Otimizando o Cálculo do Índice de Silhueta através de Paralelismo em GPU

Thiago B. Lopes[1], **Wellington S. Martins**[1]

[1]Instituto de Informatica – Universidade Federal de Goiás (UFG)
Caixa Postal 74.690-900 – Goiania – GO – Brazil

thiagoborgeslopes@discente.ufg.br, wsmartins@ufg.br

Abstract. *Meta-features are fundamental for optimizing machine learning algorithms, but the calculation of meta-features related to the clustering set is computationally costly due to the complex analyses of clustering properties. In this paper, we present an approach that employs GPU parallelism to increase the efficiency of calculating the silhouette index, a key metric in clustering, making it more viable for large datasets.*

Resumo. *Meta-características são fundamentais para otimizar algoritmos de aprendizado de máquina, mas o cálculo de meta-características relacionadas ao conjunto de agrupamento é computacionalmente custoso, devido às análises complexas de propriedades de agrupamento. Neste trabalho, apresentamos uma abordagem que emprega paralelismo em GPU para aumentar a eficiência do cálculo do índice de silhueta, uma métrica chave no agrupamento, tornando-o mais viável para grandes conjuntos de dados.*

1. Introdução

O campo de aprendizado de máquina tem visto um aumento significativo na necessidade de otimização de algoritmos, onde as meta-características desempenham um papel crucial. As meta-características, também conhecidas como características de nível superior, são usadas para descrever conjuntos de dados e podem ser calculadas a partir dos dados brutos para fornecer informações valiosas que podem ser usadas para otimizar algoritmos de aprendizado de máquina [Brazdil et al. 2009].

O índice de silhueta é uma métrica usada para avaliar a qualidade dos clusters em um conjunto de dados [Rousseeuw 1987]. No entanto, o cálculo do índice de silhueta pode ser computacionalmente custoso, especialmente para grandes conjuntos de dados. Isso se deve à necessidade de calcular a distância entre todos os pares de pontos de dados, o que resulta em uma complexidade de tempo quadrática. Portanto, há uma necessidade crescente de desenvolver métodos eficientes para calcular o índice de silhueta, a fim de torná-lo mais viável para grandes conjuntos de dados.

2. Indice Silhoueta e trabalhos relacionados

O índice de silhueta é uma métrica amplamente reconhecida para avaliar a qualidade dos agrupamentos, pois quantifica o quão bem um objeto se ajusta ao seu próprio agrupamento em relação a outros agrupamentos. O cálculo deste índice envolve a determinação das distâncias médias intra-agrupamento (a) e as distâncias médias ao agrupamento mais

próximo (b) para cada objeto de dados. O coeficiente de silhueta para cada objeto é então calculado como (b - a) / max(a, b).

No trabalho [Luna-Romera et al. 2016], é discutido como a utilização dos centroides dos clusters para calcular o índice de silhueta pode reduzir significativamente os custos computacionais. Além disso, eles também exploram o paralelismo entre máquinas usando Spark para melhorar ainda mais a eficiência. No trabalho [Silva et al. 2023] demonstraram que a implementação da biblioteca RAPIDS(GPU) resultou em um aumento considerável na velocidade do cálculo, e agora buscamos implementar isso diretamente em CUDA.

3. Ambiente de Testes e Datasets Utilizados

Os experimentos para este estudo foram realizados no ambiente Google Colab [Google 2021], uma plataforma de pesquisa baseada em nuvem que permite a execução de scripts Python e outros códigos de programação. O ambiente de execução utilizado possui uma CPU Intel(R) Xeon(R) @ 2.00GHz, 12GB de RAM e uma GPU Tesla T4 com 16GB de memória e 2.560 núcleos CUDA.

Para verificar a precisão do índice de silhueta, empregamos dois conjuntos de dados conhecidos: Iris e Digits. Iris, que contém 150 instâncias, 4 características e 3 clusters, é comumente utilizado em testes. Digits possui 10.992 instâncias, 16 características e 10 clusters.

