Universidade Federal do Ceará
Departamento de Engenharia de Teleinformática
Programa de Pós-Graduação em Engenharia de Teleinformática

César Lincoln Cavalcante Mattos

Comitês de Classificadores Baseados nas Redes SOM e Fuzzy ART com Sintonia de Parâmetros e Seleção de Atributos via Metaheurísticas Evolucionárias

Fortaleza – Ceará
Novembro 2011

César Lincoln Cavalcante Mattos

Comitês de Classificadores Baseados nas Redes SOM e Fuzzy ART com Sintonia de Parâmetros e Seleção de Atributos via Metaheurísticas Evolucionárias

Dissertação de Mestrado apresentada à Coordenação do Programa de Pós-Graduação em Engenharia de Teleinformática da Universidade Federal do Ceará como parte dos requisitos para obtenção do grau de ***Mestre em Engenharia de Teleinformática****.*

Área de Concentração: Sinais e Sistemas

Orientador : Prof. Dr. Guilherme de Alencar Barreto

Fortaleza – Ceará
Novembro 2011

Resumo

O paradigma de classificação baseada em comitês tem recebido considerável atenção na literatura científica em anos recentes. Neste contexto, redes neurais supervisionadas têm sido a escolha mais comum para compor os classificadores base dos comitês. Esta dissertação tem a intenção de projetar e avaliar comitês de classificadores obtidos através de modificações impostas a algoritmos de aprendizado não-supervisionado, tais como as redes Fuzzy ART e SOM, dando origem, respectivamente, às arquiteturas ARTIE (*ART in Ensembles*) e MUSCLE (*Multiple SOM Classifiers in Ensembles*). A sintonia dos parâmetros e a seleção dos atributos das redes neurais que compõem as arquiteturas ARTIE e MUSCLE foram tratados por otimização metaheurística, a partir da proposição do algoritmo I-HPSO (*Improved Hybrid Particles Swarm Optimization*). As arquiteturas ARTIE e MUSCLE foram avaliadas e comparadas com comitês baseados nas redes Fuzzy ARTMAP, LVQ e ELM em 12 conjuntos de dados reais. Os resultados obtidos indicam que as arquiteturas propostas apresentam desempenhos superiores aos dos comitês baseados em redes neurais supervisionadas.

Palavras-chaves: Redes Neurais Competitivas, Redes Fuzzy ART, Redes SOM, Comitês de Classificadores, Algoritmos Metaheurísticos

Abstract

The ensemble-based classification paradigm has received considerable attention in scientific literature in recent years. In this context, supervised neural networks have been the most common choice for ensembles' base classifiers. This dissertation has the intention of projecting and evaluating ensembles of classifiers built through modifications on non-supervised learning algorithms, such as the Fuzzy ART and SOM networks, originating, respectively, the ARTIE (ART *in Ensembles*) and MUSCLE (*Multiple* SOM *Classifiers in Ensembles*) models. The parameters' tunning and the feature selection of the neural networks which compose the ARTIE and MUSCLE models were tackled by metaheuristic optimization, with the proposal of the I-HPSO (*Improved Hybrid Particles Swarm Optimization*) algorithm. The ARTIE and MUSCLE models were evaluated and compared with ensembles based on Fuzzy ARTMAP, LVQ and ELM networks in 12 real world datasets. The obtained results indicate that the proposed models present performance superior to the ensembles of supervised neural networks.

Keywords: Competitive Neural Networks, Fuzzy ART Network, SOM Network, Ensembles of Classifiers, Metaheuristic Algorithms

Dedico este trabalho aos meus pais, Fernando Lincoln e Carmen,
pelo apoio indispensável e incondicional.

Agradecimentos

Aos meus pais, Fernando Lincoln e Carmen, cujos ensinamentos preciosos me permitiram ter a paciência e dedicação necessárias para mais esta realização,

À minha irmã, Fernanda, e aos amigos extra-universidade, por me permitirem momentos de lazer fundamentais para a execução deste trabalho,

Aos estudantes de pós-graduação em Engenharia de Teleinformática, pelos momentos de estudo e descontração,

Ao Professor Guilherme de Alencar Barreto, pela confiança, incentivo e dedicação constante durante todo o período de orientação, sendo apoio fundamental para a realização desta dissertação,

Aos demais professores do Departamento de Engenharia de Teleinformática, pelas importantes discussões durante o meu curso de mestrado,

À Universidade Federal do Ceará, por permitir este importante passo na minha carreira profissional.

A razão pode responder perguntas, mas a imaginação tem que perguntá-las.
Ralph Gerard

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Algoritmos

Lista de Símbolos

L	*Número de classificadores do comitê*
$g(\cdot)$	*Operador de agregação do comitê de classificadores*
N	*Número de amostras disponíveis para treinamento*
$\mathbf{a}$	*Vetor de atributos que representa um padrão*
$\mathbf{x}_i$	*i-ésimo vetor de entrada no classificador*
$\mathbf{y}_i$	*Classe do vetor de entrada* x_i
$\mathbf{W}$	*Matriz de pesos neuronais*
$\mathbf{w}_i$	*i-ésimo vetor de pesos neuronais*
ρ	*Parâmetro de vigilância*
β	*Parâmetro de escolha*
η	*Parâmetro de aprendizado*
η_0	*Valor inicial do parâmetro* η
η_f	*Valor final do parâmetro* η
$\wedge$	*Operador de mínimo fuzzy*
N_w	*Número de neurônios da rede*
N_c	*Número de neurônios por classe da rede LVQ*
$\mathbf{t}$	*Vetor de aptidões dos neurônios das redes ART*
$\mathcal{X}$	*Espaço de entrada contínuo da rede SOM*
$\mathcal{Y}$	*Espaço de saída discreto da rede SOM*
$P_1 \times P_2$	*Dimensões de uma rede SOM bidimensional*
$h(\cdot)$	*Função de vizinhança da rede SOM*
r_i	*Posição do i-ésimo neurônio na rede SOM*
σ	*Parâmetro de largura de vizinhança da rede SOM*
σ_0	*Valor inicial do parâmetro* σ
σ_f	*Valor final do parâmetro* σ
n	*Iteração de treinamento atual*
n_{MAX}	*Número máximo de iterações de treinamento*
C	*Número de classes do problema analisado*
$\mathbf{W}$	*Matriz de pesos da camada Inter-MAP da rede Fuzzy ARTMAP*
$\mathbf{w}_i$	*i-ésimo vetor de pesos da camada Inter-MAP da rede Fuzzy ARTMAP*
$f(\cdot)$	*Função objetivo*
$\mathbf{x}_i$	*Vetor de posição da i-ésima partícula* (PSO)

$\mathbf{v}_i$	*Vetor de velocidade da i-ésima partícula*
$\mathbf{p}_i$	*Vetor de melhor posição histórica da i-ésima partícula*
$\mathbf{pl}_k$	*Vetor de melhor posição histórica da k-ésima vizinhança de partículas*
$\mathbf{x}_{min}$	*Vetor dos menores valores possíveis para as variáveis de uma solução*
$\mathbf{x}_{max}$	*Vetor dos os maiores valores possíveis para as variáveis de uma solução*
c_1 *e* c_2	*Coeficientes aceleradores*
χ	*Fator de constrição*
η_{SA}	*Passo de controle do Recozimento Simulado*
t	*Parâmetro de temperatura do Recozimento Simulado*
λ	*Taxa de recozimento*
L_{PSO}	*Número de iterações da etapa PSO do algoritmo I-HPSO*
L_{SA}	*Número de iterações da etapa de Recozimento Simulado do algoritmo I-HPSO*

Lista de Siglas

ART (*Adaptive Resonance Theory*)

ARTIE (*ART in Ensembles*)

ELM (*Extreme Learning Machine*)

HPSO (*Hybrid Particles Swarm Optimization*)

I-HPSO (*Improved Hybrid Particles Swarm Optimization*)

LVQ (*Learning Vector Quantization*)

MUSCLE (*Multiple SOM Classifiers in Ensembles*)

PSO (*Particles Swarm Optimization*)

RNA (Redes Neurais Artificiais)

SA (*Simulated Annealing*)

SOM (*Self-Organizing Maps*)

Capítulo 1

Introdução

A classificação de amostras desconhecidas é um problema recorrente na pesquisa científica. Seja no diagnóstico de doenças (ROCHA NETO; BARRETO, 2009), no reconhecimento de faces (MONTEIRO, 2009), no controle de qualidade (NIEMINEN *et al.*, 2011), na detecção de invasões a sistemas computacionais (PILLAY, 2011), dentre outras, o ato de reconhecer e categorizar diferentes objetos ou situações é necessário.

A tarefa de reconhecimento de padrões pode ser definida informalmente como o processo pelo qual a uma nova amostra é atribuída uma dentre um número pré-definido de classes (HAYKIN, 2008). Enquanto seres humanos costumam ter facilidade em reconhecer diferentes padrões (e.g. faces, sons, objetos, etc.), o desenvolvimento de métodos computacionais que sejam, pelos menos em parte, capazes de tal feito é objetivo contínuo de muitos estudos.

Uma abordagem especialmente interessante consiste em modelar certas características do cérebro humano que lhe permite realizar tarefas complexas, como o reconhecimento de padrões. Esses modelos, chamados genericamente de Redes Neurais Artificiais (RNA), são capazes de adquirir e armazenar conhecimento através de um processo de aprendizado (HAYKIN, 2008).

Algumas características importantes das RNAs podem ser destacadas (SILVA; SPATTI; FLAUZINO, 2010):

- **Adaptação por experiência**: os parâmetros internos da rede são ajustados a partir da apresentação de exemplos (e.g. padrões de treinamento);

- **Habilidade de generalização**: em geral, RNAs buscam otimizar o erro empírico, ou seja, o erro no conjunto de treinamento. Apesar de não haver garantia de bom desempenho em um conjunto de teste, constata-se que RNAs são capazes de estimar soluções a partir da generalização do conhecimento adquirido;

- **Organização dos dados**: através da organização interna de sua arquitetura, uma RNA é capaz de reconhecer dados com padrões semelhantes.

Em problemas de classificação de padrões, deseja-se que a RNA consiga estimar o rótulo de uma amostra desconhecida a partir da generalização do conhecimento adquirido durante uma fase anterior de treinamento, realizada a partir de uma quantidade finita de padrões de treinamento. Dessa maneira, o projeto de um classificador neural envolve a determinação de uma arquitetura de RNA, a escolha de um algoritmo de treinamento e a disposição de um conjunto de padrões para treinamento.

Assim como um grupo de especialistas (e.g. uma junta médica) pode se reunir para, a partir de diferente pontos de vista, resolver uma questão complexa, classificadores de padrões também podem ser agrupados em comitês. Dessa maneira, surgem sistemas cuja resposta é composta pela associação de decisões de múltiplos classificadores, com a intenção de obter menores erros de generalização.

Nos últimos anos diferentes estratégias de construção de comitês de classificadores tem sido alvo de pesquisas diversas (GUNES; M.; PETITRENAUD, 2010). É de interesse desta dissertação contribuir com os estudos nesse tópico.

1.1 Motivação

Classificadores neurais são tipicamente obtidos a partir de técnicas de aprendizagem supervisionada, tais como redes MLP (*Multi-Layer Perceptron*) e RBF (*Radial Basis Function*) (HAYKIN, 2008), redes Fuzzy ARTMAP (CARPENTER; GROSSBERG; REYNOLDS, 1991), redes LVQ (*Learning Vector Quantization*) (KOHONEN, 1988a) e, mais recentemente, redes ELM (*Extreme Learning Machine*) (HUANG; ZHU; SIEW, 2004). Uma característica comum a esses métodos é a necessidade do conhecimento das classes dos exemplos usados na fase de treinamento.

Algoritmos de aprendizagem não-supervisionada, como rede SOM (*Self-Organizing Maps*) (KOHONEN, 1982), rede ART2 (*Adaptive Resonance*

Theory 2) (CARPENTER; GROSSBERG, 1987c) e rede Fuzzy ART (CARPENTER; GROSSBERG; ROSEN, 1991), costumam ser aplicados em problemas de agrupamento de dados, quantização vetorial e redução de dimensionalidade. Entretanto, é possível usar tais técnicas em tarefas de classificação de padrões a partir de modificações nos seus processos de aprendizagem. Em Monteiro *et al.* (2006) são listadas alguns desses métodos, com ênfase na rede SOM. Existe a possibilidade de aplicação dos mesmos métodos em redes não supervisionadas da família ART, assunto ainda não explorado na literatura.

As redes Fuzzy ART e SOM compartilham com as redes Fuzzy ARTMAP e LVQ o paradigma da aprendizagem competitiva, em que neurônios da rede se especializam em representar determinados grupos de padrões (HAYKIN, 2008). É de interesse desta dissertação explorar o uso de redes desse tipo em problemas de classificação de padrões, mais especificamente na composição de comitês, uma vez que este assunto não tem sido amplamente abordado na literatura especializada.

RNAs de uma maneira geral apresentam um conjunto de parâmetros a serem determinados antes da etapa de treinamento. É comum obter valores para tais parâmetros a partir de métodos de busca exaustiva, como o chamado *grid search* (LIN; CHANG; HSU, 2004). Pode-se ainda abordar o problema de escolha desses parâmetros como um problema de otimização do algoritmo de treinamento. Uma alternativa viável é o uso de técnicas de otimização estocástica (LØVBJERG, 2002). Métodos de otimização metaheurísticos, tais como Otimização por Enxame de Partículas (PSO, *Particles Swarm Optimization*) (KENNEDY; EBERHART, 1995), Algoritmos Genéticos (AG) (HOLLAND, 1975), Recozimento Simulado (SA, *Simulated Annealing*) (KIRKPATRICK *et al.*, 1983) e Otimização por Colônia de Formigas (ACO, *Ant Colony Optimization*) (DORIGO, 1992), são exemplos dessa última categoria de ferramenta, que fará parte da metodologia desta dissertação.

1.2 Objetivos

O objetivo geral desta dissertação, assim como seus objetivos específicos, são apresentados nesta seção.

1.2.1 Objetivo Geral

O principal objetivo desta dissertação consiste em construir e avaliar comitês de classificadores de padrões obtidos a partir de redes neurais competitivas

supervisionadas e não-supervisionadas. Incluído nesse objetivo está uma abordagem metaheurística para a tarefa de otimização dos parâmetros dos classificadores e seleção de atributos usados.

1.2.2 Objetivos Específicos

Os objetivos específicos desta dissertação estão listados a seguir:

1 Construir comitês de classificadores baseados nas redes neurais supervisionadas Fuzzy ARTMAP, LVQ e ELM.

2 Desenvolver um modelo de comitê de classificadores baseados na rede Fuzzy ART.

3 Desenvolver um modelo de comitê de classificadores baseados na rede SOM.

4 Propor um novo algoritmo de otimização híbrido metaheurístico para sintonia de parâmetros e seleção de atributos dos classificadores de padrões a serem usados nos comitês de classificadores.

5 Comparar os comitês de classificadores avaliados através de testes estatísticos.

1.3 Publicações

Os resultados parciais do presente trabalho foram reunidos nos artigos científicos listados a seguir.

- *"ARTIE and MUSCLE Models: Building Ensemble Classifiers from Fuzzy ART and SOM Networks"*, submetido ao periódico *Neural Computing & Applications* e aceito para publicação.
- *"On the Use of Fuzzy ART and SOM Networks in Ensemble Classifiers: A Performance Comparison"*, apresentado no VIII Encontro Nacional de Inteligência Artificial.

Sobre outra aplicação do algoritmo metaheurístico híbrido proposto, foram submetidos ainda os artigos científicos a seguir.

- *"An Improved Hybrid Particle Swarm Optimization Algorithm Applied to Economic Modeling of Radio Resource Management"*, submetido ao periódico *Memetic Computing Journal*, aguardando confirmação de aceitação.

- *"Economic Modeling of Radio Resource Management: A Novel Metaheuristic Approach"*, apresentado no XXIX Simpósio Brasileiro de Telecomunicações.

1.4 Organização da Dissertação

O restante desta dissertação está organizado da seguinte forma:

- No Capítulo 2 são descritos os passos envolvidos no processo de construção de comitês de classificadores.

- O Capítulo 3 faz um resumo dos algoritmos de aprendizado das redes neurais competitivas Fuzzy ART, Fuzzy ARTMAP, SOM e LVQ.

- O Capítulo 4 introduz duas novas arquiteturas de comitês de classificadores, uma baseada na rede SOM e outra basada na rede Fuzzy ART.

- O Capítulo 5 resume as operações e conceitos que suportam a técnica de otimização metaheurística baseada no algoritmo PSO, voltada para sintonia dos parâmetros e seleção de atributos das redes que compõem os comitês de classificadores propostos.

- O Capítulo 6 descreve as metodologias usadas para construção e avaliação dos comitês de classificadores a partir das redes neurais listadas nos Capítulos 3 e 4 e da técnica de otimização metaheurística introduzida no Capítulo 5. Também são apresentados neste capítulo os métodos de avaliação estatística a serem usados.

- No Capítulo 7 são apresentados e discutidos os resultados obtidos a partir de simulações computacionais.

- No Capítulo 8 são feitas as conclusões finais e perspectivas para futuros trabalhos.

