Dados de Saúde Processados via MASSA: Aceleração em GPU de um Classificador Baseado em Grafos

Diego Sanches Nere dos Santos[1], **Luiz Carlos Bambirra Torres**[1]

[1]Departamento de Computação e Sistemas - Universidade Federal de Ouro Preto (UFOP)
João Monlevade – MG – Brasil

diego.nere@aluno.ufop.br, luiz.torres@ufop.edu.br

Abstract. *The growth of healthcare data demands efficient processing for real-time diagnostics, yet AI algorithms fundamental to this analysis are often sequential and computationally prohibitive. This is the case for geometric classifiers such as NN-clas, based on Gabriel Graphs, face a cubic computational bottleneck ($O(N^3)$) during construction. This work presents a GPU-accelerated implementation of NN-clas specifically for healthcare data preprocessed via the MASSA flow. By utilizing a Single Instruction, Multiple Threads (SIMT) architecture and an implicit index mapping strategy, we mitigated the $O(N^2)$ memory bottleneck. Experimental results using pharmaceutical datasets demonstrated speedups of up to 2,852x compared to the sequential version, confirming the viability of the approach for low-latency clinical decision support.*

Resumo. *O crescimento de dados no setor de saúde exige processamento eficiente para diagnósticos em tempo real, mas algoritmos de IA fundamentais para essa análise costumam ser sequenciais e computacionalmente proibitivos. É o caso de classificadores geométricos como o NN-clas, baseados em Grafos de Gabriel, que enfrentam um gargalo computacional cúbico ($O(N^3)$) durante sua construção. Este trabalho apresenta uma implementação acelerada em GPU do NN-clas para dados de saúde pré-processados via fluxo MASSA. Ao utilizar uma arquitetura SIMT e uma estratégia de mapeamento implícito de índices, mitigamos o gargalo de memória $O(N^2)$. Resultados experimentais utilizando datasets farmacêuticos demonstraram speedups de até 2.852x em relação à versão sequencial, confirmando a viabilidade da abordagem para suporte à decisão clínica de baixa latência.*

1. Introdução

O processamento ágil de volumes massivos de dados tornou-se uma necessidade crítica no contexto da saúde, onde a rapidez na análise de informações biomédicas impacta diretamente a precisão e a viabilidade de diagnósticos em tempo real [Zhang et al. 2023, Çolhak et al. 2025]. Neste trabalho, o foco recai sobre dados farmacêuticos processados via fluxo MASSA (Molecular Automated Sampling Selection Algorithm) [Veríssimo et al. 2023], que organiza bases de patógenos como a *Escherichia coli*

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001

(ECOLI) e a *Klebsiella pneumoniae* (KLEB). Esses microrganismos são alvos críticos em estudos de resistência bacteriana, e a rapidez na classificação de seus perfis moleculares é vital para a escolha do tratamento clínico adequado. Tradicionalmente, muitos algoritmos de Inteligência Artificial (IA) são implementados de forma sequencial, o que limita sua aplicação em cenários de larga escala devido ao alto tempo de resposta.

Entre esses métodos, o classificador NN-clas destaca-se por utilizar a geometria dos Grafos de Gabriel para definir regiões de decisão de margem larga [Gade et al. 2018, Gabriel and Sokal 1969]. Apesar de sua robustez lógica, a construção do grafo de vizinhança impõe uma complexidade computacional de ordem cúbica ($O(N^3)$), tornando-o um gargalo crítico [Arias-Garcia et al. 2024]. Para superar essa limitação, arquiteturas massivamente paralelas, como as Unidades de Processamento Gráfico (GPUs), surgem como uma alternativa eficiente para cálculos complexos em grandes estruturas de dados [Owens et al. 2008, Zhang et al. 2023]. Ao explorar o modelo de execução *Single Instruction, Multiple Threads* (SIMT), torna-se possível acelerar o NN-clas ao paralelizar os cálculos de independência entre pares de elementos, mitigando o custo geométrico do algoritmo [NVIDIA Corporation 2025].

2. Trabalhos Relacionados

A aceleração de classificadores geométricos tem sido alvo de diferentes investigações devido ao seu alto custo computacional. Enquanto o NN-clas provou-se eficaz na definição de regiões de decisão robustas [Gade et al. 2018], sua viabilidade em larga escala permanece um desafio. Diferentes abordagens de hardware foram exploradas para mitigar o gargalo da construção do grafo; por exemplo, [Arias-Garcia et al. 2024] propuseram uma implementação em FPGA, demonstrando que a especialização de hardware é capaz de reduzir significativamente o tempo de processamento em relação a CPUs convencionais.

