

Resultados do uso do algoritmo K-médias

Augusto Ribas¹, Bruno Nazário¹ e Douglas Sorgatto¹

¹Faculdade de Computação - Universidade Federal de Mato Grosso do Sul

Resumo

Uma descrição do funcionamento do algoritmo k-médias aplicado sobre base de dados bidimensionais organizadas em grupos de 30 elementos, e implementada sem o uso de bibliotecas prontas na linguagem python. Os resultados demonstram que o algoritmo é de simples implementação, eficiente e de fácil compreensão. Os dados obtidos mostram uma eficiência superior a 90% na classificação dos dados, desde que sejam escolhidas as sementes de inicialização adequadas.

1 Introdução

Em inteligência artificial o problema de realizar agrupamentos é comum, importante e muito útil para facilitar a localização de dados, sua organização e seleção, principalmente para encontrar relações (grupos) em grandes bases de dados que, ao ser humano, parecem não possuir relação intrínseca.

Há vários algoritmos de agrupamento. Eles são divididos em dois grupos, os hierárquicos e os particionais. No primeiro grupo estão aqueles que são recursivos e retornam como informação uma árvore com as semelhanças entre todos os elementos de um conjunto. No segundo grupo, dos particionais, estão os algoritmos que separam uma grande massa de dados em grupos menores, com base em alguma função de semelhança, geralmente baseada em distâncias. Na seção 2.1 são descritos com mais detalhes o funcionamento desses algoritmos.

Este trabalho pretende avaliar a eficiência de um algoritmo particional específico, o K-médias, ou K-means. É um algoritmo básico, eficiente e com várias variantes de aplicação no agrupamento de dados.

Segue o relatório da aplicação do K-médias aos 9 conjuntos de dados fornecidos pelo professor e sua discussão.

1.1 Problema

O problema consiste em escrever o algoritmo K-médias sem utilizar as bibliotecas prontas do Sklearn [1], mas podendo utilizar as bibliotecas de manipulação de dados disponíveis, tais como panda, numpy e o uso da biblioteca matplotlib para a criação dos gráficos.

Este algoritmo deve rodar os conjuntos de dados fornecidos pelo professor e descritos na seção 2.3. Fornecendo para cada conjunto de dados o somatório do erro quadrático e o processo de agrupamento que foi realizado.

1.2 Objetivos

Implementar o algoritmo K-médias sem utilizar as bibliotecas prontas disponíveis no Scikit-Learn [1] e aplicá-lo nos conjuntos de dados fornecidos pelo professor avaliando o desempenho.

Apresentar, para cada conjunto de dados, o somatório do erro quadrático e o processo de agrupamento.

Discutir os valores encontrados e demonstrar a compreensão do processo de funcionamento dos algoritmos de agrupamento particionais.

2 Material e Métodos

Os nove conjuntos de dados foram fornecidos pelo professor da disciplina, prof Bruno Nogueira, e consistem em dados bidimensionais, organizados em grupos de 30 elementos e que devem ser separados nesses mesmos grupos quando submetidos ao algoritmo criado pelo nosso grupo.

Segue uma descrição sobre os algoritmos de agrupamento, sobre os conjuntos de dados e sobre a estrutura do algoritmo k-médias que implementamos.

2.1 Algoritmos de Agrupamento

Clustering, ou Agrupamento, é uma técnica de *Data Mining* para fazer agrupamentos automáticos de dados segundo seu grau de semelhança. O critério de semelhança faz parte da definição do problema e, dependendo, do algoritmo conforme se lê em [3].

Normalmente o usuário do sistema deve escolher *a priori* o número de grupos a serem detectados. Estes grupos são formados por equações que calculam a “semelhança” entre os dados através de funções de distância. Podem ser utilizadas diferentes tipos de função de distância: Manhattan, euclidiana, Minkowski, máximo, de Canberra, entre outras. Nós optamos por usar a função euclidiana no algoritmo implementado.

Os tipos de algoritmos de agrupamento de dados mais comuns são os: Particionais e os Hierárquicos. Os *particionais* procuram criar grupos de semelhança.

De acordo com [2], “Nos *Hierárquico* o processo de identificação de grupos (clusters) é geralmente realimentado recursivamente, utilizando tanto objetos quanto grupos já identificados previamente como entrada para o processamento. Deste modo, constrói-se uma hierarquia de grupos de objetos, no estilo de uma árvore”.

