Classificação com Redes de Bayes arborescentes

25 de Janeiro de 2022

Capítulo 1

Objectivo

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 *leave one out*.

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

Chama-se a atenção que, apesar de o exemplos a aplicar neste projeto se concentrarem em aplicações biomédicas e. em bioinformática nomeadamente **diagnóstico de doenças, classificação de organismos**, 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.

Capítulo 2

Conceitos básicos

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 da 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 $d = (d_1, \ldots, d_n)$.

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 respectiva 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.

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

2.4 Redes de Bayes

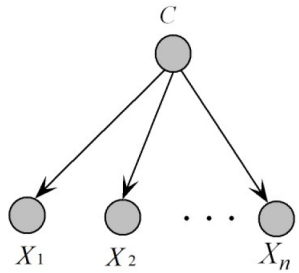
Para ultrapassar a limitação de não se possuir dados suficientemente grandes, supõe-se que existem dependências diretas entre as variáveis e que estas dependências estão descritas num grafo acíclico $G = (\mathcal{X}, E)$ onde $\mathcal{X} = \{X_1, \ldots X_n, C\}$. Para simplificar a notação, denotamos a classe C como a variável X_{n+1} e os valores que a classe são denotados pela conjunto D_{i+1}. Assim podemos decompor a distribuição de probabilidade do vector $\vec{V}$ da seguinte forma

$$\Pr(\vec{V} = (x_1, \ldots, x_n, x_{n+1})) = \prod_{i=1}^{n+1} \Pr(X_i = d_i | \Pi_i = (d_{i,1} \ldots d_{i,k_i})) \tag{2.4.1}$$

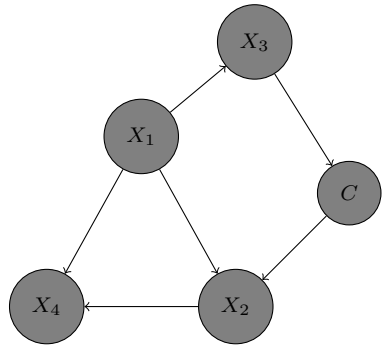
onde $\Pi_i = (X_{i,1}, \ldots, X_{i,k_i})$ é um vector constituido pelos pais de X_i no grafo G.

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, os conjuntos D_i's (o domínio da variável X_i) são finitos,e assim as variáveis $X_i|_{\Pi_i}$ são variáveis multinomiais. Neste caso já se torna possível estimar as distribuições $X_i|_{\Pi_i}$ utilizando a frequências, mesmo com dados relativamente pequenos.

Alguns classificadores baseados em redes de Bayes são a *Naive Bayes*:



Mas as dependências podem ser generalizadas



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

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

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).

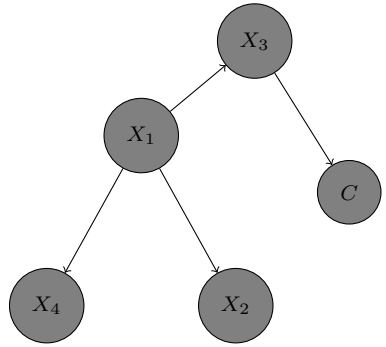
No entanto, observe que a DFO impõe que se $|T_{d_i,w_i}| = 0$ então $\Theta_{i|w_i}(d_i) = 0$, ou seja o facto de não observarmos um certo evento significa que este vai ter probabilidade 0. Isto não é considerado certo, pois os dados podem não ter dimensão suficiente para indicar que certo evento é 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 $S = 0.5$.

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 uma possibilidade é restringir a estrutura a uma árvore (Tree Bayesian network) onde todos os nós têm no máximo um pai, com excepção da raiz. Normalmente exige-se que a classe tenha um papel central, e apareça ligado a vários nós, mas neste projecto não o vamos fazer. Assim um árvore possível é:



Como se pode derivar, o grafo G que maximiza a verosimilhança de T é o grafo que maximiza

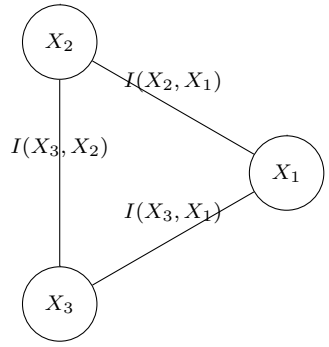
$$LL(G|T) = N \sum_{i=1}^{n+1} I_T(X_i; \Pi_i).$$

Note que $I_T(X_i; \Pi_i)$ 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

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

onde $Pr_T(x, y) = \frac{N_{x,y}}{N}$, $Pr_T(x) = \frac{N_x}{N}$, $Pr_T(y) = \frac{N_y}{N}$; e $N_{x,y}$ é o número de vezes que nos dados X toma o valor x e Y toma o valor y. De forma semelhante, N_x é o número de vezes que nos dados X toma o valor x. Considera-se no somatório acima que $0 \times \log(0) = 0$.

Como encontrar o grafo que maximiza o LL é NP-Hard, a abordagem consiste em restringir a pesquisa há árvore com peso maximal conhecida como a *Maximal Spanning Tree*. Esta árvore é obtida construindo um grafo completo pesado com $I_T(X_i, X_j)$ sobre todos os nós incluindo a classe:



Só devem ser consideradas variáveis cujo o domínio tenha cardinalidade superior a 1.

Capítulo 3

Entrega

As classes a entregar neste projeto são os seguintes:

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;
- `domain`: recebe uma posição e retorna o número de elementos possíveis da variável dessa posição;
- `count`: recebe um vector de variáveis e um vector de valores e retorna o número de ocorrências desses valores para essas variáveis na amostra;

3.2 Floresta

- `forest`: método construtor recebe um natural n e retorna uma floresta com n nós e sem arestas.
- `set_parent`: recebe dois nós n e m e e torna o pai de n o nó m;
- `treeQ`: retorna `true` sse a floresta é uma árvore;

3.3 Grafos pesados

- `grafoo`: método construtor recebe um natural n e retorna o grafo com n nós e sem arestas.
- `add_edge`: recebe dois nós e um peso e adiciona ao grafo uma aresta entre os dois nós com este peso
- `max_spanning_tree`: retorna uma árvore de extensão maximal

3.4 Redes Bayesianas Arbóreas

- `BN`: Método construtor que recebe uma árvore cuja raiz é a classe, um conjunto de dados e um `double` S e constrói a rede de Bayes com a estrutura da árvore e com as distribuições DFO amortizadas com pseudo-contagens S.

- `prob`: Recebe um vector e retorna a probabilidade desse vector, de acordo com a fórmula 2.4.1. para a rede Bayes em causa.

Deverão ser implementadas duas aplicações principais, ambas com interface gráfica:

- Uma aplicação que lê a amostra, aprende uma rede de Bayes arbórea e grava-a no disco;
- A aprendizagem é feita pelo algoritmo de Prim para encontrar a árvore de extensão maximal .
- Uma aplicação que lê a rede de Bayes do disco, permite escrever os parâmetros do amostra e classifica-o.
- As amostras a considerar devem ter S=0.5 e estarão na página da unidade curricular
- Serão detalhadas todas as componentes de avaliação.