Relatório de trabalho prático
Métodos Estatísticos
Algoritmo de K-Médias

João Paulo Pizani Flor
Mauricio Oliveira Haensch

13 de junho de 2011

1 Introdução

Para o módulo de métodos estatísticos da disciplina, o segundo trabalho prático tem como objetivo implementar o algoritmo K-Médias, que agrupa um conjunto de dados em *k clusters*, utilizando o teste de Fisher para avaliar o resultado obtido. A descrição do trabalho pode ser vista a seguir:

Implemente o método das k-Médias de maneira que o programa gerado seja capaz de ler o mesmo conjunto de dados que foi sugerido para a técnica da classificação em árvore. A escolha de k pode ser tanto um valor entrado manualmente como pode ser um valor dentro de uma faixa que você dá; se você usar uma faixa, faça o método iterar por todos os valores de k dentro da faixa, guardando em separado cada resultado. Implemente o teste de Fisher usando a seu critério o cálculo da distribuição F ou então uma tabela de valores críticos de F para validar o resultado.

2 Ferramentas utilizadas

A linguagem utilizada para implementação do programa era de livre escolha dos alunos, assim como outras ferramentas que pudessem ser utilizadas para representação gráfica dos padrões, por exemplo. As ferramentas escolhidas para a implementação do trabalho foram:

Linguagem de programação - Python: Escolhida por ser uma linguagem já bem conhecida pela equipe, que está sendo utilizada para todos os trabalhos práticos até então. Conta com boas bibliotecas e documentação, além de agilizar o desenvolvimento.

Interface gráfica - PyQt: *Binding* do framework gráfico Qt para a linguagem Python, escolhida por já ser conhecida pelos integrantes e também é o framework gráfico sendo utilizado nos demais trabalhos da disciplina.

NumPy: Biblioteca[1] da linguagem Python utilizada para computação científica, com diversas estruturas e funções otimizadas. Para este trabalho, utilizamos algumas de suas funções de álgebra linear, para fazer operações com matrizes (transposição) e cálculo de média e desvio padrão de um conjunto de dados.

O trabalho foi desenvolvido no ambiente Linux/Ubuntu e para executá-lo são necessários alguns pacotes[2] para a interface gráfica:

- pyqt4-dev-tools
- libqtgui4
- libqtcore4

[1] O pacote para essa biblioteca no ambiente Linux/Ubuntu é *python-numpy*.

[2] Os nomes dos pacotes citados são específicos para a distribuição Ubuntu, para outras distribuições devem ser procurados os pacotes correspondentes.

3 Alguns resultados

Para este trabalho, é possível carregar um conjunto de dados separados por vírgulas (.csv) para que seja utilizado para o algoritmo de K-Médias, com *k* sendo um valor passado pelo usuário. Os dados de entrada a serem carregados devem seguir o padrão demonstrado abaixo, com um caso por linha, com os valores dos atributos podendo ser inteiros ou ponto flutuante:

```
atributo1,atributo2,atributo3,atributo4
atributo1,atributo2,atributo3,atributo4
atributo1,atributo2,atributo3,atributo4
atributo1,atributo2,atributo3,atributo4
```

O algoritmo de K-Médias possui um princípio de funcionamento simples, seguindo os passos seguintes:

1. São encontrados **k** pontos aleatórios que irão definir as classes na primeira tentativa de classificação;
2. Os demais pontos são classificados em uma das categorias de acordo com alguma métrica de distância;
3. Após todos os pontos terem sido classificados, são calculados novos pontos médios para representar cada classe;
4. Até que não haja nenhuma mudança dos pontos médios, repita os dois passos anteriores.

Após aplicarmos o algoritmo de K-Médias, resultando no agrupamento dos padrões de entrada em **k** classes, devemos aplicar um teste de fisher, avaliando o quão boa foi a divisão encontrada pelo algoritmo. Com essa abordagem, é possível detectar o melhor número de *clusters* ao operar numa faixa de classes desejadas pelo usuário.

Para efeito de testes, geramos um conjunto de dados de duas dimensões para aplicarmos os algoritmos implementados e poder ser feita a visualização dos resultados de forma gráfica. Na figura 1 estão os dados sem a aplicação do algoritmo de K-Médias. Aplicando o algoritmo para encontrar duas classes, o resultado pode ser visto na figura 2.