2.1.1 K-médias

O algoritmo de agrupamento de dados K-Médias agrupa um conjunto de instâncias em k partições, sendo k um número pré-estabelecido. O arranjo dos elementos é feito de maneira que um elemento pertença a um cluster, de cujo centro o elemento é mais próximo. Dessa maneira, o algoritmo K-Médias consegue encontrar k partições disjuntas, buscando sempre minimizar a variância intra-cluster e maximizar a variância inter-cluster. Como critérios de convergência usuais, pode-se citar o número de iterações que o algoritmo executa e o número de realocações de clusters.

Para que esse algoritmo funcione centro do cluster deve ser gerado aleatoriamente, dentro do espaço de dados existente. Essa função de geração aleatória é de soma im-

portância para o sucesso do algoritmo, pois sementes de inicialização podem gerar dados aleatórios tão próximos que algum cluster fica vazio, ou acabam se sobrepondo.

2.2 Procedimentos gerais

O algoritmo que desenvolvemos é genérico para os dados fornecidos pelo professor. Ou seja, basta informar o número de grupos que pretende formar, “K”, que ele seleciona a base de dados necessária e faz os procedimentos de agrupamento, retornando as informações solicitadas pelo professor.

Ao abrir o conjunto de dados eles são transformados em uma matriz e esta matriz é separada em dois vetores: “X” com os dados da primeira coluna, e “Y” com os dados da segunda coluna.

Logo após formamos o gráfico que apresenta a distribuição dos pontos dentro do conjunto de dados. Estas primeiras imagens estão disponíveis na figura 1. Percebe-se que os dados são bem distribuídos, mas formam grupos visuais de acordo com o “K” que representam. Dessa forma o primeiro conjunto de dados possui 2 grupos de 30 elementos cada e o segundo conjunto possui 3 grupos com 30 elementos em cada, e assim sucessivamente. Nosso algoritmo tem a missão de encontrar, computacionalmente, os grupos que se apresentam visualmente.

Depois é chamada a função de *clusterização*, ou agrupamento, esta função tem como parâmetros os vetores X e Y, o K e o Seed¹ que serão utilizados para formar os grupos e um valor booleano que permite gerar os gráficos das etapas da classificação.

Para garantir que tenhamos um agrupamento eficiente é formado um laço, onde a função de classificação é chamada por 10 vezes, sempre com valores de Seed diferentes. Dessa forma, conseguimos avaliar qual das sementes produz o menor somatório de erro quadrático e, depois dessa informação, gerar os gráficos de agrupamento mais eficientes.

Dentro da função de classificação são criados os pontos aleatórios que pretendemos ter como centros dos grupos. A função escolhida é a randômica normal, que gera dados dentro do intervalo de dados que temos no conjunto de dados e separados pela variância. Essa fórmula foi escolhida por resultar em pontos bem distribuídos dentro do universo dos dados.

Com o conjunto de pontos formado inicia-se um laço que para ao ter a convergência dos pontos² ou quando o procedimento de agrupamento é realizado 50 vezes.

No interior deste novo laço é calculada a distância euclidiana de cada ponto do conjunto de dados aos pontos que serão centros de grupo. A fórmula da distância euclidiana que utilizamos é

$$d = \sqrt{(x_i - C_x)^2 + (y_i - C_y)^2}$$

Onde, x_i e y_i são todos os pontos do conjunto de dados que estão nos vetores X e Y, respectivamente. E os valores de C_x e C_y são os valores de cada ponto gerado aleatoriamente.

Com o cálculo das distâncias vêm o momento de separar os pontos do conjunto de dados para cada um dos grupos, respeitando a regra de que cada ponto deve ficar no grupo do qual o centro é o mais próximo. Neste momento o algoritmo imprime quantos

¹Seed, ou “semente”, é o parâmetro da função de geração aleatória dos pontos que serão centros dos grupos. Sua escolha é de suma importância para o sucesso do agrupamento.

²Situação na qual os centros de cada grupo não variam mais suas posições.

elementos há na proximidade de cada centro de grupo, permitindo acompanhar a evolução do agrupamento.

O próximo passo consiste em realocar os centros de grupo para o centro do grupo de pontos próximo a ele, através do cálculo da média simples de seus pontos. Para isso calculamos a média dos pontos no eixo X e depois a no eixo Y e atribuímos esse novo valor ao centro do grupo.