Por outro lado, o uso de GPUs consolidou-se como uma alternativa mais flexível e acessível para o paralelismo massivo. Trabalhos como os de [Zhang et al. 2023] evidenciam que arquiteturas SIMT são ideais para estruturas de dados extensas que exigem cálculos de distância exaustivos. No domínio da saúde e IoT, estudos comparativos entre bibliotecas de IA revelam que a migração para a GPU não apenas aumenta o throughput, mas é o fator determinante para a viabilidade de sistemas de baixa latência [Çolhak et al. 2025]. Este trabalho diferencia-se das abordagens anteriores ao combinar a eficiência das GPUs com uma estratégia de mapeamento implícito de memória, otimizando não apenas o tempo de cálculo, mas também o consumo de recursos de memória no contexto específico dos dados.

3. Metodologia

O fluxo metodológico deste trabalho integra o processamento de dados farmacêuticos via fluxo MASSA e a aceleração do *NN-clas* por meio da plataforma CUDA.

3.1. Fundamentação do NN-clas

O *NN-clas* é um classificador de margem larga que opera através da análise da vizinhança geométrica dos dados. Seu funcionamento estruturasse em três etapas principais: a construção e filtragem do Grafo de Gabriel, a identificação das arestas de suporte e, por fim, a classificação baseada nos vértices de borda.

A base do algoritmo é o Grafo de Gabriel $G = (V, A)$. Dado um conjunto de treinamento $S = \{x_i\}_{i=1}^{N}$, uma aresta (x_i, x_j) existe se, e somente se, nenhum outro ponto x_k está contido no interior da hiperesfera de diâmetro igual à distância entre x_i e x_j o que forma o grafo que pode ser observado na Figura 1 [Gabriel and Sokal 1969]. A condição matemática é expressa pela Inequação 1:

$$\|x_i - x_j\|^2 \leq \|x_i - x_k\|^2 + \|x_j - x_k\|^2, \forall x_k \in V \backslash \{x_i, x_j\} \tag{1}$$

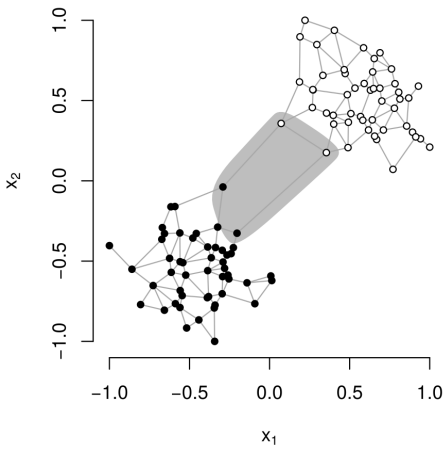

Figura 1. Exemplo de um conjunto de dados de um problema de classificação binária com o correspondente grafo de Gabriel e o subgrafo destacado. Fonte: [Gade et al. 2018].

Para garantir a robustez do modelo em cenários com sobreposição de classes ou ruído, aplica-se uma etapa de filtragem conforme proposto por [Garcia et al. 2015]. Amostras identificadas como ruidosas são removidas antes da fase de reconstrução do grafo, garantindo que a fronteira de decisão seja definida por pontos representativos.

A partir do grafo filtrado, identificam-se as Arestas de Suporte (AS), definidas como o conjunto de arestas em A cujos vértices (x_i, x_j) possuem rótulos de classes distintas. Os vértices que compõem essas arestas são denominados "vértices de borda"e delimitam a margem de separação entre as classes.

Após a filtragem de ruído e a identificação das AS, a classificação de novas amostras é realizada. O método utiliza como referência apenas os vértices que compõem o conjunto AS, os quais são referidos como "vértices de borda". Para classificar uma nova amostra de teste x, o algoritmo calcula a distância entre x e todos os vértices de borda. A amostra x recebe então o rótulo do vértice de borda que apresentar a menor distância.

3.2. Paralelização em GPU e Mapeamento de Índices

A construção do grafo exige comparar todos os pares (i, j) contra todos os pontos k. Para evitar o consumo de memória $O(N^2)$ de uma matriz de adjacência explícita, adotou-se um mapeamento implícito de índices, logo, não é armazenado uma matriz de adjacência $N \times N$, mas sim um vetor linear de tamanho $N(N-1)/2$ (triangular superior).

