Uso de GPUs na validação de agrupamentos com amostragem SkeVa

Wilson G. N. Junior[1] **, Wellington S. Martins**[1]

[1]Instituto de Informática – Universidade Federal de Goiás (UFG)
Caixa Postal 74690-900 – Goiânia – GO – Brazil

wilson2@discente.ufg.br, wsmartins@ufg.br

Resumo. *A validação de agrupamentos em grandes volumes de dados é limitada pelo custo quadrático do Índice de Dunn (DI), especialmente no cálculo da compacidade (diâmetro máximo intra-cluster). Este trabalho apresenta uma implementação paralela do DI que combina aceleração em GPU com a técnica de amostragem SkeVa (Sketch-and-Validate), que estima o diâmetro máximo usando apenas pequenas amostras. Em testes com datasets de até 1 milhão de pontos, o método alcançou speedups de 9x a 11x em relação à versão serial, preservando o valor do DI.*

Abstract. *Cluster validation on large datasets is limited by the quadratic cost of the Dunn Index (DI), particularly in computing compactness (maximum intra-cluster diameter). This work presents a parallel DI implementation that combines GPU acceleration with the SkeVa (Sketch-and-Validate) sampling technique, which estimates the maximum diameter using only small samples. In experiments with datasets up to 1 million points, the method achieved 9× to 11× speedups over the serial version while preserving the DI value.*

1. Introdução

O índice de Dunn (DI) mensura a qualidade de particionamentos maximizando a separação entre clusters e minimizando a dispersão interna de cada cluster. Na forma clássica, o DI é a razão entre a menor distância inter-clusters e o maior diâmetro intra-cluster; quanto maior o valor, melhor o particionamento. Em bases grandes, o denominador torna-se o gargalo porque envolve, no limite, comparações quadráticas entre pares de pontos. Para mitigar o custo mantendo qualidade, adotamos SkeVa (Sketch-and-Validate), que usa amostragem aleatória para estimar diâmetros com poucos pontos

2. Trabalhos Relacionados

[Ben Ncir et al. 2021] propõe o S-DI com Apache Spark e SkeVa (paralelização horizontal, sem GPU). [Grün et al. 2024] implementa o Índice de Dunn em GPU com CUDA, mas sem SkeVa. Este trabalho combina ambas as abordagens, avaliando os ganhos de SkeVa em CUDA.

3. Dunn Index

O Índice de Dunn [Dunn 1974] é uma métrica clássica de validação de clusters que avalia, ao mesmo tempo, o quão separados estão os clusters entre si e o quão compactos são

internamente. A ideia é simples: um bom particionamento tem grande separação inter-clusters e pequena dispersão intra-cluster. Por isso, o DI por centróides é definido como $\mathrm{DI} = \frac{\min_{1 \le i \ne j \le m} \delta(C_i, C_j)}{\max_{1 \le k \le m} \Delta(C_k)}$.

No numerador dessa equação temos o cálculo da Separação(δ), onde é buscada a menor distância entre dois clusters (C_i, C_j) com base nos seus respectivos centróides. No denominador, temos o cálculo da Compactação(Δ) que é aplicado para buscar a maior distância entre todos os possíveis pares de objetos de dados dentro de um cluster. O resultado final é o Dunn Index(DI). O número de pares a serem avaliados cresce quadraticamente com o tamanho do conjunto analisado, e é por isso que se torna tão custoso avaliar grandes quantidades de dados.

4. SkeVa

SkeVa (Sketch and Validate) [Traganitis et al. 2015] é uma técnica de amostragem iterativa que alterna entre (i) sketching: amostragem aleatória de um subconjunto pequeno para calcular a medida de interesse, e (ii) validation: retenção do melhor subconjunto. No contexto deste trabalho, reduz a complexidade de $O(N^2)$ para $O(R \cdot S^2)$, onde $S \ll N$ é o tamanho da amostra e R o número de repetições.

5. CUDA+SkeVa

O algoritmo CUDA+SkeVa paraleliza as duas operações mais custosas do cálculo do DI: (1) o cálculo dos centróides e (2) a estimativa dos diâmetros intra-cluster via SkeVa.

Centróides na GPU: Para cada cluster c, um kernel CUDA distribui os pontos entre blocos de threads. Cada thread acumula parcialmente as coordenadas dos pontos atribuídos ao seu bloco e, após sincronização, uma redução paralela agrega as somas parciais para obter o centróide $\mu_c = \frac{1}{|I_c|} \sum_{i \in I_c} X_i$.