Capítulo 2

Fundamentos de Comitês de Classificadores

Um comitê de classificadores de padrões pode ser analisado como uma coleção de classificadores individuais que apresentam diversidade em sua construção e que conduzem a uma maior capacidade de generalização do que quando trabalhando em separado (DIETTERICH, 2000). É sabido ainda que comitês garantem erro quadrático médio e variância menores ou iguais aos classificadores que o compõem (KROGH; VEDELSBY, 1995; HAYKIN, 2008), sendo portanto tema relevante para a área de reconhecimento de padrões.

Neste capítulo são discutidos conceitos fundamentais para a utilização de comitês, mais especificamente a combinação de classificadores, o papel da diversidade em um comitê e a caracterização dos classificadores base que o compõe quanto ao paradigma de aprendizado usado.

2.1 Combinação de resultados de classificadores

Seja uma coleção de L classificadores de padrões, ao utilizá-los em comitês é preciso definir também um método para combinar as respostas dos classificadores isolados, denominados neste trabalho *classificadores base*. Seja um determinado padrão de teste, caracterizado pelo vetor de entrada $\mathbf{x} \in \mathbb{R}^D$. Seja ainda o código da classe, ou rótulo, inferida pelo classificador base l para essa entrada definida por $Y_l^*(\mathbf{x})$. A combinação de L resultados como este pode ser descrita matematicamente

por (KUNCHEVA; JAIN, 2000)

$$Y^*(\mathbf{x}) = g(Y_1^*(\mathbf{x}), Y_2^*(\mathbf{x}) \cdots, Y_L^*(\mathbf{x})), \tag{2.1}$$

em que $Y^*(\mathbf{x})$ é a saída do comitê formado pelos L classificadores base considerados e $g(\cdot)$ é um operador de agregação.

Uma das formas mais simples de realizar a combinação anterior é por *votação* (KOTSIANTIS; ZAHARAKIS; PINTELAS, 2006). No caso de votação majoritária simples, cada classificador base tem direito a votar em uma classe, sendo a classe mais votada a escolhida para representar a amostra na entrada do comitê.

Seja $\mathbf{y}_l^*(\mathbf{x}) \in \mathbb{R}^C$ o vetor de saída do l-ésimo classificador base para uma entrada $\mathbf{x}$, em que C é o número de classes possíveis e o vetor $\mathbf{y}_l^*(\mathbf{x})$ é binário e possui somente uma componente não-nula, em geral, de valor 1.

Seja ainda $\mathbf{v}(\mathbf{x}) \in \mathbb{R}^C$ = um vetor de votos em que cada elemento $v_k(\mathbf{x}), k = 1, 2, \ldots, C$ possui um valor proporcional à frequência de escolha da classe k entre os classificadores base. O vetor $\mathbf{v}(\mathbf{x})$ pode ser calculado pela Equação (2.2) (ROCHA NETO; BARRETO, 2009).

$$\mathbf{v}(\mathbf{x}) = \sum_{l=1}^{L} \mathbf{y}_l^*(\mathbf{x}) = [v_1(\mathbf{x}), \cdots, v_C(\mathbf{x})]^T. \tag{2.2}$$

Finalmente, a classe $Y^*(\mathbf{x})$ inferida pelo comitê para a amostra $\mathbf{x}$ será dada por

$$Y^*(\mathbf{x}) = \arg \max_{k=1,2,\ldots,C} \{v_k(\mathbf{x})\}. \tag{2.3}$$

Este sistema pode ainda ser aplicado de forma ponderada, em que cada classificador base recebe antes da votação um peso proporcional ao seu nível de confiança, determinado, por exemplo, pelo seu desempenho em um conjunto de validação (KOTSIANTIS; ZAHARAKIS; PINTELAS, 2006). Nesse caso, o vetor de votos seria dados por $\mathbf{v}(\mathbf{x}) = \sum_{l=1}^{L} w_l \mathbf{y}_l^*(\mathbf{x})$, em que w_l é o peso dado ao l-ésimo classificador.

Alguns autores investigam formas de combinar não-linearmente as saídas dos classificadores base, utilizando para tanto um algoritmo de aprendizagem adicional, denominado método de meta-aprendizagem, tais como algoritmos de cascateamento

(GAMA; BRAZDIL, 2000) e árvores de meta-decisão (TODOROVSKI; DŽEROSKI, 2003).

A escolha de qual método de combinação é o mais adequado depende da metodologia de formação do comitê de classificadores, como por exemplo, o método escolhido para promover diversidade entre os classificadores base, como será visto na próxima sessão.

2.2 Diversividade em comitês de classificadores

Caso os classificadores base da Equação (2.1) fossem todos idênticos, i.e. apresentassem as mesmas saídas para um dado conjunto de entradas, o comitê resultante não teria qualquer incremento na sua capacidade de generalização em relação aos classificadores que o compõe. Por esse motivo, métodos de formação de comitês de classificadores buscam gerar diversidade nos classificadores base (ZHOU; WU; TANG, 2002).

Uma primeira abordagem consiste em construir diferentes conjuntos de treinamento para cada um dos classificadores base. Esta estratégia é explorada por algoritmos populares, como Bagging (*Bootstrap Aggregating*) (BREIMAN, 1996) e Boosting (FREUND; SCHAPIRE, 1995; SCHWENK; BENGIO, 2000).

A estrategia Bagging cria L subconjuntos de treinamento ao amostrar aleatoriamente, com reposição, exemplos de um conjunto de treinamento original. Por causa da reposição durante a amostragem, os subconjuntos formados podem conter exemplos duplicados e omissões de exemplos. Para subconjuntos de tamanho N, amostrados a partir de um conjunto de treinamento também com N exemplos, tem-se a probabilidade de $\left(\frac{N-1}{N}\right)^N$ para que uma determinada amostrada não seja selecionada em um dos subconjuntos. Para N suficientemente grande, esta probabilidade pode ser aproximada para $\left(\frac{N-1}{N}\right)^N \approx e^{-1} \approx 0,368$.

Assim como a estratégia Bagging, o objetivo das estratégias de Boosting também é gerar subconjuntos de treinamento diversos. Entretanto, nos métodos de Boosting esse procedimento é feito de formal serial, em que um classificador é treinado por vez. O algoritmo AdaBoost (*Adaptive Boosting*) (FREUND; SCHAPIRE, 1995), por exemplo, mantém um conjunto de pesos para cada um dos exemplos no conjunto de treinamento original. Esses pesos são incrementados no caso de classificação incorreta ou decrementados, no caso de classificação correta (DIETTERICH,

2000). Dessa forma, o próximo classificador é treinado ressaltando-se os exemplos incorretamente classificados pelos classificadores anteriores.

Existem diferenças substanciais entre as estratégias Bagging e Boosting. A diferença mais imediata é a capacidade de se treinar múltiplos classificadores em paralelo através de Bagging, enquanto ao aplicar Boosting o treinamento deve ser sequencial.

Nos experimentos realizados em Bauer e Kohavi (1999) chega-se à conclusão que, na presença de ruído, técnicas de Boosting são inadequadas. Essa característica é previsível, pois durante a aplicação de Boosting, os exemplos incorretamente classificados são ressaltados, buscando a minimização do erro de classificação durante a etapa de treinamento. Conclui-se ainda em Bauer e Kohavi (1999) que, apesar de técnicas de Boosting serem mais efetivas que Bagging (em média) em reduzir o erro de generalização, para conjuntos de dados ruidosos, o desempenho da estratégia Boosting é degradado em relação a um único classificador. Por outro lado, Bagging se mostrou efetivo em todos os conjuntos de dados estudados em Bauer e Kohavi (1999).

A presente dissertação pretende aplicar comitês de classificadores a problemas diversos de classificação de padrões. Dessa forma, buscando evitar degradação de desempenho na presença de ruído, esta dissertação utilizará Bagging como método promoção de diversidade em comitês de classificadores. A estratégia Bagging com decisão tomada por voto majoritário simples está descrito no Algoritmo 2.1.

Existem ainda outras técnicas para promover diversidade em comitês de classificadores. Uma possível abordagem é diversificar o processo de aprendizagem dos classificadores base ao aplicar diferentes conjuntos de parâmetros, pesos iniciais (no caso de redes neurais) ou até mesmo usar topologias diferentes (MACLIN; SHAVLIK, 1995). Outra possibilidade é a escolha de diferentes subconjuntos de atributos dos exemplos disponíveis para treinamento para cada classificador base (TSYMBAL; PECHENIZKIY; CUNNINGHAM, 2005).

2.3 Revisão Bibliográfica

Dois principais requisitos para a formação de comitês de classificadores eficientes são que seus classificadores base sejam *instáveis* e *fracos* (*weak learners*) (HANSEN; SALAMON, 2002). Um classificador é considerado instável se pequenas variações

Algoritmo 2.1 Algoritmo Bagging com voto majoritário simples.

Constantes

L: número de classificadores base

D: dimensão dos padrões de entrada

C: número de classes do problema

N: número de amostras disponíveis para treinamento

Q: número de amostras disponíveis para teste

Entradas

$\mathbf{X} = [\mathbf{x}_1\ \mathbf{x}_2\ \cdots\ \mathbf{x}_N]$: matriz de padrões de treinamento (classe conhecida), dimensão $D \times N$

$\mathbf{A} = [\mathbf{a}_1\ \mathbf{a}_2\ \cdots\ \mathbf{a}_Q]$: matriz de padrões de teste (classe desconhecida), dimensão $D \times Q$

Algoritmo

1. Para cada classificador base l **(**$l = 1, 2, \ldots, L$**)**

1.1 Criar a matriz de treinamento $\mathbf{X}_l$ a partir da amostragem com reposição de N colunas de $\mathbf{X}$

1.2 Treinar o l-ésimo classificador com os exemplos da matriz $\mathbf{X}_l$

1.3 Gerar a matriz de saída $\mathbf{Y}_l^*$ de dimensão $C \times Q$ a partir da matriz de teste $\mathbf{A}$

2. Calcular a matriz $\mathbf{V} = \sum_{l=1}^{l=L} \mathbf{Y}_l^* = [\mathbf{v}_1\ \mathbf{v}_2\ \cdots\ \mathbf{v}_Q]$ **em que** $\mathbf{v}_i = [v_1\ v_2\ \cdots\ v_C]^T$

3. Fazer $\mathbf{y}^* = [y_1^*\ y_2^*\ \cdots\ y_Q^*]^T$ **em que** $y_i^* = \arg\max_k \{\mathbf{v}_i\}$**.**

Saídas ou variáveis de interesse

$\mathbf{y}^*$: vetor de classes preditas para as Q amostras de teste

na etapa de treinamento (e.g. alterações no conjunto de treinamento, condições iniciais diferentes, etc.) implicam em grande variações no erro de classificação. Já um algoritmo de aprendizagem fraco é aquele que não garante erro arbitrariamente próximo de zero (FREUND; SCHAPIRE, 1996). Ambas são características típicas de RNAs, tais como aquelas foco da presente dissertação.

Comitês de classificadores obtidos por algoritmos de aprendizagem tradicionalmente supervisionados têm sido largamente explorados na literatura:

- **Comitês de redes MLP** (WINDEATT, 2008; KUSIAK; LI; ZHANG, 2010; CRUZ *et al.*, 2010; DAS; SENGUR, 2010; KUMAR; SELVAKUMAR, 2011);
- **Comitês de redes Fuzzy ARTMAP** (LOO *et al.*, 2006; SANTOS; CANUTO, 2008a; TRAN *et al.*, 2010);
- **Comitês de redes LVQ** (BERMEJO; CABESTANY, 2004; MADEO *et al.*, 2010; RAAFAT; TOLBA; ALY, 2011);
- **Comitês de redes ELM** (LAN; SOH; HUANG, 2009; LIU; XU; WANG, 2009; DENG *et al.*, 2010; WANG; LI, 2010);
- **Comitês de SVM (*Support Vector Machine*)** (**??**ZHOU; LAI; YU, 2010; TIAN; GU; LIU, 2011).

Entretanto, é possível adaptar técnicas de aprendizagem não-supervisionada, tais como as redes SOM e Fuzzy ART, para problemas de classificação. O próprio Kohonen, apesar de ter inicialmente proposto a rede SOM como um algoritmo não-supervisionado, introduziu em Kohonen (1988a) uma aplicação supervisionada da rede SOM ao problema de reconhecimento da fala. Desde então vários pesquisadores estudam a possibilidade de utilização da rede SOM como classificador (KANGAS; KOHONEN; LAAKSONEN, 1990; BIEBELMANN; KÖPPEN; NICKOLAY, 1996; CHO, 1997; SUGANTHAN, 1999; LAHA; PAL, 2001; CHRISTODOULOU; MICHAELIDES; PATTICHIS, 2003; HOYO; BULDAIN; MARCO, 2003; WYNS *et al.*, 2004; XIAO *et al.*, 2005; TURKY; AHMAD,).

A partir de classificadores individuais baseados na rede SOM, três abordagens de aprendizado em comitês são encontradas na literatura:

- **Comitês baseados em rede SOM para agrupamento de dados** (JIANG; ZHOU, 2004; GEORGAKIS; LI; GORDAN, 2005; CHANG *et al.*, 2008; GORGÔNIO; COSTA, 2008; BARUQUE; CORCHADO, 2010): inicialmente várias redes SOM são treinadas da maneira não-supervisionada usual para em seguida serem combinadas em uma única rede através de uma função de fusão;

- **Comitês baseados em rede SOM para classificação de padrões** (PETRIKIEVA; FYFE, 2002; CORCHADO; BARUQUE; YIN, 2007): usa-se variações na etapa de treinamento para tornar a aprendizagem da rede SOM supervisionada. Após o treinamento de várias redes SOM, a saída do comitê é decidida por votação majoritária. Três métodos capazes de tornar a rede SOM supervisionada serão apresentados no Capítulo 4;

- **Comitês baseados em redes SOM para regressão** (SCHERBART; NATTKEMPER, 2010): cada rede SOM do comitê prediz um valor de saída a partir de um modelo de regressão local associado. A saída do comitê normalmente é dada pela média das saídas das redes do comitê.

O uso de outros algoritmos de aprendizagem não-supervisionada, como por exemplo redes Fuzzy ART, em comitês de classificadores constitui um tema não amplamente explorado. Esta dissertação pretende expandir o estudo de comitês de classificadores baseados na rede SOM e iniciar a pesquisa de comitês de classificadores baseados na rede Fuzzy ART.

2.4 Conclusões

Neste capítulo foram apresentados conceitos básicos referentes à utilização de comitês de classificadores em problemas de reconhecimento de padrões. Foram apresentadas algumas das possíveis técnicas de combinação de resultados de múltiplos algoritmos de aprendizagem, assim como métodos de obtenção de diversidade entre os classificadores base.

Foram definidas ainda algumas das técnicas que serão aplicadas ao longo desta dissertação para a formação de comitês, mais especificamente o algoritmo Bagging com voto majoritário simples e a proposição de comitês com classificadores base obtidos a partir de redes neurais não-supervisionadas, mais especificamente SOM e Fuzzy ART.

O Capítulo 3 descreverá com mais detalhes as redes de aprendizagem competitiva não-supervisionadas (SOM e Fuzzy ART) e supervisionadas (Fuzzy ARTMAP e LVQ) utilizadas nesta dissertação.

Capítulo 3

Redes Neurais Competitivas

Neste capítulo serão apresentados os principais conceitos acerca de aprendizado competitivo supervisionado e não-supervisionado.

Simplificadamente, o paradigma do aprendizado competitivo em redes neurais se baseia na "competição" entre os neurônios da rede na busca por grupos de vetores similares em um processo conhecido como *clustering*[1]. Esse processo também pode ser entendido como uma busca por uma representação compacta dos padrões de entrada (quantização vetorial).

Os algoritmos apresentados nas próximas seções serão os mesmos utilizados nos comitês de classificadores propostos nesta dissertação.

3.1 Redes Neurais Competitivas Não-Supervisionadas

Técnicas de aprendizado não-supervisionado, também chamado de auto-organizado, são capazes extrair propriedades estatísticas de um conjunto de dados a partir da apresentação sucessiva de padrões. A principal diferença em relação ao aprendizado supervisionado está na ausência do rótulo dos vetores apresentados, i.e. não há uma relação previamente conhecida entre os exemplos disponíveis e a saída desejada para os mesmos. O mapeamento entrada-saída é então construído durante o processo de treinamento através de mecanismos de comparação e busca por similaridades.

Esta seção detalha as operações de duas técnicas de aprendizado

[1]Nesta dissertação os termos "*clustering*", "análise de agrupamento" e "clusterização" são usados como sinônimos.

não-supervisionado: redes Fuzzy ART e redes SOM. Ambas as redes são compostas por uma camada de neurônios e seus vetores de pesos (também chamados de vetores-protótipo, ou simplesmente protótipos) correspondentes, assim como apresentam um processo de treinamento competitivo.

Entretanto, o treinamento não-supervisionado das redes Fuzzy ART e SOM são implementados de maneiras diferentes. Redes SOM, por exemplo, apresentam mecanismos de aprendizado competitivo-cooperativo que distribuem as informações contidas em cada vetor de entrada apresentado entre um neurônio vencedor (competição) e seus vizinhos (cooperação) na rede. O efeito resultante é a formação de uma rede que, aproximadamente, preserva a topologia dos dados de entrada. A rede Fuzzy ART, por sua vez, se baseia em um mecanismo de aprendizado capaz de detectar padrões novos ou anômalos. Maiores detalhes sobre essas redes são apresentados a seguir.

