

Estudo Paramétrico de um Algoritmo Imunológico em um Problema de Classificação

Pedro Henrique Rodrigues Corrêa¹
pedrowente@gmail.com

Orientador: Prof. Dr. Eurico Luiz Prospero Ruivo

¹Faculdade de Computação e Informática – Universidade Presbiteriana Mackenzie (UPM)
– São Paulo – SP – Brazil

Abstract. *This paper presents a detailed parametric study of an immunology-inspired algorithm applied to a classification problem. The algorithm, known as the Artificial Immune Network Algorithm (AINet), has its effectiveness examined through various parameter settings on an Iris flower dataset. We analyze the performance of the parameters given to AINet and their impact on its classification accuracy as well as considering acceptable execution times. The results demonstrate that certain parameters can significantly optimize the algorithm's accuracy without overly compromising the computational efficiency compared to others. This study provides valuable data for the application of artificial immunology algorithms in classification problems.*

Key-words– *artificial immune systems, bio-inspired algorithm, bio-inspired computing, artificial immune networks, parametric analysis.*

Resumo. *Este artigo apresenta um estudo paramétrico detalhado de um algoritmo inspirado em imunologia aplicado a um problema de classificação. O algoritmo, conhecido como Artificial Immune Network Algorithm (AINet), tem sua eficácia examinada através de diversas configurações paramétricas em um conjunto de dados de flor Iris. Analisamos o desempenho dos parâmetros dados ao AINet e seu impacto em sua acurácia de classificação assim como levando em consideração tempos de execução aceitáveis. Os resultados demonstram que certos parâmetros podem otimizar significativamente a precisão do algoritmo sem comprometer excessivamente a eficiência computacional do que outros. Este estudo fornece dados valiosos para a aplicação de algoritmos de imunologia artificial em problemas de classificação.*

Palavras-chave– *sistemas imunológicos artificiais, algoritmo bio-inspirado, computação natural, redes imunológicas artificiais, análise paramétrica.*

1. Introdução

A computação inspirada na natureza faz uso de fenômenos e processos naturais para desenvolver algoritmos que resolvam problemas em várias áreas do conhecimento [Rozenberg et al. 2012, de Castro 2007, Darwish 2018]. As fontes principais de inspiração envolvem insetos, a teoria neo-darwiniana da evolução, sistemas imunológicos e redes neurais. Apesar de ser uma área relativamente nova, há poucos trabalhos na literatura que envolvam o uso de redes imunológicas artificiais aplicadas a problemas de

classificação. Isso apresenta uma lacuna científica e uma oportunidade para pesquisa, que será explorada em partes aqui, com ênfase particular na questão paramétrica e suas implicações no funcionamento do algoritmo.

Este estudo terá como base os sistemas imunológicos artificiais [de Castro and Timmis 2002, Mikherskii and Mikherskii 2021], em particular o algoritmo de redes imunológicas artificiais aiNet [de Castro and Von Zuben 2001]. O objetivo é construir um classificador que seja composto por populações específicas a cada classe da base de dados, como já fundamentado na literatura.

O objetivo é utilizar o algoritmo de redes imunológicas artificiais (aiNet) para solucionar problemas de classificação e realizar uma análise acerca do comportamento do algoritmo baseado na exploração de seus hiperparâmetros, numa tentativa de descrever o impacto de cada um desses hiperparâmetros no comportamento do algoritmo. Os resultados deste subprojeto servirão como ponto de partida para um estudo posterior mais amplo envolvendo outras técnicas inspiradas na natureza.

Os objetivos específicos são:

1. Estudar os conceitos e metodologias associados à sistemas imunológicos artificiais e algoritmos bioinspirados;
2. Enfatizar o estudo sobre o algoritmo de redes imunológicas artificiais (aiNet) que será o foco do estudo;
3. Formalizar uma metodologia sobre o uso de redes imunológicas artificiais aplicadas a problemas de classificação, assim como também uma metodologia acerca das rotinas de testes de seus parâmetros;
4. Comparar os resultados obtidos com diferentes parâmetros e seu impacto na acurácia da classificação no algoritmo assim como a viabilidade em tempo computacional.
5. Consolidar resultados alcançados com a análise paramétrica.

2. Referencial Teórico

Este trabalho se concentra no Algoritmo de Redes Imunológicas Artificiais (aiNet), um paradigma da Computação Natural, campo de estudo que abrange a intersecção da computação e da natureza. Esse campo foca na criação de algoritmos inspirados na natureza, na modelagem de fenômenos naturais e na exploração de soluções bioinspiradas para computação [de Castro 2006, de Castro 2007, Rozenberg et al. 2012]. A Computação Natural tem se destacado no desenvolvimento de algoritmos de otimização e análise de dados [Yang 2020, Nanda and Panda 2014].

Os Sistemas Imunológicos Artificiais (SIA), como a aiNet, compõem uma das principais subáreas da Computação Natural, junto com inteligência de enxame, computação evolutiva e redes neurais artificiais [de Castro 2006, Abdel-Basset et al. 2018, Bansal et al. 2019, Kar 2016, Kennedy et al. 2001, Parpinelli and Lopes 2011]. O algoritmo aiNet, inspirado na interação entre anticorpos e antígenos no sistema imunológico humano, permite o reconhecimento de padrões de patógenos ou antígenos através de anticorpos interconectados com pesos variáveis [de Castro and Von Zuben 2001].

