

TRABALHO PRÁTICO MODELAGEM E AVALIAÇÃO DE SISTEMAS

Ítalo Garcez Carvalho

Graduando em Engenharia de Computação
Pontifícia Universidade Católica de Minas Gerais - PUCMG
italoc269@gmail.com

Leonardo Augusto de Souza Filho

Graduando em Engenharia de Computação
Pontifícia Universidade Católica de Minas Gerais - PUCMG
leonardoadsfilho@gmail.com

RESUMO

A Política Nacional de Resíduos Sólidos (PNRS) é uma legislação crucial para o combate ao problema do descarte inadequado de resíduos sólidos no Brasil. A necessidade de substituir os lixões a céu aberto por aterros sanitários é uma medida essencial para proteger o meio ambiente. Para solucionar o problema de descarte de lixo eletrônico em Minas Gearis, dois algoritmos para descobrir o melhor local das unidades de tratamento de lixo eletrônico

PALAVRAS CHAVE. Política Nacional de Resíduos Sólidos, Algoritmo, Lixo eletrônico.

ABSTRACT

The National Solid Waste Policy (PNRS) is crucial legislation to combat the problem of improper disposal of solid waste in Brazil. The need to replace open-air waste with landfills is an essential measure to protect the environment. To solve the e-waste disposal problem in Minas Gearis, two algorithms to discover the best location for e-waste treatment units.

KEYWORDS. National Solid Waste Policy. Algorithm.

Electronic trash.

1. Introdução

A Política Nacional de Resíduos Sólidos (PNRS) é uma legislação crucial para o combate ao problema do descarte inadequado de resíduos sólidos no Brasil. A necessidade de substituir os lixões a céu aberto por aterros sanitários é uma medida essencial para proteger o meio ambiente. No entanto, em um cenário quase-realista em Minas Gerais, percebe-se a necessidade de construir unidades específicas para tratamento de lixo eletrônico. A quantidade de unidades necessárias e os municípios que serão atendidos ainda são desconhecidos, mas é importante que cada município fique o mais próximo possível de uma unidade de tratamento de lixo eletrônico para que possa ser devidamente atendido. A implementação de medidas como essa é fundamental para garantir a proteção ambiental e o cumprimento da PNRS.

2. Metodologia

Para este trabalho foram propostos dois algoritmos para solucionar o problema. O primeiro algoritmo é o K-Means, é um algoritmo de aprendizado não supervisionado (ou seja, que não precisa de inputs de confirmação externos) que avalia e clusteriza os dados de acordo com suas características, com as seguintes especificações:

- Cada um dos 853 municípios é um dos objetos que precisam ser clusterizados;
- Cada cidade é representada por um ponto em um plano cartesiano que representa a sua localização geográfica;
- Por simplicidade, considere a distância euclidiana entre cada cidade.
- Seja k o número de Clusters, teste o algoritmo para k variando de 2 a 10;
- Defina um limite de tempo máximo para cada execução.

O resultado do algoritmo corresponderá a uma possível solução para o problema descrito, na qual, cada Cluster representará um grupo de cidades que será atendida por uma mesma unidade de tratamento de lixo eletrônico. Ainda, o centroide de cada cluster corresponder ao local de instalação de uma destas unidades.

Para o segundo algoritmo foi escolhido o K-medoids, em contraste com o algoritmo k-means, k-medoids escolhe pontos de dados reais como centros (medoides ou exemplares) e, portanto, permite maior interpretabilidade dos centros de cluster do que em k-means, onde o centro de um cluster não é necessariamente um dos pontos de dados de entrada (é a média entre os pontos no cluster). Além disso, k-medoids podem ser usados com medidas de dissimilaridade arbitrárias, enquanto k-means geralmente requer distância euclidiana para soluções eficientes. Como os k-medoids minimizam uma soma de dissimilaridades aos pares em vez de uma soma de distâncias euclidianas ao quadrado, é mais robusto a ruído e outliers do que k-means.