3.1.1 Redes Fuzzy ART

Ao final dos anos 80 e início da década de 90, o grupo de pesquisa liderado por Stephen Grossberg introduziu as primeiras arquiteturas neurais baseadas na Teoria da Ressonância Adaptativa:

- **ART-1** (CARPENTER; GROSSBERG, 1987b);
- **ART-2** (CARPENTER; GROSSBERG, 1987a, 1988);
- **ART-2A** (CARPENTER; GROSSBERG; ROSEN, 1991);
- **ARTMAP** (CARPENTER; GROSSBERG; REYNOLDS, 1991);
- **Fuzzy ART** (CARPENTER; GROSSBERG; ROSEN, 1991);
- **Fuzzy ARTMAP** (CARPENTER *et al.*, 1992).

Essas arquiteturas foram desenvolvidas como uma possível solução para o dilema estabilidade-plasticidade (CARPENTER; GROSSBERG, 1987b) encontrado ao se projetar redes neurais para reconhecimento de padrões: ao se apresentar novos padrões a um classificador neural, é preciso adaptar os pesos da rede, adicionando uma nova parcela de conhecimento, ou seja, o sistema deve ser capaz de adquirir informação. Ao mesmo tempo, é preciso que o conhecimento acumulado referente

aos padrões previamente apresentados seja mantido, ou seja, o classificador deve ser estável[2].

A principal ideia por trás do arcabouço da ART é a seguinte: caso um dado padrão de entrada seja diferente o suficiente dos padrões já armazenados na memória de longo prazo da rede (i.e. nos seus pesos), então crie uma nova categoria e a associe a este padrão de entrada (KESKIN; ÖZKAN, 2009). Este mecanismo de detecção de novidades é especialmente efetivo na identificação de dados anômalos ou *outliers* (BARRETO; AGUAYO, 2009).

A rede Fuzzy ART estende a rede ART-1, originalmente desenvolvida para processar dados binários (CARPENTER; GROSSBERG, 1987b), com a capacidade de processar padrões analógicos. Uma das principais características dessa rede é a incorporação de operadores da Lógica Fuzzy[3], mais especificamente os operadores MAX($\vee$) e MIN($\wedge$) (ZADEH, 1965).

3.1.1.1 Arquitetura da rede Fuzzy ART

A Figura 3.1 ilustra um diagrama de blocos das partes constituintes de uma rede Fuzzy ART. A seguir são descritos em maior nível de detalhes cada componente da sua arquitetura.

Vetor de entrada. Um exemplar de treinamento, de dimensão P, é representado pelo vetor $\mathbf{a} \in \mathbb{R}^P$. Considerando-se N exemplos de treinamento, pode-se incluir índices temporais a esses vetores, indicando a ordem de apresentação à rede: $\mathbf{a}(n), \quad n = 1, 2, \ldots, N$. As P componentes dos vetores de entrada são números reais limitados entre 0 e 1, ou seja, $a_j(n) \in [0, 1], \quad j = 1, 2, \ldots, P$. Os padrões de entrada alimentam a camada F_1, chamada de *camada de apresentação*. Antes de serem apresentados, contudo, os vetores $\mathbf{a}(n)$ passam por uma etapa de codificação complementar (*complement coding*) para gerar os vetores $\mathbf{x}(n) \in \mathbb{R}^D$, em que $D = 2P$. Esse procedimento será detalhado posteriormente. Note que os vetores codificados $\mathbf{x}(n)$ passam então a serem vistos como a entrada do algoritmo de treinamento.

[2]Aqui o termo "classificador estável" possui significado diferente do usado anteriormente no contexto de comitê de classificadores, pois refere-se à capacidade do classificador de reter o conhecimento acumulado. Note que um classificador pode se estável no sentido de Grossberg mas ser instável no sentido de comitês.

[3]Apesar da existência do nome em português, lógica nebulosa ou difusa, esta dissertação adotará a nomenclatura original, lógica fuzzy.

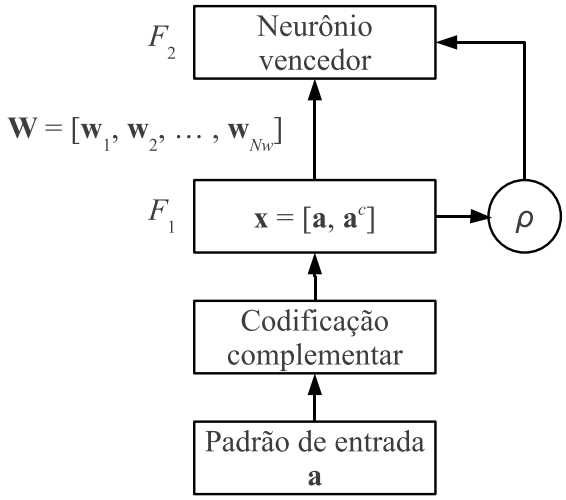


Figura 3.1: Diagrama de blocos da rede Fuzzy ART. Os índices temporais foram removidos para melhor visualização.

Matriz de pesos. Cada neurônio da rede Fuzzy ART constitui uma parcela de informação aprendida, ou seja, uma categoria, que deve ser preservada ao longo do treinamento. Nesse contexto, cada categoria corresponde a um *cluster* de dados. Enquanto um único protótipo é representado por um vetor de pesos de dimensão igual à dos padrões de entrada, $\mathbf{w}(n) \in \mathbb{R}^D$, toda a memória de longo prazo da rede pode ser representada por uma matriz de dimensão $D \times N_w$ contendo todos os N_w vetores de pesos criados até então: $\mathbf{W}(n) = [\mathbf{w}_1(n)\ \mathbf{w}_2(n)\ \cdots\ \mathbf{w}_{N_w}(n)]^T$. Neurônios não-comprometidos (*uncommitted*) têm suas componentes iniciadas com o valor 1: $\mathbf{w}(n) = \mathbf{1}^D$, em que $\mathbf{1}^D$ é um vetor de dimensão D cujos elementos são todos iguais a 1. Os vetores de pesos da rede podem ser vistos como protótipos, sendo reunidos na camada F_2, chamada de camada de representação.

Parâmetro de vigilância (ρ). No funcionamento de uma rede Fuzzy ART é preciso definir um limiar para o grau de similaridade entre a informação armazenada (protótipos) e a apresentada (vetores de entrada). O parâmetro de vigilância define esse limiar de decisão.

Parâmetro de escolha (β). Este parâmetro confere um fator de escala ao cálculo da ativação (ou memória de curto prazo) de cada neurônio, como será apresentado no algoritmo de funcionamento da rede.

Parâmetro de aprendizado (η). Um novo exemplo de treinamento contribui com uma porção de informação, que se soma à memória de longo prazo

acumulada da rede. O parâmetro η funciona como um passo de aprendizado, determinando essa parcela acrescentada a cada novo padrão apresentado.

3.1.1.2 Treinamento da rede Fuzzy ART

Para cada vetor apresentado à rede Fuzzy ART, o seu algoritmo de treinamento deve obedecer às seguintes etapas de processamento.

Codificação da entrada. O padrão de entrada $\mathbf{a}(n) \in \mathbb{R}^P$ deve ser codificado em um vetor $\mathbf{x}(n) \in \mathbb{R}^{2P}$. Isso é feito através do processo de codificação complementar

$$\mathbf{x}(n) = \begin{bmatrix} \mathbf{a}(n) \\ \mathbf{a}^c(n) \end{bmatrix} = \begin{bmatrix} \mathbf{a}(n) \\ \mathbf{1}^P - \mathbf{a}(n) \end{bmatrix}, \tag{3.1}$$

em que $\mathbf{1}^P$ é um vetor de dimensão P contendo somente elementos iguais a 1. Como até agora denotou-se a dimensão do vetor $\mathbf{x}(n)$ por D, a partir de agora tem-se $D = 2P$.

Processo de competição. Apresenta-se o vetor $\mathbf{x}(n)$ à primeira camada da rede, F_1, e, para cada um dos N_w neurônios, calcula-se a i-ésima ativação, que pode ser entendida como o nível de ressonância do protótipo:

$$t_i(n) = \frac{|\mathbf{x}(n) \wedge \mathbf{w}_i(n)|}{\beta + |\mathbf{w}_i(n)|}, \quad i = 1, 2, \ldots, N_w, \tag{3.2}$$

em que o operador $\wedge$ representa a operação de conjugação *fuzzy*, elemento a elemento, ou seja

$$x_j(n) \wedge w_{ij} \equiv \min\{x_j(n), w_{ij}(n)\}, \tag{3.3}$$

e que $|\mathbf{x}| = \sum_{j=1}^{D} |x_j|$ é a norma L_1 do vetor $\mathbf{x}$. O parâmetro β funciona como um fator de escala positivo para a ativação calculada. Por fim, busca-se pelo índice do neurônio vencedor i^* na iteração n

$$i^*(n) = \arg\max_i \{t_i(n)\}, \quad i = 1, 2, \ldots, N_w. \tag{3.4}$$

Critério de vigilância. Verifica-se se o neurônio vencedor i^* satisfaz o critério de

vigilância por meio do seguinte teste:

$$\frac{|\mathbf{x}(n) \wedge \mathbf{w}_{i^*}(n)|}{|\mathbf{x}(n)|} \geq \rho, \tag{3.5}$$

em que $0 < \rho < 1$ é o parâmetro de vigilância. Se o teste de vigilância é satisfeito, segue-se para a etapa seguinte, a de atualização dos pesos. Caso contrário, a ativação do neurônio i^* recebe o valor zero ($t_{i^*}(n) = 0$) e a busca por um novo neurônio é reiniciada usando-se a Equação (3.4). Esse processo de competição em que se verifica o grau de casamento (*matching*) entre o vetor de entrada e os protótipos da rede Fuzzy ART é chamado de *ressonância*.

Atualização dos pesos. Caso o vencedor seja um neurônio ainda não usado (i.e. $\mathbf{w}_{i^*}(n) = \mathbf{1}^D$), este recebe o padrão de entrada atual e acrescenta-se um novo neurônio à rede. Matematicamente, essa etapa é realizada por meio das seguintes regras:

$$\mathbf{w}_{i^*}(n+1) = \mathbf{x}(n), \tag{3.6}$$

$$N_w = N_w + 1, \tag{3.7}$$

$$\mathbf{w}_{N_w} = \mathbf{1}^D. \tag{3.8}$$

Caso contrário, os pesos do neurônio vencedor são atualizados

$$\mathbf{w}_{i^*}(n+1) = (1-\eta)\mathbf{w}_{i^*}(n) + \eta[\mathbf{x}(n) \wedge \mathbf{w}_{i^*}(n)], \tag{3.9}$$

em que $0 < \eta \leq 1$ corresponde a um passo de aprendizado. O treinamento é então reiniciado com a apresentação de um novo padrão.

Note que as redes da família ART foram desenvolvidas sob a ideia de "aprendizado contínuo" (*continuous learning*), não havendo portanto a separação usual entre as fases de treinamento e teste. Mesmo assim, essa separação pode ser feita para fins de comparação com outras redes neurais.

Um resumo do treinamento da rede Fuzzy ART encontra-se no Algoritmo 3.1.

3.1.1.3 Interpretação geométrica da Rede Fuzzy ART

Com o intuito de apresentar uma interpretação geométrica para o processo de atualização dos pesos utilizando-se codificação complementar, suponha-se que o

Algoritmo 3.1 Algoritmo de treinamento da rede Fuzzy ART.

Constantes

β: parâmetro de escolha, $\beta \geq 0$

ρ: parâmetro de vigilância, $0 < \rho \leq 1$

η: parâmetro de aprendizado, $0 < \eta \leq 1$

$n_{\text{MÁX}}$: número de iterações de treinamento

Entradas

$\mathbf{a}(n)$: vetor de entrada, dimensão P

$\mathbf{x}(n)$: vetor de entrada, dimensão $D = 2P$ (codificação complementar)

Algoritmo

1. Inicialização $(n = 0)$

Crie e inicialize os pesos do neurônio inicial da rede $\mathbf{w}_1(0) = \mathbf{1}^D$

2. Laço temporal $(n = 1, 2, \ldots, n_{\mathbf{MÁX}})$

2.1 Selecionar $\mathbf{x}(n)$ do conjunto de vetores de entrada

2.2 Buscar pelo índice do neurônio vencedor:

$i^*(n) = \arg\max_i\{t_i\}$, tal que $t_i = \frac{|\mathbf{x}(n) \wedge \mathbf{w}_i(n)|}{\beta + |\mathbf{w}_i(n)|}, \quad i = 1, 2, \ldots, N_w$

2.3 Teste de ressonância (critério de vigilância)

SE $|\mathbf{x}(n) \wedge \mathbf{w}_{i^*}(n)| > \rho|\mathbf{x}(n)|$, vá para o Passo 2.4

SENÃO, volte para o Passo 2.2 e busque um novo neurônio vencedor

2.4 Atualização dos pesos

SE $\mathbf{w}_{i^*}(n) = \mathbf{1}^D$ (i.e., o vencedor nunca foi ativado antes), FAÇA

$N_w = N_w + 1$ e $\mathbf{w}_{N_w} = \mathbf{1}^D$ (i.e. crie um novo neurônio)

$\mathbf{w}_{i^*}(n+1) = \mathbf{x}(n)$ (i.e. o novo neurônio armazena o novo padrão)

SENÃO FAÇA

$$\mathbf{w}_{i^*}(n+1) = (1-\eta)\mathbf{w}_{i^*}(n) + \eta(\mathbf{x}(n) \wedge \mathbf{w}^{i^*}(n))$$

Saídas

$\mathbf{w}_{i^*(n)}$: vetor de pesos do neurônio vencedor na iteração n

Observações

Tipicamente, usa-se codificação complementar para pré-processar $\mathbf{a}(n)$.

O número de neurônios é iniciado como $N_w = 1$ e incrementado ao longo das iterações.

vetor de entrada $\mathbf{a}(n)$ seja bidimensional, com componentes $[a_1(n), a_2(n)]$. Pela Equação (3.1), o vetor $\mathbf{x}(n)$ resultante é dado por

$$\mathbf{x}(n) = \begin{bmatrix} \mathbf{a}(n) \\ \mathbf{a}^c(n) \end{bmatrix}. \tag{3.10}$$

Como a entrada da rede é formada por um vetor e seu complemento, o i-ésimo vetor de pesos $\mathbf{w}_i(n)$ da rede pode ser escrito como

$$\mathbf{w}_i(n) = \begin{bmatrix} \mathbf{p}_i(n) \\ \mathbf{q}_i^c(n) \end{bmatrix}. \tag{3.11}$$

Os vetores $\mathbf{p}_i(n)$ e $\mathbf{q}_i^c(n)$ são bidimensionais e definem vértices opostos de um retângulo $R_i(n)$ (CARPENTER; GROSSBERG; ROSEN, 1991).

No caso do passo de aprendizagem ser unitário ($\eta = 1$), tem-se na primeira atualização de pesos do neurônio vencedor i^*

$$\mathbf{w}_{i^*}(n+1) = \mathbf{x}(n) \wedge \mathbf{1}^4 = \mathbf{x}(n), \tag{3.12}$$

ou seja, $\mathbf{p}_{i^*}(n+1) = \mathbf{a}(n)$ e $\mathbf{q}_{i^*}^c(n+1) = \mathbf{a}^c(n)$. O lugar geométrico definido por $\mathbf{a}(n)$ e $\{\mathbf{a}^c(n)\}^c = \mathbf{a}(n)$ equivale ao ponto $\mathbf{p} = \mathbf{a}(n)$.