Desde sua introdução, várias variantes de algoritmos imunológicos surgiram, explorando diferentes interpretações do sistema e possibilidades de otimização

[Bezerra et al. 2005]. Apesar da simplicidade conceitual e da ampla aplicação prática, esses algoritmos enfrentam desafios significativos, como a dificuldade em gerar uma população de anticorpos que efetivamente classifiquem ou cubram completamente o espaço de antígenos a serem detectados [Graaff and Engelbrecht 2006], e a parametrização, incluindo a escolha da representação dos dados de entrada e a seleção da medida de distância apropriada.

A aiNet se destaca por sua robustez diante de dados incompletos ou outliers [Guo et al. 2021], sua habilidade de generalização [Zhang et al. 2017], a preservação da diversidade da população [Brownlee and Woodward 2005], e sua capacidade de aprender e memorizar padrões [Dasgupta et al. 2003]. No entanto, seu desempenho depende muito da parametrização apropriada, o principal desafio a ser explorado neste trabalho. Assim, este projeto visa aprofundar a compreensão e o uso do algoritmo aiNet, especialmente no que se refere ao impacto da parametrização em seu funcionamento [Timmis et al. 2004].

2.1. Algoritmo aiNet

O algoritmo aiNet implementado foi baseado no pseudocódigo que pode ser encontrado na literatura [França et al. 2010]. A Figura 1 mostra este pseudocódigo.

Pseudo-code 1. Main Structure of the aiNet algorithm
<p>Input:</p> <ul style="list-style-type: none"> – N: the initial size of the population of antibodies; – n: the number of cells that should be cloned; – m: the percentage of cells to be moved to the memory set; – σ_1: antibody-antibody suppression threshold; – σ_2: antibody-antigen suppression threshold; – d: the percentage of randomly generated cells to be included in the population. <p>Randomly create the initial population of N antibodies, each one with dimension L (dimension of the data patterns);</p> <p>While stop criteria not met do</p> <p style="padding-left: 20px;">1 – For each antigen do</p> <p style="padding-left: 40px;">1.1 – Determine its affinity to all network cells;</p> <p style="padding-left: 40px;">1.2 – Select the n highest affinity network cells;</p> <p style="padding-left: 40px;">1.3 – Generate N_c clones from these n cells. The higher the affinity, the larger N_c;</p> <p style="padding-left: 40px;">1.4 – Apply hypermutation to the generated clones, with variability inversely proportional to the progenitor's fitness;</p> <p style="padding-left: 40px;">1.5 – Determine the affinity among the antigen and all clones;</p> <p style="padding-left: 40px;">1.6 – Keep only $m\%$ of the highest affinity mutated clones into the clone population;</p> <p style="padding-left: 40px;">1.7 – Eliminate all clones but one whose affinity with the antigen is inferior to a predefined threshold σ_2 (apoptosis);</p> <p style="padding-left: 40px;">1.8 – Determine the affinity among all the mutated clones and eliminate those whose affinity with each other is above a pre-defined threshold σ_1 (suppression);</p> <p style="padding-left: 40px;">1.9 – Insert the remaining mutated clones into the population;</p> <p style="padding-left: 20px;">2 – Determine the similarity among all the antibodies and eliminate those with similarity above a threshold σ_1 (suppression);</p> <p style="padding-left: 20px;">3 – Introduce a percentage $d\%$ of new randomly generated cells (random insertion);</p> <p>End While</p> <p>Eliminate the network cells, if any, which recognize no antigen.</p>

Figura 1. Pseudocódigo do algoritmo aiNet. Adaptado de [França et al. 2010].

2.1.1. Medida de Distância

A medida de distância é um componente essencial em muitos algoritmos de aprendizado de máquina, especialmente naqueles que envolvem agrupamento, classificação e técnicas

baseadas em instâncias. No contexto do algoritmo aiNet, a medida de distância é utilizada para calcular a afinidade entre células e antígenos, bem como para identificar células semelhantes durante o processo de supressão.

A escolha da medida de distância pode ter um impacto significativo no desempenho do algoritmo. Uma medida de distância deve refletir com precisão as semelhanças e diferenças entre os indivíduos no espaço de busca, levando em consideração as características particulares do problema em questão.

Neste trabalho, utilizamos a distância euclidiana, uma das medidas de distância mais comuns, como a medida de distância no algoritmo aiNet. A distância euclidiana entre dois indivíduos x e y no espaço n -dimensional é definida como:

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

Onde x_i e y_i são, respectivamente, a i -ésima característica (ou feature) de x e y .

É importante observar que, embora a distância euclidiana seja apropriada para muitos problemas, outras medidas de distância, como a distância de Manhattan, a distância de Minkowski ou a distância de Mahalanobis, podem ser mais adequadas para problemas específicos [Aggarwal and Reddy 2013]. Além disso, é sempre importante garantir que as características sejam adequadamente normalizadas ou padronizadas antes de calcular as distâncias, para evitar que características com escalas maiores dominem o cálculo da distância.

