Cálculo paralelo de índice de validação de agrupamento

Vinicius F. Barbosa[1], **Wellington S. Martins** [1]

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

vini,fleury123@discente.ufg.br, wsmartins@ufg.br

Abstract. *In meta-learning, meta-features are measures derived from the dataset that provide additional information about its properties. Extracting, or calculating, these meta-features can be costly, especially in clustering. This paper seeks to use parallelism to more quickly and efficiently compute a meta-feature, or index, of cluster validation. Our experiments show gains of up to 22 times compared to the sequential version.*

Resumo. *No meta-aprendizado, as meta-características são medidas derivadas do conjunto de dados que fornecem informações adicionais sobre suas propriedades. A extração, ou cálculo, dessas meta-características pode ser custosa, especialmente em agrupamentos. Este trabalho busca utilizar o paralelismo para calcular de forma mais rápida e eficiente a meta-característica, ou índice, de validação de agrupamentos. Nossos experimentos mostram ganhos de até 22 vezes em comparação com a versão sequencial.*

1. Introdução

No meta-prendizado, uma meta-característica é uma medida derivada do conjunto de dados que fornece informações adicionais sobre suas propriedades. Essas meta-características são usadas para selecionar algoritmos, ajustar hiperparâmetros e obter novos conhecimentos sobre a complexidade, diversidade e qualidade dos conjuntos de dados [Lemke et al. 2015]. A extração dessas meta-características pode ter um alto custo computacional, como é o caso das meta-características associadas a agrupamentos. Neste caso, o uso de paralelismo tem sido uma alternativa para minimizar este problema [Silva L. 2023, Luna-Romera et al. 2016, Zerabi et al. 2020]. Assim, este trabalho busca utilizar paralelismo a fim de realizar a extração (cálculo) dessas meta-características (índices) de maneira mais rápida e eficiente.

Dos vários índices de validação de agrupamentos, o índice Dunn é um dos mais utilizados [Rivolli et al. 2018]. Neste trabalho implementamos um cálculo aproximado do índice de Dunn, considerando um solução sequencial e uma paralela (GPU). Nossos experimentos mostram que a solução usando o índice aproximado, e explorando o paralelismo da GPU, consegue diminuir a quantidade de operações de comparação da distância entre as instâncias, e diminuir o tempo de execução total do cálculo do índice de Dunn.

2. Índice de Dunn e trabalhos relacionados

O índice de Dunn pode ser calculado seguindo três passos. O primeiro passo é achar a menor distância euclidiana entre instâncias de dois diferentes clusters, o segundo é calcular a maior distância que há entre duas instâncias em um mesmo cluster e por fim

fazer a divisão entre os dois resultados. Em contraste com a implementação original, o índice de Dunn pode ser aproximado utilizando-se dos centroides dos clusters. O cálculo da menor distância inter clusters é alterado para a extração da menor distância entre os centroides de cada cluster com o centroide global do dataset, e a maior distância intra clusters é modificado para ser a maior distância entre o centroide de um cluster e uma instância do mesmo .

Trabalhos relacionados na área [Luna-Romera et al. 2016, Zerabi et al. 2020], fazem uso de aproximações do índice de Dunn, principalmente usando centroides, e empregam paralelismo para acelerar o cálculo. Entretanto, eles exploram principalmente plataformas com escalabilidade horizontal, empregando várias máquinas interligadas com redes de alta velocidade. Em contraste, este trabalho enfatiza a escalabilidade vertical, que envolve a adição de poder computacional (GPUs) a uma única máquina existente.

3. Datasets e Ambiente de Execução

Utilizamos um dataset simples (DUT), com 2 clusters, 10 instâncias e 2 características, para verificar o bom funcionamento da implementação. O segundo dataset utilizado é o Iris, com 3 clusters, 150 instâncias e 4 características. Por último utilizamos o dataset Digits que possui 10 clusters, 1797 instâncias e 64 features. Os testes foram realizados em ambiente de nuvem, numa máquina com uma CPU de 2 núcleos, Intel(R) Xeon(R) 2GHz, memória RAM de 12 GB, e um acelerador (GPU) Tesla T4, com 2560 núcleos, e memória de 16 GB.

4. Trabalho realizado

Realizamos a implementação sequencial do cálculo do índice de Dunn utilizando a linguagem de programação Python. A implementação paralela, em CUDA, calcula as somas parciais das coordenadas dos pontos de cada cluster em paralelo. O cálculo de cada centróide é finalizado na CPU, que encontra a soma total de cada coordenada e divide pelo número de pontos do cluster.