Considera-se agora o padrão de entrada seguinte, $\mathbf{a}(n+1)$. Após a codificação complementar, tem-se o vetor de entrada

$$\mathbf{x}(n+1) = \begin{bmatrix} \mathbf{a}(n+1) \\ \mathbf{a}^c(n+1) \end{bmatrix}. \tag{3.13}$$

Considerando-se a atualização do mesmo vetor de pesos i^* analisado anteriormente, tem-se

$$\mathbf{w}_{i^*}(n+2) = \mathbf{x}(n+1) \wedge \mathbf{w}_{i^*}(n+1) = \begin{bmatrix} \mathbf{a}(n) \wedge \mathbf{a}(n+1) \\ \mathbf{a}^c(n) \wedge \mathbf{a}^c(n+1) \end{bmatrix} = \begin{bmatrix} \mathbf{a}(n) \wedge \mathbf{a}(n+1) \\ \{\mathbf{a}(n) \vee \mathbf{a}(n+1)\}^c \end{bmatrix}, \tag{3.14}$$

em que usou-se a relação $(b_1 \vee b_2)^c = b_1^c \wedge b_2^c$, versão *fuzzy* da Lei de De Morgan

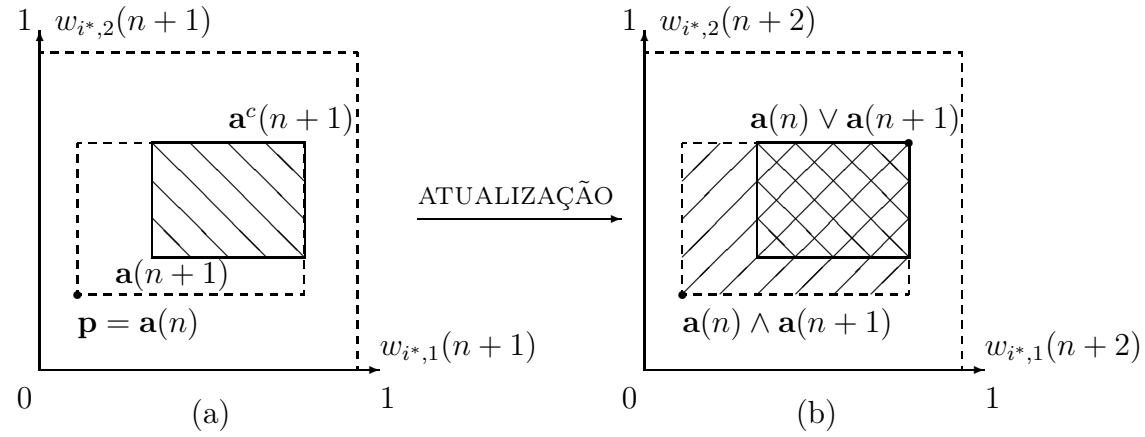


Figura 3.2: Interpretação geométrica da evolução dos pesos da rede Fuzzy ART: (a) O protótipo $\mathbf{w}_{i^*}(n+1) = [w_{i^*,1}(n+1) \;\; w_{i^*,2}(n+1)]^T$ inicialmente corresponde somente ao ponto $\mathbf{p} = \mathbf{a}(n)$. (b) O padrão de entrada $\mathbf{a}(n+1)$ expande a abrangência de $\mathbf{w}_{i^*}(n+2) = [w_{i^*,1}(n+2) \;\; w_{i^*,2}(n+2)]^T$. Modificado de Aguayo (2008).

(DUBOIS; PRADE, 1985).

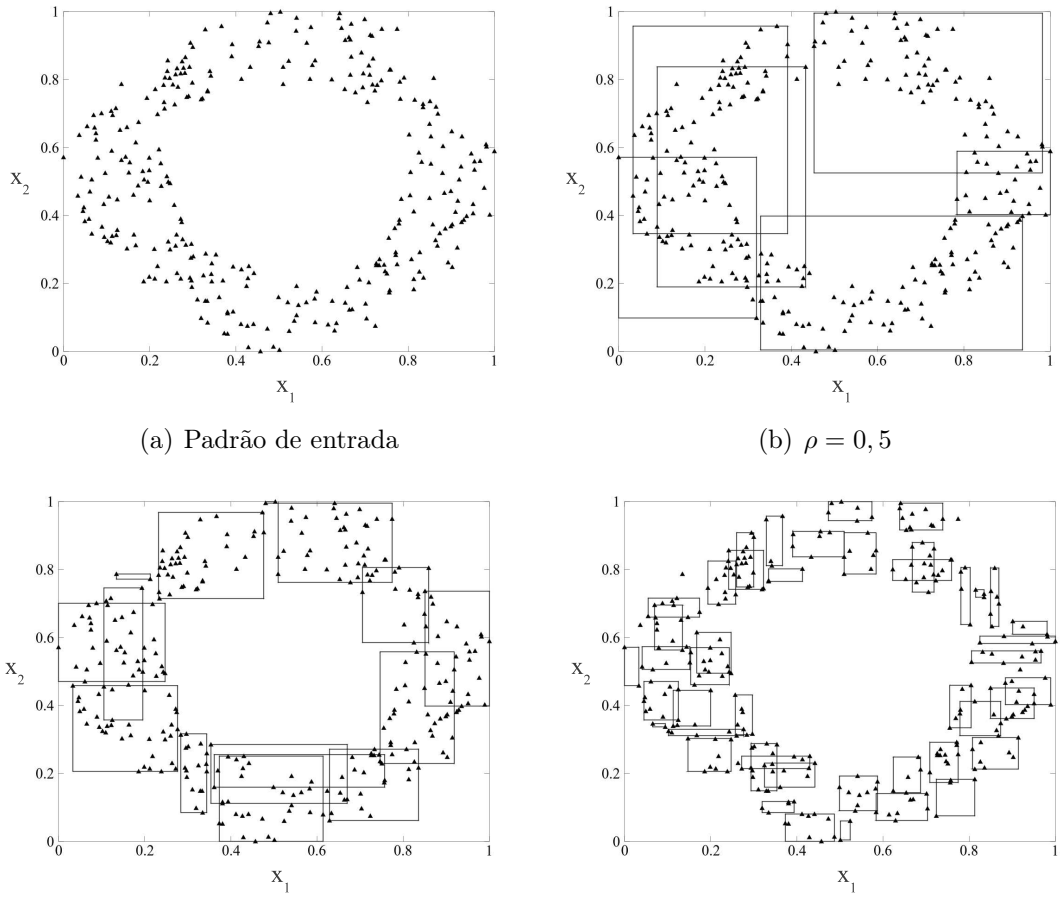
Nesse momento o lugar geométrico $R_{i^*}(n+2)$ é o retângulo formado pelos vértices $\mathbf{p}_{i^*}(n+2) = \mathbf{a}(n) \wedge \mathbf{a}(n+1)$ e $\mathbf{q}_{i^*}(n+2) = \mathbf{a}(n) \vee \mathbf{a}(n+1)$. Este retângulo acomoda tanto o padrão $\mathbf{a}(n)$ quanto o padrão $\mathbf{a}(n+1)$, ilustrando o fenômeno de plasticidade da rede que a torna capaz de aprender com novos vetores de entrada. Os procedimentos descritos podem ser visualizados na Figura 3.2.

3.1.1.4 O papel dos parâmetros ajustáveis

No algoritmo de treinamento da rede Fuzzy ART percebe-se que o parâmetro de vigilância ρ determina quando a rede deve atualizar o protótipo de um dos neurônios existentes ou adicionar um novo protótipo. Para ilustrar esse comportamento, considera-se uma distribuição de pontos no plano (x, y), como mostrado na Figura 3.3(a). Utilizou-se a codificação complementar de modo a exemplificar a interpretação geométrica em questão.

A criação progressiva de novas categorias nas Figuras 3.3(b)-3.3(d) revela uma tendência de diminuir a quantidade de exemplos representados por cada protótipo à medida que o valor de ρ tende a 1. O valor do parâmetro β é mantido fixo em $\beta = 0$.

Em Carpenter e Gjaja (1993) é feito um amplo estudo sobre o papel do parâmetro de escolha β no funcionamento do algoritmo Fuzzy ART. O estudo citado



(a) Padrão de entrada

(b) $\rho = 0,5$

(c) $\rho = 0,75$

(d) $\rho = 0,9$

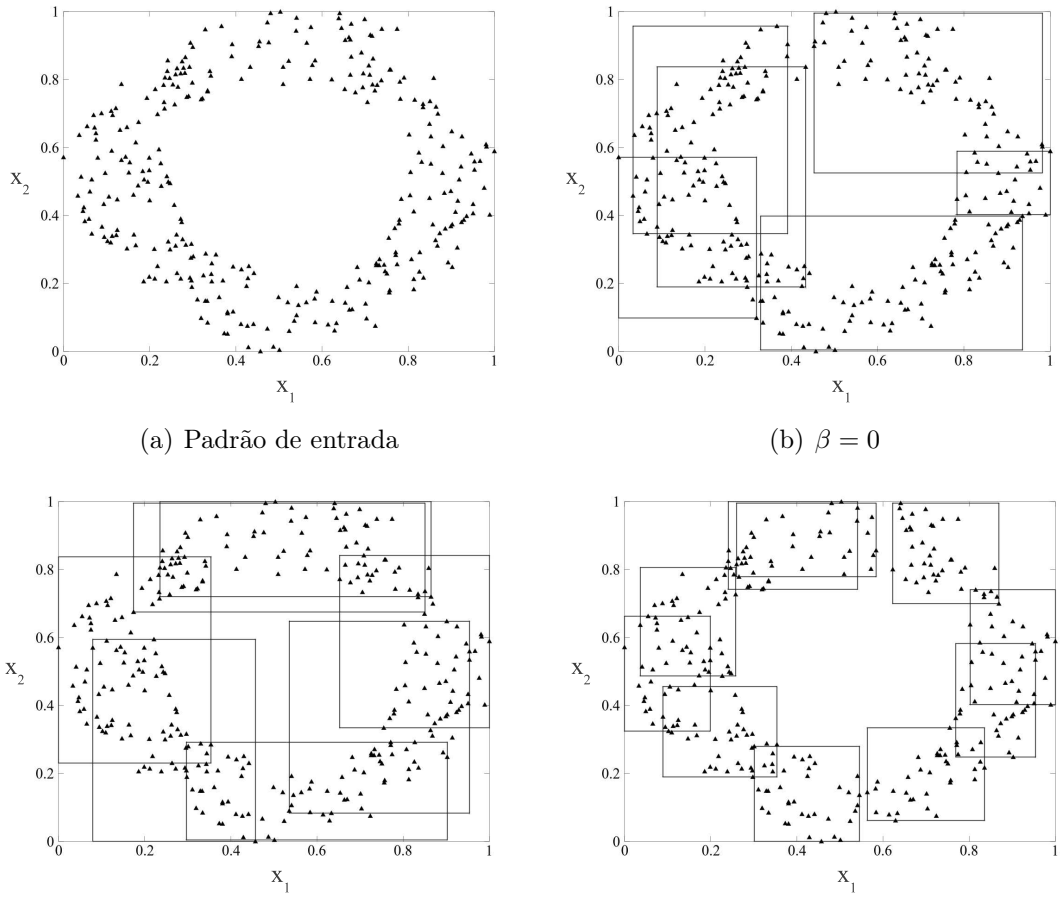
Figura 3.3: Efeito da variação do parâmetro de vigilância na rede Fuzzy ART.

conclui que o aumento no valor de β tem resultado semelhante ao de aumentar o parâmetro ρ.

Uma evidência empírica da influência do parâmetro de escolha pode ser observado na Figura 3.4, em que um maior número de categorias são criadas à medida que o parâmetro de escolha β é aumentado, para um ρ fixo (ρ = 0,5). É interessante notar ainda que a sensibilidade da rede Fuzzy ART ao parâmetro β é consideravelmente menor que ao parâmetro ρ.

É relevante lembrar que o número de protótipos necessários para representar um conjunto de dados é específico do problema considerado. Portanto, os valores dos parâmetros ρ e β devem ser determinados caso a caso para a obtenção de resultados satisfatórios.

O passo de aprendizagem η tem influência direta na parcela de informação que



(a) Padrão de entrada

(b) $\beta = 0$

(c) $\beta = 1$

(d) $\beta = 5$

Figura 3.4: Efeito da variação do parâmetro de escolha na rede Fuzzy ART.

cada vetor de entrada disponível para treinamento transfere aos pesos dos neurônios da rede. Quando seu valor é $\eta = 1$, a rede opera no modo *fast learning* (aprendizado rápido), em que o máximo de informação é extraído de cada padrão. Entretanto, o parâmetro η pode ser ajustado para controlar situações em que existem vetores para treinamento que não são considerados plenamente "confiáveis", como *outliers* e padrões ruidosos. Nesses casos, torna-se interessante a adição de uma parte da informação fornecida pelos padrões anteriormente apresentados. Como essa situação é intrínseca ao problema de interesse, o valor do parâmetro η deve ser escolhido para cada caso.

3.1.2 Redes SOM

Introduzida por Kohonen (1982), o conceito de mapas auto-organizáveis foi proposto a partir da observação que mapas corticais se formam de maneira

adaptativa e automática (KOHONEN, 1997). Por mapas corticais entende-se o mapeamento topográfico de regiões no cérebro responsáveis por recepções sensoriais específicas.

A rede SOM tem então como objetivo principal o mapeamento (ou projeção) de um espaço contínuo de dimensão possivelmente elevada em um espaço discreto de dimensão reduzida. A projeção resultante consiste de N_w neurônios dispostos em um arranjo geométrico fixo, de dimensão S. Comumente usa-se o valor $S = 2$, correspondendo a um mapa bidimensional.

Matematicamente, seja um espaço de entrada contínuo $\mathcal{X} \subset \mathbb{R}^D$ e um espaço discreto $\mathcal{Y} \subset \mathbb{R}^S$ formado por N_w vetores. Um dado vetor $\mathbf{x} \in \mathcal{X}$ será representado na rede por um vetor $\mathbf{y}_{i^*} \in \mathcal{Y}$ através do mapeamento $i^*(\mathbf{x}) : \mathcal{X} \to \mathcal{Y}$.

A preservação aproximada da topologia dos dados de entrada da rede SOM garante que os vetores de entrada próximos entre si sejam mapeados em vetores próximos no espaço discreto da rede. Essa propriedade é especialmente interessante em aplicações envolvendo visualização de dados de alta dimensão.

A rede SOM, assim como a rede Fuzzy ART, é uma rede neural competitiva que pode ser usada tanto em tarefas de agrupamento de dados quanto de quantização vetorial. A primeira consiste em separar ou encontrar grupos de vetores similares segundo um critério de similaridade. A segunda é a tarefa de substituir um conjunto de N vetores por um conjunto de N_w protótipos, em que $N_w \ll N$. Note que nem todo algoritmo de agrupamento de dados realiza quantização vetorial, mas todo algoritmo de quantização vetorial faz (ou pode fazer) análise de agrupamento.

3.1.2.1 Arquitetura Geral

A arquitetura de uma rede SOM encontra-se ilustrada na Figura 3.5. Pode-se perceber que todos os N_w neurônios recebem o padrão de entrada $\mathbf{x}(n) \in \mathbb{R}^D$ simultaneamente. Os atributos contidos em $\mathbf{x}(n)$ são ponderados pelo i-ésimo neurônio por um vetor de pesos $\mathbf{w}_i(n) \in \mathbb{R}^D$.

Reunindo os N exemplos disponíveis para o processo de aprendizagem da rede, obtém-se a matriz $\mathbf{X} = [\mathbf{x}_1 \ \mathbf{x}_2 \ \cdots \ \mathbf{x}_N]^T$, formando o conjunto de dados de treinamento. De maneira semelhante, a reunião dos vetores de pesos em colunas resulta na matriz $\mathbf{W}(n) = [\mathbf{w}_1(n) \ \mathbf{w}_2(n) \ \cdots \ \mathbf{w}_{N_w}(n)]^T$, que representa a rede SOM.

O mapeamento realizado pela rede SOM pode ser visto como um processo de

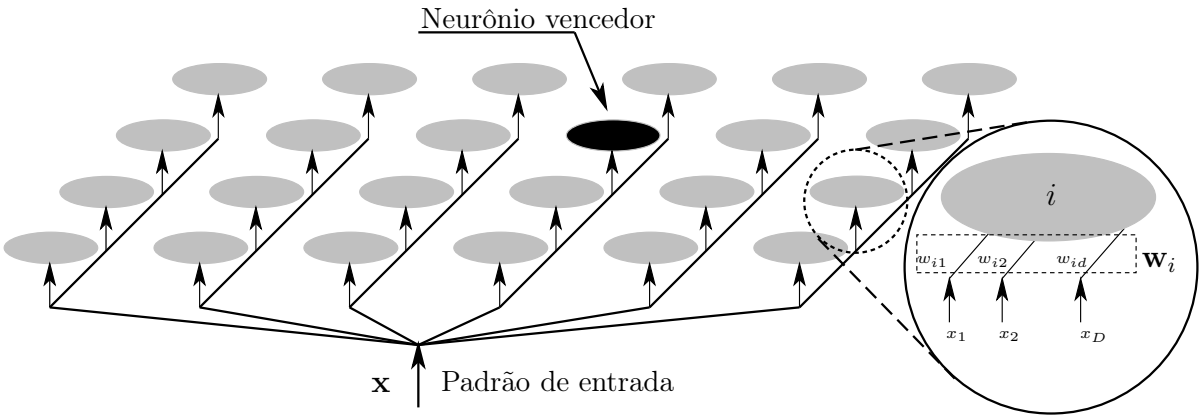


Figura 3.5: Exemplo de rede SOM bidimensional. Os vetores de entrada e de pesos são D-dimensionais. Os N_w neurônios estão uniformemente dispostos em uma grade retangular. Modificado de Aguayo (2008).

codificação da matriz de vetores de entrada $\mathbf{X}$ em que o dicionário (*codebook*) é formado pelas colunas da matriz de pesos $\mathbf{W}$. A Figura 3.6 reforça essa interpretação ao ilustrar o mapeamento entre os espaço de entrada e saída.

