Otimizando Algoritmo de Clusterização K-Means Utilizando o VNS

Cecília Flávia da Silva[2]

Abstract— **Em problemas de clusterização, o algoritmo K-means é amplamente utilizado na literatura, principalmente ao utilizar bases de dados de grande volume. Entretanto, o algoritmo é dependente essencialmente da inicialização dos centros de cada cluster, determinado pelo K-means de forma aleatória, não garantindo a resolução do problema de forma ótima. Dessa forma, esse trabalho tem como objetivo a elaboração de uma abordagem utilizando o K-means e a metaheurística Variable Neighborhood Search (VNS), afim de reduzir problemas de inicialização do agrupamento utilizando K-means. Assim, este trabalho implementou o k-means e utilizou-o como método de busca local do algoritmo VNS. Os resultados foram analisados com a literatura utilizando-se três bases distintas e comparados com o algoritmo tradicional do K-means e com o K-means++.**

I. Introdução

Clusterização é uma técnica amplamente utilizada para aplicações em diferentes áreas como aprendizado de máquina, reconhecimento de padrões, mineração de dados e neurociências. Nesta técnica, analisa-se uma determinada quantidade de dados, agrupando-se aqueles que forem mais semelhantes de acordo com uma métrica específica (e.g. distância). Nesse contexto, existem duas abordagens de agrupamento principais: heirárquica e por partição.

Na primeira abordagem, os dados são particionados sucessivamente, produzindo uma representação hierárquica dos agrupamentos, tornando-se possível visualizar as semelhanças entre os dados em diferentes níveis da árvore até as folhas, que contém o agrupamento final. Essa abordagem entretanto possui um custo elevado para uma grande quantidade de dados, de forma que atualmente as abordagens de particionamento são bastante utilizadas para esse contexto, consistindo em uma série de algoritmos cujo objetivo é minimizar o problema a partir de uma definição aleatória de particionamento.

Entre os algoritmos de particionamento, o K-means é um algoritmo baseado em distância, em que a partir da definição prévia de um número de k de clusters, calcula-se a distância média entre uma amostra e o centro de cada cluster k. A partir disso, o algoritmo identifica a amostra como pertencente ao cluster com menor distância. Dessa forma, o algoritmo possui como etapas principais:

- Inicialização: o algoritmo define, de forma aleatória, k amostras da base de dados como centro dos clusters (centroids).
- Atribuição do cluster: utilizando uma métrica de distância (e.g. euclidiana, manhattan), o algoritmo calcula a distância entre cada amostra da base de dados e os centróides definidos na primeira etapa. Após calcular a distância, cada amostra será atribuída ao cluster que retornar a menor distância.
- Realocação do centroid: com a etapa 2, k clusters são criados com diferentes densidades. Dessa forma, calcula-se a média de cada cluster obtendo-se k novos centróides.

Baseado nesse algoritmo, as etapas 2 e 3 são repetidas até uma determinada quantidade de interações, definidas pelo especialista. Entretanto, percebe-se que o algoritmo é dependente de dois fatores principais: (i) a quantidade de clusters k e (ii) a inicialização dos centróides.

Nesse contexto, usualmente experimentos com diferentes definições de k são realizados para garantir um valor de k apropriado para o problema, ou algumas abordagens como BIC podem ser utilizadas em conjunto com o mesmo propósito, evitando-se assim problemas relacionados a i). Entretanto, considerando ii), uma vez que a definição inicial do centróide é aleatória, dependendo da inicialização, o algoritmo pode retornar diferentes resultados a cada utilização, de forma que popularmente o experiento é repetido por uma quantidade n afim de encontrar a melhor combinação de grupos.

Apesar disso, ainda que a melhor combinação de grupos possa ser encontrada, não há garantias que o algoritmo encontre um mínimo global, podendo manter-se em um mínimo global. Dessa forma, na literatura diferentes algoritmos foram propostos, como o K-Means Harmônico e o K-means++, com o intuito de reduzir problemas de inicialização encontrados no K-means.