2.1.2. Normalização

A normalização é um passo crucial em muitos algoritmos de aprendizado de máquina, pois permite que todos os recursos sejam considerados de maneira equânime durante a análise. Sem normalização, os recursos com maiores valores absolutos teriam um peso maior, mesmo que não necessariamente contribuíssem mais para a informação útil no conjunto de dados [Zheng and Casari 2018].

Neste estudo, adotamos o método de normalização Min-Max. Este método redimensiona todos os valores para uma escala de 0 a 1. Faz isso subtraindo o valor mínimo e dividindo pelo intervalo dos valores (máximo - mínimo). Este procedimento é ilustrado pela seguinte fórmula:

$$x_{\text{norm}} = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}}$$

Onde x é o valor original, x_{min} é o valor mínimo da característica e x_{max} é o valor máximo da característica. A normalização Min-Max é uma escolha comum para a normalização de dados porque é simples, eficaz e tem o benefício adicional de manter exatamente os mesmos valores para todos os dados que já estão entre 0 e 1.

2.1.3. Hipermutação

A hipermutação, como o nome sugere, é um processo que causa uma alta taxa de mutação. No contexto do algoritmo aiNet, este processo é aplicado às células clonadas [de Castro and Von Zuben 2002]. A ideia por trás disso é introduzir variabilidade na população de células, permitindo que o algoritmo explore um espaço de solução mais amplo. Além disso, a taxa de mutação é inversamente proporcional à afinidade da célula progenitora. Isso significa que as células com alta afinidade têm uma taxa de mutação mais baixa, enquanto as células com baixa afinidade têm uma taxa de mutação mais alta.

A fórmula utilizada para a mutação no contexto de nosso algoritmo é dada por:

$$g_{\text{mut}} = \begin{cases} g + \text{clip}(U(-1, 1) \cdot (1 - a) \cdot 0.1, -0.3, 0.3), & \text{if } U(0, 1) < m \\ g, & \text{otherwise} \end{cases}$$

Onde:

1. g_{mut} é o valor do gene após a mutação;
2. g é o valor original do gene;
3. $U(-1, 1)$ é uma variável aleatória uniformemente distribuída entre -1 e 1;
4. $U(0, 1)$ é uma variável aleatória uniformemente distribuída entre 0 e 1;
5. a é a afinidade do anticorpo com o antígeno;
6. m é a taxa de mutação;
7. "clip" é uma função que limita o valor entre -0.3 e 0.3.

2.1.4. Supressão

A supressão é um processo chave do algoritmo aiNet que visa controlar a diversidade da população de células [de Castro and Von Zuben 2002]. O processo de supressão consiste em comparar a similaridade de cada par de células na população e remover uma delas caso a similaridade exceda um limiar pré-definido, σ_1 .

Algorithm 1 Processo de Supressão

```
1: for cada célula  $i$  na população do
2:    $is\_similar = \text{Falso}$ 
3:   for cada célula  $j$  na população onde  $j > i$  do
4:     Calcular a similaridade entre as células  $i$  e  $j$ 
5:     if similaridade  $> \sigma_1$  then
6:        $is\_similar = \text{Verdadeiro}$ 
7:       break
8:     end if
9:   end for
10:  if not  $is\_similar$  then
11:    Adicionar a célula  $i$  à nova população
12:  end if
13: end for
```

2.1.5. Apoptose

A apoptose refere-se ao processo de remoção de clones que não se mostraram úteis para o sistema [de Castro and Von Zuben 2002]. No algoritmo aiNet, a apoptose é implementada filtrando os clones cuja afinidade com o antígeno é inferior a um limite definido, σ_2 . Em termos mais gerais, a apoptose ajuda a garantir que o sistema mantenha um foco em soluções de alta qualidade, removendo clones que não contribuem efetivamente para a solução do problema.

Algorithm 2 Processo de Apoptose

```
1: for cada clone  $x$  em mutated_clone_population do
2:   if  $x[1] > \sigma_2$  then
3:     adicione  $x$  à filtered_clone_population
4:   end if
5: end for
```

2.2. Parâmetros

2.2.1. Tamanho da População

O tamanho da população inicial é crucial para determinar a diversidade das soluções que o algoritmo pode explorar. Uma população maior proporciona uma cobertura mais ampla do espaço de busca, mas requer maior poder de processamento, ainda que não implique na curva de crescimento da população, assim tendo um impacto maior no processamento das primeiras iterações.

2.2.2. Tamanho da Seleção (n)

O tamanho da seleção é um hiperparâmetro que determina o número de células de alta afinidade que serão selecionadas para clonagem e mutação em cada iteração. Essas células selecionadas são aquelas que apresentam a maior afinidade em relação ao antígeno.

Um tamanho de seleção maior pode acelerar a convergência do algoritmo para uma solução, pois permite a geração de um número maior de clones de alta afinidade. No entanto, se o tamanho da seleção for muito grande, pode limitar a exploração de novas soluções, pois um grande número de clones de alta afinidade pode dominar a população, reduzindo a diversidade.