Para que cada *thread* saiba qual par (i, j) processar a partir de seu índice global k, utilizamos um mapeamento analítico. As Equações 2 e 3 derivam da inversão da fórmula

da soma de uma progressão aritmética, permitindo que a GPU calcule as coordenadas (i, j) em tempo constante $O(1)$, sem consultas a tabelas em memória.

$$i = N - 2 - \left\lfloor \frac{\sqrt{4N(N-1) - 8k - 7}}{2} - 0.5 \right\rfloor \tag{2}$$

$$j = k - \left(\frac{N(N-1)}{2} - \frac{(N-i)(N-i-1)}{2} \right) + i + 1 \tag{3}$$

A Equação 2 resolve para i a inequação da posição da linha na matriz triangular, enquanto a 3 ajusta o deslocamento da coluna j. Essa estratégia é fundamental para manter o paralelismo massivo sem o gargalo de memória $O(N^2)$.

Para garantir a integridade e eficiência na VRAM, o algoritmo utiliza uma abordagem de duas passadas com operações atômicas:

- **Passada 1 (Marcação):** O kernel verifica a condição de Gabriel e utiliza `atomicOr` para marcar a existência da aresta em um *bitmask*. Isso reduz o consumo de memória em 32 vezes em relação ao uso de tipos `int` convencionais. Um contador global via `atomicAdd` determina o tamanho exato da alocação necessária para o vetor de arestas.
- **Passada 2 (Escrita):** As *threads* consultam o *bitmap* e realizam operações atômicas para obter posições de escrita seguras em um vetor denso final, eliminando condições de corrida e garantindo que o uso de memória seja proporcional estritamente às arestas reais do grafo.

4. Configuração Experimental

Para a avaliação, foram selecionados quatro datasets processados via fluxo MASSA (Tabela 1). O termo Características refere-se ao número de descritores moleculares de cada amostra, o que define a dimensionalidade do espaço euclidiano. Por exemplo, no conjunto MYCO, o algoritmo opera em um espaço de 7 dimensões para calcular as hiperesferas de influência do Grafo de Gabriel. O volume de Instâncias (N) determina a carga de trabalho computacional.

Tabela 1. Descrição dos conjuntos de dados farmacêuticos.

Dataset	Instâncias (N)	Características (Atributos)
ACIN	2.457	6
ECOLI	22.770	5
KLEB	8.335	5
MYCO	15.092	7

Para quantificar os ganhos de desempenho, o ambiente experimental (Tabela 2) foi estruturado para confrontar a execução sequencial em um processador convencional (*host*) com a versão paralelizada em uma GPU de alto desempenho (*device*). A infraestrutura utiliza uma GPU NVIDIA RTX 4070 Ti, equipada com 7.680 núcleos CUDA, contrapondo-se à arquitetura quad-core do AMD Ryzen 3 2200G.

Tabela 2. Especificações do ambiente de hardware e software.

Componente	Especificação
CPU	AMD Ryzen 3 2200G (4 cores, 3.5GHz)
GPU	NVIDIA RTX 4070 Ti (7680 CUDA Cores, 12GB VRAM)
Compilador	NVCC 11.5 / GCC 9.4

5. Resultados e Discussão

Os experimentos conduzidos para avaliar a viabilidade computacional da aceleração em GPU do classificador NN-clas consolidam o desempenho da solução através de métricas de eficiência e eficácia. Conforme detalhado na Tabela 3, a análise abrange o tempo de execução sequencial (TSeq) e paralelo (TPar), o fator de aceleração obtido (*Speedup*), o consumo energético em Joules (J) e a performance preditiva medida por Acurácia e F1-Score. Essa estrutura permite observar o comportamento da solução sob diferentes escalas de complexidade topológica, correlacionando o ganho de velocidade com a manutenção da precisão diagnóstica.

Tabela 3. Resultados de desempenho e métricas de classificação em GPU

Dataset	TSeq. (s)	TPar. (s)	Speedup	Energia (J)	Acc (%)	F1-Score
ACIN	25,3701	0,1168	**217,20**	2,859	75,38	0,6285
ECOLI	2127,1923	0,7458	**2852,22**	104,161	78,93	0,6614
KLEB	219,1924	0,1616	**1356,38**	9,876	72,64	0,6261
MYCO	1059,4564	0,3772	**2808,73**	45,307	64,42	0,5525

A análise do desempenho temporal revela um ganho de escala para o classificador NN-clas. O *speedup* máximo de **2.852,22x** observado na base ECOLI demonstra que o gargalo cúbico, anteriormente impeditivo para grandes volumes de dados, foi efetivamente mitigado na nova implementação. Enquanto a execução sequencial na base ECOLI exigiria aproximadamente 35 minutos de processamento, a versão acelerada concluiu a tarefa em apenas 0,74 segundos.