Além disso, abordagens também foram propostas combinando-se o K-Means Harmônico com metaheurísticas como o Variable Neighborhood Search (VNS), Particle Swarm Opmitization (PSO), colônia de formigas, multi-start local search, frefly e algoritmos genéticos.

Nesse contexto, baseando-se nas abordagens da literatura com o K-Means Harmônico, esse trabalho possui como objetivo implementar uma abordagem combinando-se o K-means e o VNS, metaheuristica que se baseia em mudanças sistemáticas de vizinhança das soluções para resolver problemas de otimização afim de identificar melhorias na inicialização. Os resultados serão comparados com o K-means tradicional, assim como utilizando o K-means++ e o K-Means Harmônico. Por fim, afim de avaliar os resultados independente de base de dados, realizou-se testes utilizando-se três bases de dados.

Dessa forma, o trabalho organiza-se da seguinte forma: seção 2 ilustra a complexidade do problema, contendo a prova que o problema é NP e uma redução a NP-difícil, seção 3 ilustra a representação da solução, contendo a metodologia

UFPB

do trabalho, seção 4 apresenta os resultaods obtidos e a seção 5 a conclusão do trabalho e trabalhos futuros.

II. Complexidade

Como ilustrado na seção 1, ao utilizar o k-means em uma quantidade finita de pontos S do espaço amostral da base de dados, o algoritmo calcula k centros utilizando uma métrica de distância entre cada um dos pontos em S e de seus vizinhos, sendo a distância euclidiana amplamente utilizada no algoritmo. Nesse contexto, tendo-se um espaço amostral com d dimensões, sabe-se a partir da literatura que a utilização da distância euclidiana no espaço d é NP-difícil a partir da utilização de 2 clusters [1], e que o problema é NP-difícil para qualquer valor de k mesmo que a dimensão seja planar [2]. Além disso, a literatura ilustra reduções da complexidade do K-means a partir da comparação com Plannar 3-Sat [3] e da complexidade geral de métodos de clusterização que utilizam soma de quadrados [4-5]

Por fim, a atribuição de um cluster no algoritmo do k-means basea-se em uma decisão chave: dado uma amostra S, essa amostra pertence ao cluster k igual 1?. Afim de identificar a decisão, o algoritmo realiza uma busca em um vetor de distâncias para encontrar a menor distância deste vetor, retornando-se a resposta SIM à decisão caso a menor distância esteja armazenada no índice que representa o cluster k igual a 1. Essa busca pode ser realizada em O(n) que é polinomial.

Dessa forma, considerando que um problema é classificado como NP (nào determinístico polinomial) se e somente se existir um algorimto não determinístico cujo reconhecimento da decisão SIM seja feita em tempo polinomial, pode-se dizer que o problema deste trabalho classifica-se como NP.

III. Representação da Solução

Para resolução do Problema de Otimização Combinatória (POC) deste trabalho, a solução possui como etapas principais:

- Implementar o algoritmo K-means
- Implementar o VND
- Implementar o VNS, utilizando-se o K-means como algoritmo de busca local
- Definir três bases de dados para realização dos testes e execução dos algoritmos
- Analisar os agrupamentos obtidos em três bases de dados distintas utilizando o VNS com o K-means
- Analisar o custo computacional obtido
- Comparar os resultados obtidos com abordagens na literatura como o K-means++ e o K-Harmônico.

Assim, para implementação dos algoritmos K-means, VND e VNS, utilizou-se a linguagem de programação Python (versão 3.6), cujo código está disponível no repositório do GitHub. Utilizou-se como base de dados (i) Iris com 150 instâncias, (ii) Wine com 178 instâncias e (iii) Breast-Cancer com 699 instâncias, uma vez que essas bases são utilizadas em trabalhos relacionados, tornando-se possível assim comparar os resultados obtidos.

Por fim, ao implementar o VND e o VNS, tornou-se necessário definir a vizinhança. Assim, esse trabalho tem-se como vizinhança as possíveis combinações entre uma amostra S e um cluster k, tendo-se como a quantidade máxima de clusters k no trabalho como igual a 5.

References