2.2.3. Memória da População (m%)

A memória da população é um hiperparâmetro que determina a porcentagem de clones que serão mantidos após a etapa de mutação com base em sua afinidade. Os clones são classificados por sua afinidade, e somente a porcentagem superior, especificada pelo valor de m%, é mantida na população. Essa porcentagem de clones mantidos constitui a "memória" do sistema.

Este mecanismo permite que o sistema mantenha uma diversidade de soluções de alta qualidade, aprendendo com as soluções anteriores e permitindo que sejam exploradas novas áreas do espaço de busca. No entanto, uma porcentagem muito alta pode limitar a exploração de novas soluções, pois manterá muitos clones das soluções anteriores, limitando a geração de novos clones mutantes.

Além disso, após essa etapa, todos os clones com afinidade com o antígeno inferior a um limite pré-definido (σ_2) são eliminados, exceto um para cada grupo de clones similares. Esse processo é conhecido como apoptose e serve para manter a diversidade e evitar a superpopulação de clones similares na população, mantendo apenas os de maior afinidade.

2.2.4. Taxa de Clonagem

A taxa de clonagem define o número de clones gerados de cada célula selecionada e está diretamente relacionada com a profundidade da busca no espaço de soluções. Uma taxa mais elevada aumenta a densidade de busca em torno das soluções promissoras.

Para cada célula selecionada, um conjunto de clones é gerado proporcional à sua afinidade, gerando assim um número N_c de clones.

Logo, a taxa de clonagem tem um papel crucial na expansão e exploração do espaço de busca, influenciando diretamente a eficiência e a eficácia do algoritmo. No entanto, uma taxa de clonagem excessivamente alta pode implicar em maior custo computacional.

2.2.5. Taxa de Mutação

A taxa de mutação determina a probabilidade de uma célula sofrer mutação após a clonagem. Uma taxa de mutação maior promove uma maior diversidade, mas também pode levar à convergência para soluções subótimas se for demasiado alta.

2.2.6. Critério de Parada

O critério de parada define quando o algoritmo deve ser interrompido. Vários critérios foram explorados, incluindo um número fixo de iterações e um tempo máximo de execução.

2.2.7. Inserção Aleatória (d%)

A taxa de inserção aleatória, d%, define a proporção de novas células geradas aleatoriamente que são inseridas na população em cada iteração. Uma taxa mais alta aumenta a diversidade da população, mas pode diluir as soluções de alta qualidade se for demasiado alta.

2.2.8. Sigma 1 (σ_1)

O hiperparâmetro σ_1 é uma medida de similaridade e é usado como um limiar no algoritmo aiNet durante o processo de supressão. Na supressão, anticorpos (ou células) muito similares são removidos da população para manter a diversidade. Esse processo acontece em duas etapas no algoritmo: uma após a mutação e seleção de clones, e outra após a atualização da população de células da rede.

2.2.9. Sigma 2 (σ_2)

O hiperparâmetro σ_2 é um limiar de afinidade usado no algoritmo aiNet durante o processo de apoptose, que é a eliminação de clones com afinidade inferior a σ_2 com o antígeno.

3. Metodologia

3.1. Base de Dados

A base de dados escolhida para a realização dos estudos paramétricos foi a base de dados Iris, que é bem conhecida na literatura de aprendizado de máquina. Esta base de dados foi introduzida por R.A. Fisher em 1936 e é frequentemente usada como um caso de teste para algoritmos de classificação [Fisher 1936]. A base contém três classes de flores de Iris (Iris setosa, Iris virginica e Iris versicolor), sendo que duas delas (Iris virginica e Iris versicolor) têm uma sobreposição considerável em suas características, enquanto a Iris setosa está mais distante das outras duas classes.

Adicionalmente, foi adotada uma estratégia para expandir a compreensão do funcionamento do algoritmo e fornecer uma visualização bidimensional de como os parâmetros afetam o comportamento do algoritmo. Para isso, implementamos um método de geração de dados bidimensionais aleatórios, dispersos em três clusters distintos.

A Figura 2 apresenta a distribuição dos dados aleatórios gerados, que foram intencionalmente agrupados em três clusters para simular cenários de classificação menos complexos.

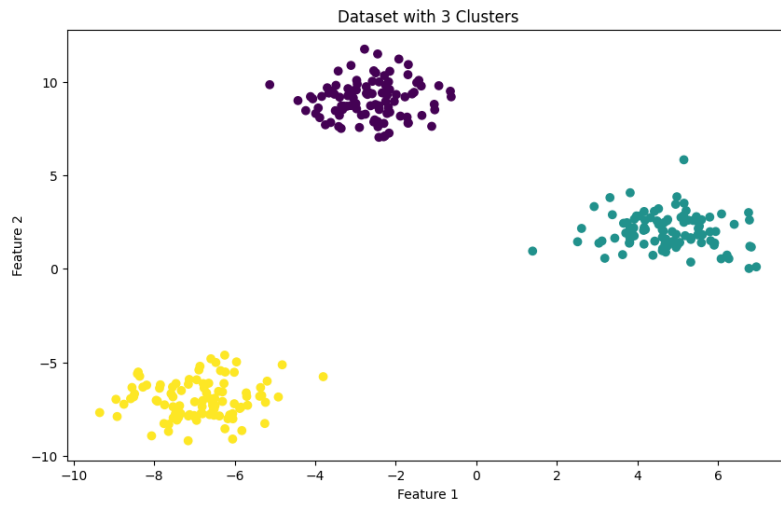


Figura 2. Distribuição dos dados aleatórios bidimensionais, dispersos em três clusters.

