

UNIVERSIDADE FEDERAL DE JUIZ DE FORA  
INSTITUTO DE CIÊNCIAS EXATAS  
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

# **Redes Imunológicas Artificiais para Classificação: SAINET com raio adaptativo**

**Ana Livia Soares Silva de Almeida**

JUIZ DE FORA  
NOVEMBRO, 2017

# Redes Imunológicas Artificiais para Classificação: SAINET com raio adaptativo

ANA LÍVIA SOARES SILVA DE ALMEIDA

Universidade Federal de Juiz de Fora

Instituto de Ciências Exatas

Departamento de Ciência da Computação

Bacharelado em Ciência da Computação

Orientador: Luciana Conceição Dias Campos

JUIZ DE FORA  
NOVEMBRO, 2017

# REDES IMUNOLÓGICAS ARTIFICIAIS PARA CLASSIFICAÇÃO: SAINET COM RAIOS ADAPTATIVOS

Ana Livia Soares Silva de Almeida

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTEGRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE BACHAREL EM CIÊNCIA DA COMPUTAÇÃO.

Aprovada por:

Luciana Conceição Dias Campos  
Doutora

Heder Soares Bernardino  
Doutor

Leonardo da Fonseca Goliatt  
Doutor

JUIZ DE FORA  
23 DE NOVEMBRO, 2017

*À minha mãe, pelo amor, incentivo e confiança desde o princípio.*

*Ao Pluto, pela companhia e carinho incondicionais enquanto construí este trabalho.*

## Resumo

Algumas áreas de estudo na Ciência da Computação têm inspiração nas mais diversas manifestações da natureza. Uma área forte de estudo são as heurísticas inspiradas em teorias evolutivas, dentre as quais pode-se citar os Algoritmos Genéticos (AG). Por outro lado, existem ferramentas da Inteligência Artificial inspiradas em sistemas presentes em organismos mais complexos, como as Redes Neurais Artificiais e os Sistemas Imunológicos Artificiais. Estes últimos, assim como os AGs, usam a estratégia de evoluir uma população inicial até alcançar o objetivo pretendido. Neste conjunto de sistemas, chama-se atenção para aqueles desenvolvidos a partir de características do Sistema Imunológico natural a fim de resolver problemas da área de Mineração de Dados. Neste trabalho apresenta-se um estudo breve sobre Sistemas Imunológicos Artificiais para classificação e agrupamento de dados, avalia-se e compara-se os resultados obtidos a partir de uma modificação proposta em uma Rede Imunológica Artificial para classificação de dados, a *Supervised Artificial Immune Network* (SAINET), com os resultados da literatura.

**Palavras-chave:** Redes Imunológicas Artificiais, classificação, raio adaptativo, SAINET, ARIA.

## Abstract

Some research fields in Computer Science found inspiration in several events in nature. A very strong field is the study of heuristics inspired on evolutive theory, from those we can cite Genetic Algorithms. On the other hand, there are tools in Artificial Intelligence that are inspired on systems found in more complex organisms, such as Artificial Neural Networks and Artificial Immune Systems. Those last, like the GAs, use the strategy of evolving a population until achieve a goal. From this set of systems, we would like to call attention to those developed from a natural immune system, using its interesting characteristics to solve Data Mining problems. The following work presents a brief case study about Artificial Immune Systems for data classification and clustering, and evaluates and compare the results due to a modification in an Artificial Immune Network for data classification, SAINET, with results present in literature.

**Keywords:** Artificial Immune Network, classification, adaptative radius, SAINET, ARIA.

## Agradecimentos

Agradeço a todos que me apoiaram e incentivaram de alguma forma até aqui, em especial à minha mãe, Rosely, por todo amor e dedicação, por sempre acreditar em mim e jamais desistir.

À professora Luciana, por aceitar me orientar nesta jornada, por toda a paciência e atenção dedicadas à mim nesses dois anos.

Aos professores do Departamento de Ciência da Computação pelos seus ensinamentos e aos funcionários do curso, que durante esses anos, contribuíram de algum modo para o nosso crescimento pessoal e profissional.

*"Maktub, participio passado do verbo Kitab. É a expressão característica do fatalismo muçulmano. Maktub significa: "estava escrito"; ou melhor, "tinha que acontecer".*

*Essa expressiva palavra dita nos momentos de dor ou angustia, não é um brado de revolta contra o destino, mas sim, a reafirmação do espírito plenamente resignado diante dos desígnios da vida."*

*Oriente*

# Conteúdo

<b>Lista de Figuras</b>	<b>7</b>
<b>Lista de Tabelas</b>	<b>8</b>
<b>Lista de Abreviações</b>	<b>9</b>
<b>1 Introdução</b>	<b>10</b>
<b>2 Fundamentação Teórica</b>	<b>13</b>
2.1 Inspiração Natural . . . . .	13
2.1.1 Sistema Imunológico Natural . . . . .	13
2.1.2 Sistema Imunológico Artificial . . . . .	14
2.1.3 Redes Imunológicas . . . . .	14
<b>3 SAINET com raio adaptativo</b>	<b>27</b>
3.1 Fase 1: Construção de conjunto de memória (rede de anticorpos . . . . .	27
3.1.1 Inicialização das variáveis . . . . .	27
3.1.2 Maturação de afinidade . . . . .	27
3.1.3 Supressão de anticorpos . . . . .	28
3.1.4 Expansão Clonal . . . . .	28
3.1.5 Supressão de Rede . . . . .	29
3.1.6 Atualização do raio $E$ da taxa de mutação $\mu$ . . . . .	30
3.2 Fase 2: Expansão Clonal . . . . .	31
3.2.1 Atualização dos pesos . . . . .	31
<b>4 Estudo de caso</b>	<b>33</b>
4.1 Bases de da Dados . . . . .	33
4.2 Desempenho do classificador proposto . . . . .	34
4.3 Comparação dos resultados com a SAINET original . . . . .	36
4.4 Adaptação das redes clusterizadoras para construção da população de anticorpos . . . . .	37
<b>5 Conclusão e trabalhos futuros</b>	<b>40</b>
<b>Bibliografia</b>	<b>42</b>

## Lista de Figuras

2.1	Identificadores moleculares e teoria da rede. (a) A porção de um antígeno que é reconhecida por um anticorpo é chamada epítipo. Enquanto os anticorpos são monoespecíficos, os antígenos podem apresentar vários epítopos distintos. (b) Molécula de anticorpo destacando paratopo e idiotopo. (c) Respostas positiva e negativa resultantes da interação de um paratopo com um idiotopo ou um epítipo. . . . .	15
2.2	Ilustração aiNet. (a) Conjunto de dados com três <i>clusters</i> densos. (b) Rede de anticorpos rotulados e conexões ponderadas. As linhas tracejadas indicam conexões a serem removidas para gerar os subgrafos que identificam os <i>clusters</i> . . . . .	17
2.3	Ilustração ARIA. (a) Posicionamento dos anticorpos após a convergência da rede com seus respectivos raios de supressão (raios dos anticorpos). (b) AGM construída sobre os anticorpos . . . . .	20
2.4	Comparação aiNet X ARIA . . . . .	23
3.1	Maturação de afinidade e expansão clonal . . . . .	29
3.2	Raio de Vizinhaça e Raio do anticorpo . . . . .	30
3.3	Supressão de Rede . . . . .	31
3.4	Heurística de expansão clonal . . . . .	32
3.5	Ajuste de pesos da rede . . . . .	32
4.1	Evolução da rede.(a)Conjunto completo da Iris identificando as classes por cores. (b) Conjunto de treinamento com 80% das amostras. (c) População inicial de anticorpos com um anticorpo por classe (em vermelho) em contraste com o conjunto de treinamento (em verde). Cada anticorpo é inicializado com a média dos atributos da classe a qual pertence. (d) População de anticorpos gerada após o término da primeira fase da SAINET (em vermelho). Total de 13 anticorpos. (e) Rede após a segunda fase do algoritmo. Não houve nesse caso necessidade de duplicação de nenhum anticorpo, não havendo assim alteração na população. (f) Aplicação do classificador ao conjunto de teste. Taxa de Classificação Correta: 96,67 %. . . . .	39

## Lista de Tabelas

2.1	Inspiração biológica e aplicações de Sistemas Imunológicos Artificiais . . .	14
2.2	Descrição dos símbolos no algoritmo ARIA . . . . .	21
2.3	Descrição dos símbolos no algoritmo SAINET . . . . .	26
4.1	Resultados da aplicação da SAINET à base da Iris . . . . .	35
4.2	Resultados da aplicação da SAINET à base de Vinhos . . . . .	35
4.3	Taxa de compressão da SAINET sobre as bases de dados . . . . .	35
4.4	Comparação entre os resultados obtidos pela SAINET original e a versão modificada sobre a base Iris . . . . .	36
4.5	Comparação entre os resultados obtidos pela versão modificada da SAINET e os resultados obtidos pelo modelo Naive Bayes sobre a base de Vinhos . .	37

## Lista de Abreviações

DCC	Departamento de Ciência da Computação
UFJF	Universidade Federal de Juiz de Fora
AG	Algoritmo Genético
PG	Programação Genética
RNA	Redes Neurais Artificiais
SI	Sistema Imunológico
SIA	Sistema Imunológico Artificial
RIA	Rede Imunológica Artificial
aiNet	Artificial Immune Network
ARIA	Adaptative Radius Immune Algorithm
SAINET	Supervised Artificial Immune Network
AGM	Árvore Geradora Mínima

# 1 Introdução

A natureza de modo geral é um exemplo de sucesso. A biologia dos seres vivos, o funcionamento dos organismos, o comportamento e as funções de cada indivíduo no meio em que estão inseridos, demonstram que de alguma forma, por coincidência ou não, existe uma receita que garante a existência de vida na Terra. Como e porque isso acontece tem motivado estudos em diversas áreas através dos séculos. O trabalho mais conhecido nessa área, recentemente reconhecido como o livro acadêmico mais influente da história (OLIVEIRA, 2015), foi publicado pelo naturalista britânico Charles Darwin em 1859, sob o título completo de *Origem das Espécies por Meio da Seleção Natural ou a Preservação de Raças Favorecidas na Luta pela Vida*. De acordo com Darwin, os indivíduos mais adaptados ao meio em que vivem possuem mais chances de sobreviver para reproduzir e passar suas características para a geração seguinte, enquanto aqueles que não se adaptam tão bem, tendem a ser eliminados antes de se reproduzirem. Dessa forma, somente as características consideradas interessantes para determinado ambiente são transmitidas, gerando uma população de indivíduos com características semelhantes e cada vez mais evoluídos (adaptados ao meio). Essa ideia é chamada Teoria da Seleção Natural, e entre outros exemplos, inspira um ramo de pesquisa dentro da Ciência da Computação, na área de Inteligência Computacional, conhecida como Computação Evolucionista ou Evolutiva. Tal ramo, consiste em um conjunto de heurísticas de busca e otimização inspiradas na evolução natural das espécies, dentre as quais podemos citar os famosos Algoritmos Genéticos (AGs), Programação Genética (PG), Colônias de Formigas, Colônias de Abelhas, Enxame de Partículas, dentre outras (TALBI, 2009). Para este trabalho, foi escolhida uma técnica baseada em um Sistema Imunológico Artificial (SIA).