Com essa nova posição calculada tem-se a segunda iteração, onde todo o processo se repete: calcula as distâncias de todos os pontos a todos os centros de grupo, reagrupa-se os pontos nos grupos de cujo centro são mais próximos e se realoca o ponto para o centro do novo grupo formado. Isso é feito até não haver mudanças nos centros dos grupos (*convergência*) ou chegar a 50 iterações.

O retorno da função é o valor do SSE, somatório do erro quadrático, que é calculado ao fim do laço interno da função e fornece uma medida que é a soma de todas as distâncias dos pontos aos seus centros dentro do conjunto de dados e que serve como parâmetro de eficiência, pois quanto menor o valor do SSE é sinal que os pontos estão o mais próximo possível dos centros de grupo. Valores maiores indicam que alguns pontos foram agrupados de forma incorreta.

A fórmula do SSE, somatório do erro quadrático dada pelo professor é usada no nosso algoritmo é

$$SSE = \sum_{i=1}^k \sum_{x \in C_i} dist^2(m_i, x)$$

Onde, k é o número de grupos, x é um ponto pertencente ao grupo C_i e m_i é o centro do grupo C_i .

Ao fim das 10 iterações com seeds diferentes são impressos o menor e o maior valor de SSE e o respectivo seed utilizado. Essas informações serão analisadas na seção 3 e fornecem uma boa ideia de porque a escolha correta do seed é muito importante para a eficiência do algoritmo.

Na figura 3 tem-se uma visão geral da evolução do SSE em cada iteração. Percebe-se que o erro varia de acordo com o seed escolhido, sendo essa variância maior para alguns valores de K.

Finalizado o processo de agrupamento é oferecido a opção de imprimir os gráficos de evolução do processo de agrupamento. Quando se responde “s” o programa solicita o seed para o qual deve rodar a função de agrupamento e passará os parâmetros necessários para isso, com a variável booleana permitindo a impressão dos gráficos.

Nos gráficos, cada centro de grupo é um círculo semitransparente e os pontos do conjunto de dados recebem cores e formas diferentes, dependendo do centro ao qual estão próximos em cada iteração. Um exemplo de evolução do processo de agrupamento para $k=5$ está na figura 3.

2.3 Conjuntos de dados

Para este trabalho, 9 conjuntos de dados foram utilizados. Todos estes conjuntos são bi-dimensionais (isto é, têm 2 atributos), com o número de grupos internos variando de 2 a 10 (o primeiro conjunto tem 2 grupos, o segundo tem 3 grupos, e assim por diante). Todos estes conjuntos de dados apresentam uma partição de referência (grupo de cada um dos pontos), sendo perfeitamente balanceados (30 exemplos por grupo), como se observa na tabela 1.

Cada um dos conjuntos de dados está disposto em um arquivo no formato CSV (“*comma separated values*”). Em cada linha há um exemplo (instância) da base, no formato:

$$ValorAtributo_1, ValorAtributo_2, Particao_de_Referencia$$

Sendo utilizado para o processamento apenas os dois primeiros atributos que, como explicado nos procedimentos gerais, são alocados em vetores para o processamento.

Tabela 1: Características gerais dos conjuntos de dados

Nome do conjunto	Valor de K	Número de instâncias	Número de grupos
artificial_2.data	2	60	2
artificial_3.data	3	90	3
artificial_4.data	4	120	4
artificial_5.data	5	150	5
artificial_6.data	6	180	6
artificial_7.data	7	210	7
artificial_8.data	8	240	8
artificial_9.data	9	270	9
artificial_10.data	10	300	10

O objetivo do algoritmo que implementamos é se aproximar da divisão apontada na terceira coluna, com cada ponto pertencendo ao grupo dado pelo professor. Portanto, criamos uma função de acurácia que permite saber se os grupos criados por nosso algoritmo consegue alocar 30 elementos em cada um dos grupos que forma ao processar o conjunto de dados.

