

Atividade 4: K-Means

K-Means é um método de quantização vetorial que tem como objetivo separar amostras, de acordo com suas características. Este trabalho apresenta uma implementação do método em linguagem Java 1.8, para a disciplina de tópicos especiais em aprendizagem.

Introdução

k-Means clustering é um método de quantização vetorial que tem como objetivo separar amostras, de acordo com suas características, em um determinado número de clusters. Para alcançar esse objetivo, o algoritmo inicia com K pontos (que devem ser selecionados pelo usuário) centrais de clusters e executa dois passos. O primeiro consiste em separar um conjunto de amostras que estejam mais próximas a um ponto central, formando assim os clusters. O segundo ponto calcula a média das amostras que estão em cada cluster para formar um novo ponto central. O processo é repetido até que o problema convirja.

Métodos

Kmeans matematicamente, para um conjunto k de centros m_1, \dots, m_k a fórmula para a etapa de atribuição se dá por:

$$S_i^{(t)} = \{x_p : \|x_p - m_i^{(t)}\|^2 \leq \|x_p - m_j^{(t)}\|^2 \forall j, 1 \leq j \leq k\}$$

As médias são calculadas a partir da fórmula nas atualizações:

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j$$

Desenvolvimento

Os métodos foram implementados acertivamente para os problemas propostos. Diferentes métodos foram criados que em conjunto implementam o método dos mínimos quadrados.

Matriz de Dispersão

```
for (int c=0; c<ClassVector.size(); c++){

    int N = ClassVector[c].size();

    Sb = this->somaMatrix( Sb,

    this->multiplicaMatrix( N,

    this->multiplicaMatrix( this->transpose(this->subMa
    trix(meanSample, meanClass[c])),

    this->transpose(this->transpose(this->subMatrix(me
    anSample, meanClass[c])))

    )

    )

    );
```

```
}
```

K-Means

```
for(int i = 0; i < total_values; i++){

    sum += pow(clusters[0].get_central_value(i) -
        point.get_value(i),2.0);

    min_dist = sqrt(sum);

    for(int i = 1; i < K; i++){

        double dist;

        Sum = 0.0;

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

            sum += pow(clusters[i].get_central_value(j) - point.get_value(j),
                2.0);

        dist = sqrt(sum);

        if(dist < min_dist){

            min_dist = dist;

            id_cluster_center = i;

        }

    }

}
```

Calculo da Matriz Inversa

```
public static double[][] calculaInversa(double[][] matriz, int
    tamanho) {

    double[][] inversa = new double[tamanho][tamanho];

    double[][] adjunta = calculaAdjunta(matriz, tamanho);

    double determinante = calculaDeterminante(matriz,
        tamanho);

    for (int i = 0; i < tamanho; i++){

        for (int j = 0; j < tamanho; j++){

            inversa[i][j] = (1/determinante) * adjunta[i][j];

        }

    }

    return inversa;

}
```

Calculo da Matriz Transposta

```
public static double[][] calculaTransposta(double[][] matriz, int
    tamanho) {

    double[][] transp = new double[tamanho][tamanho];

    for (int i = 0; i < tamanho; i++){

        for (int j = 0; j < tamanho; j++){

            transp[i][j] = matriz[j][i];

        }

    }

    return transp;

}
```

```

for (int i = 0; i < tamanho; i++) {

    for (int j = 0; j < tamanho; j++) {

        transp[j][i] = matriz[i][j];

    }

}

return transp;
}

```

Resultados

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

Exemplo retirado do Slide

Cluster 1

Valor do cluster: 2, 62857 6, 5
total de pontos: 7

Cluster 2

Valor do cluster: 1, 36667 2, 15
total de pontos: 6

Cluster 3

Valor do cluster: 4, 775 3, 05
total de pontos: 4

Iris Dataset

Cluster 1

Valor do cluster: [5, 88 2,74 4,39
1,43]
iris-versicolor: 47

iris-virginica: 14
iris-setosa: 0
total de pontos: 61

Cluster 2

Valor do cluster: [6, 86 3,08 5,72
2,05]
iris-versicolor: 3
iris-virginica: 36
iris-setosa: 0
total de pontos: 39

