Algoritmo Competitivo de Agrupamento Não Supervisionado Baseado em Heurística de Propagação Viral

Bassem Youssef Makhoul Junior, Eduardo Furtado de Simas Filho
Programa de Engenharia Eltétrica e de Computação
Universidade Federal da Bahia
Salvador, Brasil
bassemmakhoul@ufba.br
eduardo.simas@ufba.br

***Resumo*—Agrupamento (*clustering*) é uma das tarefas centrais do aprendizado não supervisionado, sendo comumente limitada pela necessidade de definir previamente o número de grupos. Este trabalho propõe o algoritmo *Virus Propagation Clustering* (VPC), uma abordagem bioinspirada baseada em propagação e competição viral. O método permite identificar agrupamentos de forma adaptativa, sem exigir conhecimento prévio da quantidade de grupos, e ainda destaca regiões densas nos dados de forma visual. O algoritmo foi avaliado em conjuntos de dados públicos por meio de validação cruzada com cinco partições, utilizando métricas como *accuracy*, *precision*, *recall* e *F1-score*. Os resultados mostram que o VPC apresenta desempenho competitivo frente a métodos consagrados, como *K-means* e *Gaussian Mixture*, oferecendo, adicionalmente, interpretabilidade e adaptabilidade superiores em estruturas de dados complexas.**

***Index Terms*—Propagação de Vírus, Competição e Cooperação, Aprendizado Não-Supervisionado.**

I. Introdução

O agrupamento (*clustering*) é uma das tarefas fundamentais do aprendizado não supervisionado, com aplicações em descoberta de padrões, segmentação de dados e pré-processamento para modelos supervisionados. Apesar da ampla adoção de algoritmos como *K-means*, *Gaussian Mixture Models* e métodos baseados em densidade (como DBSCAN), uma limitação recorrente é a necessidade de conhecer previamente o número de grupos nos dados [1], [2]. Esse requisito torna-se especialmente problemático em cenários de alta dimensionalidade e em domínios com estruturas complexas [3]–[5].

Com o crescimento exponencial de dados gerados em tempo real, especialmente em contextos como cidades inteligentes, internet das coisas e redes complexas [6], [7], o volume e a heterogeneidade da informação tornam inviável rotular todos os dados manualmente. Nesses casos, o aprendizado não supervisionado torna-se essencial [8], com destaque para algoritmos capazes de identificar padrões de forma adaptativa.

Entre as abordagens não supervisionadas, os algoritmos de agrupamento competitivo se destacam por simular dinâmicas bioinspiradas, nas quais representações (como neurônios ou vetores) competem por pontos de dados e ajustam suas posições em direção às regiões mais densas [9], [10]. Um exemplo clássico é o *Self-Organizing Map* (SOM), que utiliza princípios de competição neural para organizar os dados no espaço [9]. De forma semelhante, algoritmos inspirados em comportamentos biológicos, como redes neurais [11] e algoritmos genéticos [12], têm sido amplamente aplicados em tarefas de otimização e representação de dados.

Neste trabalho, propomos o algoritmo *Virus Propagation Clustering* (VPC), uma abordagem bioinspirada que modela o processo de propagação viral em ambientes com recursos limitados. Diferentes tipos de vírus competem ou cooperam por pontos de dados, formando grupos que evoluem ao longo de iterações. Ao final, apenas os vírus mais adaptados sobrevivem, representando as regiões de maior densidade no espaço amostral [13]–[15]. Para refinar as regiões obtidas, o VPC utiliza uma etapa de *bagging*, permitindo gerar representações robustas que podem ser posteriormente refinadas com algoritmos como *Mean Shift* [16] ou *Kernel Density Estimation* [17].

Diferentemente do *Virus Optimization Algorithm* [18], o VPC não se destina à otimização com função objetivo explícita, mas sim ao agrupamento não supervisionado com foco em visualização e adaptabilidade. O algoritmo se destaca por não exigir definição prévia do número de grupos e por proporcionar uma visualização interpretável das regiões densas nos dados.

Este artigo está organizado da seguinte forma: a Seção II descreve o funcionamento do algoritmo proposto, dividido em três fases principais. A Seção III apresenta a metodologia e os conjuntos de dados utilizados para avaliação. A Seção IV discute os resultados obtidos. Por fim, a Seção V traz as conclusões e perspectivas futuras.

II. VPC

O algoritmo proposto simula a dinâmica de interações entre grupos virais em um ambiente com recursos limitados. Ele opera em três fases principais, resultando em regiões densas no espaço de dados, cujos centróides podem ser estimados posteriormente por métodos baseados em densidade. Esta seção descreve detalhadamente cada uma das fases. A Tabela I apresenta a notação matemática utilizada.