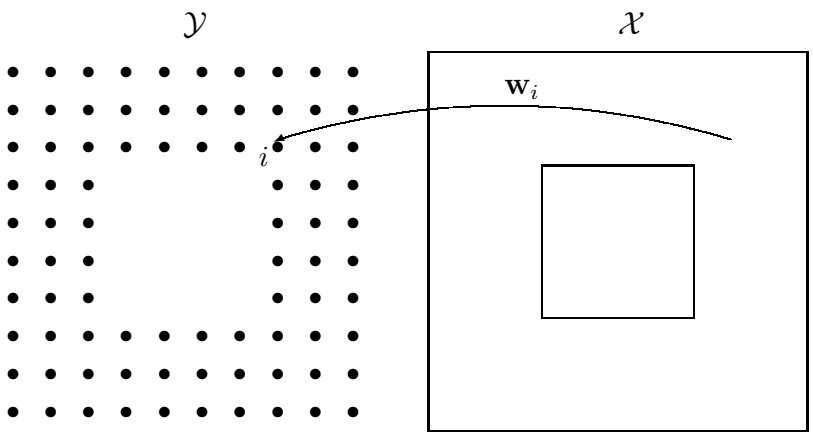


Figura 3.6: Mapeamento entre os espaços $\mathcal{X}$ e $\mathcal{Y}$ realizado pela rede SOM. Modificado de Aguayo (2008).

3.1.2.2 Treinamento da rede SOM

No início do treinamento da rede SOM, seus N_w neurônios são dispostos de forma regular em uma malha de dimensão $P_1 \times P_2$, considerando um mapa bidimensional. Nesse momento os vetores de pesos possuem valores aleatórios, normalmente pequenos. Para cada vetor apresentado à rede na iteração n do

algoritmo de treinamento, são realizadas as etapas de processamento a seguir.

Processo de competição. Inicialmente verifica-se qual o neurônio i^* é o mais próximo da entrada $\mathbf{x}(n)$ pela expressão

$$i^*(n) = \arg\min_{\forall i} \|\mathbf{x}(n) - \mathbf{w}_i(n)\|, \tag{3.15}$$

em que $\|\cdot\|$ denota o cálculo da distância euclidiana. A métrica de dissimilaridade pode ser outra, mas a distância euclidiana é a escolha mais comum.

Processo de cooperação. O vetor de pesos associado ao neurônio vencedor $i^*(n)$, assim como aos seus neurônios vizinhos, são simultaneamente atualizados pela seguinte regra de aprendizagem:

$$\mathbf{w}_i(n+1) = \mathbf{w}_i(n) + \eta(n)h(i^*, i; n)[\mathbf{x}(n) - \mathbf{w}_i(n)], \tag{3.16}$$

em que $0 < \eta(n) \leq 1$ corresponde ao valor do parâmetro de aprendizagem na iteração n do algoritmo e a função $h(i^*, i; n)$ é chamada *função de vizinhança*. Esta função define a vizinhança de influência do neurônio vencedor ao determinar quais neurônios terão seus pesos atualizados de modo mais intenso. Uma escolha comum para $h(i^*, i; n)$ é a função gaussiana:

$$h(i^*, i; n) = \exp\left(-\frac{\|\mathbf{r}_i - \mathbf{r}_{i^*}\|^2}{2\sigma^2(n)}\right), \tag{3.17}$$

em que $\mathbf{r}_i$ e $\mathbf{r}_{i^*}$ são, respectivamente, as coordenadas dos neurônios i e i^* na grade de saída. O parâmetro $\sigma(n) > 0$ refere-se à largura da vizinhança considerada: quanto maior seu valor, maior o número de neurônios atualizados em torno do neurônio vencedor.

A Equação (3.16) pode ser reescrita ao substituir $\eta^*(n) = \eta(n)h(i^*, i; n)$, tornando-se similar à Equação (3.9):

$$\mathbf{w}_i(n+1) = (1 - \eta^*(n))\mathbf{w}_i(n) + \eta^*(n)\mathbf{x}(n). \tag{3.18}$$

Como a equação de atualização dos pesos, Equação (3.16) ou Equação (3.18), depende da proximidade dos neurônios em relação ao neurônio vencedor, existe a tendência do surgimento de regiões específicas na rede SOM sensíveis a

determinadas variações nos padrões de entrada. Essa característica constitui a já citada capacidade da rede SOM de preservar, de forma aproximada, a topologia dos dados após o treinamento.

O processo de treinamento da rede SOM encontra-se no Algoritmo 3.2.

3.1.2.3 Sobre a convergência da rede SOM

Para garantir a convergência dos pesos da rede SOM a valores estáveis durante o algoritmo de treinamento, é preciso reduzir o passo de aprendizado e o parâmetro de espalhamento ao longo das iterações do método (RITTER; SCHULTEN, 1988). Esse precedimento tem como objetivo reduzir gradualmente a influência dos pesos iniciais.

Nesta dissertação, optou-se pelo decaimento exponencial dos parâmetros $\eta(n)$ e $\sigma(n)$:

$$\eta(n) = \eta_0 \left(\frac{\eta_f}{\eta_0}\right)^{(n/n_{\text{MÁX}})}, \tag{3.19}$$

$$\sigma(n) = \sigma_0 \left(\frac{\sigma_f}{\sigma_0}\right)^{(n/n_{\text{MÁX}})}, \tag{3.20}$$

em que $n_{\text{MÁX}}$ é o total de iterações de treinamento, $\eta(1) = \eta_0$, $\eta(n_{\text{MÁX}}) = \eta_f$, $\sigma(1) = \sigma_0$ e $\sigma(n_{\text{MÁX}}) = \sigma_f$. Os valores iniciais η_0 e σ_0, assim como os valores finais η_f e σ_f, constituem parâmetros a serem especificados para cada problema estudado.

A Figura 3.7 apresenta exemplos de decaimento do parâmetro η para diferentes valores de η_f, considerando-se um valor fixo $\eta_0 = 0{,}9$. Note que as curvas para o parâmetro σ seriam semelhantes.

A Figura 3.8 ilustra um exemplo de uma rede SOM durante a etapa de treinamento. Nesse exemplo, tem-se um conjunto de dados bidimensionais (os mesmos exemplificados na Seção 3.1.1) que devem ser mapeados pela rede. Percebe-se que, apesar da quantidade de neurônios ser menor que a quantidade de amostras, ao longo das épocas[4] a rede SOM é capaz de obter uma representação condensada dos dados treinados.

A Figura 3.9 apresenta a evolução do erro de quantização médio, calculado para

[4]Uma época consiste de uma apresentação de todos os vetores de treinamento à rede.

Algoritmo 3.2 Algoritmo de treinamento da rede SOM.

Constantes

N_w: número de neurônios da rede

η_0: valor inicial do parâmetro de aprendizado η

σ_0: valor inicial do parâmetro de espalhamento σ

D: dimensão de entrada

$D_1 e D_2$: dimensões do mapa

$n_{\text{MÁX}}$: número de iterações de treinamento

Entradas

$\mathbf{x}(n)$: vetor de entrada, dimensão D

Algoritmo

1. Inicialização

Iniciar os pesos $\mathbf{w}_i(0)$ com valores pequenos aleatórios $(i = 1, 2, \ldots, N_w)$

Fazer $\eta(1) = \eta_0$ e $\sigma(1) = \sigma_0$

2. Laço temporal $(n = 1, 2, \ldots, n_{\text{MÁX}})$

2.1 Selecionar $\mathbf{x}(n)$ do conjunto de vetores de entrada

2.2 Armazenar o índice do neurônio vencedor

$$i^*(n) = \arg\min_{\forall i} \|\mathbf{x}(n) - \mathbf{w}_i(n)\|$$

2.3 Atualizar os pesos do vencedor e da vizinhança

Para $i = 1, 2, \ldots, N_w$, calcular

$$h(i^*, i; n) = \exp\left(-\frac{\|\mathbf{r}_i - \mathbf{r}_{i^*}\|^2}{2\sigma^2(n)}\right)$$

$$\mathbf{w}_i(n+1) = \mathbf{w}_i(n) + \eta(n) h(i^*, i; n)[\mathbf{x}(n) - \mathbf{w}_i(n)]$$

Decair os parâmetros $\eta(n)$ e $\sigma(n)$

Saídas

Saída a cada iteração: coordenada $\mathbf{r}_{i*(n)}$ do neurônio vencedor e seu vetor de pesos $\mathbf{w}_{i*(n)}$

Resultado do treinamento: $\mathbf{W}(n_{\text{MÁX}})$: matriz dos pesos dos neurônios (dimensão $D \times N_w$)

Observações

Durante o uso da rede SOM, pós-treinamento, o Passo 2.3 não é necessário

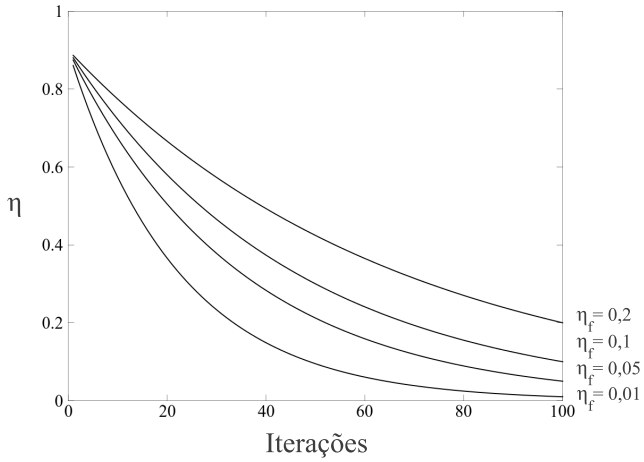


Figura 3.7: Exemplos de decaimento do parâmetro η da rede SOM.

a k-ésima época pela Equação (3.21).

$$eqm(k) = \frac{1}{N}\sum_{n=1}^{N} \|\mathbf{x}(n) - \mathbf{w}_{i^*}(n)\|^2, \tag{3.21}$$

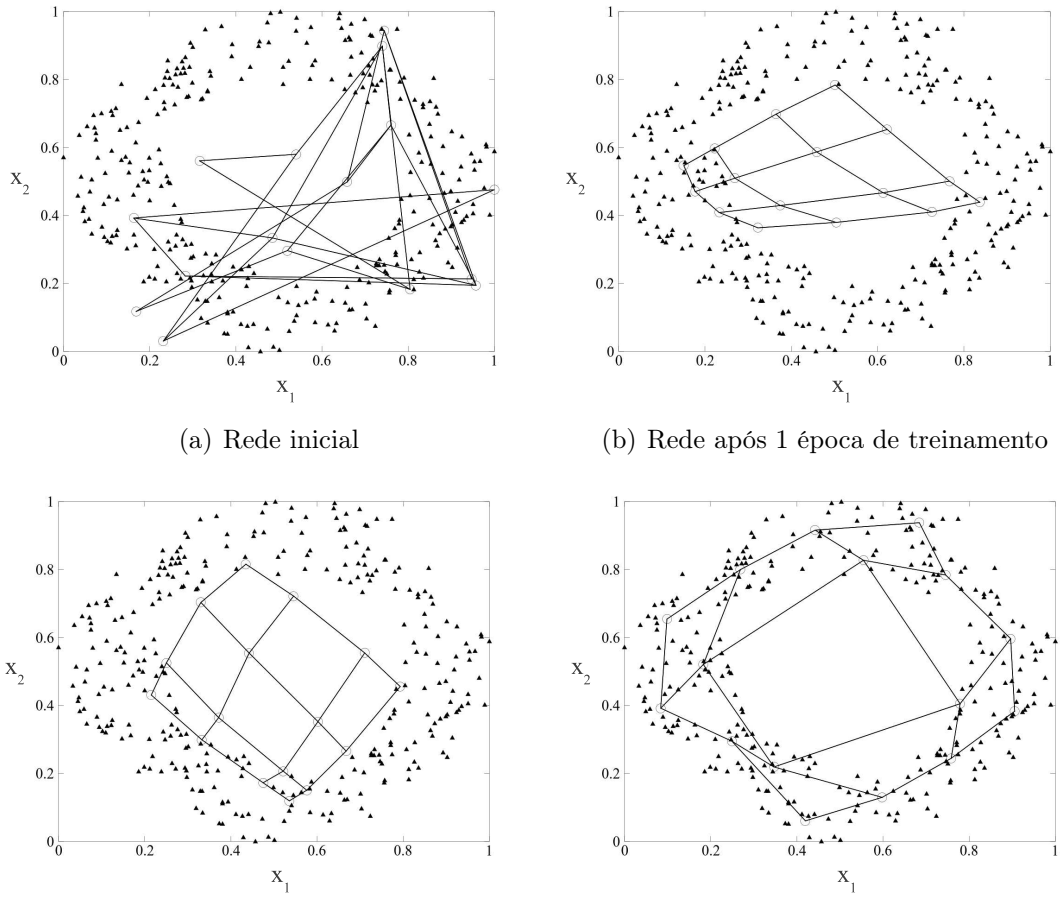
em que N é o número de vetores de treinamento. A convergência do erro de quantização médio, conforme ilustrado na Figura 3.9, indica que o mapeamento final é de fato satisfatório.

3.1.3 Comparação entre as redes Fuzzy ART e SOM

Redes Fuzzy ART e redes SOM compartilham semelhanças em suas arquiteturas e algoritmos de treinamento. Entretanto, algumas diferenças importantes podem ser ressaltadas. A Tabela 3.1 apresenta um resumo comparativo das características das redes Fuzzy ART e SOM.

Ambas as redes podem ser usadas em tarefas de agrupamento de dados e quantização vetorial. No entanto, a rede SOM é capaz de preservar aproximadamente a topologia dos dados de entrada devido sua etapa cooperativa, tornando-a útil em problemas de visualização de dados de dimensões elevadas. A rede Fuzzy ART não possui essa propriedade, pois não dispõe seus neurônios de forma regular nem usa uma função de vizinhança na sua equação de atualização de pesos.

Por outro lado, a rede Fuzzy ART, por criar novos protótipos ao longo do treinamento, é capaz de aprender padrões não-estacionários, característica comum às



(a) Rede inicial

(b) Rede após 1 época de treinamento

(c) Rede após 10 épocas de treinamento

(d) Rede após 100 épocas de treinamento

Figura 3.8: Efeito do treinamento da rede SOM nos pesos dos neurônios.

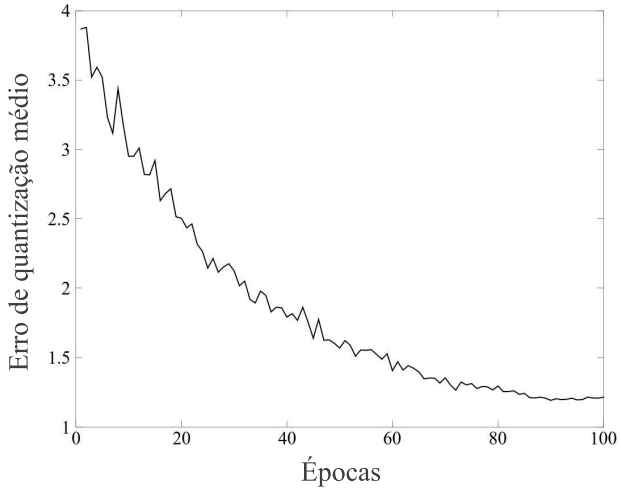


Figura 3.9: Exemplo de convergência do erro médio de quantização durante o treinamento da rede SOM.

Tabela 3.1: Comparação entre as redes Fuzzy ART e SOM.

	Fuzzy ART	**SOM**
Paradigma de aprendizado	Competitivo	Competitivo e cooperativo
Número de neurônios	Incrementado quando necessário	Especificado no início do treinamento
Agrupamento de dados	Sim	Sim
Quantização vetorial	Sim	Sim
Preservação de topologia	Não	Sim
Aprendizado de dados não-estacionários	Sim	Não

redes neurais da família ART (CARPENTER; GROSSBERG; ROSEN, 1991). Já a rede SOM perde a capacidade de aprender com o passar das iterações de treinamento, pois seu passo de aprendizado precisa ser reduzido para garantir convergência. Esse fenômeno revela sua baixa plasticidade e a torna inadequada para aprendizado de distribuições não-estacionárias, pois, neste caso, a rede SOM "esquece" os dados anteriores.

3.2 Redes Neurais Competitivas Supervisionadas

Diferente dos algoritmos de aprendizagem não-supervisionada, redes supervisionadas recebem como entrada a classe correspondente aos padrões de treinamento.

Antes do início da fase de treinamento supervisionado, é preciso dispor de N pares $\{\mathbf{x}(n), \mathbf{y}(n)\}$, $n = 1, \ldots, N$, em que $\mathbf{x}(n) \in \mathbb{R}^D$ é o padrão de entrada da iteração n, $\mathbf{y}(n) \in \mathbb{R}^C$ é o rótulo associado ao padrão $\mathbf{x}(n)$ e C é o número total de classes. Todas as componentes de $\mathbf{y}(n)$ possuem valor 0, com exceção daquela cuja posição corresponde à classe de $\mathbf{x}(n)$, que possui o valor 1.

Por exemplo, para $C = 3$, tem-se três possíveis escolhas para $\mathbf{y}(n)$: $[1\ 0\ 0]^T$,

$[0\ \ 1\ \ 0]^T$ e $[0\ \ 0\ \ 1]^T$, em que cada uma representa uma classe diferente. Essa codificação, chamada 1-*out-of-C*, será a abordagem usada nesta dissertação.

Esta seção detalha as operações de duas técnicas de aprendizado competitivo supervisionado: redes Fuzzy ARTMAP e redes LVQ. Ambas as redes apresentam processos de treinamento competitivo similares aos das redes Fuzzy ART e SOM. Entretanto, além de procurarem similaridades nos padrões apresentados, em ambas as redes o aprendizado é guiado pelo conhecimento das classes associadas aos exemplos de treinamento.