Já a Figura 3 mostra a formação das populações de células geradas pelo algoritmo a partir dos dados de entrada. Esta visualização é essencial para entender como o algoritmo responde a diferentes cenários e parâmetros.

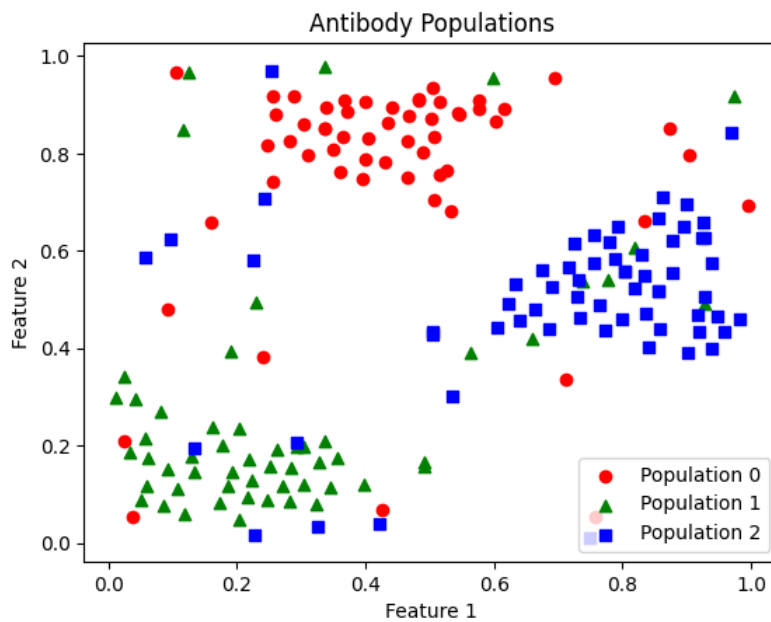


Figura 3. Formação das populações de células a partir dos dados de entrada.

Através dessas duas representações, é possível entender melhor como os parâmetros do algoritmo afetam sua capacidade de classificação e de lidar com diferentes estruturas de dados.

3.2. Análise Paramétrica

Durante a fase de desenvolvimento do projeto, observou-se que certos hiperparâmetros tinham um impacto significativo no tempo de processamento, muitas vezes tornando-o

inviável. Com base nisso, foram realizadas várias iterações do algoritmo com o objetivo de entender melhor o papel de cada hiperparâmetro no desempenho do algoritmo.

Após uma análise aprofundada, dois hiperparâmetros, nomeadamente σ_1 e σ_2 (ver Figura 1), foram identificados como sendo de particular interesse para este estudo. Foram selecionados por seu potencial impacto tanto na eficácia da classificação quanto no tempo de execução do algoritmo.

Para os restantes hiperparâmetros, foram definidos limites superiores e inferiores que permitiam um tempo de execução razoável, facilitando assim o estudo de σ_1 e σ_2 .

Os valores selecionados para cada hiperparâmetro podem ser vistos na Tabela 1.

Variável	Mínimo	Máximo
Tamanho da População	80	500
Tamanho da Seleção	15	50
Conjunto de Memória	5%	20%
Taxa de Clonagem	5%	15%
Taxa de Mutação	10%	30%
Inserção Aleatória ($d\%$)	5%	15%
Condição de Parada	10	100
Sigma 1 (σ_1)	0.8	0.92
Sigma 2 (σ_2)	0.8	0.7

Tabela 1. Alcance dos hiperparâmetros utilizados em múltiplas execuções

Após extensas execuções exploratórias dentro dos limites detalhados na Figura 4, o conjunto de parâmetros que apresentou o melhor desempenho foi selecionado. Para uma avaliação robusta e confiável deste conjunto ótimo de parâmetros, empregamos uma validação cruzada de 10-fold para estabelecer o desempenho genuíno do algoritmo.

3.3. Método K-Nearest Neighbors (K-NN)

Na Figura 3, observa-se indivíduos distantes dos seus respectivos clusters. Esses são os indivíduos gerados nas últimas iterações do algoritmo aiNet, sendo menos especializados por passarem por menos etapas do processo.

Com base em [Qing et al. 2010], a classificação foi aprimorada combinando o algoritmo aiNet com o método K-Nearest Neighbors (K-NN), aumentando a robustez da classificação. Nesse método, a classificação de um antígeno considera o consenso entre seus K vizinhos mais próximos, ao invés de sua proximidade com um único indivíduo. Nas execuções realizadas, K foi fixado em 3.

4. Resultados Obtidos

Este estudo destaca o papel crítico dos hiperparâmetros no algoritmo aiNet, abordando especificamente a influência de σ_1 e σ_2 na eficácia computacional e no desempenho do algoritmo.