3 Resultados e Discussão

Tabela 2: Comparação de eficiência - Menor SSE

K	Seed	SSE	Convergir	Min	Max	Fora	Acurácia
2	2	1.17178486551	5	29	31	1	0.98333
3	2	1.82592591238	5	30	30	0	1.00000
4	0	2.40978786268	9	28	32	2	0.98333
5	6	2.47892858032	6	29	31	2	0.98666
6	10	3.38661981209	8	28	34	5	0.97222
7	4	3.64855687389	8	29	31	3	0.98571
8	10	4.62062310715	5	25	35	9	0.96250
9	16	4.6656708739	13	28	32	4	0.98518
10	10	5.70866779082	11	28	32	5	0.98333

Tabela 3: Comparação de eficiência - Maior SSE

K	Seed	SSE	Convergir	Min	Max	Fora	Acurácia
2	0	1.18583121013	4	27	33	3	0.95000
3	8	4.18779991225	7	30	30	0	1.00000
4	12	7.00024804537	4	28	32	2	0.98333
5	8	7.47642260367	50	0	61	60	0.60000
6	0	18.7881874706	50	0	123	120	0.33333
7	12	23.488383509	50	0	115	120	0.42875
8	18	79.7031992156	50	0	240	210	0.12500
9	18	122.065360325	50	0	270	240	0.11111
10	0	62.8291736825	50	0	156	240	0.20000