Para avaliar a eficiência computacional do algoritmo, três conjuntos de dados foram gerados aleatoriamente em um ambiente de laboratório: Random1, consistindo de 1 milhão de instâncias; Random10, consistindo de 10 milhões de instâncias; e Random20, consistindo de 20 milhões de instâncias. Todos os conjuntos de dados têm 16 características e 10 clusters. Esses conjuntos de dados de grande escala foram utilizados para avaliar a capacidade do algoritmo de lidar com quantidades significativas de dados.

4. Trabalho Realizado

Neste estudo, implementamos dois algoritmos baseados em CUDA para calcular o índice de silhueta, cada um utilizando uma abordagem diferente.

O primeiro algoritmo(Quadrático) segue uma abordagem generalista para calcular o índice de silhueta. Utilizamos um kernel que lança uma thread para cada ponto no conjunto de dados. Cada thread é responsável por realizar todo o cálculo do índice para o ponto específico. Após a execução do kernel, a média do índice de silhueta é calculada na CPU.

O segundo algoritmo(Linear) é baseado na abordagem que utiliza os centroides dos clusters para calcular o índice de silhueta. Nesta abordagem, um kernel é lançado para cada cluster no conjunto de dados para calcular parcialmente os centroides. Os valores retornam para a CPU, onde o cálculo do centroide e a média das distâncias inter-cluster (entre centroides) são realizados. Em seguida, é realizado o cálculo da média das distâncias intra-cluster (do ponto para o centroide) com um kernel para cada cluster. Finalmente, o cálculo do índice de silhueta é realizado na CPU.

5. Resultados

Ao analisar a Tabela 1, observamos pequenas variações nos valores do índice de acordo com o conjunto de dados. Essas pequenas variações não comprometem a eficácia da abordagem proposta.

A Tabela 2, observa-se um ganho de speedup relevante entre os algoritmos Quadrático e Linear, especialmente em cenários que exigem análise de grandes volumes de dados dentro de de um tempo hábil. Enquanto o algoritmo Quadrático enfrentou dificuldades em lidar com grandes volumes de dados (indicados pela marcação "X"), o algoritmo Linear se mostrou mais eficiente em termos de tempo de execução.

	Iris	Digits
Quadrático	0.587166	0.313712
Linear	0.601920	0.135583

Tabela 1. Valores do índice silhueta

	Random1	Random10	Random20
Quadrático	1171.15	X	X
Linear	0.0098	0.085	0.16

Tabela 2. Tempo de execução do índice silhueta em segundos

6. Conclusões

Com base nos resultados apresentados, podemos concluir que a eficiência e a precisão dos dois algoritmos implementados variam. É evidente que o algoritmo Linear é significativamente mais rápido que o algoritmo Quadrático, especialmente para conjuntos de dados de grande escala. No entanto, os resultados do índice de silhueta podem variar dependendo das características do conjunto de dados. Portanto, é crucial considerar tanto a eficiência computacional quanto a precisão do índice de silhueta ao selecionar um algoritmo para essa tarefa.

Referências

Brazdil, P., Giraud-Carrier, C., Soares, C., and Vilalta, R. (2009). Metalearning: Applications to data mining. Springer Publishing Company.

Google (2021). Google colaboratory. https://colab.research.google.com. Accessed: 2023-05-25.

Luna-Romera, J. M., Martínez Ballesteros, M., García-Gutiérrez, J., and Riquelme, J. (2016). An approach to silhouette and dunn clustering indices applied to big data in spark. Advances in Artificial Intelligence. CAEPIA 2016. Lecture Notes in Computer Science(), vol 9868. Springer, Cham.

Rousseeuw, P. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics 20, 53-65.

Silva, L. L., Franco, R., Carvalho, A., and Martins, W. (2023). Gpu acceleration of clustering meta-feature extractionusing rapids. XXII Workshop em Desempenho de Sistemas Computacionais e de Comunicacão, (wperformance 2023) edition.