Tabela I
NOTAÇÕES E DESCRIÇÕES UTILIZADAS NO ALGORITMO

Notação	Descrição
$\mathbf{X} = \{x_1, x_2, \ldots, x_N\}$	Conjunto de instâncias dos dados
$\mathbf{x}_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,d})$	Instância d-dimensional (i-ésima instância)
$\mathbf{W} = \{w_1, w_2, \ldots, w_n\}$	Conjunto com posições iniciais dos vírus
$\mathbf{w}_i = (w_{i,1}, w_{i,2}, \ldots, w_{i,d})$	Representação d-dimensional da posição do i-ésimo vírus
$\mathbf{G} = (g_1, g_2, \ldots, g_n)$	Dicionário de pontos infectados por cada vírus
$g_i = \{\mathbf{x}_1^i, \mathbf{x}_2^i, \ldots\}$	Conjunto com os pontos infectados pelo vírus i
$\lvert g_i \rvert$	Quantidade de pontos infectados por i
N	Número total de pontos no conjunto de dados
n	Hiperparâmetro: número de vírus
n_{min}	Hiperparâmetro: número mínimo de pontos para que um vírus sobreviva
n_{max}	Número máximo de pontos infectados por um vírus na fase 1
n_m	Número de movimentções realizadas na fase 3
α	Hiperparâmetro: taxa de aprendizado para atualização de posições
ϵ	Hiperparâmetro: quantidade extra de pontos infectáveis na fase 1
δ	Distância mínima entre ponto e vírus na fase 1
δ'	Distância de iteração para a fase 2
t_{max}	Número máximo de iterações no processo competitivo

A. Fase 1: Infecção Inicial

A Fase 1 tem como objetivo gerar os grupos virais iniciais que irão competir nas etapas seguintes. O número total de vírus é definido pelo hiperparâmetro n. Cada vírus é representado por uma posição no espaço d-dimensional, em que $w_i \in \mathbb{R}^d$, e por um conjunto de pontos infectados (g_i). Inicialmente, os vetores w_i são amostrados aleatoriamente do conjunto $\mathbf{X}$ e normalizados no intervalo $[0, 1]$ via escala *min-max*.

A infecção ocorre de forma iterativa. A cada passo, calcula-se a distância de Minkowski entre o vetor w_i e um ponto não infectado x_j:

$$d(w_i, x_j) = \left(\sum_{k=1}^{d} |w_i^p - x_j^p|^r \right)^{1/r}. \quad (1)$$

em que o índice p está relacionado com a dimensão dos dados. Se $d(w_i, x_j) < \delta$ (em que δ é a média das distâncias aos 5 vizinhos mais próximos de cada vírus) e $|g_i| < n_{\max}$, o ponto x_j é considerado infectado e adicionado ao grupo g_i. Após a avaliação de todos os pontos, a posição w_i é atualizada para a média dos elementos de seu grupo:

$$w_i = \frac{1}{|g_i|} \sum_{x_j \in g_i} x_j. \quad (2)$$

A Figura 1 ilustra esse processo, análogo ao utilizado nos mapas auto-organizáveis de Kohonen. Inicialmente, mapeamos cada ponto a um respectivo vírus de forma aleatória, após isto, cada vírus tentará infectar os pontos mais próximos, respeitando o limite de distância δ e a capacidade máxima $n_{\max}$. Caso o ponto não satisfaça ambos os critérios, ele será descartado nesta fase. Ao final, os grupos virais formados representam regiões preliminares de concentração no espaço de dados. A seguir, δ é recalculado com base na média das cinco menores distâncias entre os grupos resultantes.

O valor de $n_{\max}$ é definido por:

$$n_{\max} = \left\lfloor \frac{|X|}{n} \right\rfloor + \epsilon, \quad (3)$$

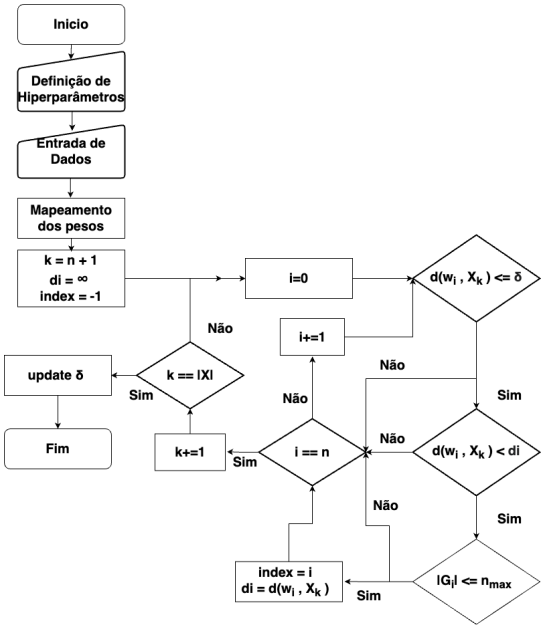

Figura 1. Diagrama da Fase 1: infecção inicial dos pontos.

onde ϵ é um hiperparâmetro empírico que ajusta a tolerância de infecção. A complexidade computacional dessa fase é $\mathcal{O}(n(N-n))$, uma vez que cada ponto é avaliado por todos os vírus.

1) Atualização de δ: Para limitar o alcance dos vírus nas próximas fases, recalcula-se δ como a média das cinco menores distâncias entre cada vetor w_i e seus vizinhos mais próximos, definindo a nova δ'. O número de cinco vizinhos foi definido empiricamente, com base no desempenho observado durante os testes iniciais.

B. Fase 2: Competição e Cooperação

Nesta etapa, os grupos de vírus interagem entre si por meio de coexistência [19], [20] ou competição [21], [22]. A cada iteração, um vírus i é selecionado aleatoriamente. Ele tentará

competir com outro grupo ou, caso não consiga, buscará fundir-se com ele.