3.2.1 Redes Fuzzy ARTMAP

A rede Fuzzy ARTMAP foi proposta em Carpenter *et al.* (1992) como uma variante da rede ARTMAP (CARPENTER; GROSSBERG; REYNOLDS, 1991) que utiliza os operadores fuzzy da rede Fuzzy ART. Desde então o algoritmo Fuzzy ARTMAP tem sido o mais popular representante da família ART para problemas de aprendizagem supervisionada.

O arquitetura original da rede Fuzzy ARTMAP envolve o treinamento simultâneo de dois módulos Fuzzy ART, sendo cada um deles responsável por associar dois espaços vetoriais distintos, porém relacionados. Em problemas de classificação de padrões um dos espaços é o espaço dos rótulos, enquanto o outro é o espaço de entrada (RAJASEKARAN; PAI, 2000; PALANIAPPAN; ESWARAN, 2009). Em Kasuba (1993) foi realizada uma simplificação na notação da rede Fuzzy ARTMAP, através da redução de redundâncias na arquitetura original. Essa versão é normalmente chamada de *Simplified Fuzzy ARTMAP* (SFAM) e será a utilizada nesta dissertação. Mesmo assim, a denominação Fuzzy ARTMAP será mantida.

3.2.1.1 Arquitetura da rede Fuzzy ARTMAP

A rede Fuzzy ARTMAP usa dois módulos Fuzzy ART, denotados ART_a e ART_b, interligados por uma matriz de pesos. Como apresentado na Seção 3.1.1, cada módulo Fuzzy ART possui duas camadas principais, F_1 e F_2. Entretanto, para fins de classificação de padrões, o módulo ART_b da rede Fuzzy ARTMAP pode ser simplificado em uma única camada F_2^b (CARPENTER, 2003).

A Figura 3.10 ilustra a arquitetura geral da rede Fuzzy ARTMAP. A seguir são detalhados os principais componentes da arquitetura.

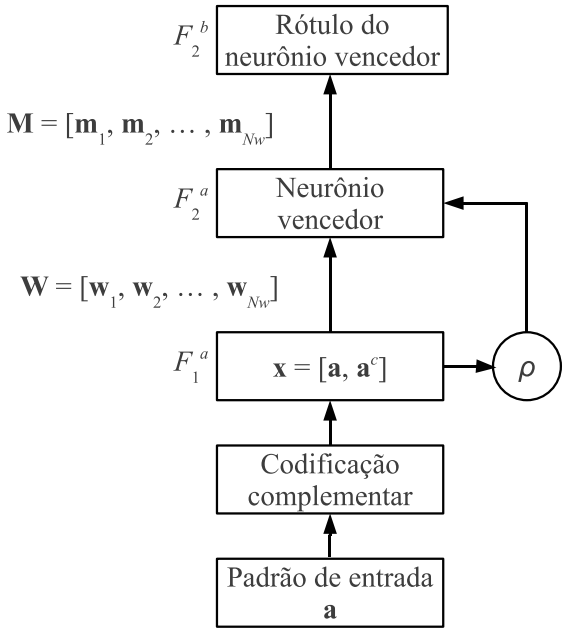


Figura 3.10: Diagrama de blocos da rede Fuzzy ARTMAP. Os índices temporais foram removidos para melhor visualização.

Sinal de entrada. Assim como descrito na Seção 3.1.1, a entrada da rede Fuzzy ARTMAP constitui N vetores $\mathbf{a}(n) \in \mathbb{R}^P$, normalmente transformados via codificação complementar em vetores $\mathbf{x}(n) \in \mathbb{R}^D$, $n = 1, 2, \ldots, N$, em que $D = 2P$. Assim como na rede Fuzzy ART, as D componentes dos vetores de entrada possuem valores limitados entre 0 e 1. Os vetores $\mathbf{x}(n)$ constituem entrada da camada F_1^a do módulo ART_a. Por ser um algoritmo supervisionado, a rede também recebe como entrada os N rótulos das classes associadas aos N padrões de treinamento. Esses rótulos são representados na forma vetorial $\mathbf{y}(n) \in \mathbb{R}^C$, $n = 1, 2, \ldots, N$, em que C é o número de classes possíveis.

Módulo ART_a. O módulo ART_a possui duas camadas, F_1^a e F_2^a. Os vetores de entrada são apresentados à rede pela primeira. A segunda camada é formada pelos protótipos criados ao longo do treinamento. O i-ésimo neurônio corresponde ao vetor de pesos $\mathbf{w}_i(n) \in \mathbb{R}^D$ na iteração n da fase de treino.

Módulo ART_b. O módulo ART_b se resume à camada F_2^b, onde é definida a classe do padrão de entrada atual.

Matriz Inter-MAP. Entre as camadas F_2^a e F_2^b existe uma matriz de pesos $\mathbf{M}$, chamada Inter-MAP, que associa aos protótipos de F_2^a uma classe na camada F_2^b. Seja N_w o número de protótipos em F_2^a na iteração n do treinamento,

a matriz Inter-MAP $\mathbf{M}(n) = [\mathbf{m}_1(n)\ \mathbf{m}_2(n)\ \cdots\ \mathbf{m}_{N_w}(n)]$ possui dimensão $C \times N_w$. As colunas da matriz $\mathbf{M}(n)$ determinam na camada F_2^b a classe do padrão de entrada, possuindo construção idêntica aos vetores de rótulo de entrada $\mathbf{y}(n)$, ou seja, apenas uma componente do vetor $\mathbf{m}_i(n)$ é igual a 1, enquanto as outras são iguais a zero.

3.2.1.2 Treinamento da rede Fuzzy ARTMAP

O processo de treinamento da rede Fuzzy ARTMAP possui semelhanças com o da rede Fuzzy ART. De fato, as etapas de codificação da entrada, processo de competição e critério de vigilância são realizadas como descritas na Seção 3.1.1. Os passos seguintes são descritos a seguir.

Critério de predição. Depois de um neurônio vencedor i^* passar no critério de vigilância (ver Seção 3.1.1), verifica-se o critério de predição. Esse novo teste consiste em verificar se o vetor $\mathbf{m}_{i^*}(n)$ prediz exatamente a saída desejada $\mathbf{y}(n)$ para a entrada atual, $\mathbf{x}(n)$. Se o teste falhar, o valor do parâmetro de vigilância é modificado através da adição de uma pequena constante $\epsilon \to 0$, ou seja,

$$\rho = \frac{|\mathbf{x}(n) \wedge \mathbf{w}_{i^*}(n)|}{|\mathbf{x}(n)|} + \epsilon. \tag{3.22}$$

Os testes de vigilância e de predição são repetidos até que um neurônio vencedor passe em ambos os testes ou todos os neurônios da rede tenham sido testados, caso em que um neurônio ainda não usado (*uncommitted*) é escolhido como vencedor.

Atualização dos pesos. Caso um neurônio já usado tenha sido escolhido como vencedor, os seus pesos são atualizados pela seguinte equação:

$$\mathbf{w}_{i*}(n+1) = \eta\left(\mathbf{w}_{i^*}(n) \wedge \mathbf{x}(n)\right) + (1-\eta)\mathbf{w}_{i^*}(n). \tag{3.23}$$

Caso o neurônio vencedor não tenha ainda sido usado (i.e. $\mathbf{w}_{i^*}(n) = \mathbf{1}^D$), o mesmo recebe o vetor de entrada atual:

$$\mathbf{w}_{i^*}(n+1) = \mathbf{x}(n), \tag{3.24}$$

sua coluna correspondente em $\mathbf{M}(n)$ é atualizada, ou seja,

$$\mathbf{m}_{i^*}(n+1) = \mathbf{y}(n), \tag{3.25}$$

e um novo protótipo é adicionado à rede:

$$N_w = N_w + 1, \tag{3.26}$$
$$\mathbf{w}_{N_w} = \mathbf{1}^D, \tag{3.27}$$
$$\mathbf{m}_{N_w} = \mathbf{0}^C. \tag{3.28}$$

Classificação de um vetor de entrada. Na fase de teste da rede Fuzzy ARTMAP, determina-se um neurônio vencedor em relação a um vetor de entrada desconhecido a partir da Equação (3.4). Após o protótipo $\mathbf{w}_{i^*}$ ser escolhido como vencedor, a classe inferida é aquela determinada por $\mathbf{m}_{i^*}$.

Os passos de treinamento são repetidos para todos os N pares $\{\mathbf{x}(n), \mathbf{y}(n)\}$ disponíveis, sempre retornando o parâmetro de vigilância ρ a um valor base $\overline{\rho}$ ao se apresentar um novo padrão. Um resumo do treinamento da Fuzzy ARTMAP encontra-se no Algoritmo 3.3.

3.2.1.3 Interpretação geométrica da Rede Fuzzy ARTMAP

Analisando o algoritmo de treinamento da rede Fuzzy ARTMAP, verifica-se que trata-se do algoritmo da rede Fuzzy ART adicionado de um critério de predição que leva em consideração a matriz Inter-MAP e os rótulos conhecidos dos padrões de entrada.

A análise geométrica feita na Seção 3.1.1 para a rede Fuzzy ART também pode ser feita para a rede Fuzzy ARTMAP. A Figura 3.11 ilustra um exemplo bidimensional em que a rede foi treinada de maneira a diferenciar padrões de duas classes. Os parâmetros usados foram $\beta = 0$, $\rho_0 = 0{,}6$ e $\eta = 1$. Observa-se que os protótipos criados são vinculados a rótulos específicos, representados na Figura 3.11 por cores diferentes e no algoritmo de treinamento pelas colunas da matriz Inter-MAP $\mathbf{M}$.

É relevante mencionar que a análise feita na Seção 3.1.1 sobre o efeito dos parâmetros ρ, α e η no treinamento da rede Fuzzy ART também se aplicam à rede Fuzzy ARTMAP. Além disso, a rede Fuzzy ARTMAP compartilha das propriedades

Algoritmo 3.3 Algoritmo de treinamento da rede Fuzzy ARTMAP.

Constantes

β: parâmetro de escolha, $\beta \geq 0$

ρ_0: valor base do parâmetro de vigilância ρ, $0 < \rho_0 \leq 1$

η: parâmetro de aprendizado, $0 < \eta \leq 1$

$n_{\text{MÁX}}$: número de iterações de treinamento

Entradas

$\mathbf{a}(n)$: vetor de entrada, dimensão P

$\mathbf{x}(n)$: vetor de entrada, dimensão $D = 2P$ (codificação complementar)

$\mathbf{y}(n)$: rótulo do padrão $\mathbf{x}(n)$, dimensão C

Algoritmo

1. Inicialização ($n = 0$)

Criar e inicializar os pesos do neurônio inicial da rede $\mathbf{w}_1(0) = \mathbf{1}^D$

Criar e inicializar os pesos da matriz Inter-MAP, $\mathbf{M}(0) = \mathbf{m}_1(0) = \mathbf{0}^C$

2. Laço temporal ($n = 1, 2, \ldots, n_{\text{MÁX}}$)

2.1 Selecionar $\mathbf{x}(n)$ do conjunto de vetores de entrada

2.2 Buscar pelo índice do neurônio vencedor:

$i^* = \arg\max_i\{t_i\}$, em que $t_i(n) = \frac{|\mathbf{x}(n) \wedge \mathbf{w}_i(n)|}{\beta + |\mathbf{w}_i(n)|}, \quad i = 1, 2, \ldots, N_w$

2.3 Teste de ressonância (critério de vigilância)

SE $|\mathbf{x}(n) \wedge \mathbf{w}_{i^*}(n)| > \rho|\mathbf{x}(n)|$, ir para o Passo 2.5

SENÃO, voltar para o Passo 2.2 e buscar um novo neurônio vencedor

2.4 Teste de predição

SE $\mathbf{m}_{i^*}(n) = \mathbf{y}(n)$, ir para o Passo 2.4

SENÃO, fazer $\rho = \frac{|\mathbf{x}(n) \wedge \mathbf{w}_{i^*}(n)|}{|\mathbf{x}(n)|} + \epsilon$ e voltar para o Passo 2.2

2.5 Atualização dos pesos

SE $\mathbf{w}_{i^*}(n) = \mathbf{1}^D$ (i.e., o vencedor nunca foi ativado antes), FAZER

$\mathbf{w}_{i^*}(n+1) = \mathbf{x}(n)$ (i.e., armazena o novo padrão)

$\mathbf{m}_{i^*}(n+1) = \mathbf{y}(n)$ (i.e., armazena a classe do novo padrão)

$N_w = N_w + 1$, $\mathbf{w}_{N_w} = \mathbf{1}^D$ e $\mathbf{m}_{N_w} = \mathbf{0}^C$

SENÃO FAÇA

$$\mathbf{w}_{i*}(n+1) = \eta\left(\mathbf{w}_{i^*}(n) \wedge \mathbf{x}(n)\right) + (1-\eta)\mathbf{w}_{i^*}(n)$$

Saídas

$\mathbf{m}_{i^*}(n)$: vetor de pesos que codifica o rótulo da classe do vetor de entrada na iteração n

Observações

Tipicamente, usa-se a técnica de codificação complementar para pré-processar $\mathbf{a}(n)$.

O número de neurônios é iniciado como $N_w = 1$ e incrementado ao longo das iterações.

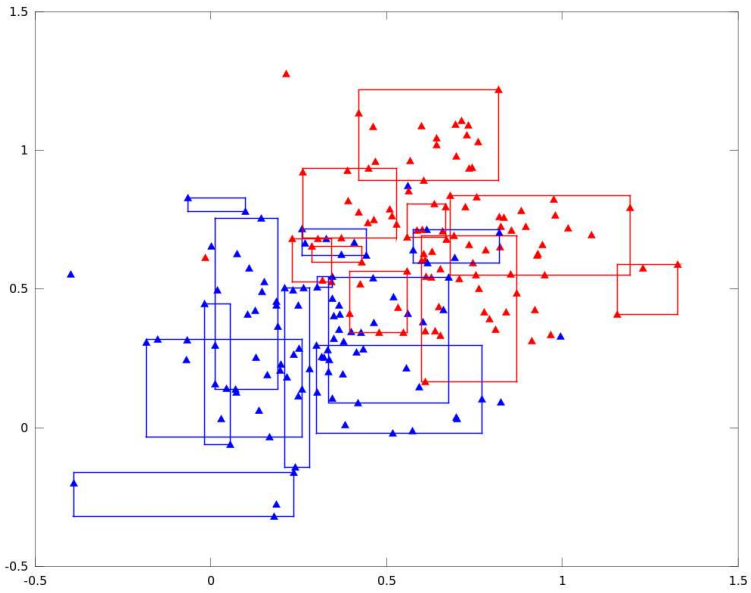


Figura 3.11: Exemplo de operação da rede Fuzzy ARTMAP. As regiões retangulares representam as áreas de influência de cada protótipo da rede. As cores diferentes representam classes diferentes.

das redes neurais da família ART, como a capacidade de realizar aprendizado contínuo e a capacidade de lidar com distribuições não-estacionárias.

3.2.2 Redes *Learning Vector Quantization* (LVQ)

Também proposta por Kohonen (1988a), uma rede LVQ, assim como a rede SOM, promove competição entre seus neurônios, cuja quantidade é definida logo no início do projeto da rede. Entretanto, o aprendizado de uma rede LVQ é supervisionado, pois conta com o conhecimento *a priori* das classes dos exemplos de treinamento.

Os algoritmos de treinamento de redes LVQ não apresentam uma etapa de cooperação como na rede SOM, em que uma vizinhança em torno do neurônio vencedor é definida. Dessa maneira, não é esperada uma organização espacial dos neurônios que compõem à rede LVQ, mas sim o aprendizado de um dicionário (*codebook*) que represente de forma compacta os dados de entrada considerando os seus rótulos.

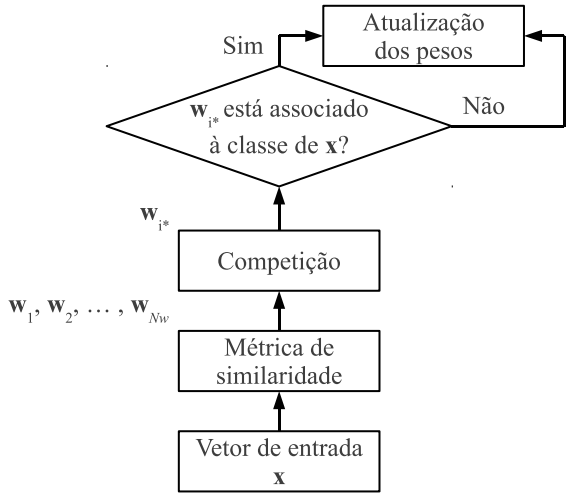


Figura 3.12: Diagrama de blocos de uma rede LVQ. Os índices temporais foram omitidos para melhor visualização.

3.2.2.1 Arquitetura geral das redes LVQ

O padrão de entrada $\mathbf{x}(n) \in \mathbb{R}^D$ no instante n é recebido simultaneamente na rede LVQ por todos os seus N_w neurônios, definidos pelos vetores de pesos $\mathbf{w}_i(n) \in \mathbb{R}^D$.

