Parte II

Projeto

Greedy Hill Climber para Aprendizagem de BNC's

SECÇÃO 1.1

Objetivo

O objetivo do projeto é desenvolver um classificador baseado em redes de Bayes. O classificador é aprendido a partir de dados públicos que são fornecidos na página da disciplina. Estes dados provêm do *UCI machine learning repository*.[1]

A qualidade do classificador será avaliada por intermédio de um método chamado *stratified cross validation*.

Chama-se a atenção que, apesar de o exemplo a aplicar neste projeto se concentrarem em aplicações biomédicas nomeadamente **diagnóstico de doenças e funcionalidade de fármacos**, o domínio de aplicação do mesmo é muito mais extenso, incluindo por exemplo: OCR, previsão da bolsa de valores e de resultados de eventos desportivos, etc.

[1] `http://archive.ics.uci.edu/ml/`

SECÇÃO 1.2

Conceitos básicos

1.2.1 Classificador

Um *classificador* sobre um domínio D é simplesmente um mapa $f : D \to C$ onde C é chamado o *conjunto de classes*. Por exemplo, para o caso da base de dados *Cancer*, o conjunto de classes é $C = \{\text{benign, malignant}\}$ e um elemento em D corresponde a um tuplo de dez medições sobre o tumor. Nos casos de interesse, o domínio é sempre estruturado da seguinte forma: $D = \prod_{i=1}^{n} D_i$ onde n é o número de medições e D_i é o domínio da i-ésima medição. Assim, um elemento $d \in D$ é da forma $d = (d_1, \ldots, d_n)$.

1.2.2 Dados

O classificador é construído (ou aprendido) a partir de um conjunto de dados T. Os dados são uma amostra de elementos do domínio e respetiva classe ou seja $T = \{T_1, \ldots, T_m\}$ e $T_j = (d_{1j}, \ldots, d_{nj}, c_j)$ onde m é a dimensão dos dados, $d_{i,j} \in D_i$, $c_j \in C$ para todo o $1 \leq i \leq n$ e $1 \leq j \leq m$. Como os dados são discretizados, isto é $D_i \subseteq \mathbb{N}$, podemos ver os dados como uma matriz $m \times (n+1)$ de entradas naturais.

1.2.3 Classificar vs estimar

Uma maneira simples de classificar consiste em inferir a distribuição que gera os dados (há muitas outras maneiras). Sejam $X_1 \ldots X_n$ e C variáveis aleatórias para as quais os dados T são uma amostra multinomial do vector aleatório $\vec{V} = (X_1, \ldots, X_n, C)$. O objectivo de classificar pode-se reduzir a inferir a distribuição deste vector da seguinte forma

$$f(d_1, \ldots, d_n) = c$$

tal que $\Pr(\vec{V} = (d_1, \ldots, d_n, c)) > \Pr(\vec{V} = (d_1, \ldots, d_n, c'))$ para $c' \neq c$.

Por outras palavras, sabendo a distribuição do vector $\vec{V}$, classificar um elemento do domínio reduz-se a escolher o elemento da classe que maximiza a probabilidade de observar o elemento do domínio

com este elemento da classe (ou seja f é o estimador de máxima verosimilhança para a classe dado o elemento do domínio).

Note que a dimensão do domínio D cresce exponencialmente com o número de variáveis, e portanto inferir a distribuição (multinomial) do vector V utilizando a lei dos grandes números[2] requer dados de dimensão exponencial no número de variáveis para obter distribuições próximas das distribuições reais. Nestas condições, quando se utilizam dados pequenos, a distribuição obtida fica muito enviesada aos dados, fenómeno a que se dá o nome de *overfitting*.