A base para as variações dos parâmetros é detalhada na Tabela da Figura ??, com as modificações sublinhadas nas seções subsequentes. Destacamos que cada alteração é feita de maneira isolada, mantendo todos os outros parâmetros constantes para fornecer uma análise robusta da influência de cada hiperparâmetro na performance do algoritmo.

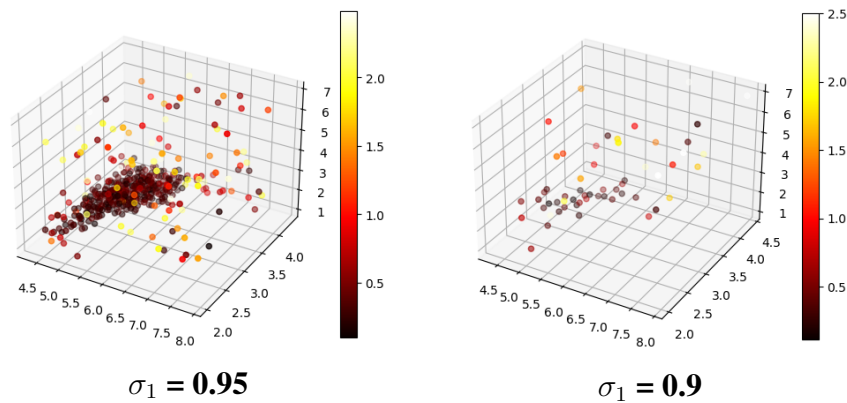


Figura 4. Influência do valor de σ_1 na formação dos clusters (dataset Iris)

4.1. Influência do Parâmetro Sigma 1 (σ_1)

A Figura 4 ilustra a influência do hiperparâmetro σ_1 , que determina o limiar de similaridade, na dinâmica populacional e na formação de clusters. Observa-se que a redução de σ_1 promove uma população mais dispersa e de menor tamanho. Isso ocorre porque com um limiar de similaridade menor, mais células são consideradas similares e, portanto, eliminadas do processo.

Contrariamente, ao aumentar o valor de σ_1 , a tolerância à similaridade entre células aumenta, possibilitando a formação de clusters mais densos e promovendo um crescimento populacional mais acelerado. Entretanto, esse crescimento acelerado também pode levar a um aumento exponencial da complexidade computacional, já que mais células precisam ser processadas.

Essas observações destacam a importância de um equilíbrio adequado para o hiperparâmetro σ_1 , de forma a evitar a formação de populações excessivamente grandes e, ao mesmo tempo, garantir a diversidade das soluções.

4.2. Influência do Parâmetro Sigma 2 (σ_2)

A Figura 5 destaca o efeito do hiperparâmetro σ_2 , que atua como um limiar de afinidade, sobre a formação de clusters e a dinâmica da população. Nota-se que, com um alto valor de σ_2 , o algoritmo mantém apenas clones com maior afinidade para o antígeno, originando clusters mais concentrados em torno desses antígenos e controlando o crescimento populacional.

Entretanto, ao reduzirmos o valor de σ_2 , o algoritmo se torna mais permissivo para clones com afinidades mais baixas, levando a clusters mais dispersos. Isto resulta em uma maior cobertura do espaço de solução e impulsiona o crescimento populacional.

Esses resultados enfatizam a importância da seleção adequada de σ_2 . O hiperparâmetro deve ser ajustado para balancear a necessidade de manter a diversidade do sistema e de explorar de maneira eficiente o espaço de solução, sem, contudo, promover um crescimento populacional excessivo que poderia aumentar a complexidade computacional.

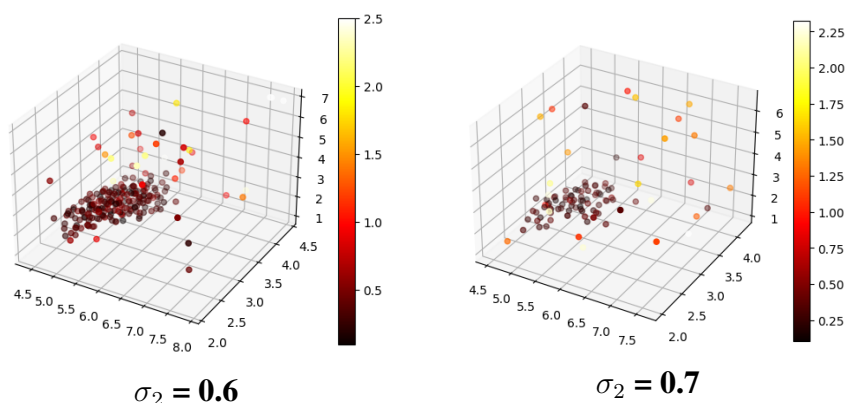


Figura 5. Influência do valor de σ_2 na formação dos clusters (dataset Iris)

4.2.1. Parâmetros (σ)

Em suma, σ_1 e σ_2 são parâmetros cruciais no algoritmo aiNet que controlam a diversidade de anticorpos na população e a afinidade dos clones com os antígenos, respectivamente.