Diâmetro via SkeVa: Em cada repetição r, uma amostra aleatória S_c de tamanho S é selecionada. O kernel de distâncias pareadas mapeia cada par (p, q) de S_c a uma thread, que calcula $\|X_p - X_q\|_2$. Uma redução em memória compartilhada identifica a distância máxima δ entre os pares. O algoritmo retém o maior δ ao longo das R repetições como estimativa de Δ_c.

Algoritmo 1: Dunn CUDA-SkeVa

Input: $X \in \mathbb{R}^{N \times NF}$ (dados), $y \in \{0, \ldots, K-1\}$ (rótulos),
K (nº de clusters), NF (nº de atributos), S (tamanho do *sketch* por cluster),
R (repetições SkeVa por cluster), T (threads por bloco CUDA)
Output: $\{\mu_c\}$(centróides), $\delta_{\min}$, $\{\Delta_c\}$, $\Delta_{\max}$, DI

Construir $I_c = \{i : y_i = c\}$.
Copiar X para a GPU; computar $\mu_c = \dfrac{1}{|I_c|} \displaystyle\sum_{i \in I_c} X_i$ em paralelo (CUDA).
for $c = 0$ **to** $K - 1$ **do**
    $\Delta_c \leftarrow 0$;
    for $r = 1$ **to** R **do**
        amostrar $S_c \subset I_c$, $|S_c| = S$;
        $\delta \leftarrow \max_{p,q \in S_c} \|X_p - X_q\|_2$ (GPU);
        $\Delta_c \leftarrow \max(\Delta_c, \delta)$
$\delta_{\min} \leftarrow \min_{c \ne c'} \|\mu_c - \mu_{c'}\|_2$; $\Delta_{\max} \leftarrow \max_c \Delta_c$; $DI \leftarrow \delta_{\min} / \Delta_{\max}$;

6. Resultados

Realizamos os experimentos em um notebook com Intel Core i7-13620H, 8 GB de RAM e GPU NVIDIA GeForce RTX 3050 (6 GB), em Linux com CUDA. Em todos os experimentos utilizamos 30% do dataset em cada iteração e um total de 8 iterações do SkeVa.

Os datasets sintéticos utilizados (A, B e C) possuem 1M, 500k e 100k pontos respectivamente, mantendo 3 features e 10 clusters. A Tabela 1 apresenta os resultados comparativos, destacando que a implementação CUDA+SkeVa alcança ganhos de desempenho entre 9,16x e 11,33x (speedup) em relação ao código serial, enquanto preserva o valor do Índice de Dunn (DI) com aproximações mínimas.

Tabela 1. Comparação entre o código serial e a versão CUDA com SkeVa.

Dataset	Serial (s)	CUDA+SkeVa (s)	Speedup	DI (Serial)	DI (CUDA+SkeVa)
A	112,4	9,94	11,31	0,1348	0,1348
B	28,1	2,48	11,33	0,1419	0,1422
C	1,1	0,12	9,16	0,1495	0,1612

7. Conclusões e trabalhos futuros

A combinação da amostragem SkeVa com a paralelização em GPU demonstrou ser uma solução escalável e eficiente para a validação de agrupamentos. Nossa abordagem elimina o gargalo quadrático do Índice de Dunn (DI), resultando em speedups de 9x a 11x, ao mesmo tempo que mantém a precisão do índice. Isso valida uma alternativa prática para a avaliação confiável de agrupamentos em larga escala.

Como trabalhos futuros, pretendemos realizar experimentos com conjuntos de dados de maior volume, avaliar a sensibilidade aos parâmetros do SkeVa (tamanhos de sketch e número de iterações), e aplicar a validação acelerada para determinar o número ideal de clusters (K), transformando-a em uma ferramenta de otimização de agrupamentos.

Referências

Ben Ncir, Chiheb-Eddine, Hamza, Abdallah, and Bouaguel, Waad (2021). Parallel and scalable dunn index for the validation of big data clusters. *Parallel Computing*, 102.

Dunn, J. C. (1974). Well-separated clusters and optimal fuzzy partitions. *Journal of Cybernetics*, 4(1):95–104.

Grün, Eduardo S., Martins, Wellington S., and Franco, Ricardo (2024). Acelerando o cálculo do índice dunn de validação de agrupamento. In *Escola Regional de Alto Desempenho do Centro-Oeste (ERAD-CO)*, Goiânia, GO, Brasil. SBC.

Traganitis, Panagiotis A., Slavakis, Konstantinos, and Giannakis, Georgios B. (2015). Sketch and validate for big data clustering. *IEEE Journal of Selected Topics in Signal Processing*.