1.2.4 Redes de Bayes

Para ultrapassar a limitação de não se possuir dados suficientemente grandes, supõe-se que existem dependências diretas entre as variáveis e que estas dependências estão descritas num grafo acíclico $G = (\mathcal{X}, E)$ onde $\mathcal{X} = \{X_1, \dots X_n, C\}$ tal que $(C, X_i) \in E$ para $1 \leq i \leq n$. O facto de todas as variáveis X_i dependerem de C prende-se com o facto de que, em princípio, X_i não é independente de C, pois caso contrário X_i não serve para classificar (ou estimar) C. Assim podemos decompor a distribuição de probabilidade do vector $\vec{V}$ da seguinte forma

$$\Pr(\vec{V} = (d_1, \dots, d_n, c)) = \Pr(C = c) \prod_{i=1}^{n} \Pr(X_i = d_i | \Pi_i = (d_{i,1} \dots d_{i,k_i}, c)) \tag{1.2.1}$$

onde $\Pi_i = (X_{i,1}, \dots, X_{i,k_i}, C)$ é um vector constituído pelos pais de X_i no grafo G.

Assim para obter a distribuição de $\vec{V}$ basta conhecer as distribuições C e $X_i|_{\Pi_i}$. Note que como todo os dados estão discretizados, D_C (o domínio da variável classe C) e D_i (o domínio da variável X_i) são finitos, as variáveis C e $X_i|_{\Pi_i}$ são variáveis multinomiais. Neste caso já se torna possível estimar as distribuições C e $X_i|_{\Pi_i}$, utilizando a lei dos grande números, mesmo com dados relativamente pequenos.

Com generalidade, uma rede de Bayes é um tuplo (G, Θ) onde $\Theta = \{\Theta_{i|w_i}\}_{i \in N, w_i \in D_{\Pi_i}}$ e $\Theta_{i|w_i}$ é uma distribuição multinomial para a variável X_i e D_{Π_i} é o domínio dos pais de X_i em G. Fixado um grafo G as distribuições multinomiais em Θ que maximizam a verosimilhança dos dados T são dadas por

$$\Theta_{i|w_i}(d_i) = \frac{|T_{d_i,w_i}|}{|T_{w_i}|}$$

[2] $\text{Prob}(\vec{V} = (d_1, \dots, d_n, c)) = \lim_{m \to \infty} \frac{|\{i \leq m : T_i = (d_1, \dots, d_n, c)\}|}{m}$ e T é uma amostra arbitrariamente grande.

onde T_{d_i,w_i} é o conjunto de amostras de T onde a variável X_i toma o valor d_i e os seus pais tomam o valor w_i e, de forma semelhante T_{w_i} é o conjunto de amostras de T onde os pais de X_i tomam o valor w_i. Caso T_{w_i} seja vazio, $\Theta_{i|w_i}$ deverá ser uniforme. Esta distribuição é chamada a distribuição das frequências observadas (DFO).

No entanto, observe que a DFO impõe que se $|T_{d_i,w_i}| = 0$ então $\Theta_{i|w_i}(d_i) = 0$, ou seja o facto de não observarmos um certo evento significa que este vai ter probabilidade 0. Isto não é considerado certo, pois os dados podem não ter dimensão suficiente para indicar que certo evento é impossível. Assim sendo considera-se que todos os eventos ocorreram pelo menos S vezes (a este S chama-se pseudo-contagem) e estima-se que

$$\Theta_{i|w_i}(d_i) = \frac{|T_{d_i,w_i}| + S}{|T_{w_i}| + S \times |D_i|}.$$

Assim eventos raros nunca têm probabilidade 0. Tipicamente considera-se $S = 0.5$.

1.2.5 Aprendizagem de Redes de Bayes

Pelo o que foi apresentado anteriormente, para aprender redes Bayes dado T basta aprender o grafo orientado G já que Θ é obtido das DFO's. Encontrar o grafo que maximiza a verosimilhança de T é um problema NP-completo e para o qual não se espera haver solução eficiente. Mais, ao maximizar a verosimilhança obtêm-se grafos completos e não grafos esparsos. Mas mais uma vez, para grafos acíclicos completos as DFO's associadas a dados pequenos fazem overfitting. A solução é restringir a aprendizagem a grafos com estruturas mais simples, e no caso deste projeto só serão aprendidos grafos com *grau de entrada* até um certo limite k com $k \leq 4$.