Eles devem ser cuidadosamente ajustados para garantir que o algoritmo seja capaz de explorar eficientemente o espaço de busca, mantendo ao mesmo tempo a capacidade de convergir para soluções de alta qualidade, pois como observado certas combinações de σ_1 e σ_2 podem tanto resultar em explosões populacionais, com uma taxa de crescimento inviável computacionalmente como o inverso também pode ocorrer e o algoritmo não ser capaz de manter um crescimento populacional com a devida diversidade de indivíduos resultando em uma cobertura insuficiente no espaço.

4.3. Melhoria do Desempenho com KNN

A implementação do método KNN no algoritmo aiNet, conforme descrito na seção 3.3, trouxe uma melhora significativa na precisão da classificação, conforme evidenciado pelos resultados apresentados na Tabela 2.

K	Execução 1 (%)	Execução 2 (%)	Execução 3 (%)	Execução 4 (%)	Execução 5 (%)
1	91	81	94	85	93
3	100	100	98	96	100

Tabela 2. Precisão obtida com a aplicação do método KNN no algoritmo aiNet

Quando $K=1$, o algoritmo KNN se comporta de maneira clássica, considerando o vizinho mais próximo para a classificação. Este comportamento, apesar de simples, provou ser eficaz, como evidenciado pelos resultados obtidos.

Por outro lado, com $K=3$, o algoritmo considera os três vizinhos mais próximos para a classificação. Esta abordagem adiciona um nível adicional de robustez ao algoritmo, resultando em uma precisão de classificação ainda maior.

Estes resultados demonstram que a inclusão do método KNN trouxe uma robustez adicional ao algoritmo aiNet, melhorando a precisão das classificações realizadas.

Estes resultados são consistentes com as previsões da literatura de otimização da

AINet, conforme discutido por Qing [Qing et al. 2010], que sugere que a adição de técnicas de aprendizado de máquina, como o KNN, pode fornecer melhorias significativas na eficiência e precisão do algoritmo aiNet.

4.4. Validação 10-folds

Os resultados indicaram um alto grau de eficácia na classificação das classes 0, 1 e 2, com uma precisão média de 95%, conforme apresentado na Tabela 3. Este relatório de classificação revela que o algoritmo aiNet foi capaz de distinguir com sucesso entre as diferentes classes, mostrando um equilíbrio entre precisão, recall e F1-score.

Métrica	Classe 0	Classe 1	Classe 2	Média/Total
Precisão	1,00	0,94	0,92	0,95
Recall	1,00	0,92	0,94	0,95
F1-score	1,00	0,93	0,93	0,95
Suporte	50	50	50	150

Tabela 3. Relatório de Classificação

A matriz de confusão, apresentada na Figura 6, fornece uma visualização mais detalhada dos resultados de classificação. Esta matriz é uma ferramenta essencial para analisar a precisão das previsões para cada classe, permitindo entender de maneira mais clara o desempenho do algoritmo aiNet.

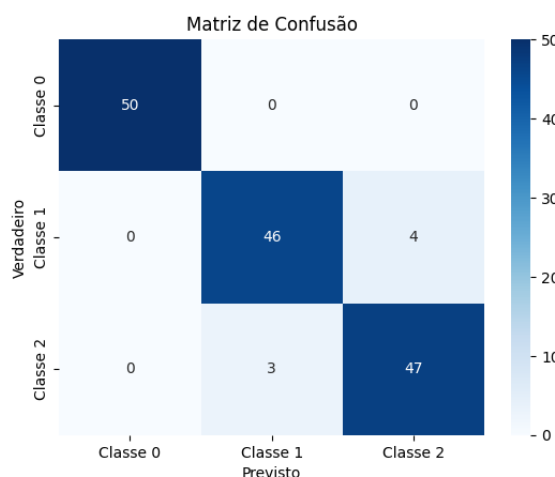


Figura 6. Matriz de Confusão

Observando a matriz de confusão, notamos que os erros de classificação ocorreram entre as classes Iris versicolor (Classe 1) e Iris virginica (Classe 2), que são conhecidas por terem sobreposição na base de dados Iris. Este fenômeno de sobreposição significa que essas duas classes possuem características similares, assim causando que anticorpos de classes diferentes dividam um espaço mais próximo.

Este resultado é esperado e consistente com o que é conhecido sobre o conjunto de dados Iris. As classes Iris versicolor e Iris virginica, em particular, possuem características semelhantes que dificultam a sua distinção. Dessa forma, apesar de alguns erros de classificação entre essas duas classes, o algoritmo aiNet demonstrou um desempenho

robusto, sendo capaz de fazer distinções complexas entre classes que possuem sobreposição.

A validação cruzada em 10-folds, uma técnica amplamente aceita para avaliar a eficácia de modelos de aprendizado de máquina, foi utilizada para fornecer uma estimativa mais confiável do desempenho do modelo em dados não vistos. O alto desempenho alcançado neste teste sugere que o algoritmo aiNet, com os hiperparâmetros selecionados conforme demonstrado na Tabela 4, é capaz de lidar efetivamente com a tarefa de classificação.