A competição ocorre quando $|g_i| \geq |g_j|$. A probabilidade de w_i vencer é dada por:

$$P(w_i \text{ vence } w_j) = \frac{|g_i|}{|g_i| + |g_j|}. \tag{4}$$

Caso vença, um ponto aleatório de g_j é transferido para g_i, e w_i é atualizado tomando a média aritmética dos pontos infectados. Caso contrário, w_j vence e absorve um ponto de g_i. Este processo pode ser observado na Figura Y.

Na coexistência, os grupos g_i e g_j podem se juntar se a soma dos seus tamanhos não ultrapassar o tamanho do maior grupo viral, e se:

$$\frac{|g_i|}{|g_i| + |g_j|} \geq p,$$

onde p é um número aleatório no intervalo $[0, 1]$. A nova posição resultante é calculada pela média aritmética do novo conjunto de pontos, em que este é o resultado da combinação dos dois conjuntos, tal pode ser observado na Figura 3. A Figura 2 apresenta o diagrama dessa fase.

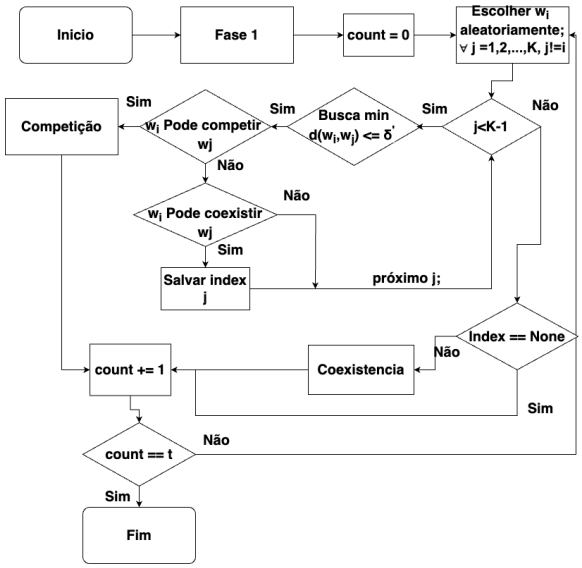

Figura 2. Diagrama da Fase 2: competição e cooperação entre vírus.

Após $t_{\max}$ iterações, os vírus que acumularem mais pontos sobrevivem e avançam para a próxima fase. A Figura 3 ilustra os operadores utilizados neste processo. A complexidade desta fase é da ordem de $\mathcal{O}(t_{\max} \cdot n \cdot \log n)$, uma vez que a cada iteração, um vírus interage com seu vizinho mais próximo, exigindo atualizações locais de posição e redistribuição de pontos.

C. Fase 3: Consolidação com Bagging

Na fase final, emprega-se o método de *bagging* sem reposição para consolidar os agrupamentos, tornando o algoritmo com complexidade $\mathcal{O}(nN + n \log n)$. A cada iteração,

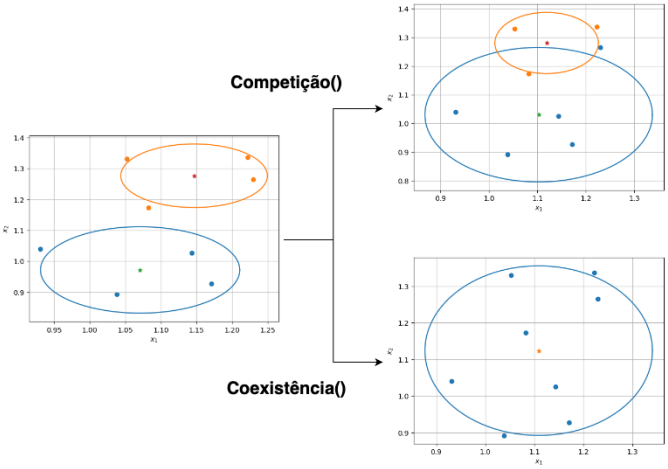

Figura 3. Exemplo dos operadores de competição e coexistência. À esquerda, dois grupos iniciais com suas médias (estrelas) e elipses de dispersão. À direita, Na imagem superior é aplicado a competição, eliminando o ponto de um grupo e adicionado a outro; a coexistência funde ambos em um único grupo.

um subconjunto de $\mathbf{X}$ é amostrado e submetido novamente às Fases 1 e 2. Os grupos gerados são organizados de forma decrescente em relação ao número de pontos, e serão tomados apenas os $n_{\min}$ primeiros grupos, com pelo menos 10 pontos cada, isto irá garantir que grupos pequenos e afastados dos demais não irão participar do agrupamento final. O processo é ilustrado na Figura 4.

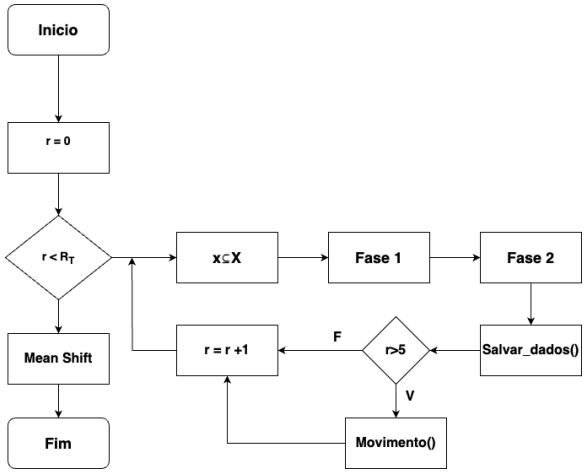

Figura 4. Diagrama da Fase 3: consolidação dos agrupamentos.