Computacionalmente, o que torna interessante o Sistema Imunológico, Imunitário ou Imune, dos seres vertebrados é a capacidade de distinguir elementos que são próprios do organismo de outros que não sejam próprios do organismo (agentes patogênicos), e eliminá-los sem prejudicar os elementos próprios. Além disso, o sistema é capaz de recordar encontros passados com agentes patogênicos e, em contato com o mesmo agente

no futuro é capaz de combatê-lo de forma mais eficiente. Tais habilidades são conhecidas como reconhecimento de próprio e não-próprio e memória imunológica, respectivamente.

Os primeiros estudos publicados sobre os SIAs surgiram no início dos anos 90, e ainda há muito a ser desenvolvido. Os SIAs podem ser classificados de acordo com sua inspiração biológica: algoritmos inspirados em seleção clonal são indicados para problemas de busca e otimização, enquanto os inspirados na capacidade de reconhecimento de próprio e não-próprio são indicados para detecção de falhas e anomalias em segurança computacional, por exemplo. A inspiração do presente trabalho tem origem na memória imunológica do Sistema Imunológico e no conceito de Rede Imunológica Artificial (RIA). Dessa inspiração, originaram-se diversos métodos computacionais para resolução de problemas na área de Mineração de Dados. A Mineração de Dados pode ser caracterizada como uma área multidisciplinar que utiliza várias técnicas para extrair informações de um conjunto de dados. Algumas definições formais são apresentadas em (CAMILO; SILVA, ). Uma das tarefas mais populares é a classificação, que tem como intuito identificar a qual classe pertence um determinado registro. Para isso, é preciso apresentar o modelo classificador as classes existentes no conjunto de dados e “ensiná-lo” a identificar as características de cada classe de acordo com as amostras para que, dadas novas entradas de dados com o mesmo padrão de informação, o mesmo seja capaz de identificar automaticamente a qual classe pertence um novo elemento. Passa-se para o classificador um conjunto de dados com informação sobre as classes das amostras, para que o mesmo encontre uma relação entre os dados de entrada e as classes. Esse processo é conhecido como aprendizado ou treinamento e é um processo supervisionado. A classificação serve para determinar valores de variáveis do tipo categórico, como por exemplo, indivíduos que pertencem a um grupo de risco para desenvolvimento de determinada doença, quais turmas em uma escola são indicadas para um determinado aluno, identificar qual o tipo de vinho a partir de informações sobre a sua composição (CAMILO; SILVA, ).

Existem vários métodos considerados populares para classificação de dados (CAMILO; SILVA, ): Árvores de Decisão, *Support Vector Machine* (SVM), Redes Neurais Artificiais (RNAs), Algoritmos Genéticos (AGs), dentre outros. Aplicações para problemas de classificação também são encontradas nas mais diversas áreas de conhecimento e

negócios, desde a biologia, estatística, a bancos de dados de empresas. Dados de aplicações se apresentam nas mais diversas configurações, podendo apresentar um padrão uniforme ou não. Essa configuração pode afetar diretamente o resultado da classificação dependendo da ferramenta classificadora utilizada, caso esta não possua sensibilidade à diversidade dos dados a serem tratados. Surge assim a necessidade de se estudar novas técnicas de classificação ou aprimorar as técnicas já existentes a fim de melhorar os resultados.

Um dos primeiros algoritmos imuno-inspirados foi o *Artificial Immune Network* (aiNet). Na versão original, o algoritmo se propõe a resolver problemas de agrupamento (clusterização), porém outras versões foram implementadas com intuito de resolver problemas de outras naturezas como classificação e otimização, por exemplo. Apesar do sucesso, a aiNet não apresenta resultados satisfatórios para casos em que os dados não apresentam as mesmas dispersões de dados e/ou há grande desbalanceamento de amostras entre as classes. Da motivação em resolver este problema, surge em 2005, um novo algoritmo para clusterização com uma abordagem que apresenta melhores resultados para clusterização nesse tipo de situação, *Adaptive Radius Immune Algorithm* (ARIA) (BEZERRA et al., 2005).

Para problemas de classificação, é possível encontrar várias opções de algoritmos imunológicos (KNIDEL, 2006). Destaca-se uma rede supervisionada chamada SAINET (KNIDEL, 2006). A primeira fase do algoritmo da SAINET é constituída por um processo de clusterização feito pelo aiNet. Porém como já foi dito, tal algoritmo apresenta problemas para clusterização de amostras que não apresentam a mesma dispersão de dados, enquanto o ARIA apresenta resultados melhores.

Este trabalho tem como objetivo propor uma modificação no algoritmo da SAINET e comparar os resultados obtidos com outros modelos classificadores, sendo um deles a própria versão original da SAINET.

## 2 Fundamentação Teórica

De maneira semelhante a outras heurísticas evolutivas, os SIAs são técnicas bio-inspiradas. Partindo da conceituação e apresentação de informações importantes com relação ao Sistema Imunológico natural, este capítulo apresenta conceitos e informações sobre a estrutura, funcionalidade e aplicações dos Sistemas Imunológicos Artificiais.

### 2.1 Inspiração Natural

#### 2.1.1 Sistema Imunológico Natural

Para que um organismo funcione de maneira adequada, são necessárias estruturas ou elementos que garantam que as necessidades básicas do mesmo sejam supridas. Além da capacidade de se manter funcionando por si só, o organismo necessita de mecanismos para se defender de ameaças à sua integridade, sejam elas advindas do meio exterior ou do próprio organismo. Ao conjunto de defesas do organismo dá-se o nome de Sistema Imunológico, Imunitário ou Imune. Nos vertebrados, o SI pode ser dividido em duas categorias: inato e adaptativo. O primeiro é constituído das defesas naturais com as quais o indivíduo já nasce, como pele, saliva, lágrima, mucosas, enzimas e algumas células brancas e moléculas originárias do sangue. Já o segundo, é formado por células chamadas linfócitos e seus produtos como os anticorpos, responsáveis por identificar e combater elementos estranhos e prejudiciais ao organismo aos quais este pode ser exposto em algum momento da vida, denominados agentes patogênicos ou antígenos.

Diferentemente da imunidade inata que possui caráter generalizado, a imunidade adaptativa desenvolve anticorpos específicos para cada agente patogênico detectado (FIGUEREDO; BERNARDINO; BARBOSA, 2013). Ao ser exposto a um antígeno, um tipo especial de células brancas entra em ação, os linfócitos. Os linfócitos identificam os antígenos e produzem anticorpos para combatê-los (ABBAS A. K., ). Durante esse processo, anticorpos são selecionados naturalmente de modo que aqueles que tenham mais

chance de combater de maneira eficaz o antígeno sobrevivam, e os linfócitos são aprimorados de modo a produzir anticorpos melhores. Isso faz com que o número de anticorpos no sangue para combater um determinado tipo de antígeno aumente e caia logo em seguida, quando o agente patogênico estiver neutralizado. Caso o organismo venha a ser exposto novamente ao mesmo agente patogênico, aumenta-se rapidamente a concentração dos anticorpos produzidos anteriormente neutralizando-o sem grandes dificuldades (LOPES, 2010). Anticorpos e antígenos podem ser vistos como figuras que se encaixam, sendo feitos sob medida. Os anticorpos são formados de modo que suas características físico-químicas possibilitem interações com os antígenos. O espaço onde tais moléculas se encontram e interagem é formalmente chamado espaço de formas S.

### 2.1.2 Sistema Imunológico Artificial

Inspirados no Sistema Imune natural, os Sistemas Imunológicos Artificiais (SIAs) se baseiam em propriedades específicas do sistema natural para resolver problemas computacionais diversos. Na Tabela 2.1, extraída de (LOPES, 2010) é apresentada uma relação entre a inspiração biológica e os problemas e aplicações típicos abordados.

Aspecto Imunológico	Problema Computacional	Aplicações Típicas
Reconhecimento de Próprio e Não-Próprio do organismo	Detecção de mudanças ou anomalias	Segurança de computador Detecção de falha
Teoria de Rede Imunológica e Memória Imunológica	Aprendizagem (supervisionada ou não)	Classificação Clusterização Análise de Dados Mineração de Dados
Seleção Clonal	Busca, otimização	Otimização de Funções
Mobilidade e Distribuição	Processamento Distribuído	Arquiteturas de agentes Controle robótico descentralizado
Imunidade Inata	Teoria de Perigo	Segurança de redes

Tabela 2.1: Inspiração biológica e aplicações de Sistemas Imunológicos Artificiais

### 2.1.3 Redes Imunológicas

A Teoria de Rede Imunológica (JERNE, 1974) propõe novas ideias sobre a produção de anticorpos e discute sobre seleção do repertório pré-imune, diferenciação entre próprio e

não-próprio, tolerância ao próprio, evolução e memória do SI. A Teoria de Rede propõe que o sistema está ativo o tempo todo e não precisa de um estímulo externo para começar a agir além de ser capaz de reconhecer também as células e moléculas do próprio organismo. (KNIDEL, 2006).

(KNIDEL, 2006) ainda define formalmente o SI de acordo com a Teoria de Rede como sendo um conjunto de paratopos e idiotopos que se reconhecem e são reconhecidos entre si. Paratopos e idiotopos são elementos contidos na superfície dos anticorpos, sendo idiotopo um receptor e paratopo um marcador que se liga à superfície de um antígeno em regiões chamadas epítomos. Para a teoria de rede, não só as moléculas são relevantes, mas também as suas interações. Quando uma molécula é reconhecida e identificada como nociva, ocorre o que se chama resposta positiva, resultando em uma proliferação de células, na ativação e secreção de anticorpos para atacar a molécula. Caso contrário, ocorre que se chama resposta negativa, sendo tal molécula tolerada. A Figura 2.1 extraída também de (KNIDEL, 2006) ilustra esta interação.