O kernel utilizado para o cálculo das somas parciais faz uso de um número de blocos de threads igual ao total de pontos do cluster dividido por 128 (threads). Este número de threads por bloco foi encontrado experimentalmente. O kernel também faz uso da memória compartilhada do bloco, para acelerar o processo, e realiza uma redução paralela para combinar as somas parciais calculadas por cada bloco. O trabalho conjunto da CPU e GPU permite um ganho consider[avel] de desempenho, como mostrado na seção seguinte.

5. Resultados

O tempo de execução dos códigos foi medido a partir do momento em que a primeira função de cálculo do índice é chamada, e corresponde à média de 10 execuções. Além disso, determinamos a quantidade de operações de distância entre instâncias, permitindo identificar qual implementação realiza menos operações para obter o resultado desejado. Os resultados são apresentados a seguir. Foi realizada uma comparação entre as implementações sequencial e paralela desenvolvidas. Não foi possível fazer uma comparação direta dos nossos resultados com os trabalhos relacionados já que são plataformas completamente distintas.

Na Tabela 1 apresentamos os resultados obtidos quando comparamos as implementações sequencial e paralela para ao cálculo do índice de Dunn. A terceira coluna mostra o valor do índice de Dunn calculado, enquanto o tempo requerido para seu cálculo é apresentado na coluna seguinte. As colunas cinco e seis apresentam o número de operações realizadas usando o índice de Dunn tradicional e o aproximado respectivamente. Na última coluna temos o speedup alcançado em cada caso. Podemos observar que, à medida que temos a elevação das instâncias e poucas características (Iris), o tempo de execução tem uma queda significativa, indicando um desempenho elevado. À medida que aumentamos o número de características (Digits) a execução não alcança um elevado desempenho.

Tabela 1. Comparação no cálculo de Dunn: sequencial vs paralelo

Dataset	Algoritmo	Valor	Tempo	#OpDunn	#OpDunnAprox	SpeedUp
DUT	**Sequencial**	**0.43**	**3.94 ms**	**100**	**12**	**16.62**
	Paralelo	**0.44**	**0.237 ms**			
Iris	**Sequencial**	**0.288**	**9.87 ms**	**30000**	**153**	**22.38**
	Paralelo	**0.29**	**0.441 ms**			
Digits	**Sequencial**	**0.26**	**55 ms**	**5813121**	**1807**	**7.8**
	Paralelo	**0.26**	**7.047 ms**			

6. Conclusões e Novos desafios

O trabalho conseguiu fazer um bom uso dos aceleradores no problema de se calcular o índice de Dunn para validação de agrupamentos. Foram obtidos speedups expressivos, chegando a mais de 22x para o caso da base de dados Iris. Como trabalho futuro, pretendemos aprimorar o código desenvolvido, implementar outros índices de validacão, e utilizar múltiplas GPUs para acelerar ainda mais o cálculo destes índices.

Referências

Lemke, C., Budka, M., and Gabrys, B. (2015). Metalearning: a survey of trends and technologies. *Artificial intelligence review*, 44:117–130.

Luna-Romera, J. M., del Mar Martinez-Ballesteros, M., Garcia-Gutierrez, J., and Riquelme-Santos, J. C. (2016). An approach to silhouette and dunn clustering indices applied to big data in spark. In *Advances in Artificial Intelligence: 17th Conference of the Spanish Association for Artificial Intelligence, CAEPIA 2016, Salamanca, Spain, September 14-16, 2016. Proceedings 17*, pages 160–169. Springer.

Rivolli, A., Garcia, L. P., Soares, C., Vanschoren, J., and de Carvalho, A. C. (2018). Characterizing classification datasets: a study of meta-features for meta-learning. *arXiv preprint arXiv:1808.10406*.

Silva L., Franco R., C. A. M. W. (2023). Gpu acceleration of clustering meta-feature extraction using rapids. XXII Workshop em Desempenho de Sistemas Computacionais e de Comunicação, (wperformance 2023) edition.

Zerabi, S., Meshoul, S., and Boucherkha, S. C. (2020). Models for internal clustering validation indexes based on hadoop-mapreduce. *International Journal of Distributed Systems and Technologies (IJDST)*, 11(3):42–67.