Como derivado no quadro da aula teórica, o grafo G que maximiza a verosimilhança de T é o grafo que maximiza

$$LL(G|T) = N \sum_{i=1}^{n} I_T(X_i; \Pi_i^*|C)$$

onde Π^* representa o conjunto de pais sem a classe.

Note que $I_T(X_i; \Pi_i^*|C)$ é a informação mútua condicional de X_i e do vector aleatório Π_i^* dado C medida com a distribuição de probabilidade obtida pela DFO (sem pseudo-contagens). A expressão para a

informação mútua condicional é dado por

$$I_T(X;Y|C) = \sum_{x,y,c} Pr_T(x,y,c) \log \left(\frac{Pr_T(x,y,c)Pr_T(c)}{Pr_T(y,c)Pr_T(x,c)} \right)$$

onde $Pr_T(x,y,c) = \frac{N_{x,y,c}}{N}$, $Pr_T(x,c) = \frac{N_{x,c}}{N}$, $Pr_T(y,c) = \frac{N_{y,c}}{N}$ e $Pr_T(c) = \frac{N_c}{N}$; e $N_{x,y,c}$ é o número de vezes que nos dados X toma o valor x, Y toma o valor y e C toma o valor c (e semelhante para os outros).

Como encontrar o grafo que maximiza o LL é NP-Hard, a abordagem consiste em gerar um conjunto de grafos aleatórios (com *grau de entrada* até k) e depois adicionar/retirar arestas de forma a aumentar ao máximo o LL em cada passo. O procedimento pára quando não houver mais ganho a fazer. Como o número possível de arestas a acrescentar/retirar é polinomial, o algoritmo corre em tempo polinomial em cada passo. A este algoritmo chama-se *Greedy Hill Climber* (GHC). Um facto reconhecido é que o GHC apenas encontra máximos locais, não se garantindo serem máximos globais. Por atenuar este efeito, torna-se necessário iniciar o processo com um grafo aleatório. Um dos grafos que deve ser sempre escolhido como ponto de partida é o grafo sem aresta nenhuma entre os nós X_i, e que tem apenas as arestas de C para X_i, (dito *Naive Bayes Classifier*).

1.2.6 Minimum Description Length (MDL)

Uma propriedade indesejável da verosimilhança é que

$$I_T(X_i;\Pi_i^*|C) \leq I_T(X_i;\tilde{\Pi}_i^*|C)$$

se $\Pi_i^* \subseteq \tilde{\Pi}_i^*$. Uma consequência é que para a verosimilhança, o GHC irá sempre acrescentar arestas (e nunca retirar). Pior é que a rede Bayes obtida fica demasiado enviesada (overfitting) aos dados, por ter excesso de pais por nó. A solução é considerar a abordagem da navalha de Occam (Occam's razor): a explicação mais simples é a melhor. Para tal deriva-se uma penalização (dita *minimum description length*) baseada em teoria da informação que penaliza estruturas muito complexas. O MDL de uma rede de Bayes é dada por

$$MDL(G|T) = \frac{\log_2 N}{2}|\Theta| - N\sum_{i=1}^{n} I_T(X_i;\Pi_i^*|C) \tag{1.2.2}$$

onde $|\Theta| = |D_C| + \sum_{i=1}^{n}(k_i - 1) \times q_i$ é o número de parâmetros de uma rede de Bayes e $k_i = |D_i|$ e $q_i = |D_{\Pi_i}| = |D_C| \times \prod_{X_j \in \Pi_i^*} |D_j|$.