Então para o k-means utilizamos as seguintes configurações:

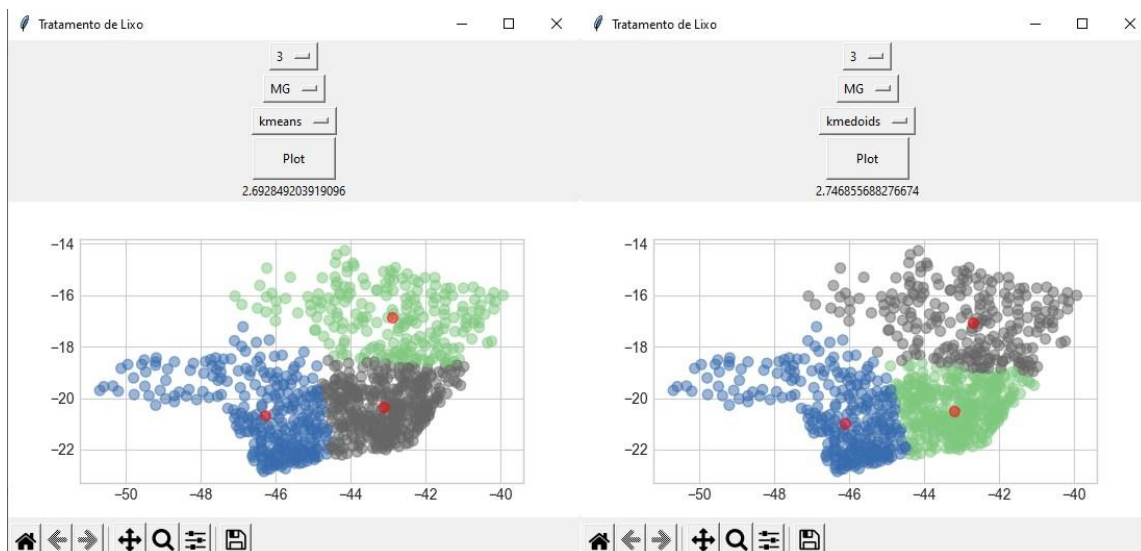
- Pontos iniciais escolhidos aleatoriamente.
- Busca de minimizações da distância até o cluster.

Enquanto que para o k-medoids as configurações foram:

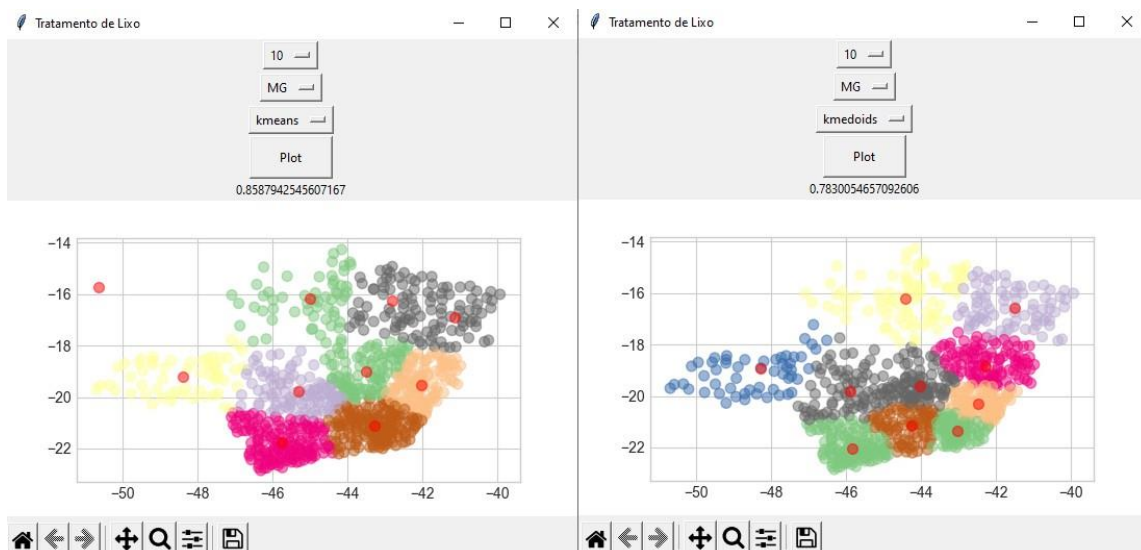
- Pontos iniciais escolhidos dentre os conjuntos das cidades.
- Redução da dissimilaridade em cada cluster.