Nas cinco primeiras iterações, os grupos permanecem estáticos para formar uma população inicial. A partir da sexta iteração, suas posições são atualizadas em direção ao vizinho mais próximo:

$$w_i = w_i + \alpha \cdot (w_j - w_i), \tag{5}$$

onde w_j é o vizinho mais próximo de w_i.

Durante esse processo, todos os vírus se movem simultaneamente em direção aos respectivos vizinhos mais próximos.

Para evitar sobreposição de movimentos, os índices já movimentados são registrados a cada passo. A taxa de aprendizado α também é atualizada dinamicamente conforme:

$$\alpha = \alpha + (\alpha_{\max} - \alpha) \cdot \frac{r}{R}, \tag{6}$$

sendo $\alpha_{\max} = 0{,}5$ e r/R o fator de progressão da iteração corrente.

Esse processo de movimentação e refinamento pode ser aplicado n_m vezes, produzindo agrupamentos robustos, assim, facilitando a aplicação de métodos como *Mean Shift* para a extração final dos centróides. *Mean Shift* é um algoritmo de agrupamento não supervisionado que tem como objetivo encontrar regiões com maior densidade de dados. A cada passo, ele calcula a média dos pontos vizinhos em relação a um ponto i e move este para essa média.

III. Metodologia

Como o algoritmo proposto é não supervisionado e tem como um de seus principais objetivos a identificação de centróides, sua avaliação será realizada por meio da comparação com dois algoritmos clássicos de agrupamento: *K-means* (KM) [23] e *Gaussian Mixture* (GM) [24]. Já o algoritmo *Mean Shift* será aplicado na etapa final do método, utilizando uma largura de banda (*bandwidth*) fixa de 0,1, que define a distância máxima de influência entre os pontos.

O objetivo é ajustar adequadamente os hiperparâmetros (n, n_{min}, α) de modo que todas as regiões distintas presentes nos dados sejam corretamente identificadas. Esses valores foram definidos empiricamente com base em experimentação, testando-se combinações de $n \in \{5, 10, 20\}$, $\alpha \in \{0{,}05, 0{,}2\}$ e $n_{min} \in \{3, 5, 7\}$, conforme detalhado na seção de resultados. Após a consolidação das regiões por meio do algoritmo proposto, o algoritmo *Mean Shift* foi aplicado para refinar a estrutura dos agrupamentos.

Os conjuntos de dados utilizados para avaliação incluem os tradicionais *Iris Dataset*, *Wine Dataset* e *Breast Cancer Dataset* [25], amplamente empregados como benchmarks na literatura de aprendizado não supervisionado. Todos os dados foram normalizados para o intervalo $[0, 1]$ por meio de escala min-max.

A fim de garantir robustez na avaliação, foi aplicada validação cruzada com 5 partições (*5-fold cross-validation*). Embora o método seja não supervisionado, as métricas foram calculadas com base na correspondência com os rótulos reais disponíveis nos conjuntos de dados. As médias das seguintes métricas de desempenho foram registradas: *accuracy* (exatidão), *precision* (precisão), *recall* (revocação) e *F1-score* [26].

Para os métodos comparativos baseados em partição, como K-Means e Gaussian Mixture Models (GMM), o número de clusters k foi fixado em 3, correspondente ao número de classes reais no conjunto *Iris*. Embora seja sabido que métodos como o K-Means podem, por vezes, favorecer agrupamentos com $k = 2$ em alguns conjuntos, optou-se por manter k igual ao número "real"de grupos de forma a permitir uma comparação direta entre os centróides gerados pelo método proposto e os centróides obtidos por métodos tradicionais sob uma segmentação comum. Tal escolha visa avaliar o alinhamento estrutural dos agrupamentos produzidos, mais do que otimizar a performance de cada algoritmo individualmente.

IV. Resultados

Primeiro, serão apresentados os resultados do modelo proposto para diferentes valores de hiperparâmetros no conjunto de dados Iris. Em seguida, as métricas mencionadas acima serão comparadas para os algoritmos de aprendizado não supervisionado em alguns conjuntos de dados do *Toy Dataset*.

A. Análise de Hiperparâmetros

Nesta seção, analisamos o desempenho do algoritmo proposto frente a diferentes configurações de hiperparâmetros e o comparamos com métodos clássicos de agrupamento em três conjuntos de dados amplamente utilizados. Inicialmente, apresentamos uma análise exploratória da influência dos hiperparâmetros no conjunto *Iris*, seguida de uma avaliação quantitativa em múltiplos conjuntos de dados.

B. Análise de Hiperparâmetros

No Apêndice A, as Figuras 8-12 ilustram os efeitos dos hiperparâmetros nas regiões de agrupamento formadas. Em todos os experimentos, os valores de $\epsilon = 2$ e $n_m = 4$ foram definidos empiricamente.

Observa-se que não existe uma combinação única de hiperparâmetros que produza os melhores resultados em todos os casos. No entanto, a maioria das configurações avaliadas gerou agrupamentos coesos e bem separados. Por exemplo, as Figuras 8, 9 e 11 apresentam resultados superiores aos observados nas Figuras 10 e 12, no sentido de que os grupos formados exibem menor granularidade, isto é, menor dispersão dos pontos em relação à média dos respectivos grupos, além de uma separação mais evidente entre os agrupamentos.