Inicialmente os N_w vetores de pesos correspondentes aos neurônios da rede recebem valores aleatórios. Cada neurônio é então aleatoriamente rotulado com uma das classes do problema. É comum escolher uma distribuição uniforme do número de neurônios entre as classes disponíveis. Alternativamente pode-se escolher aleatoriamente amostras de treinamento de diferentes classes para inicializar os pesos da rede. Cada neurônio é então previamente associado a uma das C classes existentes no conjunto de dados analisado. Note que podem haver múltiplos neurônios associados a uma mesma classe.

A Figura 3.12 apresenta um diagrama de blocos de uma rede LVQ. Percebe-se que após o processo de competição é verificado se o neurônio vencedor está associado ou não à classe do vetor de treinamento. A atualização dos pesos do neurônio vencedor é feita nos dois casos, mas de maneira diferente, como será detalhado na próxima seção.

Considerando a medida de similaridade como a distância euclidiana quadrática e $d_i(n)$ a distância quadrática entre o vetor de entrada e o protótipo $\mathbf{w}_i(n)$, tem-se

a seguinte métrica:

$$d_i(n) = \|\mathbf{x}(n) - \mathbf{w}_i(n)\|^2, \qquad \forall i = 1, 2, \ldots, N_w. \tag{3.29}$$

A Equação (3.29) pode ser expandida:

$$\begin{aligned} \|\mathbf{x}(n) - \mathbf{w}_i(n)\|^2 &= (\mathbf{x}(n) - \mathbf{w}_i(n))^T(\mathbf{x}(n) - \mathbf{w}_i(n)), & (3.30)\\ &= \mathbf{x}(n)^T\mathbf{x}(n) - 2\mathbf{w}_i(n)^T\mathbf{x}(n) + \mathbf{w}_i(n)^T\mathbf{w}_i(n). & (3.31) \end{aligned}$$

Mas o valor $\mathbf{x}(n)^T\mathbf{x}(n)$ é constante para todos os protótipos $\mathbf{w}_i(n)$. Dessa maneira, a métrica usada na etapa de competição, denotada por $\zeta_i(n)$ será dada por

$$\begin{aligned} \zeta_i(n) &= -2\mathbf{w}_i(n)^T\mathbf{x}(n) + \|\mathbf{w}_i(n)\|^2, & (3.32)\\ &= \mathbf{a}_i(n)^T\mathbf{x}(n) + b_i(n), & (3.33) \end{aligned}$$

em que $\mathbf{a}_i(n)^T = -2\mathbf{w}_i(n)^T$ e $b_i(n) = \|\mathbf{w}_i(n)\|^2$. Note que a Equação (3.33) é a equação de um hiperplano. Como $\zeta_i(n)$ é linear em relação ao vetor de entrada, pode-se afirmar que o classificador resultante de uma rede LVQ é linear.

À medida que os vetores de pesos referentes aos neurônios são atualizados, formam-se regiões no espaço de dados de entrada. Estas regiões são delimitadas por hiperplanos individualmente lineares (cujas equações são semelhantes à Equação (3.33)), como em diagramas de Voronoi (KOHONEN, 1997). Na Figura 3.13 é ilustrado um exemplo de diagrama de Voronoi tradicional. Pode-se observar que as regiões são divididas por segmentos de reta, consequência dos dados serem bidimensionais. Tais segmentos são equidistantes em relação a protótipos vizinhos, garantindo sua máxima separação.

Em seu livro, Kohonen (1997) apresenta diversas variantes do método de treinamento LVQ original (denominado LVQ1), tal como OLVQ1 (*Optimized*-LVQ1), LVQ2, LVQ2.1 e LVQ3. Após testes preliminares, decidiu-se que nesta dissertação será usado o algoritmo OLVQ1, detalhado na seção seguinte.

3.2.2.2 Algoritmo OLVQ1

Dado o par de entrada $\{\mathbf{x}(n), \mathbf{y}(n)\}$ da n-ésima iteração, são realizados os passos abaixo.

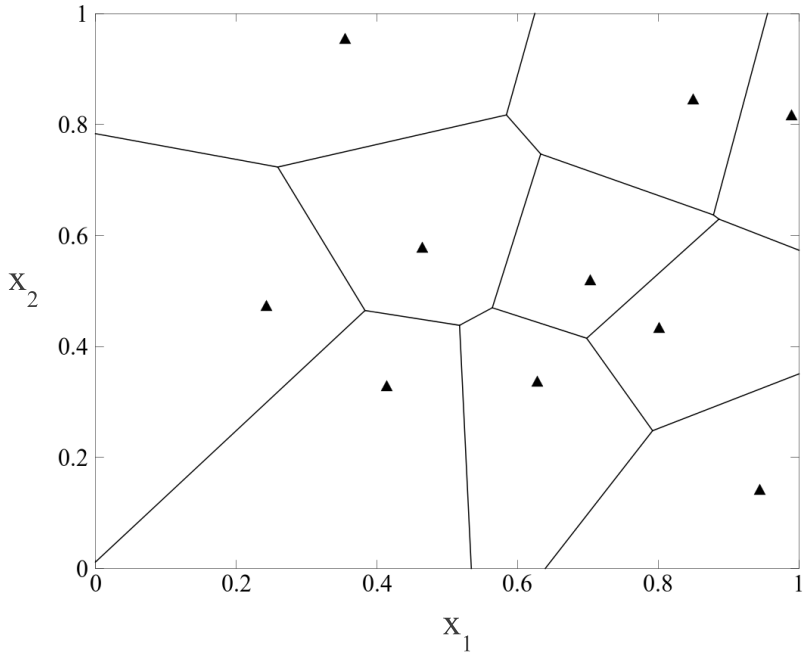


Figura 3.13: Exemplo de diagrama de Voronoi para dados bidimensionais. As regiões poderiam, por exemplo, representar classes diferentes caracterizadas pelos seus protótipos.

Processo de competição. Verifica-se qual o neurônio i^* é o mais próximo da entrada $\mathbf{x}(n)$ de acordo com a seguinte regra:

$$i^*(n) = \arg\min_{\forall i} \|\mathbf{x}(n) - \mathbf{w}_i(n)\|, \tag{3.34}$$

em que $\|\cdot\|$ denota a norma euclidiana.

Atualização dos pesos. Somente os pesos do neurônio vencedor são atualizados segundo a expressão abaixo

$$\mathbf{w}_{i^*}(n+1) = \mathbf{w}_{i^*}(n) + s(n)\eta_{i^*}(n)[\mathbf{x}(n) - \mathbf{w}_{i^*}(n)], \tag{3.35}$$

em que $s(n) = 1$, caso o rótulo associado ao vetor $\mathbf{w}_{i^*}(n)$ seja $\mathbf{y}(n)$, i.e. o mesmo associado a $\mathbf{x}(n)$, ou $s(n) = -1$, caso o neurônio vencedor e o padrão de entrada não pertençam à mesma classe.

No algoritmo LVQ1 original existe apenas um parâmetro de aprendizagem $0 < \eta \leq 1$, comum a todos os neurônios. Na versão OLVQ1, cada neurônio possui um parâmetro próprio, $0 < \eta_i(n) \leq 1$. O parâmetro referente ao neurônio

vencedor é atualizado em cada iteração por meio da seguinte expressão:

$$\eta_{i^*}(n+1) = \frac{\eta_{i^*}(n)}{1 + s(n)\eta_{i^*}(n)}, \tag{3.36}$$

em que $\eta_{i^*}(n+1)$ deve ser mantido abaixo do valor unitário. Essa condição garante uma convergência mais veloz durante o treinamento (KOHONEN, 1997). O valor inicial $\eta_i(0) = \eta_0$ deve ser especificado antes da execução do algoritmo.

A Figura 3.14 ilustra um exemplo em que dados bidimensionais divididos em duas classes são treinados por uma rede OLVQ1 em que foram usados 4 protótipos por classe, $\eta_0 = 0{,}3$ e uma época de treinamento. Percebe-se que os protótipos (representados por asteriscos) tendem a se posicionar próximos aos centróides dos grupos (*clusters*) da classe que representam. Os diagramas de Voronoi formados são semelhantes aos ilustrados na Figura 3.13, mas na Figura 3.14 as regiões convexas são associadas ao rótulo do protótipo que contêm.

A etapa de teste da rede OLVQ1 consiste em encontrar o protótipo mais próximo do vetor de entrada desconhecido $\mathbf{x}_{novo}$ através da Equação (3.34) e atribuir a este vetor a classe associada ao neurônio vencedor, ou seja,

$$\text{Se } \|\mathbf{x}_{novo} - \mathbf{w}_{i^*}\| < \|\mathbf{x} - \mathbf{w}_i\|, \qquad \forall i \neq i^* \tag{3.37}$$

$$\text{Então } classe(\mathbf{x}_{novo}) = classe(\mathbf{w}_{i^*}). \tag{3.38}$$

O Algoritmo 3.4 resume o processo de treinamento da rede OLVQ1.

3.3 Conclusões

Neste capítulo foram apresentadas algumas das redes neurais competitivas mais conhecidas: Fuzzy ART, SOM, Fuzzy ARTMAP e LVQ. Seus algoritmos de treinamento foram detalhados e suas propriedades mais importantes foram ressaltadas.

Enquanto as redes não-supervisionadas Fuzzy ART e SOM realizam o processo de treinamento somente com base nos padrões encontrados a partir dos vetores de atributos na entrada, as redes supervisionadas Fuzzy ARTMAP e LVQ são orientadas durante o treinamento pelas classes associadas aos exemplos disponíveis.

Algoritmo 3.4 Algoritmo de treinamento da rede OLVQ1.

Constantes

N_w: número de neurônios da rede

η_0: valor inicial dos parâmetros de aprendizado $\eta_i(n),\ i = 1, 2, \ldots, N_w$

$n_{\text{MÁX}}$: número de iterações de treinamento

Entradas

$\mathbf{x}(n)$: padrão de entrada, dimensão D

$\mathbf{y}(n)$: rótulo do padrão $\mathbf{x}(n)$, dimensão C

Algoritmo

1. Inicialização

Inicializar os vetores $\mathbf{w}_i(0)$ com vetores selecionados aleatoriamente do conjunto de treinamento

Faça $\eta_i(1) = \eta_0\ (i = 1, 2, \ldots, N_w)$

2. Laço temporal ($n = 1, 2, \ldots, n_{\text{MÁX}}$)

2.1 Selecionar $\mathbf{x}(n)$ do conjunto de vetores de entrada

2.2 Encontrar o índice do neurônio vencedor

$$i^*(n) = \arg\min_{\forall i} \|\mathbf{x}(n) - \mathbf{w}_i(n)\|$$

2.3 Atualizar pesos do neurônio vencedor

Caso $classe(\mathbf{w}_{i^*}(n)) = \mathbf{y}(n)$, faça $s(n) = 1$,

caso contrário, $s(n) = -1$

$$\mathbf{w}_{i^*}(n+1) = \mathbf{w}_{i^*}(n) + s(n)\eta_{i^*}(n)[\mathbf{x}(n) - \mathbf{w}_{i^*}(n)]$$

Atualizar o passo de aprendizado do neurônio vencedor

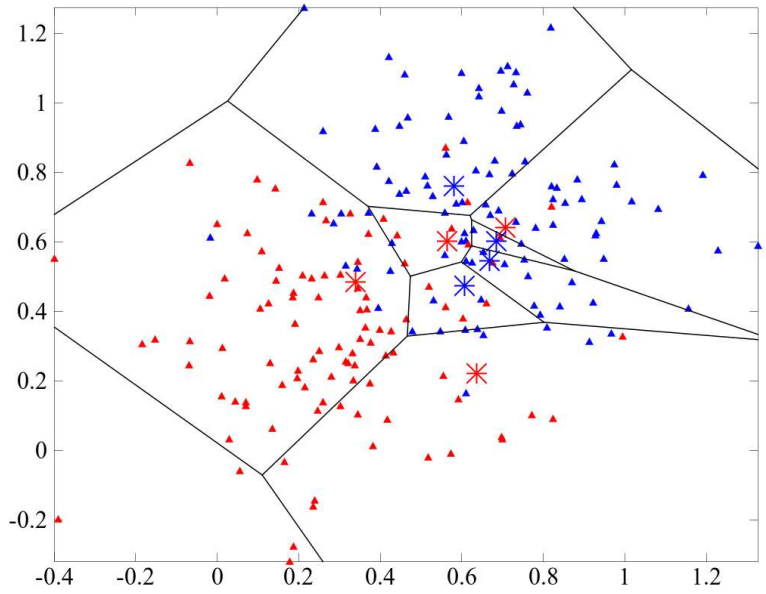
$$\eta_{i^*}(n+1) = \frac{\eta_{i^*}(n)}{1+s(n)\eta_{i^*}(n)}$$

Saídas

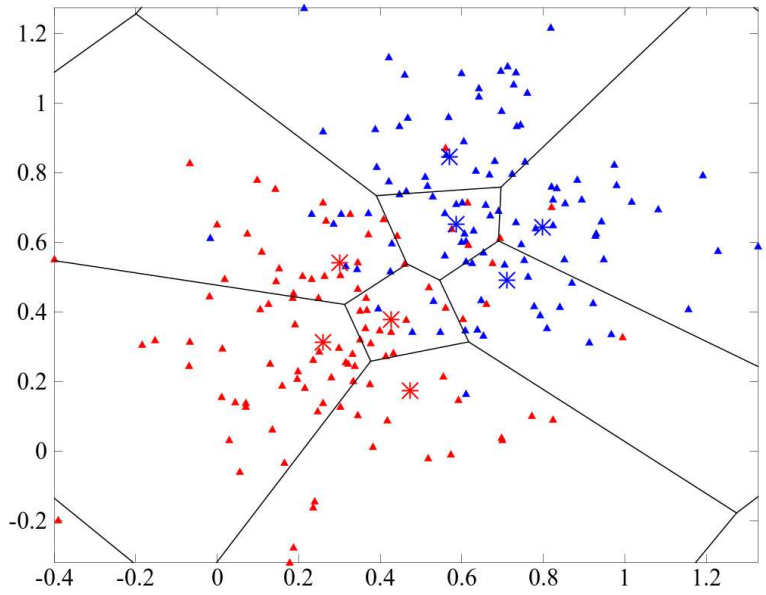
$classe(\mathbf{w}_{i^*}(n))$: rótulo do neurônio vencedor na iteração n

Observações

Durante a fase de teste, ou seja, com a apresentação de vetores de entrada desconhecidos, o Passo 2.3 não é realizado.



(a) Rede inicial.



(b) Rede após treinamento.

Figura 3.14: Exemplo de aplicação da rede OLVQ1 a um conjunto de dados bidimensionais. As cores representam classes diferentes, enquanto os segmentos de reta separam as regiões mapeadas pela rede. Os asteriscos representam as posições dos protótipos da rede.

No Capítulo 4 serão discutidas técnicas que tornam possível algoritmos não-supervisionados, como as redes Fuzzy ART e SOM, serem usados em problemas de classificação de padrões. Mais adiante, no Capítulo 6, serão detalhadas as diretivas usadas na construção de comitês de classificadores a partir dos métodos de aprendizagem competitiva comentados neste capítulo.

Capítulo 4

Arquiteturas ARTIE e MUSCLE

Neste capítulo serão apresentados alguns métodos que permitem a aplicação de algoritmos de aprendizagem não-supervisionada em problemas de classificação supervisionada de padrões. Os métodos descritos são Rotulação *a Posteriori* por Voto Majoritário, Rotulação *a Priori* por Redes Individuais e Rotulação Auto-Supervisionada.

Este capítulo detalha ainda uma das propostas desta dissertação, que é a utilização das redes SOM e Fuzzy ART em comitês de classificadores, dando origem às arquiteturas MUSCLE e ARTIE.

4.1 Redes Neurais Não-Supervisionadas para Classificação

Antes da aplicação de redes neurais não-supervisionadas em problemas de classificação, é preciso adicionar etapas extras ao seus algoritmos de treinamento. As técnicas apresentadas neste capítulo foram exploradas por outros pesquisadores em redes SOM (MONTEIRO *et al.*, 2006), mas seu uso em redes Fuzzy ART não está documentado.