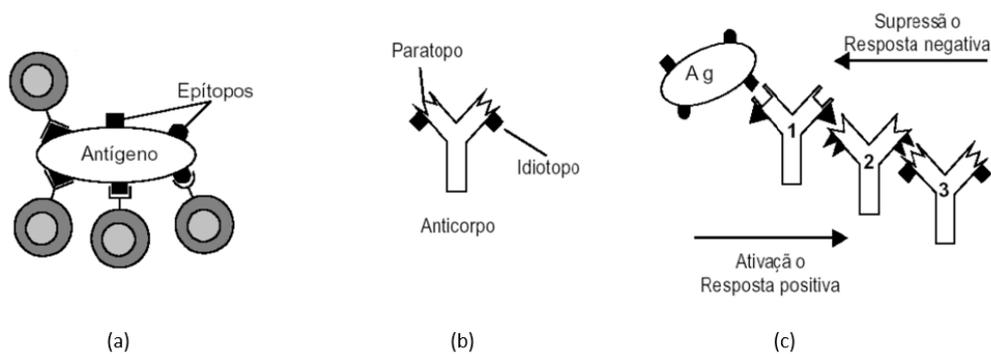


Figura 2.1: Identificadores moleculares e teoria da rede. (a) A porção de um antígeno que é reconhecida por um anticorpo é chamada epítomo. Enquanto os anticorpos são mono-específicos, os antígenos podem apresentar vários epítomos distintos. (b) Molécula de anticorpo destacando paratopo e idiotopo. (c) Respostas positiva e negativa resultantes da interação de um paratopo com um idiotopo ou um epítomo.

As principais características de uma rede imunológica são:

- Estrutura: responsável pela descrição dos padrões de conexão e interação dos componentes da rede
- Dinâmica: variação da concentração e da estrutura dos anticorpos ao longo do tempo
- Metadinâmica: produção contínua de novos anticorpos e morte de anticorpos pouco

estimulados. Introduce diversidade e garante eficiência ao SI.

A seguir, são introduzidas três Redes Imunológicas Artificiais, sendo as duas primeiras para clusterização e a terceira para classificação.

### aiNet

O primeiro algoritmo para uma RIA foi proposto por (CASTRO; ZUBEN, 2002). Conhecido pelo nome de aiNet (*Artificial Immune Network*), tinha por objetivo inicial resolver problemas de clusterização, e era capaz de identificar automaticamente um número arbitrário de pontos. A aiNet faz uso além da Teoria de Rede, dos princípios de Seleção Clonal e de maturidade de afinidade. A primeira versão do algoritmo ainda introduziu novos aspectos no estudo como as interações da rede entre as soluções e a adaptação dinâmica do conjunto de soluções candidatas.

A partir de 2002, outras versões da aiNet foram desenvolvidas para aplicações em problemas de áreas diversas da computação como classificação, otimização, problemas combinatórios, bioinformática, entre outras. Referências para alguns destes trabalhos e outros podem ser encontradas em (FRANÇA et al., 2010).

A aiNet pode ser formalmente definida como um grafo com conexões ponderadas, não necessariamente totalmente interconectado, composto por um conjunto de nós, denominados anticorpos, e conjuntos de pares de nós, chamados conexões, com um valor característico associado, chamado de peso da conexão ou simplesmente peso (CASTRO; ZUBEN, 2002). A Figura 2.2a mostra o exemplo de um conjunto de dados com 3 *cluster* densos e a Figura 2.2b mostra a arquitetura de uma rede hipotética gerada pelo algoritmo aiNet para este conjunto, ambas extraídas de (CASTRO; ZUBEN, 2002).

O pseudocódigo é apresentado no Algoritmo 1 e as etapas do algoritmo podem ser resumidas da seguinte forma:

- **Inicialização:** criação de uma população aleatória de anticorpos.
- **Apresentação dos antígenos:**

**Seleção clonal e expansão:** determinar a afinidade para cada anticorpo da rede, e selecionar um número de anticorpos com alta afinidade para serem clonados de

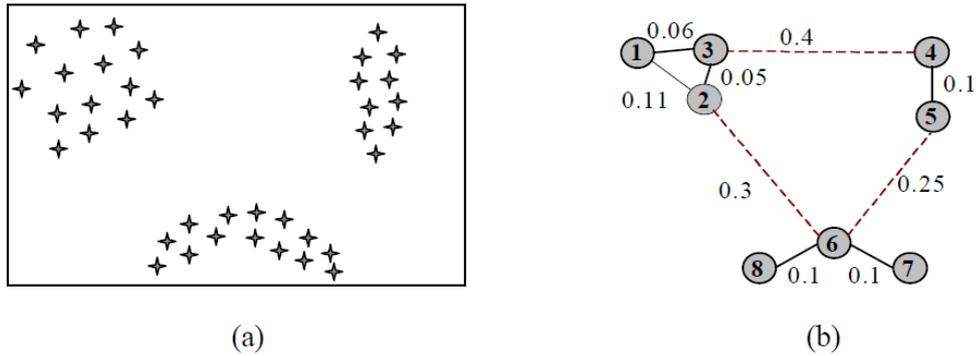


Figura 2.2: Ilustração aiNet. (a) Conjunto de dados com três *clusters* densos. (b) Rede de anticorpos rotulados e conexões ponderadas. As linhas tracejadas indicam conexões a serem removidas para gerar os subgrafos que identificam os *clusters*

forma proporcional à sua afinidade;

**Maturação e afinidade:** aplicar a cada um dos clones uma mutação inversamente proporcional à sua afinidade;

**Morte dos elementos não estimulados:** eliminar da memória os clones cuja afinidade com o antígeno seja menor que um limiar estabelecido;

**Interações clonais:** determinar a afinidade de todos os elementos da rede

**Supressão clonal:** eliminar os clones cuja afinidade entre si seja menor que um limiar estabelecido;

**Construção da rede:** incorporar os elementos restantes do conjunto de clones de memória aos anticorpos da rede.

- **Interações da rede:** determinar a similaridade entre cada par de células da rede.
- **Supressão da rede:** eliminar da rede todos os anticorpos cuja afinidade em relação a um anticorpo específico seja maior que um limiar pré-definido. Este anticorpo específico deve ter afinidade a antígenos do que os anticorpos a serem eliminados.
- **Introdução de novos elementos:** introduzir na rede um número fixo de anticorpos gerados aleatoriamente.
- **Ciclo:** repetir os passos 2 a 5 até atingir um número máximo de iterações pré-estabelecido.

A primeira versão da aiNet apresentava algumas características que podem ser

**Algoritmo 1:** Algoritmo aiNet

---

**Entrada:**  $N$  : tamanho da população inicial de anticorpos  
**n:** número de células a serem clonadas  
**m:** porcentagem de células a serem mantidas no conjunto de memória  
 $\sigma_d$ : limiar de supressão anticorpo-antígeno  
 $\sigma_s$ : limiar de supressão anticorpo-anticorpo  
**d:** porcentagem de anticorpos gerados aleatoriamente a serem incluídos na população  
**Saída:** população de anticorpos da rede

```

1 início
2   construir população inicial com N anticorpos abs gerados aleatoriamente
3   while não atingir o critério de parada do
4     for cada antígeno ag do
5       determinar a sua afinidade para cada ab da rede;
6       selecionar os n abs com afinidade mais alta;
7       gerar Nc clones a partir dos n abs selecionados. Quanto maior a
          afinidade, maior será Nc:
          
$$Nc = \sum_{i=1}^N \text{round}(N - D_{ab_i, ag_j} \cdot N)$$

          sendo  $D_{ab_i, ag_j}$  a distância
          entre o anticorpo  $ab_i$  e o antígeno  $ag_j$ 
          aplicar a hipermutação ao conjunto C de clones gerados, com
          variabilidade inversamente proporcional à afinidade do ab pai
          segundo
          
$$c_k^* = c_k + \alpha_k(ag - c_k) \quad ;$$

          onde  $\alpha$  é a taxa de mutação de cada clone c definida por:
          
$$\alpha_k = e^{-\rho \cdot f_i / f_{max}} \quad ;$$

          determinar a afinidade entre ag e cada um dos clones;
          manter apenas m% dos clones com maior afinidade na
          população;
          eliminar todos os clones cuja afinidade para com ag seja menor
          que  $\sigma_d$  (apoptosis);
          determinar a afinidade entre si dos clones mutados e eliminar
          aqueles cuja afinidade seja maior que  $\sigma_s$  (supressão);
          concatenar os clones e o conjunto de anticorpos de memória ;
8     end for
9     eliminar da rede abs que não reconheçam nenhum ag;
10    introduzir uma porcentagem d de novos anticorpos na população
11  end while
12 fim

```

---

aprimoradas para aumentar sua eficiência e estabilidade, e possibilitar a aplicação de RIAs a uma gama maior de problemas. Posteriormente, uma nova rede imuno-inspirada chamada ARIA (*Adaptative Radius Immune Algorithm*) foi proposta por (BEZERRA et al., 2005) apresentando a capacidade de lidar com a densidade variável de dados entre *clusters*, algo que não era possível com o aiNet.

## ARIA

Grande parte dos algoritmos imuno-inspirados para clusterização são baseados em compressão de dados e redução, e funcionam posicionando os anticorpos em áreas mais representativas do conjunto de dados (posicionamento de protótipos) e podem ser separados em *clusters* com a ajuda de ferramentas de visualização, o que reduz a complexidade do procedimento. Porém, essa estratégia pode ser problemática em algumas situações por exemplo, quando os *clusters* identificados estão muito próximos, quando as densidades de dados variam de *cluster* para *cluster*, ou ainda quando as fronteiras dos *clusters* se sobrepõem. Nesse caso, a compressão de dados pode prejudicar a representação de dados fundamentais para a identificação dos *clusters*. Isso acontece porque o posicionamento de protótipos não leva em conta a densidade dos dados, resultando numa diferença entre a distância entre os anticorpos na representação dos dados pelo modelo e a distância entre dados reais que pode afetar drasticamente a performance do particionamento (BEZERRA et al., 2005).