Note que minimizar a Equação 1.2.3 é equivalente a maximizar

$$MDL^s(G|T) = N \sum_{i=1}^{n} I_T(X_i; \Pi_i^*|C) - \frac{\log_2 N}{2}|\Theta|. \tag{1.2.3}$$

SECÇÃO 1.3

Tipos de dados para a primeira entrega

Os tipos de dados a serem utilizados neste projeto são os seguintes:

1.3.1 Amostra

- `add`: recebe um vector e acrescenta o vector à amostra;
- `length`: retorna o comprimento da amostra;
- `element`: recebe uma posição e retorna o vector da amostra;
- `count`: recebe um vector de variáveis e um vector de valores e retorna o número de ocorrências desses valores para essas variáveis na amostra;
- `join`: recebe uma amostra e concatena-a à amostra;

1.3.2 Grafos orientados

- `grafoo`: método construtor recebe um natural n e retorna o grafo com n nós e sem arestas.
- `add_edge`: recebe dois nós e adiciona ao grafo uma aresta de um nó para outro.
- `add_edge`: recebe dois nós e retira ao grafo uma aresta de um nó para outro.
- `parents`: recebe um nó e retorna a lista de nós que são pais do nó.
- `MDLdelta`: recebe uma amostra e dois nós e retorna a variação de MDL causada por retirar ou colocar aresta entre os nós.
- `MDL`: recebe uma amostra e retorna o MDL score da amostra.

1.3.3 Redes Bayesianas

- `BN`: Método construtor que recebe um grafo, um conjunto de dados e um `double` S e retorna a rede de Bayes com as distribuições DFO amortizadas com pseudo-contagens S.
- `prob`: Recebe uma rede de Bayes e um vector e retorna a probabilidade desse vector.

2ª Entrega

Na segunda entrega deverão ser implementadas duas aplicações principais, ambas com interface gráfica:

- Uma aplicação que lê a amostra, aprende uma rede de Bayes e grava-a no disco;
- A aprendizagem é feita por um algoritmo *greedy* que começa com um grafo aleatório e vai adicionando ou removendo arestas até maximizar o MDL. O número máximo de pais deve ser um parâmetro da aplicação gráfica, bem como o número de grafos aleatórios com que se inicializa o processo de aprendizagem. O grafo totalmente desconexo deve ser sempre considerado como um ponto inicial.
- Uma aplicação que lê a rede de Bayes do disco, permite escrever os parâmetros do paciente e classifica-o.
- Deverá ser submetido um relatório com a explicação das opções tomadas e alterações realizadas na 1ª parte do projecto.
- As amostras a considerar devem ter S=0.5 e estão na página da unidade curricular: Breast Cancer; Diabetes; Hepatitis; Parkinsons; Thyroid.

SECÇÃO 1.4

Descrição da avaliação

A Avaliação é partida da seguinte forma, condicionada à escolha e eficiência das soluções propostas.

- Amostra 0.5 val;

- Grafo pesados 2.5 val;
- Redes de Bayes 1 val;
- Aplicações principais 4 valores
  - Aplicação de leitura e aprendizagem escrita da rede de Bayes: 3 dos 4 valores
  - Aplicação de avaliação da rede de Bayes: 1 dos 4 valores.
- Documentação 1 val
- Utilização correcta do paradigma de OO 1 val

Os alunos, durante a oral do projecto, entregam uma ficha de auto-avaliação que será disponibilizada na última semana de aulas.

Bibliografia

[CSRL01] T. Cormen, C. Stein, R. Rivest, and C. Leiserson. *Introduction to Algorithms*. McGraw-Hill Higher Education, 2nd edition, 2001.

[CSS+04] J. Carmo, A. Sernadas, C. Sernadas, F. M. Dionísio, and C. Caleiro. *Introdução à Programação em Mathematica – Segunda Edição (Introduction to Programming in Mathematica – Second Edition)*. IST Press, 2004.

[Eck02] B. Eckel. *Thinking in Java*. Prentice Hall Professional Technical Reference, 3rd edition, 2002.

[MS13] P. Mateus and A. Souto. *Aulas de Algoritmos e Modelação Matemática*. IST - em preparação, 2013.