Cluster 3

Valor do cluster: [5, 01 3,42 1,46
0,24]
iris-versicolor: 0
iris-virginica: 0
iris-setosa: 50
total de pontos: 50

Seeds Dataset

Cluster 1

Valor do cluster: [11, 9644 13,
2748 0, 8522 5, 22929 2, 87292
4, 75974 5, 08852]
Canadian: 68
Kama: 9
Rosa: 0
total de pontos: 77

Cluster 2

Valor do cluster: [14, 6485 14,
4604 0, 879167 5, 56378 3,
2779 2, 64893 5, 19232]
Canadian: 2
Kama: 60
Rosa: 10
total de pontos: 72

Cluster 3

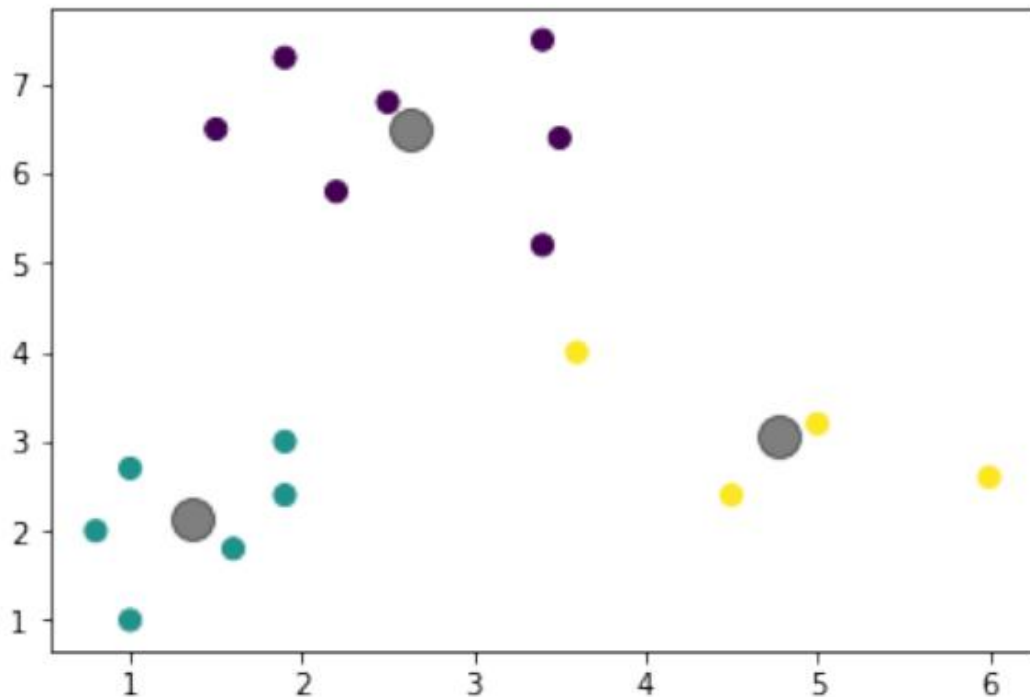


Figura 1. Resultado plotado

Valor do cluster: [18, 7218 16, 2974 0, 885087 6, 20893 3, 72267 3, 60359 6, 0661]

Canadian: 0

4Kama: 1

Rosa: 60

total de pontos: 61

Lenses Dataset

Cluster 1

Valor do cluster: [1, 46154 1, 46154 1, 07692 3]

hard-lenses: 4

soft-lenses: 4

no-lenses: 5

total de pontos: 13

Cluster 2

Valor do cluster:[2 2 2 3]

hard-lenses: 0

soft-lenses: 1

no-lenses: 1

total de pontos: 2

Cluster 3

Valor do cluster: [1, 44444 1, 44444 2 1, 55556]

hard-lenses: 4

soft-lenses: 3

no-lenses: 2

total de pontos: 9

Conclusão

A partir do algoritmo implementado foi possível executar os exemplos indicados com sucesso, absorvendo os conceitos teóricos e as aplicações.

Referências

1. Trevor Hastie, Robert Tibshirani, and Jerome Friedman. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer, 2001.