A quantidade de vírus, representada por n, define o número inicial de grupos a serem gerados. Valores baixos desse parâmetro favorecem a alocação de vírus em regiões mais densas, como pode ser observado na Figura 10. Por outro lado, valores elevados de n tendem a gerar grupos mais dispersos ao longo do conjunto de dados, conforme ilustrado nas Figuras 9 e 11.

O parâmetro que define o número mínimo de elementos por grupo ($n_{\min}$) exerce forte influência na granularidade dos agrupamentos. Valores muito baixos tendem a gerar grupos espúrios ou ruidosos, uma vez que poucos grupos serão mantidos após a segunda fase do algoritmo. Isso pode contribuir para que alguns agrupamentos fiquem levemente mais dispersos, como ilustrado na Figura 10. Por outro lado, valores muito elevados podem eliminar estruturas relevantes, pois impedem a formação de pequenos grupos, os quais, apesar de reduzidos, podem representar padrões distintos. Essa situação é exemplificada na Figura 8. Em nossos experimentos, observou-se que a escolha de $n_{\min} = 5$ resultou em divisões mais consistentes e representativas.

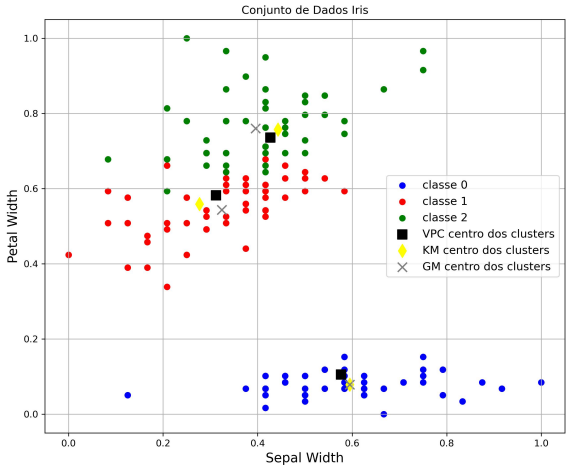

Figura 5. Conjunto de pontos do *Iris* em que cada classe é demarcada por uma cor diferente, juntamente com os centroides encontrados pelos algoritmos não supervisionados.

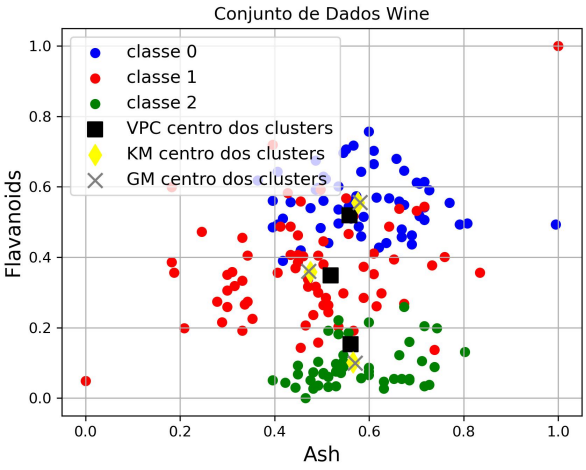

Figura 6. Conjunto de pontos do *Wine* em que cada classe é demarcada por uma cor diferente, juntamente com os centroides encontrados pelos algoritmos não supervisionados.

A taxa de aprendizagem α também impacta a consolidação dos agrupamentos. Valores altos resultam em deslocamentos abruptos, que podem desestabilizar a formação dos grupos. Já valores muito baixos dificultam a convergência. As Figuras 8 e 10 destacam essa sensibilidade. Em geral, é difícil estimar os parâmetros ideais para obter a melhor divisão entre os grupos finais. Valores baixos de n podem ser compensados por valores mais altos de α. Observou-se, contudo, que valores de $\alpha \in [0{,}05,\ 0{,}1]$ apresentaram bom desempenho, e uma estimativa eficaz para a quantidade inicial de vírus foi $n \approx |X|/3$.

C. Comparação com Métodos Clássicos

Para realizar as comparações entre os algoritmos, foram empregados os valores de $n \approx |X|/3$, $\alpha = 0{,}1$ e $n_{min} = 5$. As Figuras 5–7 apresentam os centróides obtidos pelos algoritmos *K-means*, *Gaussian Mixture* e pelo VPC para os conjuntos de dados *Iris*, *Wine* e *Breast Cancer*. Visualmente, o VPC localiza regiões de alta densidade de forma eficaz, mesmo sem conhecer o número de grupos a priori.

As Tabelas II, III e IV mostram as métricas médias e desvios padrão obtidos após validação cruzada com 5 partições. Nota-se que o VPC atinge desempenho competitivo, apesar de ligeiramente inferior aos métodos tradicionais em alguns casos. Essa diferença é atribuída principalmente à escolha fixa do parâmetro de largura de banda no *Mean Shift*, que poderia ser ajustado dinamicamente ao final da Fase 3.

	f1	acurácia	precisão	recall
VPC	0,925±0,055	0,933±0,047	0,932±0,055	0,922±0,055
KM	0,883±0,048	0,886±0,049	0,898±0,055	0,884±0,042
GM	0,959±0,044	0,960±0,489	0,967±0,035	0,960±0,043

Tabela II
TABELA DE MÉTRICAS PARA O CONJUNTO DE DADOS *Iris*, COM SEUS RESPECTIVOS DESVIOS.