3. Resultados

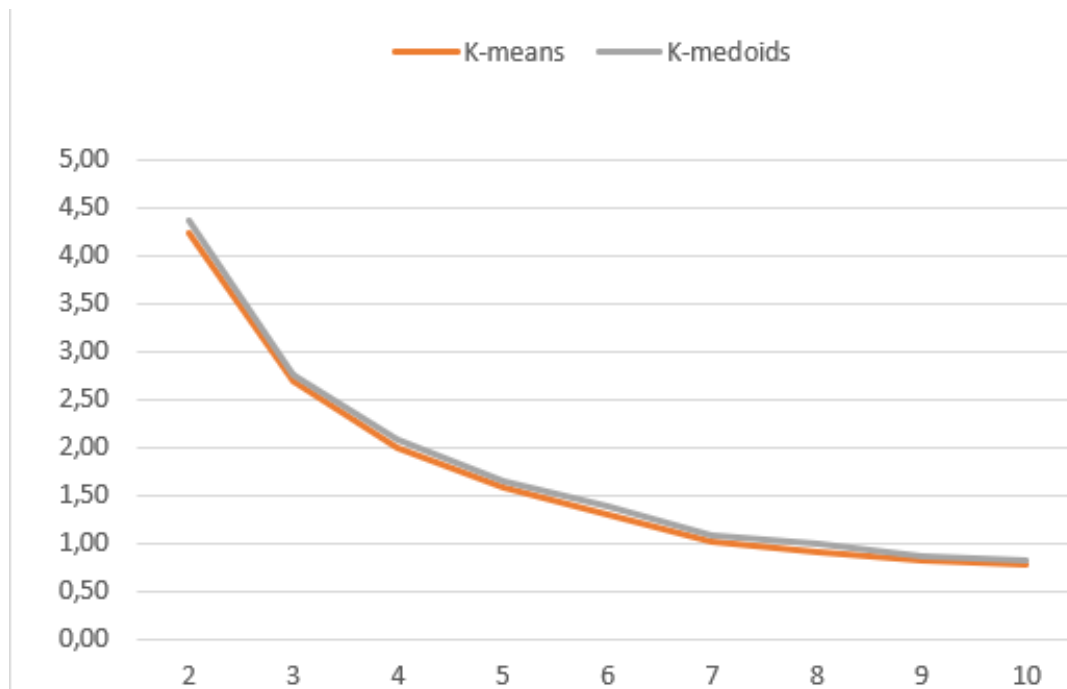
Para os dois algoritmos utilizamos a mesma quantidade de clusters e calculamos o erro quadrático médio de cada um. Vemos abaixo os resultados do k-means e k-medoids para 3 clusters com seus respectivos erros:



Abaixo temos os resultados da execução dos algoritmos k-means e k-medoids para 10 clusters e cada um com seus respectivos erros:



Por fim um gráfico comparativo de ambos os algoritmos, com o erro quadrático médio para cada quantidade de clusters:



4. Conclusão

Com os resultados obtidos vemos que o algoritmo k-medoids não mostra mais eficiente que o k-means. Um ponto que o algoritmo k-medoids oferece uma melhora é na seleção dos centroides, o algoritmo k-means por selecionar os primeiros centroides de maneira totalmente aleatória pode gerar centroides extremamente distantes das cidades, que acabam não fazendo parte de nenhum cluster. Já no k-medoids como os primeiros medoids escolhidos são alguma das cidades, sempre todos os medoids pertencem a algum cluster. Para uma melhora da eurística um ponto a ser analisado é a utilização de algum outro método para determinar a distância entre as cidades, como por exemplo a distância de Minkowski, que é uma métrica em um espaço vetorial normado, a qual pode ser considerada como uma generalização de ambas as distâncias euclidiana e Manhattan. Assim para uma melhor eurística fundamental a escolha dos primeiros centroides dentro da amostragem das posições e um cálculo de distância mais robusto.

Referências

- Wikipedia. (2023). Web page. <https://en.wikipedia.org/wiki/K-medoids>. Acessado: 2023-05-07.
- Wikipedia. (2023). Web page. https://en.wikipedia.org/wiki/K-means_clustering. Acessado: 2023-05-07.
- Wikipedia. (2023). Web page. https://en.wikipedia.org/wiki/K-means_clustering. Acessado: 2023-05-07.
- Angel , D. (2023). <https://towardsdatascience.com/k-medoid-clustering-pam-algorithm-in-python-with-solved-example-c0dcb35b3f46>. Acessado: 2023-05-07.
- Scikiti-learn. (2023). <https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html>. Acessado: 2023-05-07.
- Scikiti-learn. (2023). <https://scikit-learn.org/stable/modules/classes.html#module-sklearn.cluster>. Acessado: 2023-05-07.