CENTRO UNIVERSITÁRIO FEI
Leonardo Contador Neves – 118315-1

Tópicos Especiais em Aprendizagem:

Classificador Naive Bayes

São Bernardo do Campo, SP

2018

SUMÁRIO

1 INTRODUÇÃO

Este relatório busca a implementação do algoritimo Naive Bayes, um algoritmo que faz uso do teorema de Bayes e aplica probabilidades sobre acontecimentos para prever eventos não supervisionados, atuando como classificador Bayesiano.

2 REVISÃO BIBLIOGRÁFICA

O classificador Naive Bayes, que aplica o teorema de Bayes, está entre os métodos de classificação mais usados hoje em aprendizagem de máquina. O modelo do classificador é chamado de ingênuo (Naive), por assumir que os atributos de entrada são todos condicionamente independentes, ou seja, podemos falar que as informações de um evento não são informativas para outro evento. Assim, mesmo com a premissa de ingenuidade do modelo ser simplista, podemos obter excelentes resultados dentre outros classificadores.

Até meados do século XVIII, problemas relacionados a probabilidade de certos eventos, dadas certas condições, estavam bem resolvidos. Por exemplo, dado um número específico de bolas negras e brancas em uma urna, qual é a probabilidade de eu sortear uma bola preta? Tais problemas são chamados de problemas de “forward probability”. Porém, logo, o problema inverso começou a chamar a atenção dos matemáticos da época: Dado que uma ou mais bolas foram sorteadas, o que pode ser dito sobre o núumero de bolas brancas e pretas na urna?

Thomas Bayes, um ministro ingês do século XVIII, foi o primeiro a formalizar uma teoria para problemas desta natureza que foi vista como revolucionária no meio científico da época.

3 METODOLOGIA

Para trabalharmos com o classificador Bayesiano, vamos primeiramente entender como o teorema de Bayes pode ser aplicado. O conceito inicial é o cálculo de algumas propabilidades para os eventos observados. Na equação à baixo, podemos aplicar diretamento o teorema de Bayes para saber a probabilidade de uma hipótese h acontecer dado uma evidência D.

Coma aplicação, para chegar nesse resultado probabilistico, temos que multiplicar a propabilidade de observar a evidência D, dado que a hipótese h aconteceu, pela probabilidade

à priori da hipótese h acontecer e ambas divididas pela probabilidade a priori dos dados de treinamento D (nossas evidências).

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

Equação 1: Teorema de Bayes.

Com o teorema de Bayes já exposto, precisamos agora chegar em mais três conceitos para construir nosso classificador Bayesiano. O primeiro deles fala sobre a ocorrência de An eventos mutuamente exclusivos e que formam uma partição do evento B, denominado teorema da probabilidade total, podendo ser representado pela equação a seguir.

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

Equação 2: Teorema da probabilidade total.

O conceito seguinte é sobre a ocorrência simuntânea de mais de um evento e o calculo da probabilidade envolvido. Assim, a equação a seguir, permite calcular a probabilidade de ocorrência simultânea de vários eventos a partit das probabilidades condicionais.

$$P(A_1 \cap \cdots \cap A_n) = P(A_n / A_1 \cap \cdots \cap A_{n-1})P(A_1 \cap \cdots \cap A_{n-1})$$

$$\Downarrow$$

$$P(A_1 \cap \cdots \cap A_n) = P(A_n / A_1 \cap \cdots \cap A_{n-1}) \cdots P(A_2 / A_1)P(A_1)$$

Equação 3: Teorema da multiplicação de probabilidades.

O último tópico é o conceito de probabilidade máxima à posteriori (MAP) , esse conceito permite estimar a probabilidade máxima definindo o valor que ocorre com maior frequência em um conjunto de dados da distribuição a posteriori. Assim, podemos usar o

MAP para obter uma estimativa de uma quantidade não observada com base em dados empíricos.

Para o cálculo, vamos primeiro calcular para cada hipótese h a probabilidade a posteriori pela equação 1 (Teorema de Bayes) e a seguir vamos escolher a hipótese hMAP de maior probabilidade à posteriori, como na equação a seguir, assim encontramos a maior probabilidade de uma hipótese ocorrer para os dados observados.

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

Equação 4: Escolha da maior probabilidade a posteriori.

Dados esses conceitos podemos então chegar no classificador Bayesiano, onde devemos calcular as probabilidades para n eventos e usamos a teoria do MAP para chegar na maior probabilidade para aquela hipótese. A equação a seguir ilustra o funcionamento do classificador usando os conceitos anteriormente descritos.

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

Equação 5: Classificador Bayesiano Ingênuo.