Figura 1: Visão geral dos dados antes do agrupamento

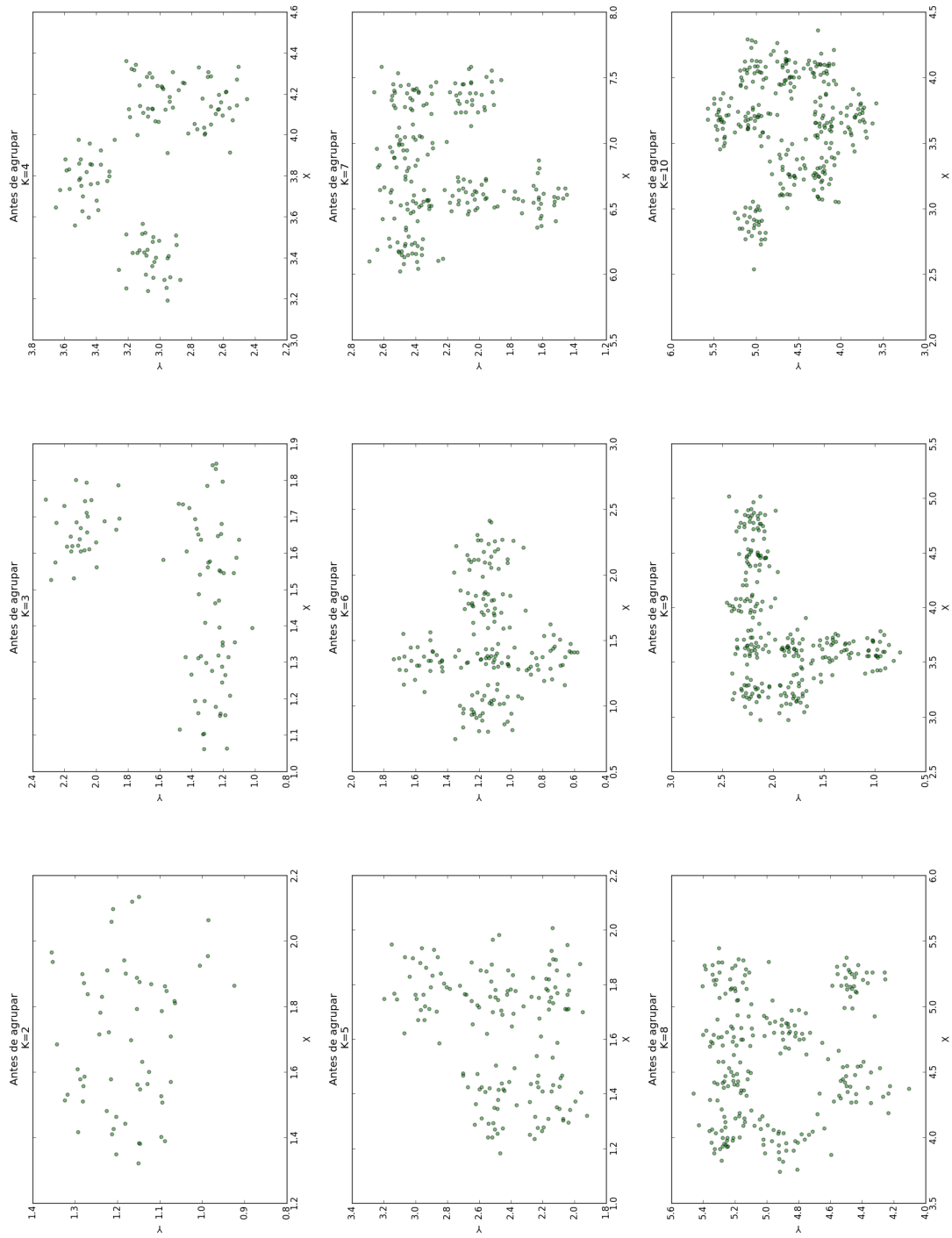


Figura 2: Evolução dos Somatórios de Erro Quadrático - SSE

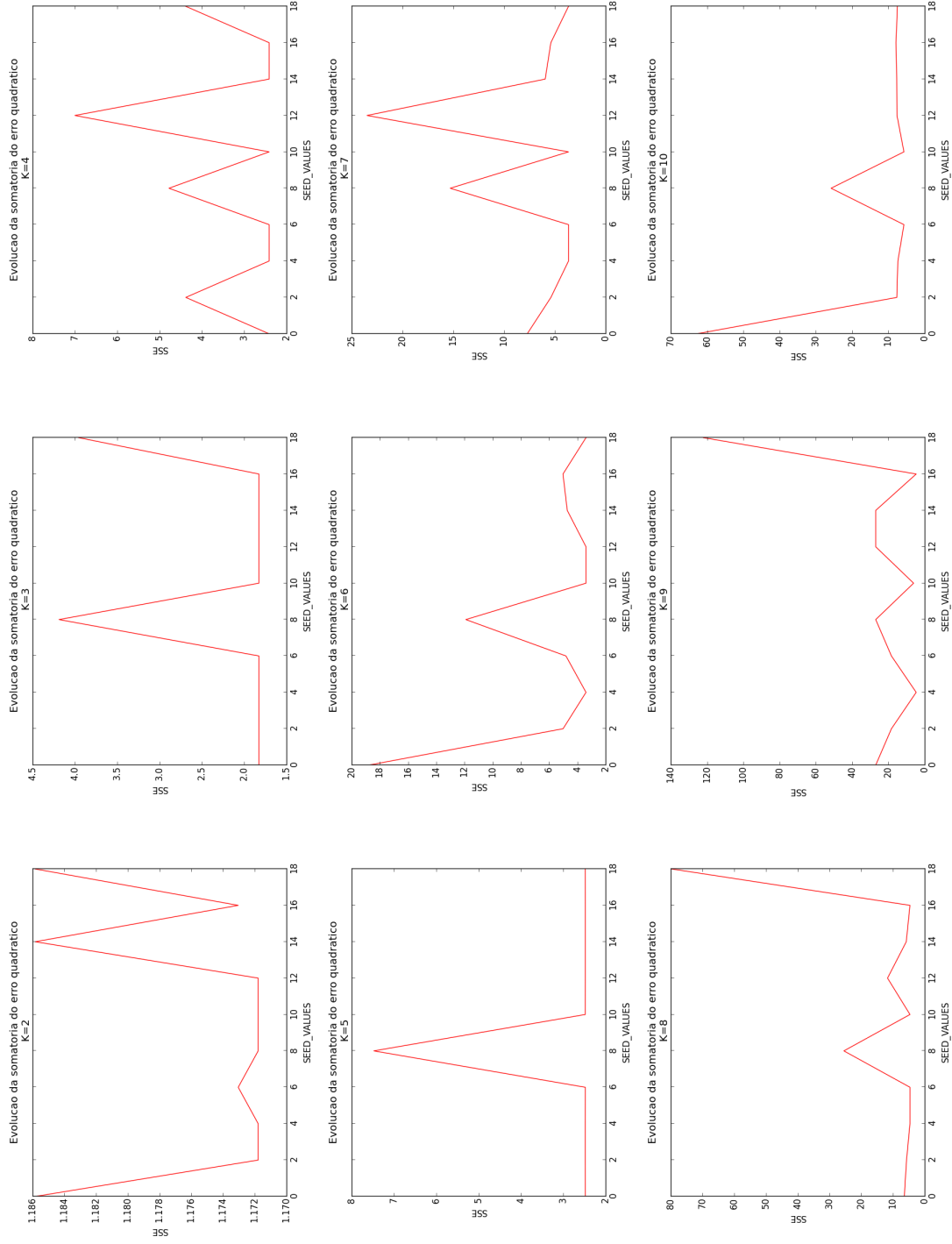
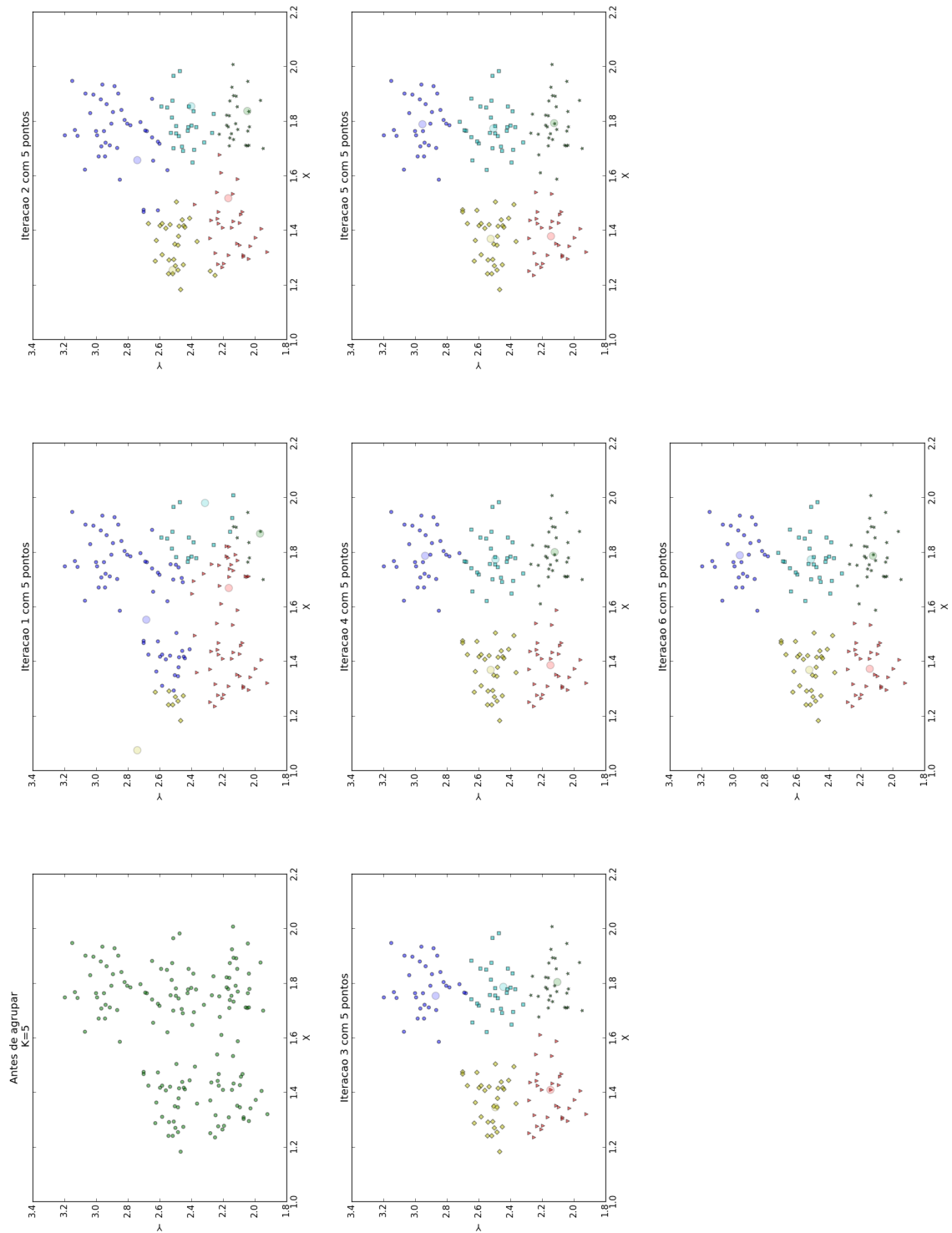


Figura 3: Exemplo de evolução do processo de agrupamento para $k = 5$



Referências

- [1] Scikit-learn: Machine Learning in Python.
- [2] Sopa de Letrinhas. Algoritmos de análise de agrupamentos: Método hierárquico – agnes e diana. 2012.
- [3] Wikipedia. Clustering. 2015.