No cenário descrito acima, surge a formulação de um algoritmo imuno-inspirado que usa expansão clonal aliada à informação de densidade presente no conjunto de dados afim de produzir uma representação mais correta dos dados reais, chamado de ARIA (*Adaptative Radius Immune Algorithm*) (BEZERRA et al., 2005). Sendo rápido em termos de execução e simples de compreender e implementar, ARIA faz uso de uma supressão adaptativa do raio (distância entre os anticorpos) que é inversamente proporcional à densidade local para cada vizinho do anticorpo, mantendo assim a aglomeração ou esparsidade dos dados. Preservando a densidade após a compressão, o ARIA gera uma representação mais fiel dos dados reais, garantindo assim mais eficiência na clusterização (BEZERRA et al., 2005). A representação dos dados por anticorpos e seus raios pode ser vista na Figura 2.3a. Nesta figura extraída de (BEZERRA et al., 2005), é possível a rede construída pelo algoritmo para uma base de dados com dois *clusters* com densidades diferentes. A Figura 2.3b mostra a Árvore Geradora Mínima (AGM) obtida sobre a população de anticorpos, que deve ser particionada a fim de identificar os *clusters*.

ARIA é um algoritmo adaptativo que pode ser dividido em três etapas (BEZERRA et al., 2005):

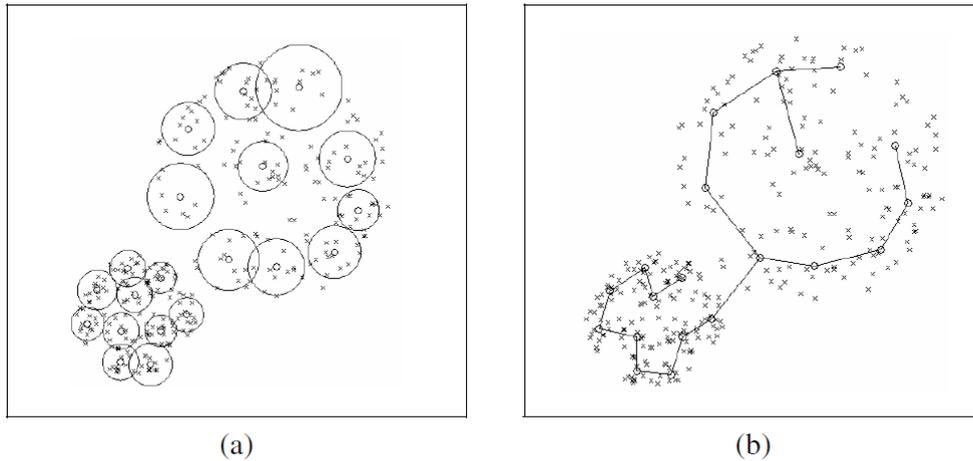


Figura 2.3: Ilustração ARIA. (a) Posicionamento dos anticorpos após a convergência da rede com seus respectivos raios de supressão (raios dos anticorpos). (b) AGM construída sobre os anticorpos

1. Maturação de Afinidade (mutação): os dados, representados por antígenos, são apresentados aos anticorpos. Os anticorpos sofrem hipermutação a fim de se adaptarem aos antígenos (interação antígeno-anticorpo);
2. Expansão Clonal: os anticorpos mais estimulados são selecionados para serem clonados, expandindo assim a rede;
3. Supressão de Rede: a interação entre anticorpos é quantificada, e se um anticorpo reconhece o outro, um deles é removido do pool de células (interação anticorpo-anticorpo).

O Algoritmo 2 apresenta o pseudocódigo do ARIA e uma descrição dos símbolos utilizados (BEZERRA et al., 2005) na Tabela 2.2.

### aiNet X ARIA

Em termos dos parâmetros de cada algoritmo, dos oito parâmetros importantes da aiNet, seis (entrada) são definidos pelo usuário (além do critério de parada), e apenas dois são calculados em função da população (taxa de mutação  $\alpha$  e o tamanho  $N_c$  do conjunto de clones), o que torna o algoritmo muito sensível aos parâmetros de entrada. Além disso,  $\alpha$  depende ainda de um outro fator  $\rho$  que não é especificado por (CASTRO; ZUBEN, 2002). Já no algoritmo ARIA, dos sete parâmetros indicados na Tabela 2.2, apenas o raio mínimo  $r$  e o fator de decaimento da taxa de mutação *decay* (além do critério de parada,

**Algoritmo 2:** ARIA

---

**Entrada:** conjunto de dados,  $r$ ,  $gen$   
**Saída:** população de anticorpos da rede

```

1 início
2   inicializar variáveis;
3   for cada antígeno Ag do
4     |   selecionar melhor anticorpo Ab;
5     |   mutar Ab segundo  $\mu$ ;
6   end for
7   matar anticorpos não estimulados;
8   clonar anticorpos que reconhecem antígenos a uma distância maior que
   R;
9   calcular a densidade local para cada Ab;
10  calcular calcular o limiar de supressão de Ab fazendo:
    $R_{Ab} = r * (den_{max}/den)^{\frac{1}{dim}}$ 
11  suprimir anticorpos dando prioridade de sobrevivência para aqueles
   com menor R;
12  Fazer  $E = media(R)$ ;
13  if geração corrente >  $gen/2$  then
14    |    $\mu = \mu * decay$ ;
15  end if
16 fim

```

---

Símbolo	Descrição
R	Raio de cada anticorpo (limite de supressão)
$r^*$	Raio multiplicador. Determina o tamanho do raio
$\mu$	Taxa de Mutação
$decay^*$	Multiplicador constante usado para diminuir a taxa de mutação
E	Raio que define a vizinhança para estimativa de densidade
$gen^*$	número de iterações
dim	Dimensão dos dados de entrada

Tabela 2.2: Descrição dos símbolos no algoritmo ARIA

que neste caso é número de gerações  $gen$ ), são definidos pelo usuário. Informações sobre como definir o valor de  $r$  podem ser encontradas em (AZZOLINI; ZUBEN, 2010).

Ainda sobre os parâmetros, dos listados acima, alguns são claramente equivalentes, como taxa de mutação e número de gerações. Entretanto, é interessante atentar para os limiares de supressão adotados por ambos. A aiNet adota um limiar de supressão  $\sigma_d$  para eliminar anticorpos com baixa afinidade para com os antígenos, fase chamada de *apoptosis*, e um outro limiar  $\sigma_s$  para eliminar os anticorpos que tenham alta afinidade com outros anticorpos, ou seja, que reconheçam outros anticorpos, fase conhecida como

supressão de rede. A apoptose no algoritmo da ARIA é feita suprimindo os anticorpos que não foram estimulados por nenhum antígeno durante a maturação de afinidade, não havendo portanto a necessidade de um limiar para decidir se anticorpo é ou não mantido. A supressão de rede adota de forma semelhante um limiar de supressão, que é o próprio raio do anticorpo, sendo este calculado pelo próprio algoritmo em função do parâmetro  $r$  e da densidade que é dada em função do raio de vizinhança  $E$  que também é atualizado ao longo das iterações.

A mutação se dá de forma semelhante em ambos algoritmos: aplicando-se um operador mutação sobre o anticorpo selecionado levando em conta a afinidade para com o antígeno, uma taxa de mutação e os atributos de ambos. Por outro lado, o processo de clonagem ocorre de forma bem distinta na ARIA em relação à aiNet. Enquanto a clonagem na aiNet ocorre antes da mutação e se resume a gerar cópias idênticas do anticorpo pai, na ARIA a clonagem consiste em aplicar o operador de mutação novamente à uma cópia feita do anticorpo pai no momento em que este é selecionado para sofrer mutação, caso o antígeno esteja localizado a uma distância do anticorpo maior que seu raio. Em outras palavras, cada clone nada mais é do que uma cópia mutada de seu anticorpo pai (BEZERRA et al., 2005). A ARIA limita o número de clones permitido por anticorpo a, no máximo 1. Desta forma, limita-se a quantidade possível de descendentes de cada anticorpo: uma cópia fiel do mesmo mantida na próxima geração, uma versão mutada e (ou não) uma cópia mutada, caso este seja estimulado. Enquanto na aiNet, o número de clones de cada anticorpo é proporcional à sua afinidade para com o antígeno considerado.

A saída de ambos algoritmos tem o mesmo formato: um conjunto de anticorpos de memória que representam o conjunto de dados (antígenos) de forma reduzida no número objetos, porém sem perda de informação.

A Figura 2.4 apresenta um quadro comparativo entre os algoritmos da aiNet e da ARIA, destacando diferenças e semelhanças entre o funcionamento de ambos.

## SAINET

A partir da versão original do aiNet, várias adaptações do algoritmo destinadas a resolver problemas de classificação foram desenvolvidas. Além destas, novas propostas de redes

	<b>aiNet</b>	<b>ARIA</b>
Parâmetros de entrada (definidos pelo usuário)	$\sigma d, \alpha, N, n, m, d, \text{gen}$	$r, \text{decay}, \text{gen}$
Parâmetros calculados pelo algoritmo	$N_c, \alpha$	$E, R, \mu, \text{dim}$
Limiares de supressão	$\sigma d, \alpha$	$R$
Mutação	aplica-se um operador Mutação sobre o anticorpo selecionado levando em conta a afinidade para com o antígeno, uma taxa de mutação e os atributos de ambos	aplica-se um operador Mutação sobre o anticorpo selecionado levando em conta a afinidade para com o antígeno, uma taxa de mutação e os atributos de ambos
Clonagem	Ocorre antes da mutação; gera cópias idênticas ao anticorpo pai em quantidade proporcional à afinidade do anticorpo para com o antígeno considerado	Ocorre depois da mutação; aplicação do operador de mutação em uma cópia idêntica do anticorpo pai e se limita a, no máximo, 1 clone por anticorpo selecionado
Supressão clonal (apoptosis)	Os clones cuja afinidade para com o antígeno considerado seja menor que $\sigma d$ são removidos	Os anticorpos que não foram estimulados durante a Maturação de afinidade são eliminados da população intermediária
Supressão de Rede	<ol style="list-style-type: none"> <li>1) Elimina da rede anticorpos que tenham afinidade para com outros anticorpo menor que <math>\alpha</math></li> <li>2) Elimina da rede todos os anticorpos que não reconhecem nenhum antígeno</li> </ol>	Elimina da rede anticorpos que se reconhecem, ou seja, anticorpos localizados a uma distância menor que o raio de algum deles

Figura 2.4: Comparação aiNet X ARIA

neuro-imunológicas, redes imunológicas híbridas de Redes Neurais Artificiais (RNAs) com intuito de aprimorar os resultados. Alguns exemplos de redes supervisionadas para classificação são apresentados e discutidos por (KNIDEL, 2006). A Supervised Artificial Immune Network (SAINET) é um algoritmo para classificação de dados desenvolvido a partir do aiNet original. O funcionamento da SAINET é constituído de três fases (KNIDEL, 2006):

1. Aplicação do aiNet ao conjunto original de dados analisado para construir um conjunto de memória que reconheça e represente a organização estrutural dos dados: o número de anticorpos é controlado por um limiar de supressão relacionado à especificidade dos anticorpos da rede. Por ser uma abordagem supervisionada, há necessidade de rotular os anticorpos responsáveis pela classificação, utiliza-se então um vetor para armazenar o rótulo ao qual cada anticorpo pertence e, quando um anticorpo é clonado na expansão, o rótulo do clone é o mesmo do anticorpo que o gerou.
2. Verificação da necessidade de duplicação do anticorpo: feita através de uma heurística que apura se os anticorpos são capazes de reconhecer uma quantidade de padrões não-próprios maior que um limiar estabelecido pelo usuário (LRPNP – Limiar de

Reconhecimento de Padrões Não-Próprios). Em outras palavras, avalia a capacidade de reconhecer padrões diferentes da classe à qual o anticorpo pertence. Caso essa condição seja satisfeita, o anticorpo é duplicado, sendo que, os atributos do novo anticorpo são a média aritmética das amostras pertencentes à classe considerada.