Podemos ainda, dados os presupostos e teoremas apresentados anteioromente, fazer uma modificação na nossa equação do classificador de Bayes ingênuo para usar em dados contínuos no tempo. O primeiro ponto antes de fazer essas modificações é assumir que os dados que serão usados podem ser distribuídos de acordo com uma distribuição gaussiana. Com base nesse presuposto, vamos precisar calcular a média e o desvio padrão do nosso conjunto de atributos.

Depois dos cálculos mencionados, podemos aplicara equação a seguir, que vai calcular a probabilidade do dado X pertencer à conjunto ou classe Cn e assim podemos usar a teoria MAP diante de cada cálculo para cada classe para nos dizer a classe que o dado tem a maior probabilidade de pertencer.

$$\hat{P}(X_j \mid C = c_i) = \frac{1}{\sqrt{2\pi}\sigma_{ji}} \exp\left(-\frac{(X_j - \mu_{ji})^2}{2\sigma_{ji}^2}\right)$$

Equação 6: Teorema da probabilidade para dados contínuos.

4 DESENVOLVIMENTO

A implementação do algoritmo de classificação Bayesiano Ingênuo foi feita neste trabalho através de uma classe na linguagem de programação *Python* que tem um métodos principal. O método *predict()*, faz o cálculo de todas as equações do classificador e tem como saída a classe que o dado pode ser agrupado e a probabilidade.

4.1 CLASSE PARA CÁLCULO DO CLASSIFICADOR NAIVE BAYES

4.1.1 Método de predição P*redict*

Esse é o método principal da implementação do classificador Bayesinano ingênuo. Como entrada temos o conjunto de informações (divididas anteriormente em dados para treinamento e dados para teste proporção de 20 por cento para teste) e o valor que vamos prever a que classe ele está.

Primeiramente temos que cálcular a média de cada grupo do meu conjunto de informações seguido pelo cálculo do desvio padrão. A imagem a seguir representa a criação das listas de média e desvio padrão para cada classe, *mean_classes* e *variance_classes* respectivamente.

```
def predict(self, data, value, targets_name='target'):

    self.mean_classes = []
    self.variance_classes = []
    self.classes = max(data[targets_name]) + 1
    dim = len(data.keys()) - 1

    for x in range(self.classes):
        self.mean_classes.append(np.mean(data[data[targets_name] == x].iloc[:,0:dim].values))
        self.variance_classes.append(np.std(data[data[targets_name] == x].iloc[:,0:dim].values))
```

Com as médias e desvios padrões clculados para cada classe vamos aplicar diretamente a equação 6 anteriormente descrita, onde ela nos retornará, para cada classe, o valor da probabilidade da distribuição. A imagem a seguir mostra a criação da variável *probabilities* que vai armazenar cada um desses cáculos em uma lista de probabilidades.

```
probabilities = []
self.probabilities = []
for x in range(self.classes):
    exponent = (-(np.power(value-self.mean_classes[x],2)/(2*self.variance_classes[x]*self.variance_classes[x])))
    probabilities.append((np.power((1 / (np.sqrt(2*math.pi)*self.variance_classes[x])),exponent)[0]))
```

Para cada argumento de entrada, vamos pegar a resposta dos cálculos feitos anteriormente e armazenar na variável auxiliar *c_aux* a produtória das probabilidades para cada argumento do meu conjunto de informações, de modo que todos os argumentos compoem a resposta final pela equação 3.

```
c_aux = []
for x in probabilities:
    prob = x/sum(probabilities)
    self.probabilities.append(prob)
    a = 1
    for i in prob:
        a *= i
    c_aux.append(a)
```

Por fim, o método faz uso da função *max* para retornar o maior valor de probabilidade e qual classe esse valor pertence com a operação *index*. A imagem a seguir mostra essas operações sobre a lista de porbabilidades *c_aux.*

```
return c_aux.index(max(c_aux)), self.probabilities[c_aux.index(max(c_aux))]
```

5 RESULTADOS

Para a resolução dos exercícios propostos, que ao todo são 2, foram usadas ambas as equações 1 e 6. Para o primeiro exercício, vamos usar a aplicação direta do teorema de Bayes sobre a tabela a seguir para dizer se uma pessoa possui um computador Mac, qual telefone ela deve possuir.

name	laptop	phone
Kate	PC	Android
Tom	PC	Android
Harry	PC	Android
Annika	Mac	iPhone
Naomi	Mac	Android
Joe	Mac	iPhone
Chakotay	Mac	iPhone
Neelix	Mac	Android
Kes	PC	iPhone
B'Elanna	Mac	iPhone

Tabela 1: Dados do exercício 1.

