

Atividade 6: Naive-Bayes

Naive Bayes é uma técnica utilizada para classificação de dados baseada na probabilidade de Bayes. Este trabalho apresenta uma implementação do método em linguagem Java 1.8, para a disciplina de tópicos especiais em aprendizagem.

Introdução

Este artigo mostra como a aplicação que implementa Naive-Bayes foi codificada. Para isto, uma breve explicação é feita sobre cada método desenvolvido, que compõe o exercício. O objetivo do trabalho é implementar um modelo simples que atenda os requisitos da disciplina.

Métodos

Os métodos Naive Bayes são um conjunto de algoritmos de aprendizagem supervisionados baseado na aplicação do teorema de Bayes com a suposição "ingênua" de independência entre cada par de recursos. Dada uma variável de classe y e um vetor de função dependente x_1 a x_n , o teorema de Bayes indica a seguinte relação:

$$P(y | x_1, \dots, x_n) = \frac{P(y)P(x_1, \dots, x_n | y)}{P(x_1, \dots, x_n)}$$

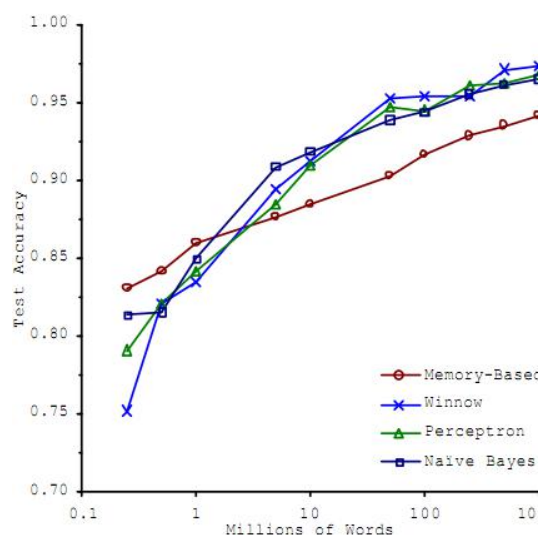
Usando a suposição de independência ingênua:

$$P(x_i | y, x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) = P(x_i | y),$$

Por tudo isso, essa relação é simplificada para:

$$P(y | x_1, \dots, x_n) = \frac{P(y) \prod_{i=1}^n P(x_i | y)}{P(x_1, \dots, x_n)}$$

Apesar de seus pressupostos aparentemente simplificados, os classificadores Bayes ingênuos trabalharam bastante bem em muitas situações do mundo real, famosa classificação de documentos e filtragem de spam. Eles exigem uma pequena quantidade de dados de treinamento para estimar os parâmetros necessários. Além disso, com dados suficientes, os resultados obtidos são primos:



Desenvolvimento

Os métodos foram implementados na linguagem de programação Java. Diferentes métodos foram criados que em conjunto implementam Naive

Bayes.

Classe Matriz

```
// create M-vs-N matrix of 0's

public Matrix(int M, int N) {

    this.M = M;

    this.N = N;

    data = new double[M][N];

}

// create matrix based on input array

public Matrix(double[][] data) {

    M = data.length;

    N = data[0].length;

    this.data = new double[M][N];

    for (int i = 0; i < M; i++)

        for (int j = 0; j < N; j++)

            this.data[i][j] = data[i][j];

}

public static Matrix random(int M, int N) {

    Matrix A = new Matrix(M, N);

    for (int i = 0; i < M; i++)

        for (int j = 0; j < N; j++)

            A.data[i][j] = Math.random();

}
```

```
        return A;

    }

    public int rowsLen() {

        return this.M;

    }

    public int colLen() {

        return this.N;

    }

    // create and return the N-vs-N identity matrix

    public static Matrix identity(int N) {

        Matrix I = new Matrix(N, N);

        for (int i = 0; i < N; i++)

            I.data[i][i] = 1;

        return I;

    }

}
```

Formula NaiveBayes

```
double x = input[input.length - 1];

double prob = 0, fator1 = 1, fator2 = 1, fator3 = 1,
f1, f3;

Matrix probAtrib = new Matrix(1, A.colLen());

// Calcula a prob de cada atributo
```

```

for (int j = 0; j < A.colLen(); j++) {

    f1 = 0;

    f3 = 0;

    for (int i = 0; i < A.rowsLen(); i++) {

        if (A.data[i][j] == input[j]) {

            f3++;

            if (A.data[i][A.colLen() - 1] == x)

        {

            f1++;

        }

    }

}

f1 = f1 / A.rowsLen();

f3 = f3 / A.rowsLen();

fator1 = fator1*f1;

fator3 = fator3*f3;

}

for (int i = 0; i < A.rowsLen(); i++) {

    if (A.data[i][A.colLen() - 1] == x) {

        fator2++;

    }

}

```

```

}

fator2 = fator2 / A.rowsLen();

System.out.println(fator1);

System.out.println(fator2);

System.out.println(fator3);

prob = (fator1 * fator2) / fator3;

return prob;

```

Resultados

Abaixo estão representados de forma matricial os resultados obtidos a partir dos dados de exemplo de entrada.

Dados de treinamento

Id	Casa própria	EstCivil	Rendim.	Mau Pagador
1	S	Solteiro	alto	NÃO
2	N	Casado	médio	NÃO
3	N	Solteiro	baixo	NÃO
4	S	Casado	alto	NÃO
5	N	Divorc.	médio	SIM
6	N	Casado	baixo	NÃO
7	S	Divorc.	alto	NÃO
8	N	Solteiro	médio	SIM
9	N	Casado	baixo	NÃO
10	N	Solteiro	médio	SIM

Utilizando os dados de treinamento acima, foi possível indicar o resultado para o exercício proposto:

Casa Própria	Estado Civil	Rendim.	Mau pagador
N	Divorc.	médio	?

Foi gerada a Matriz de equivalencia abaixo, de forma a permitir que os dados sejam interpretados de forma genérica:

Naive Bayes

1.0000	0.0000	2.0000
0.0000		
0.0000	1.0000	1.0000
0.0000		
0.0000	0.0000	0.0000
0.0000		
1.0000	1.0000	2.0000
0.0000		
0.0000	-1.0000	1.0000
1.0000		
0.0000	0.0000	0.0000
0.0000		
1.0000	-1.0000	2.0000
0.0000		
0.0000	0.0000	1.0000
1.0000		
0.0000	1.0000	0.0000
0.0000		
0.0000	0.0000	1.0000
1.0000		

É mau pagador?

R: Sim

Dados de treinamento

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

Dada a tabela ao lado, se uma pessoa possui um mac, qual telefone ela deve possuir?

R. iPhone

Conclusão

A partir do algoritmo Naive Bayes implementado foi possível executar os exemplos indicados com sucesso, absorvendo os conceitos teóricos e as aplicações. Seus resultados se mostraram sólidos, corroborando com o método de Bayes, obtendo resultados razoáveis mesmo quando os dados de entrada possuíam certa correlação.

Além do exercício, foi realizado um teste com chatbot, para classificação de palavras de interesse, caso considerado não recomendado, por conta da correlação existente entre as palavras, ainda assim, foi possível obter uma acurácia de 84% para os casos testados,

mostrando o bom desempenho do algoritmo.

Referências

1. Scikit. 1.9, Naive Bayes - Disponível em: http://scikit-learn.org/stable/modules/naive_bayes.html. Acesso em: 01 de Nov. de 2017
2. Naive Bayes classifier - Disponível em (https://en.wikipedia.org/wiki/Naive_Bayes_classifier). Acesso em: 01 de nov. de 2017