3. Atualização dos pesos: nesta fase se encontra a aprendizagem supervisionada. Os pesos da rede são atualizados de forma semelhante à adota na Aprendizagem por Quantização Vetorial (KOHONEN, 2010), segundo a equação 3.3, apresentada na seção 3.2.1,

O Algoritmo 3 apresenta o pseudocódigo da SAINET e uma breve descrição dos parâmetros utilizados no código (KNIDEL, 2006) é mostrada na Tabela 2.3.

---

**Algoritmo 3: SAINET**

---

**Entrada:** conjunto de dados,  $gen, N_c, T_s, T_r, \alpha, \sigma$ **Saída:** AB, R,

```

1 início
2   inicializar a rede com um anticorpo  $ab$  para cada classe presente nos
   dados de treinamento com a média dos atributos da classe da classe a
   qual ele for atribuído;
3   while iteração for menor que máximo de gerações do
4     for cada antígeno  $ag$  do
5       apresentar um antígeno escolhido aleatoriamente à rede, sem
       repetição ;
6       calcular a distância euclidiana entre o  $ag$  apresentado e todos os
        $abs$  da rede;
7       clonar o anticorpo vencedor (o mais próximo do  $ag$  apresentado)
       produzindo  $N_c$  clones ;
8       atualizar o vetor de atributos do anticorpo com
9        $ab_k = ab_k + \alpha * (ag - ab_k$ 
10      onde  $k$  é o índice do  $ab$  vencedor e  $\alpha$  é a taxa de aprendizagem;
11     end for
12     atualizar o fator de aprendizagem:  $\alpha = \sigma * \alpha$  ;
13     suprimir os anticorpos que se auto reconhecem dentro do limiar de
       supressão  $T_s$ 
14   end while
15   for para cada anticorpo  $ab$  do
16     for cada classe diferente da classe de  $ab$  do
17       if razão entre o número de padrões que reconhecidos por  $ab$  que
       peretencem a uma classe diferente da sua, pelo número de
       padrões que reconhecidos por  $ab$  que peretencem à mesma classe
       de  $ab$  for maior que  $T_r$  then
18         duplicar  $ab$  e atribuir à cópia a média dos atributos da
         classe de  $ab$ 
19       end if
20     end for
21   end for
22   while iteração menor que o número máximo de gerações do
23     apresentar um antígeno ag escolhido aleatoriamente à rede, sem
     repetição;
24     encontrar o anticorpo  $ab$  vencedor e atualizar seu vetor de
     atributos de acordo com a equação 3.3
25   end while
26 fim

```

---

Símbolo	Descrição
gen	número de gerações
Nc	número de clones gerados
Ts	limiar de supressão usado na eliminação de anticorpos que se auto-reconhecem
Tr	limiar de reconhecimento de padrões não-próprios
$\alpha$	taxa de aprendizagem
$\sigma$	fator que controla o decaimento da taxa de aprendizagem
Matriz  AB	conjunto de anticorpos (protótipos)
Vetor  R	armazena os rótulos (classes) dos anticorpos da matriz  AB

Tabela 2.3: Descrição dos símbolos no algoritmo SAINET

## **3 SAINET com raio adaptativo**

A rede escolhida para realizar a classificação de dados neste trabalho foi a SAINET (KNIDEL, 2006). O funcionamento da rede implementada neste trabalho é explicado nas seções que seguem, e que são organizadas de acordo com as etapas do algoritmo.

### **3.1 Fase 1: Construção de conjunto de memória (rede de anticorpos)**

Na formulação original da SAINET a primeira fase consiste na execução da aiNet, apresentada na Seção 2.1.3, para contruir um conjunto de anticorpos de memória. Neste trabalho, a aiNet foi substituída pela ARIA, também apresentada na Seção 2.1.3. As fases do algoritmo ARIA são detalhadas a seguir.

#### **3.1.1 Inicialização das variáveis**

A população inicial de anticorpos é inicializada com valores aleatórios. O tamanho da população inicial não interfere no desempenho do algoritmo uma vez que esta cresce e diminui durante as iterações. Entretanto, por se tratar de uma abordagem supervisionada, é necessário manter a informação da classe a qual cada anticorpo pertence. Dessa forma, a rede é inicializada com um anticorpo para cada classe conhecida no conjunto de dados, sendo os atributos de cada anticorpo inicialmente definidos como a média dos atributos da classe a qual este pertence. Nos passos seguintes, à medida que as operações de mutação e clonagem são aplicadas, a população cresce e os anticorpos filhos gerados têm atributos diferentes mas mantêm a classe do pai.

#### **3.1.2 Maturação de afinidade**

Cada um dos antígenos é comparado com todos os anticorpos que compõem a população atual, sendo selecionado para ser mutado o anticorpo que possui maior afinidade com o

antígeno, onde a afinidade é definida como a menor distância euclidiana (pode-se aplicar qualquer outra métrica ou cálculo de distância compatível com o tipo dos dados do problema em questão). Para cada anticorpo selecionado, é feita uma cópia, antes da mutação, que poderá ou não ser clonada mais a frente. Os anticorpos mutados compõem a população intermediária construída e manipulada a cada geração. O operador mutação é definido como segue na equação 3.1, onde  $Ab'$  é o anticorpo resultante da mutação,  $Ab$  é o anticorpo selecionado,  $\mu$  é a taxa de mutação,  $rand$  é um valor aleatório uniformemente gerado entre 0 e 1,  $Ag$  é o antígeno considerado, e  $Ag - Ab$  é a Distância Euclidiana entre  $Ab$  e  $Ag$ .

$$Ab' = Ab + \mu * rand * (Ag - Ab) \quad (3.1)$$

### 3.1.3 Supressão de anticorpos

Os anticorpos que não foram estimulados na fase anterior são eliminados, ou seja, não integram a população intermediária.

### 3.1.4 Expansão Clonal

As cópias de cada anticorpo selecionado para mutação são avaliadas. Vale ressaltar que as cópias são feitas antes da mutação, sendo portanto cópias idênticas do anticorpo pai. Além disso, para cada anticorpo estimulado só é permitida uma única cópia. Tendo em vista que um mesmo anticorpo pode ser estimulado por antígenos diferentes, a clonagem é feita na direção do primeiro antígeno a estimulá-lo. O operador de clonagem é definido exatamente como o operador de mutação apresentado na equação 2.2.

A clonagem é feita sobre a cópia do anticorpo pai, e acontece se e somente se a distância entre o anticorpo e o antígeno for maior do que o raio de supressão do anticorpo. Caso ocorra a clonagem, o clone é incorporado à população intermediária.

Em resumo, um clone é uma cópia mutada, que pode ou não ocorrer, de um anticorpo estimulado na Maturação de Afinidade.

A Figura 3.1 ilustra de forma compacta as etapas de maturação de afinidade e expansão clonal, mostrando como a população intermediária de anticorpos é construída

durante uma geração. Quando um anticorpo é selecionado, antes que ele sofra mutação é feita uma cópia dele para, possivelmente, sofrer clonagem posteriormente. O anticorpo mutado e o anticorpo original (caso este último já não esteja na população intermediária) são inseridos na população intermediária. Caso ocorra a clonagem, o anticorpo resultante também é inserido na população intermediária.

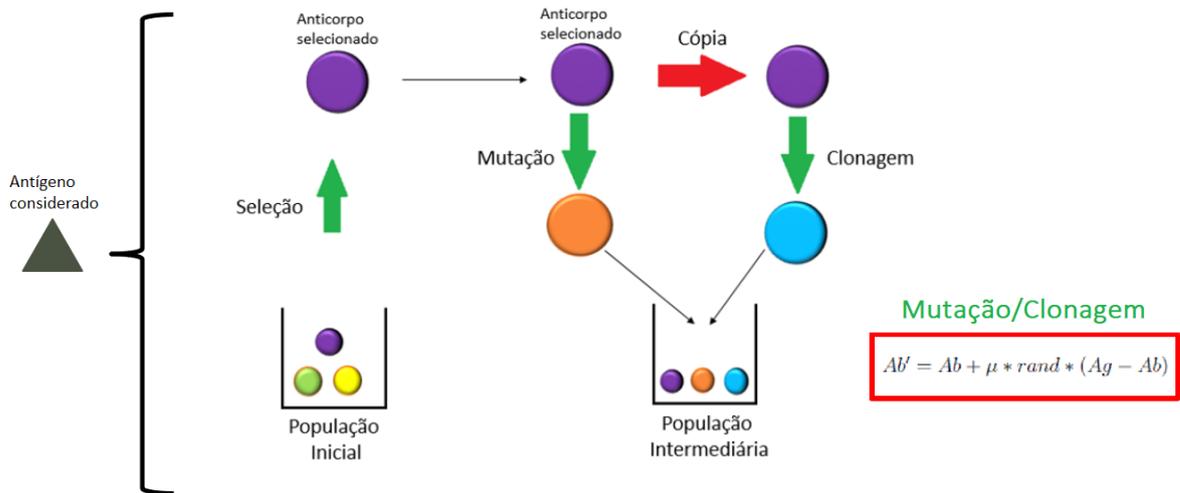


Figura 3.1: Maturação de afinidade e expansão clonal

### 3.1.5 Supressão de Rede

A supressão de rede consiste em eliminar da população intermediária anticorpos que se reconhecem. Para isso, é necessário atualizar o raio  $R$  de cada anticorpo. A densidade de cada anticorpo é definida como o número de antígenos que ele reconhece, ou seja, quantidade de antígenos localizados na região delimitada pelo raio de vizinhança  $E$  do anticorpo. O raio do anticorpo é então atualizado em função da sua densidade, de acordo com a equação 3.2. Na Figura 3.2, mostram-se, para o exemplo dado, o raio de vizinhança  $E$  e o raio do anticorpo calculado pela equação 3.2.

$$R_{ab} = r * (den_{max}/den_{ab})^{(1/dim)} \quad (3.2)$$

onde  $R_{ab}$  é o raio do anticorpo  $ab$ ,  $r$  é o raio mínimo (menor valor que  $R_{ab}$  pode assumir),  $den$  é a densidade do anticorpo  $ab$ .