Este desempenho sub-segundo em todas as bases testadas possui implicações práticas diretas no domínio da saúde. Em cenários de diagnóstico laboratorial de patógenos como *E. coli* e *K. pneumoniae*, a latência reduzida permite que o profissional de saúde receba uma classificação do perfil molecular quase instantaneamente após a coleta e o pré-processamento. Tal agilidade pode ser crucial para a prescrição imediata de antibióticos assertivos, combatendo a resistência bacteriana em estágios críticos da infecção.

Além da velocidade, a eficiência energética observada (ex: 2,859 J para ACIN) sugere que o modelo pode ser integrado a dispositivos de *Edge Computing* em ambientes clínicos, operando com baixo custo operacional. Embora a base MYCO tenha apresentado o maior desafio preditivo (F1-Score 0,5525), possivelmente devido à maior sobreposição geométrica das classes, a manutenção da acurácia próxima a 75% nas demais bases, aliada à redução drástica de tempo, valida a solução como um motor de inferência robusto para suporte à decisão clínica em tempo real.

6. Conclusão e Trabalhos Futuros

Este trabalho demonstrou que a aceleração em GPU do classificador NN-clas é uma solução eficaz para o processamento de grandes volumes de dados farmacêuticos provenientes do fluxo MASSA. A estratégia de paralelização fundamentada no modelo SIMT, integrada ao mapeamento implícito de índices, permitiu superar o gargalo cúbico da construção do Grafo de Gabriel, reduzindo o custo de memória sem comprometer a integridade lógica do algoritmo.

A viabilidade técnica da abordagem é confirmada pela redução drástica do tempo de resposta para a escala sub-segundo, tornando o NN-clas apto para aplicações de suporte à decisão clínica em tempo real. Como desdobramentos desta pesquisa, pretende-se:

- Explorar a escalabilidade em ambientes multi-GPU e técnicas de *tiling* para conjuntos de dados que excedam a capacidade de memória de um único dispositivo;
- Realizar um *benchmarking* comparativo com outros algoritmos de estado da arte em classificação geométrica e redes neurais leves;
- Otimizar o perfil energético, visando a consolidação de algoritmos de computação verde adaptados para infraestruturas hospitalares e dispositivos de borda.

Referências

Arias-Garcia, J., de Souza, A. C., Gade, L., Yudi, J., Coelho, F., Castro, C. L., Torres, L. C. B., and Braga, A. P. (2024). Improved design for hardware implementation of graph-based large margin classifiers for embedded edge computing. *IEEE Transactions on Neural Networks and Learning Systems*, 35(1):1320–1329.

Gabriel, K. R. and Sokal, R. R. (1969). A new statistical approach to geographic variation analysis. *Systematic Biology*, 18(3):259–278.

Gade, L., Castro, C., Torres, L., Coelho, F., Braga, A., Arias García, J., and Sill Torres, F. (2018). Nn-clas: classificador geométrico de margem larga baseado na regra do vizinho mais próximo. pages 1–12.

Garcia, L. P., de Carvalho, A. C., and Lorena, A. C. (2015). Effect of label noise in the complexity of classification problems. *Neurocomputing*, 160:108–119.

NVIDIA Corporation (2025). *CUDA C++ Programming Guide*. NVIDIA Corporation. Version 13.0. Acessado em: 27 out. 2025.

Owens, J. D., Houston, M., Luebke, D., Green, S., Stone, J. E., and Phillips, J. C. (2008). Gpu computing. *Proceedings of the IEEE*, 96(5):879–899.

Veríssimo, G. C., Pantaleão, S. Q., Fernandes, P. O., Gertrudes, J. C., Kronenberger, T., Honorio, K. M., and Maltarollo, V. G. (2023). Massa algorithm: an automated rational sampling of training and test subsets for qsar modeling. *Journal of Computer-Aided Molecular Design*, 37(12):735–754.

Zhang, Y., He, G., Ma, L., et al. (2023). A gpu-based computational framework that bridges neuron simulation and artificial intelligence. *Nature Communications*, 14:5798.

Çolhak, F., Coşkun, H., Tsafac Nkombong, R. C., Hoxa, T., Ecevit, M. , and Aydin, M. N. (2025). Accelerating iov intrusion detection: Benchmarking gpu-accelerated vs cpu-based ml libraries. `https://arxiv.org/abs/2504.01905`. Available at: `https://arxiv.org/abs/2504.01905`.