Variável	Valor
Tamanho da População	500
Tamanho da Seleção	30
Conjunto de Memória	10%
Taxa de Clonagem	5%
Taxa de Mutação	30%
Inserção Aleatória (d)	5%
Condição de Parada	10
Sigma 1 (σ_1)	0,92
Sigma 2 (σ_2)	0,7

Tabela 4. Valores dos hiperparâmetros utilizados na execução 10-folds

Entretanto, é importante ressaltar que este estudo foi realizado com a base de dados Iris, que é uma base de dados pequena e relativamente simples. Portanto, embora os resultados sejam promissores, experimentos adicionais com bases de dados maiores e mais complexas são necessários para avaliar mais completamente a eficácia e a eficiência do algoritmo aiNet.

O código para este projeto pode ser encontrado online no repositório GitHub do autor [Corrêa 2023] <https://github.com/pedrordrigs/Tr-aiNET>.

5. References

Referências

- Abdel-Basset, M., Abdel-Fatah, L., and Sangaiyah, A. K. (2018). *Metaheuristic Algorithms: A Comprehensive Review*. Academic Press.
- Aggarwal, C. C. and Reddy, C. K. (2013). *Data Clustering: Algorithms and Applications*. CRC Press.
- Bansal, J. C., Singh, P. K., and Pal, N. R. (2019). *Evolutionary and Swarm Intelligence Algorithms*. Springer.
- Bezerra, G. B., Barra, T. V., de Castro, L. N., and Von Zuben, F. J. (2005). Adaptive radius immune algorithm for data clustering. In Jacob, C., Pilat, M. L., Bentley, P. J., and Timmis, J. I., editors, *Artificial Immune Systems*, pages 290–303, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Brownlee, J. and Woodward, C. (2005). Tuning of clonal selection algorithm parameters using taguchi methods. In *Congress on Evolutionary Computation*, volume 1, pages 102–109.

- Corrêa, P. H. R. (2023). Tr-ainet. Acessado: 16/06/2023.
- Darwish, A. (2018). Bio-inspired computing: Algorithms review, deep analysis, and the scope of applications. *Future Computing and Informatics Journal*, 3(2):231–246.
- Dasgupta, D., Yu, S., and Nino, F. (2003). Recent advances in artificial immune systems: Models and applications. *Applied Soft Computing*, 3(2):113–132.
- de Castro, L. N. (2006). *Fundamentals of Natural Computing: Basic Concepts, Algorithms, and Applications*. Chapman Hall/CRC.
- de Castro, L. N. (2007). Fundamentals of natural computing: an overview. *Physics of Life Reviews*, 4(1):1–36.
- de Castro, L. N. and Timmis, J. I. (2002). *Artificial Immune Systems: A New Computational Intelligence Approach*. Springer.
- de Castro, L. N. and Von Zuben, F. J. (2001). ainet: An artificial immune network for data analysis.
- de Castro, L. N. and Von Zuben, F. J. (2002). Learning and optimization using the clonal selection principle. *IEEE Transactions on Evolutionary Computation*, 6(3):239–251.
- Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188.
- França, F. O. d., Coelho, G. P., Castro, P. A. D., and Von Zuben, F. J. (2010). Conceptual and practical aspects of the ainet family of algorithms. *International Journal of Natural Computing Research*, 1(1):6.
- Graaff, A. and Engelbrecht, A. (2006). Optimized coverage of non-self with evolved lymphocytes in an artificial immune system. *International Journal of Computational Intelligence Research*, 2(2):127–150.
- Guo, P., Xu, X., Liu, Y., Hu, X., and Wang, T. (2021). Immunological intelligent systems: Based on the immune system. *IEEE Transactions on Cybernetics*, 51(5):2486–2501.
- Kar, A. K. (2016). Bio inspired computing – a review of algorithms and scope of applications. *Expert Systems with Applications*, 59:20–32.
- Kennedy, J., Eberhart, R. C., and Shi, Y. (2001). *Swarm Intelligence*. Morgan Kauffman Publishers.
- Mikherskii, R. M. and Mikherskii, M. R. (2021). Analysis of the use of artificial immune systems. In *IOP Conference Series: Materials Science and Engineering*, volume 1069, page 012025.
- Nanda, S. J. and Panda, G. (2014). A survey on nature inspired metaheuristic algorithms for partitional clustering. *Swarm and Evolutionary Computation*, 16:1–18.
- Parpinelli, R. S. and Lopes, H. S. (2011). New inspirations in swarm intelligence: a survey. *International Journal of Bio-Inspired Computation*, 3(1):1–16.
- Qing, J., Liang, X., Bie, R., and Gao, X. (2010). A new clustering algorithm based on artificial immune network and k-means method. In *2010 Sixth International Conference on Natural Computation*, volume 6, pages 2826–2830. IEEE.

- Rozenberg, G., Bäck, T., and Kok, J. N. (2012). *Handbook of Natural Computing*. Springer.
- Timmis, J., Neal, M., and Hunt, J. (2004). An overview of artificial immune systems and their applications. *Artificial Immune Systems and Their Applications*.
- Yang, X.-S. (2020). Nature-inspired optimization algorithms: Challenges and open problems. *Journal of Computational Science*, 46:101104.
- Zhang, Y., Gong, D., Zhang, Y., and Liu, X. (2017). Artificial immune network-based multi-objective optimization algorithm. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(7):1292–1305.
- Zheng, A. and Casari, A. (2018). *Feature Engineering for Machine Learning*. O'Reilly Media, Inc.