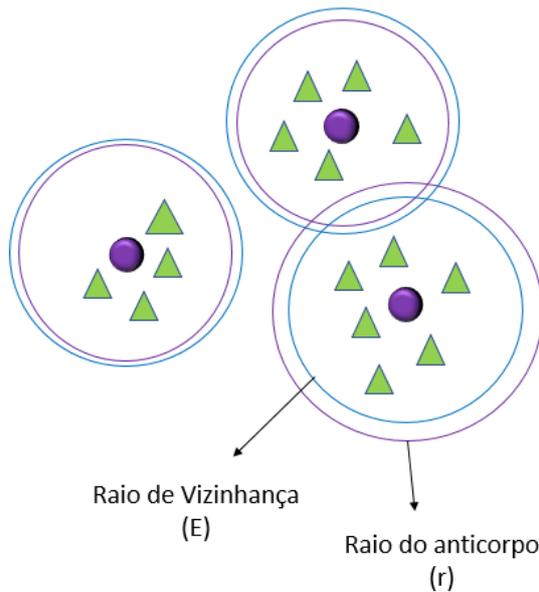
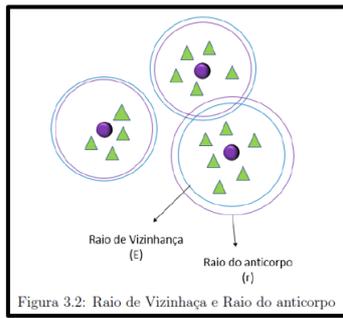


Figura 3.2: Raio de Vizinhaça e Raio do anticorpo

Em seguida, os anticorpos são comparados entre si. Se a distância entre dois anticorpos for menor que o raio de algum deles, aquele que possuir maior raio deve ser eliminado. A Figura 3.3 ilustra como funciona essa etapa. Um dos anticorpos está posicionado dentro da região alcançada por outro anticorpo, sendo assim um anticorpo reconhece o outro e aquele que possui maior raio é eliminado. Note que não há problema algum em dois raios de anticorpos se interceptarem, como acontece com os dois mais à direita na Figura. A eliminação do anticorpo com raio maior deve ser feita a fim de evitar raios muito grandes, que incluam muitos antígenos e regiões vazias do espaço de dados, o que pode levar a uma má representação dos dados.

### 3.1.6 Atualização do raio $E$ da taxa de mutação $\mu$

Para a próxima geração, o raio de vizinhaça  $E$  é atualizado com a média dos raios dos anticorpos. A taxa de mutação atual é multiplicada pela taxa de decaimento caso  $n \geq gen/2$ , onde  $n$  é a geração atual e  $gen$  é o número máximo de gerações.



$$R_{ab} = r * (den_{max}/den_{ab})^{(1/dim)}$$

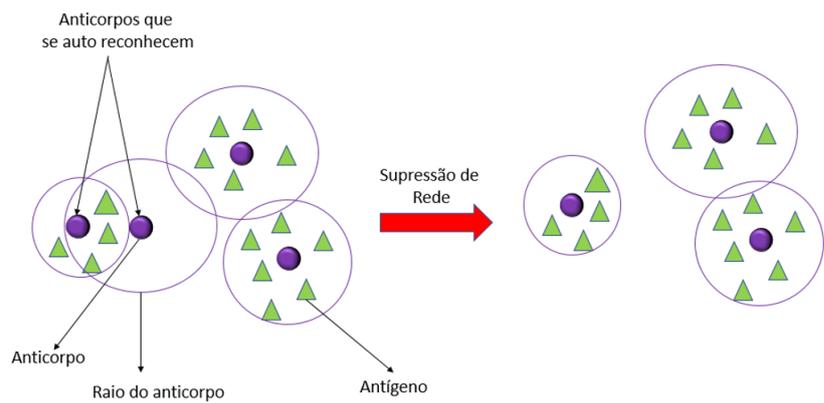


Figura 3.3: Supressão de Rede

## 3.2 Fase 2: Expansão Clonal

A população gerada na fase anterior é submetida à uma heurística ilustrada na Figura 3.4 que verifica a necessidade de duplicação de cada anticorpo com base nos padrões reconhecidos por ele. Cada anticorpo desde o início já tem a informação sobre a classe a qual pertence, uma vez que a população inicial foi construída com um representante de cada classe presente no conjunto de dados e os anticorpos filhos mantêm o rótulo de classe do pai nas operações de clonagem e mutação. Dessa forma, sendo conhecidas também as classes de cada antígeno, calcula-se a razão entre o número de antígenos reconhecidos pelo anticorpo que pertencem à mesma classe deste, e número de antígenos reconhecidos pelo anticorpo que pertencem à classes diferentes. Caso esta razão exceda o valor definido para o Limiar de Reconhecimento de Padrões Não-Próprios (LRPNP) definido, o anticorpo é então duplicado sendo os atributos do novo anticorpo gerado a média dos atributos da classe a qual o original pertence. Se o anticorpo reconhecer padrões de mais de uma classe além da sua, esta análise deve ser feita de forma independente para cada classe envolvida.

### 3.2.1 Atualização dos pesos

A última fase do algoritmo constitui o aprendizado supervisionado da rede de fato. Os pesos, neste caso, os atributos dos anticorpos são atualizados pela equação 3.3

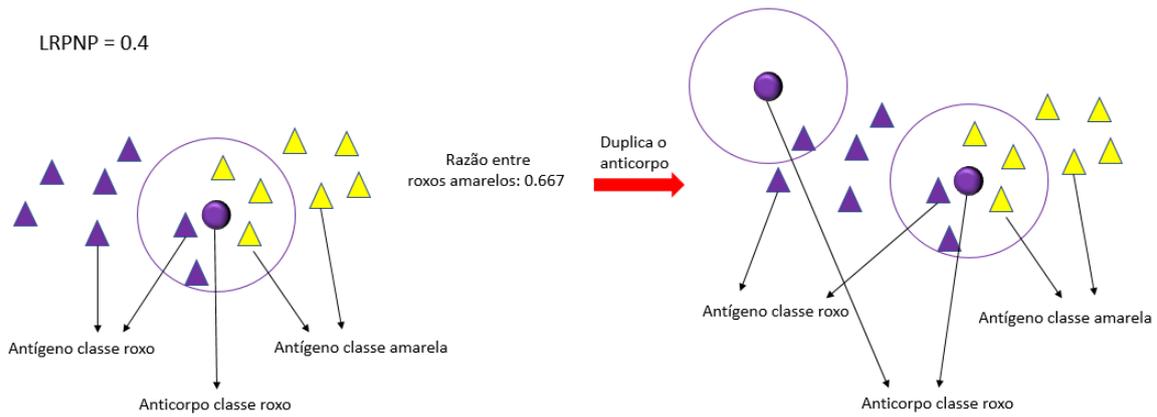


Figura 3.4: Heurística de expansão clonal

$$Ab_k(t+1) = \begin{cases} Ab_k(t) + \alpha * (Ag - Ab_k(t)), & \text{se } Classe(Ab_k) = Classe(Ag) \\ Ab_k(t) - \alpha * (Ag - Ab_k(t)), & \text{caso contrário} \end{cases} \quad (3.3)$$

onde  $Ab_k$  é o anticorpo em análise,  $Ag$  é um antígeno, e  $\alpha$  é a taxa de mutação.

Atualização dos pesos causa uma espécie de perturbação na rede, movendo anticorpos de lugar. Esta perturbação corrige a posição dos anticorpos gerados na fase anterior pela eventual duplicação de alguns anticorpos, que têm seus atributos definidos como a média dos atributos dos dados (antígenos) da classe à qual pertence. Esta é a situação mostrada na Figura 3.5. O anticorpo surgido na Figura 3.4 se aproxima dos antígenos da mesma classe que ele com ajuste dos pesos.

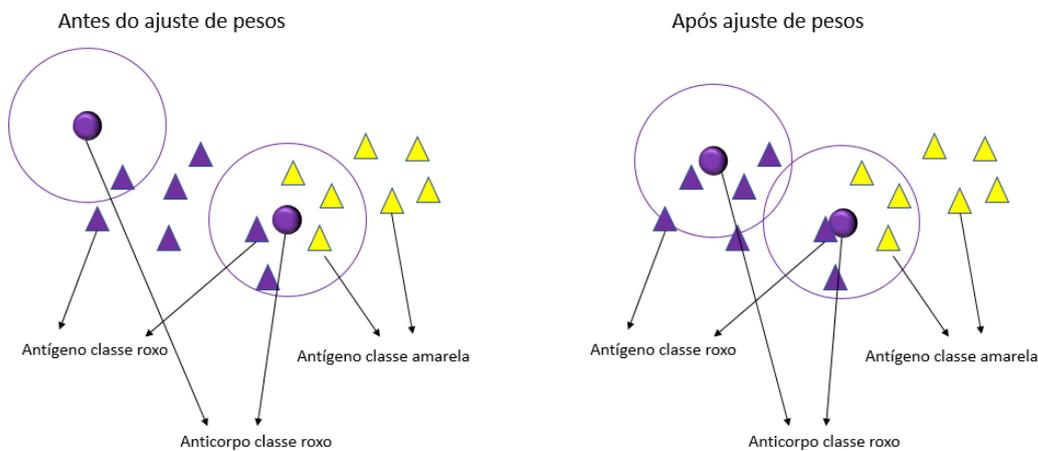


Figura 3.5: Ajuste de pesos da rede

## 4 Estudo de caso

### 4.1 Bases de da Dados

Para realização do estudo de caso, foram escolhidas duas bases de dados reais.