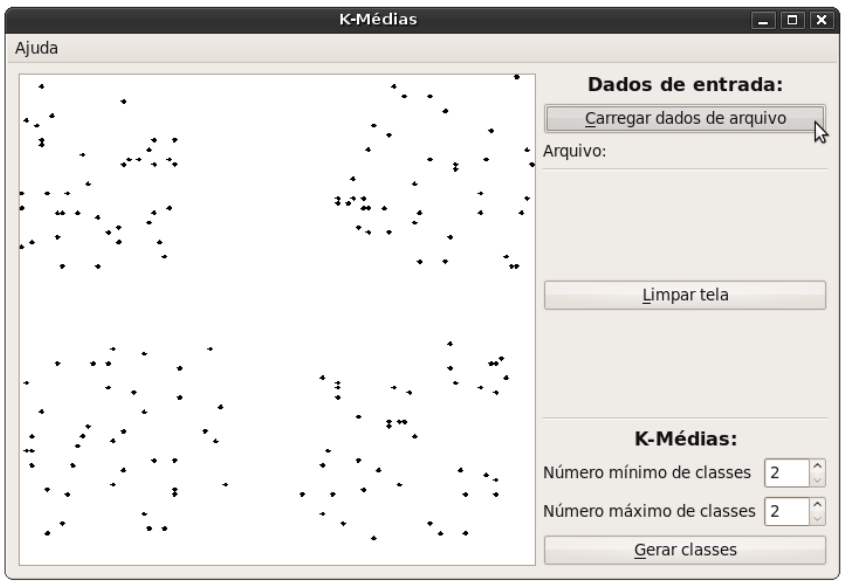

Figura 1: Dados carregados, antes da classificação.

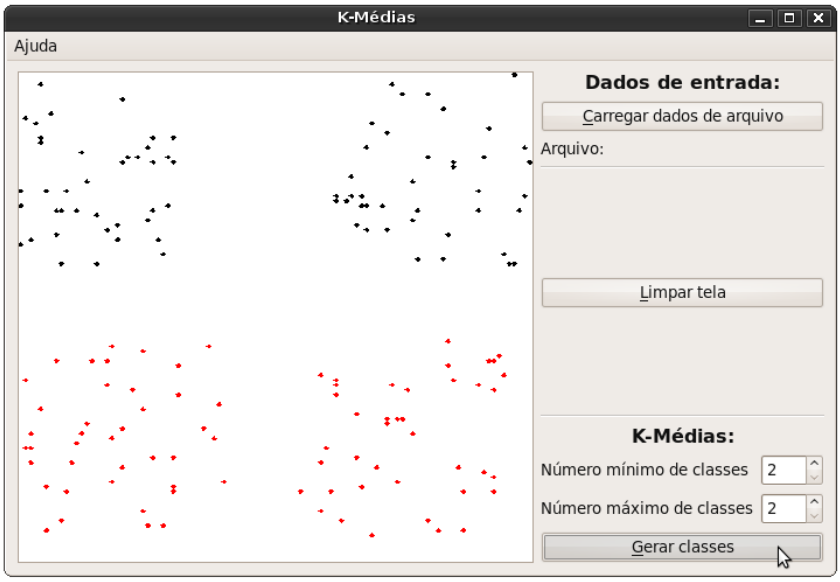

Figura 2: Dados agrupados em 2 classes.

O teste de Fisher foi calculado em separado para cada uma das coordenadas, e então consideramos o pior caso entre as coordenadas, ou seja, o máximo dos valores obtidos.

Esse cálculo foi repetido para cada um dos n (número de classes) na faixa definida pelo usuário, e o n com menor valor do teste de Fisher é aceito como o melhor particionamento do conjunto de dados.

Por exemplo, na figura 3, o usuário requisita um particionamento do conjunto de dados entre 3 a 7 classes, e o melhor particionamento encontrado foi em 3 classes distintas, cada uma representada por uma cor na figura.

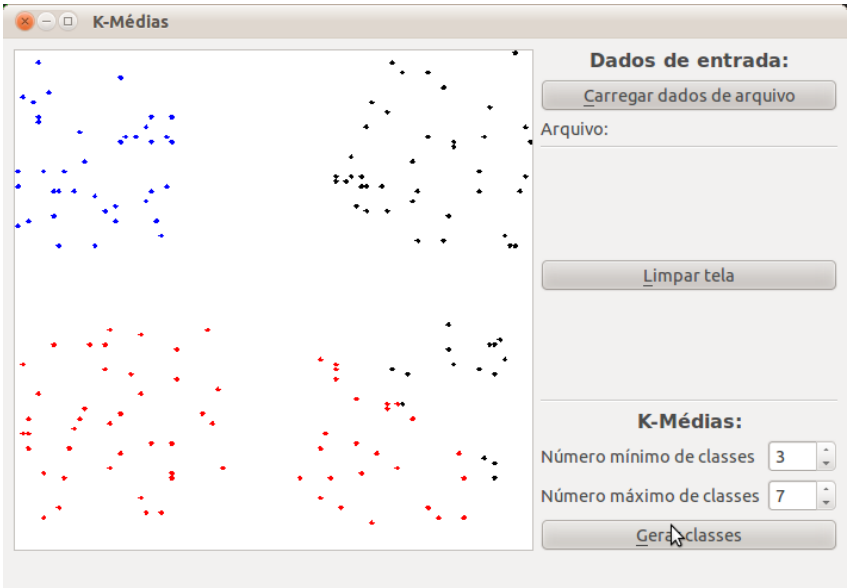

Figura 3: Dados agrupados em 3 classes.

4 Referências

- Python - http://www.python.org/
- Qt - http://doc.qt.nokia.com/
- pyQt - http://www.riverbankcomputing.co.uk/software/pyqt/intro
- Numpy - http://numpy.scipy.org/