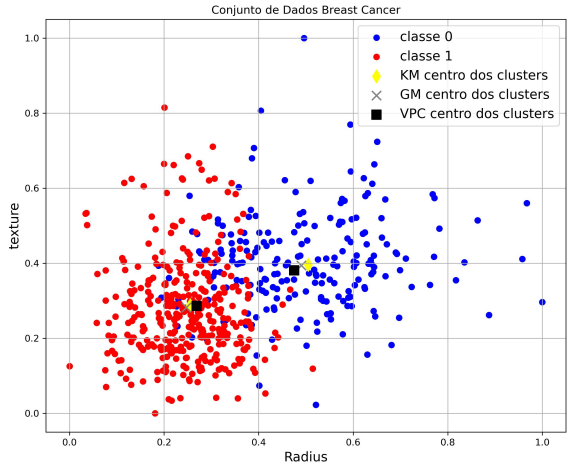

Figura 7. Conjunto de pontos do *Breast Cancer* em que cada classe é demarcada por uma cor diferente, juntamente com os centroides encontrados pelos algoritmos não supervisionados.

	f1	acurácia	precisão	recall
VPC	0,876±0,050	0,886±0,041	0,890±0,052	0,887±0,051
KM	0,943±0,036	0,948±0,033	0,941±0,043	0,957±0,022
GM	0,959±0,024	0,960±0,022	0,960±0,023	0,961±0,026

Tabela III
TABELA DE MÉTRICAS PARA O CONJUNTO DE DADOS *Wine*, COM SEUS RESPECTIVOS DESVIOS.

	f1	acurácia	precisão	recall
VPC	0,868±0,011	0,873±0,012	0,863±0,012	0,886±0,012
KM	0,921±0,028	0,927±0,026	0,933±0,0224	0,913±0,031
GM	0,934±0,017	0,938±0,011	0,931±0,016	0,938±0,018

Tabela IV
TABELA DE MÉTRICAS PARA O CONJUNTO DE DADOS *Breast Cancer*, COM SEUS RESPECTIVOS DESVIOS.

Uma vantagem importante do algoritmo proposto é sua capacidade de estimar automaticamente o número de agrupamentos, sem necessidade de informação prévia. Além disso, o VPC permite visualizar regiões densas do espaço de dados com alta frequência de vírus vencedores, o que facilita a interpretação dos agrupamentos.

Diferentemente de algoritmos clássicos, que exigem a definição antecipada do número de grupos, o VPC pode atuar como uma etapa de pré-processamento. Ele identifica regiões centrais dos dados e sugere, de forma interpretável, a estrutura de agrupamento, podendo ser combinado posteriormente com algoritmos como *Gaussian Mixture* ou *K-means* para refinamento supervisionado.

V. Conclusões

Este trabalho apresentou o algoritmo *Virus Propagation Clustering* (VPC), um método de agrupamento não supervisionado inspirado em processos de propagação e competição viral. O algoritmo foi descrito em detalhes, evidenciando suas três fases principais: infecção inicial, competição/coexistência e consolidação via bagging. Através de experimentos com conjuntos de dados amplamente utilizados, demonstrou-se que o VPC é capaz de identificar centróides de forma competitiva em relação a métodos clássicos como *K-means* e *Gaussian Mixture*, com o diferencial de não requerer conhecimento prévio do número de grupos.

É importante destacar que o VPC não se propõe a superar algoritmos tradicionais como K-means ou Gaussian Mixture em desempenho absoluto. Em vez disso, sua principal contribuição está na capacidade de fornecer uma visualização preliminar da estrutura dos dados, sugerindo regiões densas e possíveis agrupamentos que podem posteriormente ser refinados por métodos mais robustos. Essa característica torna o VPC uma ferramenta útil em fases exploratórias do pipeline de análise, especialmente em cenários com ruído ou estrutura complexa. Mesmo não havendo valores ótimos dos hiperparâmetros, o método ainda é capaz de identificar clusters bem definidos tomando as aproximações empregadas, mesmo na presença de pontos levemente dispersos. Essa resiliência evidencia a robustez do VPC frente a variações nos dados e reforça sua aplicabilidade em cenários com estruturas complexas e ruído moderado.

Apesar de depender de mais hiperparâmetros do que algoritmos clássicos como o K-means, o VPC demonstrou convergência estável quanto ao número final de agrupamentos. Mesmo com variações em n, α e $n_{\min}$, os agrupamentos produzidos tenderam a refletir a estrutura real dos dados, o que indica robustez frente a configurações alternativas. As interações entre vírus foram mantidas intencionalmente simples nesta versão inicial do VPC, visando facilitar a análise e o controle do comportamento emergente. Futuramente, operadores mais sofisticados poderão ser explorados para ampliar a adaptabilidade e expressividade do modelo em dados com estruturas mais complexas. Além disto, visa-se a paralelização da terceira fase para ganho de eficiência computacional, e a aplicação do modelo em contextos de detecção de anomalias. Em que neste último caso, vírus que não conseguem expandir seus agrupamentos podem ser interpretados como indicadores de regiões ruidosas ou pontos fora do padrão, permitindo uma abordagem natural para filtragem de outliers via votação interativa. Essas extensões posicionam o VPC como uma ferramenta promissora para exploração, segmentação e pré-processamento de dados não rotulados em cenários reais.