1. Iris:

Uma das bases de dados mais conhecidas na literatura para reconhecimento de padrões, consiste em um conjunto de dados com 150 amostras contendo três classes com 50 instâncias cada. Cada classe corresponde a um tipo de flor Íris: Íris Setosa, Íris Versicolor e Íris Virgínica, sendo que uma delas é linearmente separável das outras duas. A classificação é feita observando os seguintes critérios:

- Comprimento da sépala em cm
- Largura da sépala em cm
- Comprimento da pétala em cm
- Largura da pétala em cm

2. Vinhos:

Base de dados composta por um conjunto de 178 resultados de análises químicas realizadas em amostras de vinhos produzidas em uma mesma região da Itália por três produtores distintos. As análises determinam as quantidades de 13 componentes encontrados nas amostras:

- Álcool
- Ácido málico
- Cinza
- Alcalinidade da cinza
- Magnésio
- Total de fenóis

- Flavonóides
- Fenóis não flavanóides
- Proantocianidinas
- Intensidade de cor
- Hue
- OD280/OD315 de vinhos diluídos
- Proline

Segundo sua descrição em (HETTICH S.AND BLAKE; MERZ, 1998), a base é considerada bem comportada na separação das classes.

Os arquivos com informações sobre as instâncias podem ser encontrados no repositório da *University of California, Irvine* (UCI) (HETTICH S.AND BLAKE; MERZ, 1998), para máquinas de aprendizado, onde estão disponíveis para download gratuito.

## 4.2 Desempenho do classificador proposto

Os resultados mostrados na Tabela 4.1 correspondem aos testes feitos para a base da IRIS e a Tabela 4.2 apresenta os resultados sobre a base de Vinhos. As Tabelas apresentam a Taxa de Classificação Correta (TCC) e o número de anticorpos final (NA) da rede para cada uma das bases, variando os parâmetros de entrada da SAINET implementada, o raio mínimo dos anticorpos ( $r$ ) e o Limiar de Reconhecimento de Próprio e Não-Próprio (LRPNP). Para ambas as bases de dados, foi adotado o número de gerações igual a 40. Foram utilizados 80% das amostras selecionadas aleatoriamente para o treinamento da rede, e os 20% restantes para teste. A população inicial para ambas as bases tem tamanho 3 (um anticorpo para cada classe presente no conjunto de dados). As bases de dados não foram normalizadas antes da classificação. Para a base da Iris, a melhor combinação de parâmetros foi  $r=0.5$  e  $LRPNP=0.8$ . Já para a base de vinhos, dentre os resultados apresentados, a combinação com melhor desempenho foi  $r=0.5$  e  $LRPNP=0.4$ . Foram realizadas 10 execuções do algoritmo para cada combinação dos parâmetros descritas acima. Os testes foram realizados desta forma para seguir o mesmo padrão utilizado por

(KNIDEL, 2006) a fim de facilitar a comparação entre os resultados, exceto pelo valor de LRPNP=0.2 que foi descartado nos testes aqui realizados pela não convergência da rede para este valor.

$r=0.5$			$r=0.6$	
LRPNP	TCC	NA	TCC	NA
0.4	72.55±6.21	16.8±2.78	82.0±12.04	13.3±1.67
0.6	90.33±4.58	14,8±2,4	87.33±7.57	13±2,57
0.8	93.0±2.84	16,3±2,45	90.00±7.30	12,3±2,68

Tabela 4.1: Resultados da aplicação da SAINET à base da Iris

$r=0.5$			$r=0.6$	
LRPNP	TCC	NA	TCC	NA
0.4	92.33±5.38	142±0.0	71.11±4.15	142±0.0
0.6	72.55±6.21	142±0.0	66.65±6.08	142±0.0
0.8	68.38±6.79	142±0.0	70.27±6.34	142±0.0

Tabela 4.2: Resultados da aplicação da SAINET à base de Vinhos

A Tabela 4.3 indica as taxas de compressão de dados para as bases consideradas, analisando o tamanho do conjunto de treinamento e o número de anticorpos da rede. Para estes cálculos, foram tomadas as médias dos números de anticorpos totais obtidos em cada teste feito. Para a base da Iris a taxa de compressão é satisfatória, já para a base de Vinhos não houve compressão uma vez que o conjunto de anticorpos tem o mesmo tamanho do conjunto de treinamento. Sobre esta última, ainda não há uma conclusão sobre o porquê deste número de anticorpos em todas as execuções realizadas nos teste apesar da variação dos parâmetros.

Base de Dados	Conjunto de treinamento	Número de anticorpos	Taxa de compressão
Iris	120	14.42±1.69	87.98%
Vinhos	142	142±0.0	0.0%

Tabela 4.3: Taxa de compressão da SAINET sobre as bases de dados

A Figura 4.1 ilustra a evolução do classificador proposto sobre a base de dados da Iris, escolhida pela facilidade maior de visualização.

## 4.3 Comparação dos resultados com a SAINET original

Por ser uma das bases utilizadas por (KNIDEL, 2006) para seus testes, a Iris foi escolhida para este trabalho a fim de comparar os resultados obtidos neste trabalho com os apresentados por ele. Os testes foram realizados buscando manter os mesmos critérios de avaliação quando possível, salvo as variações entre os algoritmos usados para construção da população de anticorpos e outras adversidades encontradas durante os testes.

De um modo geral, os resultados obtidos se assemelham bastante aos apresentados na tese. Porém, para facilitar a comparação, é necessário pontuar algumas semelhanças e diferenças observadas entre o classificador original e a modificação proposta. A modificação proposta neste trabalho apresentou resultados satisfatórios em comparação com a versão original da SAINET, tanto no que se refere ao tamanho da população de anticorpos, quanto ao desempenho do classificador em si. Apesar dos valores elevados de desvio padrão em comparação com os apresentados por (KNIDEL, 2006) indicando uma oscilação maior nos resultados, os valores médios para a base da Iris de número de anticorpos, taxa de classificação correta e taxa de compressão se mativeram próximo do esperado, mesmo com a diferença de parâmetros entre os algoritmos da ARIA e da aiNet. Na Tabela 4.4 é apresentada uma comparação entre os valores da taxa de classificação correta (TCC), número de anticorpos (NA) e a taxa de compressão (TC) médios obtidos neste trabalho e os obtidos por (KNIDEL, 2006), obtidos de forma análoga aos dados apresentados na Tabela 4.3. Para esta comparação, foram considerados apenas os resultados obtidos por ambos para valores de LRPNP de 0.4 a 0.8.

Autor	TCC	NA	TC
SAINET	93.62±1.97	10.76±3.39	86.3±0.7%
SAINET com Raio Adaptativo	94.1±4.33	14.39±2.96	87.89±2.91%

Tabela 4.4: Comparação entre os resultados obtidos pela SAINET original e a versão modificada sobre a base Iris

A base de Vinhos não foi citada por (KNIDEL, 2006), portanto não há como fazer uma comparação com esta referência. Entretanto, em termos do desempenho do classificador proposto, é possível fazer uma comparação dos resultados obtido utilizando algum

outro classificador conhecido sobre a mesma base de dados. Na Tabela 4.5 é apresentada uma comparação entre os resultados obtidos da aplicação do modelo proposto e os resultados obtidos aplicando o modelo Naive Bayes (LUCCA et al., 2013) com distribuição Multinomial sobre a base de vinhos. O Naive Bayes foi escolhido por ser um modelo amplamente conhecido no âmbito da mineração de dados e por apresentar bons resultados em diversas aplicações. Foram realizadas 10 execuções do algoritmo disponível no módulo *Scikit Learning* (COURNAPEAU, 2007), uma biblioteca de aprendizado de máquinas disponível em várias versões do Python. A divisão dos dados foi feita como descrito na seção 4.2. Embora sejam dois modelos com abordagens bem distintas, o objetivo de ambos é o mesmo: classificar amostras de dados corretamente. Portanto, a comparação resultados da rede com resultados do Naive Bayes não é inválida, apesar dos resultados da rede se mostrarem inferiores nesse caso, tomando a acurácia como referencial para análise.

Modelo	TCC (acurácia)
SAINET modificada	69.73±6.442
Naive Bayes Multinomial	81.71±0.059

Tabela 4.5: Comparação entre os resultados obtidos pela versão modificada da SAINET e os resultados obtidos pelo modelo Naive Bayes sobre a base de Vinhos

## 4.4 Adaptação das redes clusterizadoras para construção da população de anticorpos

Algumas modificações nos algoritmos originais usados na primeira fase da SAINET são feitas para adaptar a rede clusterizadora ao processo de classificação. Estas modificações são destacadas nesta seção a fim de evitar que má interpretação das referências leve a erros que foram cometidos durante o desenvolvimento deste trabalho. A primeira modificação ocorre já na construção da população inicial de anticorpos. Tanto ARIA quanto aiNet utilizam um construtor aleatório e aceitam um tamanho arbitrário para a população inicial, porém o trecho do algoritmo da SAINET correspondente à atuação destas redes indica que a população inicial deve conter um anticorpo que represente cada classe do conjunto de dados, onde os atributos deste sejam a média dos atributos da classe. Isso ocorre porque o

processo de classificação, ao contrário do agrupamento é um processo supervisionado, ou seja, é necessário saber *a priori* quais elementos pertencem a que classes. Neste cenário, caso a população inicial seja construída sem a informação das classes dos anticorpos não é possível inferí-la posteriormente sem interferência direta no algoritmo, o que não é interessante uma vez que o algoritmo deve identificar automaticamente os limites das classes. Além disso, analisar extrinsecamente a população para inferir as classes dos anticorpos é um processo que leva em conta características da base de dados atando o algoritmo à uma aplicação específica e levando à perda de generalidade do mesmo.

A segunda modificação é consequência da primeira: a ausência da árvore geradora mínima. Para o processo de clusterização, a AGM e suas partições são fundamentais para identificação dos *clusters*. Essa informação se faz desnecessária no contexto da classificação apresentada, justamente pelo fato de a quantidade de classes ser definida no pré-processamento dos dados.

