *k* o ponto pode ser classificado como pertencendo à classe vermelha (1-NN) ou à azul (3-NN).
- No caso de existir um empate, sorteia-se a classe a atribuir entre as que empataram.
- Para um problema de 2 classes e para evitar empates, escolhe-se um valor de *k* ímpar.

Leitura recomendada

- Russell e Norvig, de sec. 20.1 até 20.2.2.