Agradecimentos

Este trabalho foi financiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), nº 145035/2021-2. Este estudo também foi financiado parcialmente pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

Referências

[1] Theodoridis, S., & Koutroumbas, K. (2006). *Pattern recognition.* Elsevier.

[2] El Abbassi, M., Overbeck, J., Braun, O. et al. *Benchmark and application of unsupervised classification approaches for univariate data.* Commun Phys 4, 50 (2021). https://doi.org/10.1038/s42005-021-00549-9

[3] D. Charte, F. Charte and F. Herrera, "Reducing Data Complexity Using Autoencoders With Class-Informed Loss Functions,"in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 12, pp. 9549-9560, 1 Dec. 2022, doi: 10.1109/TPAMI.2021.3127698

[4] G. -J. Qi and J. Luo, "Small Data Challenges in Big Data Era: A Survey of Recent Progress on Unsupervised and Semi-Supervised Methods,"in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 4, pp. 2168-2187, 1 April 2022, doi: 10.1109/TPAMI.2020.3031898.

[5] A. Maalouf, I. Jubran and D. Feldman, "Fast and Accurate Least-Mean-Squares Solvers for High Dimensional Data,"in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 12, pp. 9977-9994, 1 Dec. 2022, doi: 10.1109/TPAMI.2021.3139612. keywords

[6] Q. Zhang, L. T. Yang, and Z. Chen, "Deep computation model for unsupervised feature learning on big data," *IEEE Transactions on Services Computing*, vol. 9, no. 1, pp. 161–171, 2015. doi: 10.1109/TSC.2015.2420631.

[7] Chen, X. W., & Lin, X. (2014). *Big data deep learning: challenges and perspectives.* IEEE access, 2, 514-525.

[8] G. -J. Qi and J. Luo, "Small Data Challenges in Big Data Era: A Survey of Recent Progress on Unsupervised and Semi-Supervised Methods,"in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 4, pp. 2168-2187, 1 April 2022, doi: 10.1109/TPAMI.2020.3031898

[9] Kohonen, T. (1990). *The self-organizing map. Proceedings of the IEEE*, 78(9), 1464-1480.

[10] N. Masuyama, Y. Nojima, C. K. Loo and H. Ishibuchi, "Multi-Label Classification via Adaptive Resonance Theory-Based Clustering,"in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 7, pp. 8696-8712, 1 July 2023, doi: 10.1109/TPAMI.2022.3230414

[11] R. E. Uhrig, *Introduction to artificial neural networks*, Proceedings of IECON '95 - 21st Annual Conference on IEEE Industrial Electronics, Orlando, FL, USA, 1995, pp. 33-37 vol.1, doi: 10.1109/IECON.1995.483329.

[12] Q. Ye, A. A. Amini and Q. Zhou, *Optimizing Regularized Cholesky Score for Order-Based Learning of Bayesian Networks*, in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, no. 10, pp. 3555-3572, 1 Oct. 2021, doi: 10.1109/TPAMI.2020.2990820.

[13] U. Sommer and B. Worm, *Competition and Coexistence*. Springer, 2002.

[14] Levin, B. R., Stewart, F. M., & Chao, L. (1977). *Resource-limited growth, competition, and predation: a model and experimental studies with bacteria and bacteriophage*. The American Naturalist, 111(977), 3-24.

[15] Harper, D. (2011). *Viruses: biology, applications, and control*. Garland Science.

[16] D. Comaniciu and P. Meer, *"Mean shift: a robust approach toward feature space analysis,"* in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 5, pp. 603-619, May 2002, doi: 10.1109/34.1000236. keywords: Robustness;Pattern recognition;Convergence;Density functional theory;Kernel;Smoothing methods;Image segmentation;Image resolution;Image analysis;Image color analysis,

[17] Davis, R. A., Lii, K. S., & Politis, D. N. (2011). *Remarks on some nonparametric estimates of a density function*. In Selected Works of Murray Rosenblatt (pp. 95-100). New York, NY: Springer New York.

[18] Liang, Y. C., & Cuevas Juarez, J. R. (2016). *A novel metaheuristic for continuous optimization problems: Virus optimization algorithm. Engineering Optimization*, 48(1), 73-93.

[19] Du, Y., Wang, C., & Zhang, Y. (2022). *Viral coinfections. Viruses*, 14(12), 2645.

[20] Seabloom EW, Borer ET, Gross K, Kendig AE, Lacroix C, Mitchell CE, Mordecai EA, Power AG. *The community ecology of pathogens: coinfection, coexistence and community composition.* Ecol Lett. 2015 Apr;18(4):401-15. doi: 10.1111/ele.12418. Epub 2015 Mar 2. PMID: 25728488.

[21] Tamborindeguy, C., Hata, F. T., Molina, R. D. O., & Nunes, W. M. D. C. (2023). *A new perspective on the co-transmission of plant pathogens by hemipterans.* Microorganisms, 11(1), 156.

[22] Kadyrov, S., Haydarov, F., Mamayusupov, K., & Mustayev, K. (2025). *Endemic coexistence and competition of virus variants under partial cross-immunity.* Electronic Research Archive, 33(2).

[23] MacQueen, J. (1967, January). *Some methods for classification and analysis of multivariate observations*. In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics (Vol. 5, pp. 281-298). University of California press.