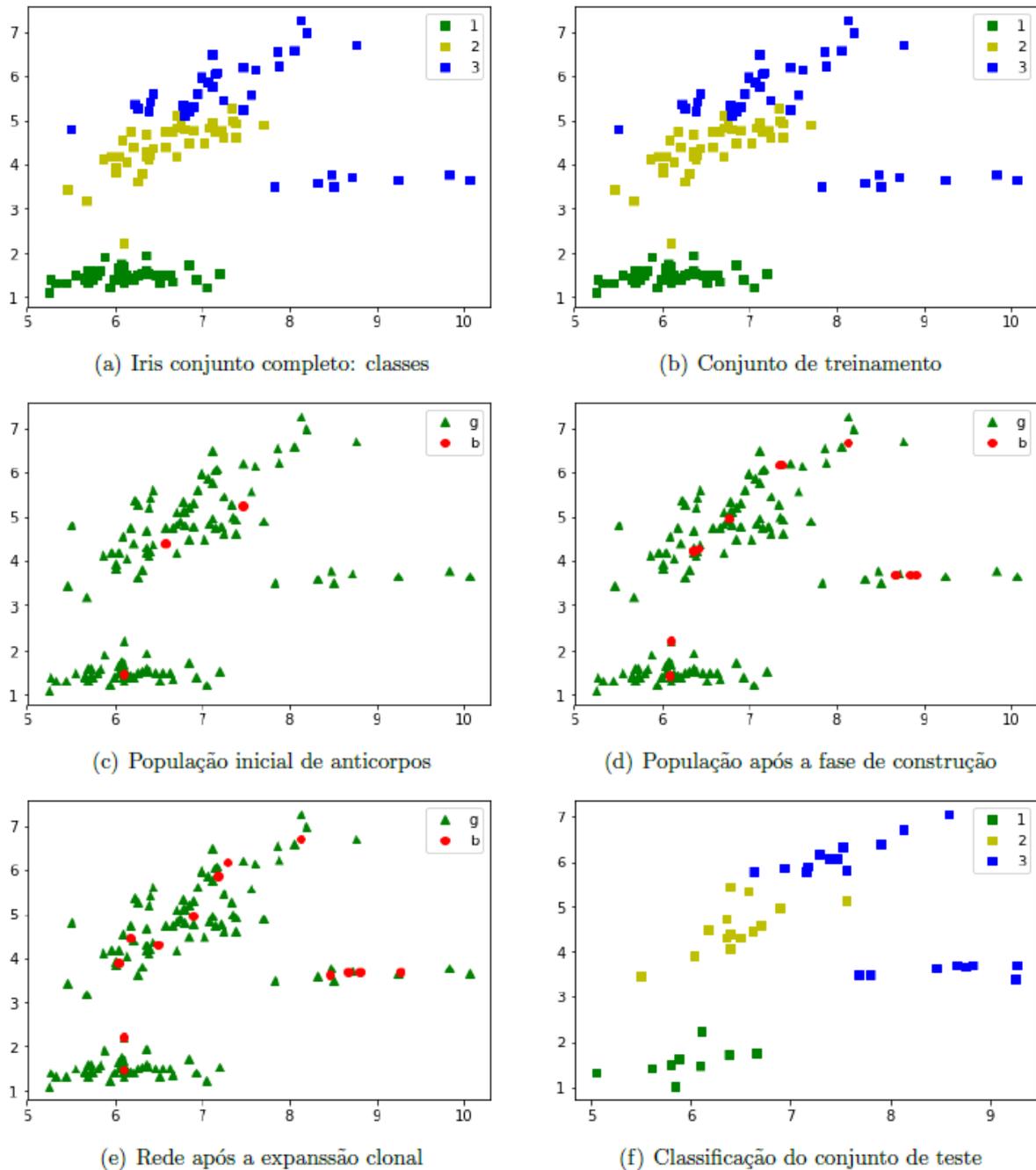


Figura 4.1: Evolução da rede. (a) Conjunto completo da Iris identificando as classes por cores. (b) Conjunto de treinamento com 80% das amostras. (c) População inicial de anticorpos com um anticorpo por classe (em vermelho) em contraste com o conjunto de treinamento (em verde). Cada anticorpo é inicializado com a média dos atributos da classe a qual pertence. (d) População de anticorpos gerada após o término da primeira fase da SAINET (em vermelho). Total de 13 anticorpos. (e) Rede após a segunda fase do algoritmo. Não houve nesse caso necessidade de duplicação de nenhum anticorpo, não havendo assim alteração na população. (f) Aplicação do classificador ao conjunto de teste. Taxa de Classificação Correta: 96,67 %.

## 5 Conclusão e trabalhos futuros

Dentre as várias abordagens populacionais e evolutivas conhecidas dentro da área de Inteligência Computacional para resolver problemas de otimização, a metaheurística escolhida para este trabalho foi um tipo de Sistema Imunológico Artificial baseado na Teoria de Rede Imunológica. Tendo em vista as aplicações possíveis para este tipo de SIA dentro da Mineração de Dados, destacaram-se as tarefas de clusterização e classificação, sendo esta última o foco principal deste trabalho. Uma RIA específica foi escolhida como referência para este trabalho, que tinha como objetivo analisar e comparar os resultados obtidos a partir da substituição da RIA utilizada para construção da população de anticorpos por outra RIA.

ARIA, a rede escolhida para construção da população de anticorpos neste trabalho, apresenta melhores resultados quando utilizada no agrupamento de dados em comparação com a aiNet de acordo com (BEZERRA et al., 2005). Entretanto como a finalidade do uso de qualquer um das redes na parte que lhes cabe na SAINET não está associada ao agrupamento dos dados, não há ganho considerável em termos de desempenho ao utilizar uma ou outra. Porém, vale ressaltar que a implementação da ARIA é mais simples, rápida e menos sensível ao ajuste de parâmetros pelo usuário que a aiNet, indicando uma alternativa para implementação da SAINET a fim de facilitar o estudo das RIAs.

O classificador proposto não se mostra melhor nem pior classificador original no que tange à acurácia da classificação de amostras, usando ambos da mesma abordagem computacional. A comparação entre o modelo proposto e o modelo Naive Bayes, bastante conhecido na área de mineração de dados, mostra que a rede implementada tem desempenho inferior a este, porém não necessariamente o classificador proposto é de fato inferior. Erros foram cometidos ao rodar o algoritmo implementado da SAINET, negligenciando o ajuste dos parâmetros  $r$  e LRPNP para a base de Vinhos. Para que uma comparação mais precisa e verossímil seja feita, é necessário corrigir estes erros.

Alguns detalhes não foram devidamente explorados neste trabalho, como por

exemplo o fato de não haver compressão na representação da base de Vinhos pela população de anticorpos. Tal fato ainda não foi justificado, levantando hipóteses sobre a distribuição dos dados, posicionamento dos anticorpos e o ajuste do parâmetro  $r$ , raio mínimo dos anticorpos, influencia diretamente nos resultados. (AZZOLINI; ZUBEN, 2010) apresenta um estudo sobre a influência deste parâmetro no desempenho da ARIA que não foi levado em consideração neste trabalho. Tal estudo define métricas para determinar automaticamente um valor para  $r$  adequado para cada base de dados utilizada, uma vez que bases diferentes podem demandar ajustes diferentes para  $r$  a fim de obter resultados satisfatórios.

Um outro ponto tido como objetivo que não foi alcançado é a visualização correta dos anticorpos. A visualização ideal para anticorpos e antígenos inclui a representação, na mesma Figura, de anticorpos com seus respectivos raios, como nas Figuras 3.2 a 3.5. Tal feito não foi possível pelo insucesso na plotagem do raio de todos os anticorpos na mesma figura, juntamente com seus centros(anticorpos) e os antígenos alcançados por eles.

Para trabalhos futuros, sugere-se estudar e implementar o ajuste automático de  $r$  e plotar os anticorpos com seus raios na mesma Figura.

## Bibliografia

- ABBAS A. K., L. A. H. P. S. Imunologia celular e molecular. In: *Imunologia Celular e Molecular*. [S.l.]: Elsevier. p. 4–6.
- AZZOLINI, R. P. V. V. A. G.; ZUBEN, F. J. V. Density preservations and vector quantization in immune-inspired algorithm. In: . [S.l.: s.n.], 2010.
- BEZERRA, G. et al. Adaptive radius immune algorithm for data clustering. In: *Adaptive radius immune algorithm for data clustering*. [S.l.: s.n.], 2005. p. 290–303.
- CAMILO, C.; SILVA, J. Mineração de dados: Conceitos, tarefas, métodos e ferramentas. In: . [S.l.: s.n.].
- CASTRO, L. N. de; ZUBEN, F. J. V. ainet: An artificial immune network for data analysis. In: *In Abbass, H.; Sarker, R. A.; Newton, C. S. (eds), Data Mining: a Heuristic Approach*. [S.l.]: Idea Group Publishing, 2002. p. 231–259.
- COURNAPEAU, D. Scikit learn: Machine learning in python. In: . [S.l.: s.n.], 2007.
- FIGUEREDO, G.; BERNARDINO, H.; BARBOSA, H. Introdução aos sistemas imunológicos artificiais. In: *Introdução aos Sistemas Imunológicos Artificiais*. [S.l.: s.n.], 2013.
- FRANÇA, F. O. de et al. Conceptual and practical aspects of the ainet family of algorithms. In: *Conceptual and Practical Aspects of the aiNet Family of Algorithms*. [S.l.: s.n.], 2010. p. 1–35.
- HETTICH S.AND BLAKE, C. L.; MERZ, C. J. Uci repository of machine learning databases. In: . [S.l.: s.n.], 1998.
- JERNE, N. K. Towards a network theory of the immune system. In: *Towards a network theory of the immune system*. [S.l.: s.n.], 1974. p. 378–389.
- KNIDEL, H. Extensões e aplicações de redes neuro-imunológicas. In: . [S.l.]: Universidade Federal de Campinas, 2006. p. 39–41,62–65.
- KOHONEN, T. . Self-organizing maps. In: *Self-organizing maps*. [S.l.]: Springer, 2010.
- LOPES, G. A. de M. imarag: Um algoritmo baseado em sistemas imunológicos artificiais para treinamento de redes mlp. In: *iMaRag: Um algoritmo baseado em Sistemas Imunológicos Artificiais para treinamento de redes MLP*. [S.l.]: Universidade de Pernambuco, 2010.
- LUCCA, G. et al. Uma implementação do algoritmo naïve bayes para classificação de texto. In: . [S.l.: s.n.], 2013.
- OLIVEIRA, A. J. de. Esses 5 livros acadêmicos foram eleitos os mais influentes de todos os tempos. In: . [S.l.]: Editora Globo, 2015.
- TALBI, E. Metaheuristics: From design to implementation. In: *Metaheuristics: From Design to Implementation*. [S.l.]: Wiley, 2009. p. 4–6.