Com o conjunto de informações, basicamente temos duas classes para nossos dados, a classe "iPhone" e a classe "Android" e temos uma variável de entrada, o computador. Precisamos agora calcular a probabilidade do argumento para cada classe. Calculando para cada tipo de argumento na coluna laptop a classe pertencente, temos que:

laptop	Phone = iPhone	Phone = Android
Mac	4/5	2/5
PC	1/5	3/5

E a probabilidade do Celular ser "iPhone" ou "Android" são, respectivamente, ½ e ½.

Assim, para um novo dado de entrada, temos que o indivíduo qualquer tem um computador MAC. Para calcular quais as probabilidades de ele ter algum dos celulares, vamos multiplicar 4/5 vezes ½, resultando em 0,4 para o caso do celular iPhone e multiplicar 2/5 por ½, resultando em 0,2 para o caso do "Android". Usando o teorema do MAP em {0,4 ; 0,2}, pegamos o maior valor de probabilidade, que representa o celular "iPhone" como resposta para esse caso.

Para os testes com as bases solicitadas no exercício 2, vamos fazer a aplicação do teorema contínuo da equação do classificador de Bayes. A primeira base testada é a base Iris, onde um valor de teste foi posto e o algoritmo em 78% dos casos mostrou acerte na resposta a que classe pertencia o dado. A imagem a seguir mostra a resposta do algoritmo para uma das entredas testadas para as seguintes bases: Iris, Qualidade de Vinhos

```
leonardo@leonardo:~/Documents/Mestrado/SpecialTopicsinLearning$ python3 cla
ss5NaiveBayes.py
Entrada: [[4.7 3.2 1.6 0.2]] Target: Class [[0]]

Class: 0
Probability: [0.68030456 0.51890858 0.32630625 0.18397199]

leonardo@leonardo:~/Documents/Mestrado/SpecialTopicsinLearning$
```

```
leonardo@leonardo:~/Documents/Mestrado/SpecialTopicsinLearning$ python3 cla
ss5NaiveBayes.py
Entrada: [[ 8.      0.71    0.      2.6     0.08   11.     34.      0.9976
 3.44
   0.53    9.5   ]] Target: Class [[2]]

Class: 2
Probability: [1.63755610e-01 1.59381087e-01 1.54065993e-01 1.69517773e-01
 1.54703978e-01 1.38828371e-01 6.83087474e-06 1.61306750e-01
 1.72052403e-01 1.58108409e-01 1.53064043e-01]

leonardo@leonardo:~/Documents/Mestrado/SpecialTopicsinLearning$
```

```
leonardo@leonardo:~/Documents/Mestrado/SpecialTopicsinLearning$ python3 cla
ss5NaiveBayes.py
Entrada: [[ 7.5      0.71     0.       1.6      0.092   22.      31.
0.99635
   3.38     0.58    10.     ]] Target: Class [[3]]

Class: 2
Probability: [1.66474061e-01 1.59381087e-01 1.54065993e-01 1.64901547e-01
 1.54798833e-01 1.27563988e-02 6.84822420e-05 1.61298670e-01
 1.71912709e-01 1.58467095e-01 1.48692357e-01]

leonardo@leonardo:~/Documents/Mestrado/SpecialTopicsinLearning$
```

5 CONCLUSÃO

O algoritmo de classificação Bayes ingênuo se mostrou bastante rápido em termos computacionais, mesmo quando um conjunto de informação de mais de 200 mil linhas foi posto em teste. A eficiência também foi um ponto forte nos resultados, que mesmo para um conjunto de 10 classes, o algoritmo teve um bom acerto. O único ponto observado foi a necessidade de um tratamento anterior junto à base de dados Iris, visto que uma das classes não foi bem classificada pelo algoritmo, tendo probabilidades bem próximas das outras classes. No geral, seu ainda grande uso nos dias de hoje é notado por esses pontos fortes podendo assim ser usado em tempo real por sua grande velocidade e seu uso de poucas observações.

REFERÊNCIAS

LEUNG, K. Ming. Naive bayesian classifier. **Polytechnic University Department of Computer Science/Finance and Risk Engineering**, 2007.

RISH, Irina et al. An empirical study of the naive Bayes classifier. In: **IJCAI 2001 workshop on empirical methods in artificial intelligence**. New York: IBM, 2001. p. 41-46.

SPIEGEL, Murray R.; STEPHENS, Larry J. **Estadística/Theory and problems of statistics**. McGraw-Hill,, 2009.

NumPy Reference. Disponível em: <https://docs.scipy.org/doc/numpy-1.15.1/reference/index.html>. Acesso em: 10 set. 2018.