[24] Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). *Maximum likelihood from incomplete data via the EM algorithm*. Journal of the royal statistical society: series B (methodological), 39(1), 1-22.

[25] Asuncion, A., Newman, D. (2007, November). UCI machine learning repository.

[26] Sokolova, M., & Lapalme, G. (2009). *A systematic analysis of performance measures for classification tasks*. Information processing & management, 45(4), 427-437.

APÊNDICE A
ANÁLISE VISUAL DOS AGRUPAMENTOS COM VARIAÇÃO DE HIPERPARÂMETROS

Os gráficos abaixo exibem os agrupamentos formados, tais correspondem a uma combinação específica dos hiperparâmetros (n, α, n_{min}), indicados no título de cada figura. Essas visualizações permitem observar como diferentes escolhas afetam a formação dos grupos, sua dispersão e coesão.

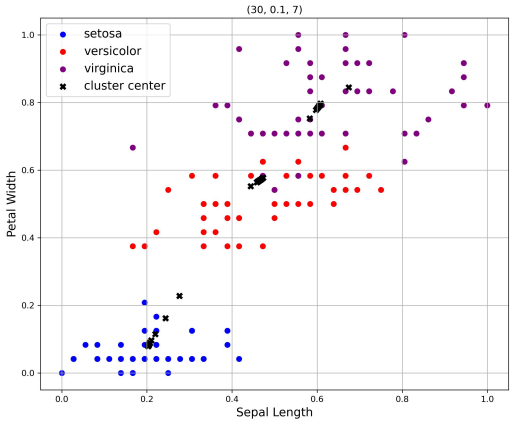

Figura 8. Os valores dos hiperparâmetros usados foram $n = 30$, $\alpha = 0,05$ e $n_{min} = 7$

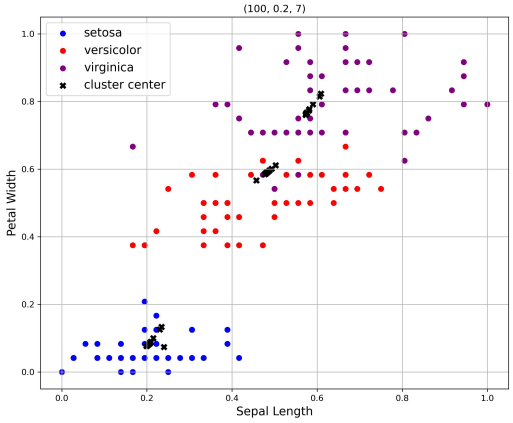

Figura 9. Os valores dos hiperparâmetros usados foram $n = 100$, $\alpha = 0,2$ e $n_{min} = 7$

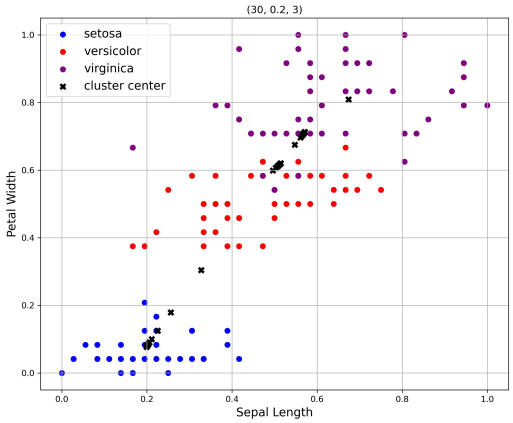

Figura 10. Os valores dos hiperparâmetros usados foram $n = 30$, $\alpha = 0,2$ e $n_{min} = 3$

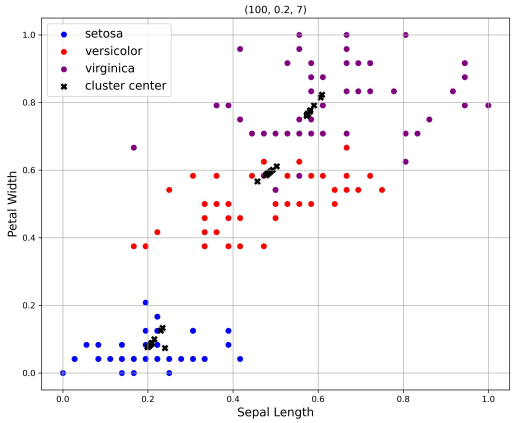

Figura 11. Os valores dos hiperparâmetros usados foram $n = 100$, $\alpha = 0,02$ e $n_{min} = 7$

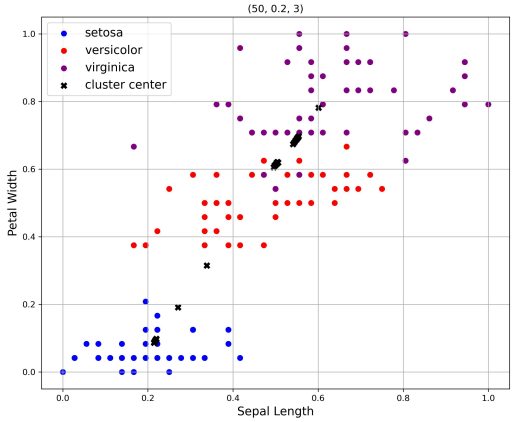

Figura 12. Os valores dos hiperparâmetros usados foram $n = 50$, $\alpha = 0,2$ e $n_{min} = 3$