Assim como no Capítulo 3, a informação de rótulo será representada por um vetor binário $\mathbf{y}(n) \in \mathbb{R}^C$ (considerando-se C classes possíveis) de comprimento unitário, i.e., apenas um de seus elementos possui o valor 1, enquanto os demais possuem o valor 0. O índice do elemento com valor 1 corresponde à classe do padrão representado pelo vetor de entrada $\mathbf{x}(n) \in \mathbb{R}^D$. Por exemplo, se existem três classes ao todo, ou seja, $C = 3$, então três vetores de rótulos são possíveis: um para a primeira classe, $[1\ 0\ 0]^T$, outro para a segunda classe, $[0\ 1\ 0]^T$ e um para a terceira

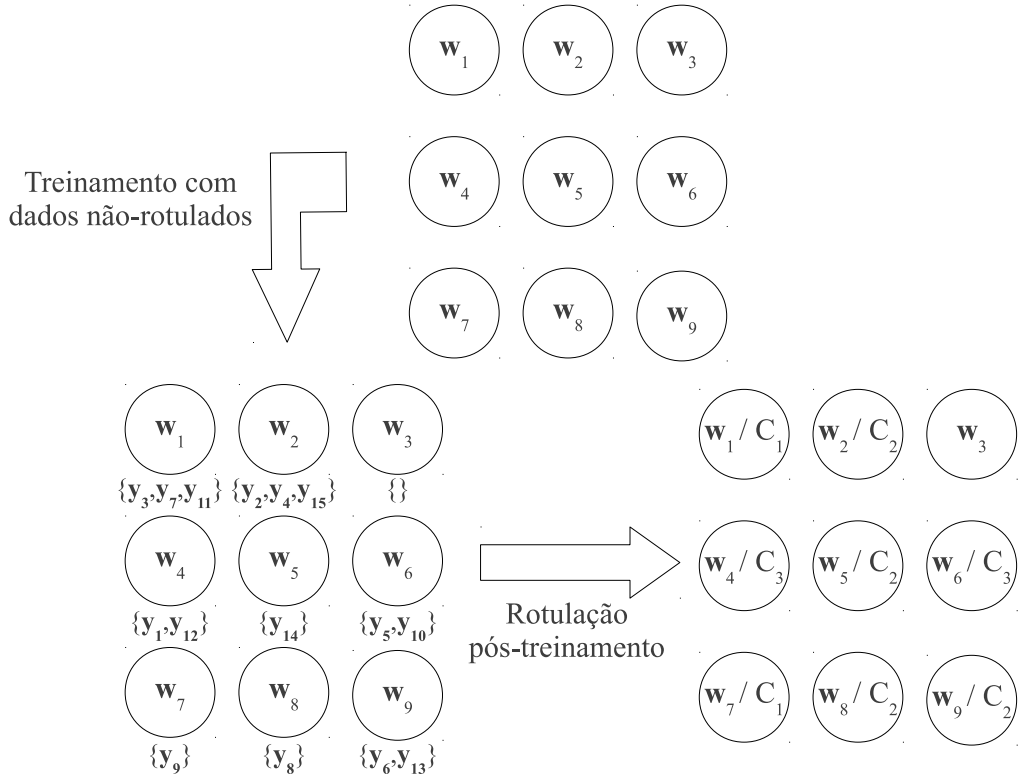


Figura 4.1: Ilustração da rotulação *a posteriori* por voto majoritário. Após o treinamento não-supervisionado com os dados não-rotulados, cada neurônio é associado a uma lista de vetores de entrada $\mathbf{x}(n)$, cujos rótulos são dados pelos vetores $\mathbf{y}_n$. A rotulação pós-treinamento dos neurônios segue a regra do voto majoritário, resultando em uma classe sendo associada a cada neurônio. Note que pode haver neurônios que não são associados a nenhuma classe.

classe, $[0\ 0\ 1]^T$.

As três estratégias apresentadas a seguir serão identificadas posteriormente pelos sufixos Ci, i $\in \{1, 2, 3\}$.

4.1.1 Rotulação *a Posteriori* por Voto Majoritário (C1)

Nesse método as redes SOM ou Fuzzy ART são treinadas inicialmente da maneira não-supervisionada usual. Em seguida, um processo de rotulação dos neurônios (protótipos) pós-treinamento é feito apresentando-se os exemplos de treinamento novamente à rede e determinando os neurônios vencedores para cada vetor de entrada. Esse processo é feito de acordo com a Equação (3.15), para redes SOM, ou com a Equação (3.4), para redes Fuzzy ART. Note que na etapa de rotulação os pesos dos neurônios não são alterados. A Figura 4.1 ilustra o efeito da rotulação descrita.

Seja $\mathcal{X}$ um conjunto de pares $\{\mathbf{x}(n), \mathbf{y}(n)\}$, em que $n = 1, \ldots, N$. Seja $\mathcal{X}_i$ o conjunto de n_i pares de treinamento que foram mapeados no i-ésimo neurônio (pela expressão $i = \arg\min_{\forall l}\{\|\mathbf{x}_j - \mathbf{w}_l\|\}$), ou seja,

$$\mathcal{X}_i = \left\{(\mathbf{x}_1^{(i)}, \mathbf{y}_1^{(i)}), (\mathbf{x}_2^{(i)}, \mathbf{y}_2^{(i)}), \cdots, (\mathbf{x}_{n_i}^{(i)}, \mathbf{y}_{n_i}^{(i)})\right\}. \tag{4.1}$$

Note que os vetores $\mathbf{y}_j^{(i)}$ devem possuir apenas uma componente com o valor unitário, sendo o resto igual a zero.

Seja ainda o vetor de agregação $\mathbf{y}^{(i)} \in \mathbb{R}^C$ definido para o i-ésimo neurônio:

$$\mathbf{y}^{(i)} = \sum_{j=1}^{n_i} \mathbf{y}_j^{(i)} = \begin{bmatrix} y_1^{(i)} \\ y_2^{(i)} \\ \vdots \\ y_k^{(i)} \\ \vdots \\ y_C^{(i)} \end{bmatrix} \tag{4.2}$$

em que C é o número de classes.

A classe associada ao i-ésimo neurônio é aquela com maior número de ocorrências em $\mathcal{X}_i$. Matematicamente, tem-se a seguinte regra de atribuição:

$$classe(\mathbf{w}_i) = \arg\max_{\forall k}\{y_k^{(i)}\}. \tag{4.3}$$

Por exemplo, seja o conjunto $\mathcal{X}_1 = \left\{(\mathbf{x}_1^{(1)}, \mathbf{y}_1^{(1)}), (\mathbf{x}_2^{(1)}, \mathbf{y}_2^{(1)}), (\mathbf{x}_3^{(1)}, \mathbf{y}_3^{(1)})\right\}$ referente ao neurônio $\mathbf{w}_1$, em que $n_1 = 3$. Sejam os vetores $\mathbf{y}_n^{(1)}$ dados por:

$$\mathbf{y}_1^{(1)} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{y}_2^{(1)} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{y}_3^{(1)} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \tag{4.4}$$

Assim, o vetor de agregação $\mathbf{y}^{(1)}$ é dado por

$$\mathbf{y}^{(1)} = \sum_{j=1}^{n_1} \mathbf{y}_j^{(1)} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ 1 \end{bmatrix}. \tag{4.5}$$

Daí, tem-se

$$classe(\mathbf{w}_1) = \arg \max_{k=1,2,3} \{y_k^{(1)}\} = 1. \tag{4.6}$$

Caso haja empate na Equação (4.3), em geral, uma das classes concorrentes é aleatoriamente escolhida. É possível ainda rotular o neurônio em que houve o empate com base no critério do vizinho mais próximo ou ainda não rotulá-lo, denotando-o como classe de rejeição durante a fase de teste..

Como exemplificado na Figura 4.1, existe a possibilidade de haver neurônios não rotulados por não serem sido escolhidos como vencedores durante a fase de rotulação. Caso um desses neurônios seja escolhido na etapa de teste, o mesmo é rejeitado e a busca por um neurônio rotulado continua.

Em relação às redes SOM, essa estratégia tem sido utilizada em problemas de classificação de padrões em vários trabalhos (SUGANTHAN, 1999; LAHA; PAL, 2001; CHRISTODOULOU; MICHAELIDES; PATTICHIS, 2003; WYNS *et al.*, 2004; MONTEIRO *et al.*, 2006). Em redes Fuzzy ART, no entanto, não se tem conhecimento de trabalhos que utilizem esta estratégia de rotulação de neurônios.

4.1.2 Rotulação *a Priori* por Redes Individuais (C2)

Nessa segunda abordagem, uma rede SOM (ou Fuzzy ART) é treinada para cada classe disponível no problema em questão. Cada rede é treinada separadamente, de forma independente e da maneira não-supervisionada usual, usando somente os dados (vetores) de treinamento daquela classe. Antes da fase de treinamento, no entanto, é preciso separar os exemplos disponíveis por classe. Em seguida todos as amostras de treino são direcionadas para a rede neural correspondente. A Figura 4.2 ilustra a etapa de treinamento descrita.

Sejam $\mathbf{x}(n), n = 1, \ldots, N$, os vetores de treinamento disponíveis. Inicialmente separa-se os conjuntos $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_C$, em que $\mathcal{X}_i = \{\mathbf{x}(n) | classe(\mathbf{x}(n)) = C_i\}$ e C é o número de classes.

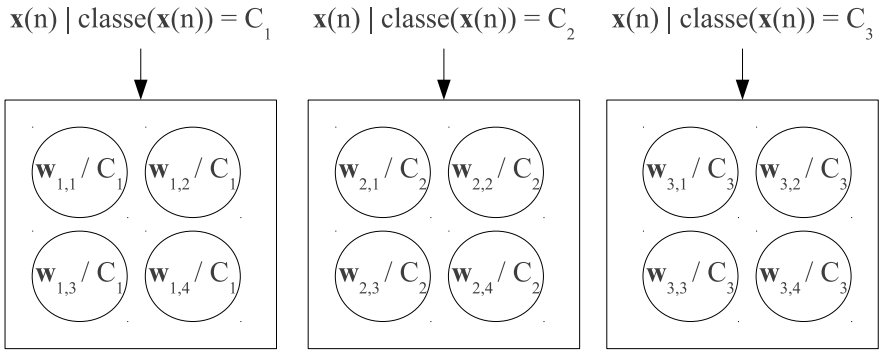


Figura 4.2: Ilustração da etapa de treinamento da rotulação *a priori* por redes individuais. Note que os neurônios de uma mesma rede estão associados todos a uma mesma classe desde antes do processo de treinamento. Além disso, cada rede só é treinada com os vetores de entrada associados à classe que representa.

Sejam ainda as matrizes de pesos $\mathbf{W}_i = [\mathbf{w}_{i,1}\ \mathbf{w}_{i,2}\ \cdots\ \mathbf{w}_{i,N_{wi}}], i = 1, \ldots, C$, referentes a C redes distintas. Treina-se os N_{wi} pesos da i-ésima rede com os vetores do i-ésimo subconjunto $\mathcal{X}_i$ de maneira não-supervisionada. Note que a informação de classe dos vetores de entrada não é mais necessária, pois já foi utilizada para separar os subconjuntos $\mathcal{X}_i$. Daí o termo *rotulação a priori*, pois os neurônios das redes são rotulados antes do treinamento.

A etapa de teste consiste em realizar a seguinte atribuição à classe do padrão desconhecido $\mathbf{x}^*$:

$$k^* = classe(\mathbf{x}^*) = \arg\min_{\forall k}\{\|\mathbf{x}^* - \mathbf{w}_k^*\|\}, \tag{4.7}$$

em que $\mathbf{w}_k^*$ é o neurônio vencedor da k-ésima rede, encontrado pela Equação (3.15), no caso da rede SOM, ou pela Equação (3.4), no caso da rede Fuzzy ART. A Figura 4.3 ilustra essa fase de teste.

Assim como a estratégia C1, a estratégia C2 também tem sido usada já há algum tempo no projeto de classificadores de padrões baseados na rede SOM (SOUZA JÚNIOR; BARRETO; VARELA, 2011; BIEBELMANN; KÖPPEN; NICKOLAY, 1996); entretanto, essa técnica não tem sido usada em conjunção com redes Fuzzy ART, ou em qualquer outra rede não-supervisionada da família ART.

4.1.3 Rotulação Auto-Supervisionada (C3)

Uma terceira estratégia consiste em tornar o treinamento de uma rede SOM (ou Fuzzy ART) supervisionado ao adicionar a informação de rótulo a cada vetor do conjunto de treinamento. Tem-se agora o vetor de entrada $\mathbf{x}(n)$ formado pela

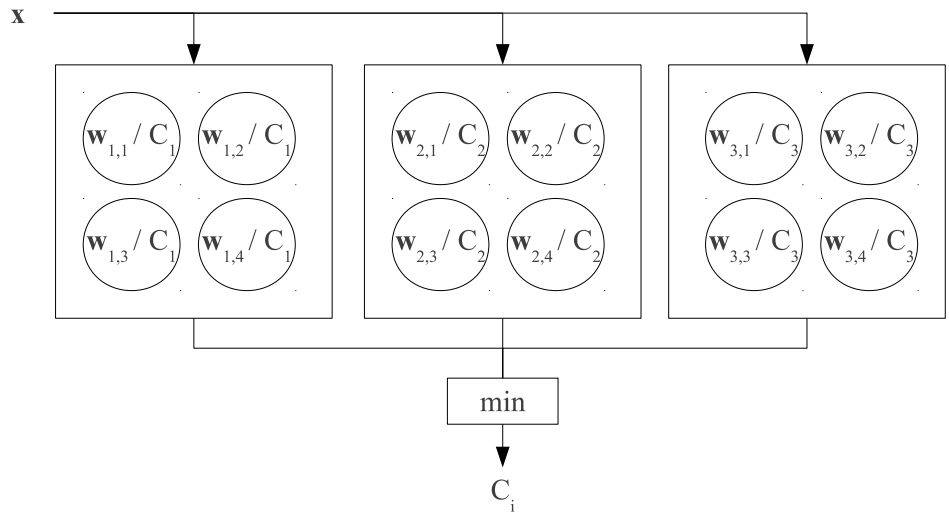


Figura 4.3: Ilustração da etapa de teste da rotulação *a priori* por redes individuais. O neurônio vencedor é buscado em todas as redes, sendo a classe deste escolhida para estimar a classe do padrão desconhecido.

concatenação de dois outros vetores: $\mathbf{x}^p(n)$ e $\mathbf{x}^l(n)$, em que $\mathbf{x}^p(n) \in \mathbb{R}^D$ é o próprio vetor de atributos e $\mathbf{x}^l(n) \in \mathbb{R}^C$ é o seu rótulo correspondente, ou seja, $\mathbf{x}^l(n)$ possui valor 1 somente na componente referente à classe de $\mathbf{x}^p(n)$ e zero nas demais.

Antes do algoritmo de treinamento ser iniciado, os dois vetores mencionados são concatenados, formando um vetor de entrada aumentado:

$$\mathbf{x}(n) = \begin{bmatrix} \mathbf{x}^p(n) \\ \mathbf{x}^l(n) \end{bmatrix}, \qquad \mathbf{x}(n) \in \mathbb{R}^{D+C}. \tag{4.8}$$

Os vetores de pesos correspondentes são formados de maneira semelhante:

$$\mathbf{w}_i(n) = \begin{bmatrix} \mathbf{w}_i^p(n) \\ \mathbf{w}_i^l(n) \end{bmatrix}, \qquad \mathbf{w}_i(n) \in \mathbb{R}^{D+C}. \tag{4.9}$$

Tais vetores são ajustados da maneira usual durante o treinamento da rede SOM (ou Fuzzy ART).

Durante a etapa de reconhecimento de um vetor desconhecido $\mathbf{x}$, busca-se por um neurônio vencedor na rede através da seguinte equação:

$$i^* = \arg\min_{\forall i} \left\{ \|\mathbf{x}^p - \mathbf{w}_i^p\| \right\}. \tag{4.10}$$

A classe do padrão desconhecido é estimada por

$$j^* = classe(\mathbf{x}^p) = \arg\max_{\forall j} \left\{ w^l_{i^* j} \right\}, \tag{4.11}$$

em que $w^l_{i^* j}$ é a j-ésima componente do vetor $\mathbf{w}^l_{i^*}$

É interessante perceber que, apesar da estratégia de rotulação auto-supervisionada ter acesso aos rótulos conhecidos no início da fase treinamento, estes são vistos como conjuntos de atributos, de mesma importância que os atributos do padrão de entrada. Essa consideração constitui uma diferença importante em relação ao uso do conhecimento do rótulo em algoritmos de treinamento supervisionado, como na rede Fuzzy ARTMAP, por exemplo.

Assim como as abordagens C1 e C2, a estratégia C3 já foi usada em classificadores SOM por outros autores (KOHONEN, 1988b; KANGAS; KOHONEN; LAAKSONEN, 1990; HOYO; BULDAIN; MARCO, 2003; XIAO *et al.*, 2005), mas não existem trabalhos sobre seu uso em redes Fuzzy ART.

4.2 Arquitetura ARTIE: ART *in Ensembles*

A rede Fuzzy ARTMAP, como descrita na Seção 3.2.1, é uma técnica de aprendizagem supervisionada formada a partir de módulos do algoritmo Fuzzy ART. Por causa da sua natureza supervisionada, redes Fuzzy ARTMAP constituem uma escolha natural para classificadores base em comitês de classificadores. Esta foi a abordagem utilizada, por exemplo, por Santos e Canuto (2008b) e Loo *et al.* (2006). Entretanto, como formalizado anteriormente, as estratégias C1, C2 e C3 permitem que redes Fuzzy ART sejam aplicadas em problemas de classificação de padrões. Dessa maneira, redes Fuzzy ART também podem ser escolhidas como classificadores base de um comitê de classificadores. A utilização das variantes supervisionadas da rede Fuzzy ART em comitês leva à proposição do modelo ARTIE, que, de acordo com o método usado para tornar o algoritmo Fuzzy ART supervisionado, possui três denominações:

- **ARTIE-C**1: comitê de classificadores baseados na rede Fuzzy ART com rotulação *a posteriori* por voto majoritário;
- **ARTIE-C**2: comitê de classificadores baseados na rede Fuzzy ART com rotulação *a priori* por redes individuais;

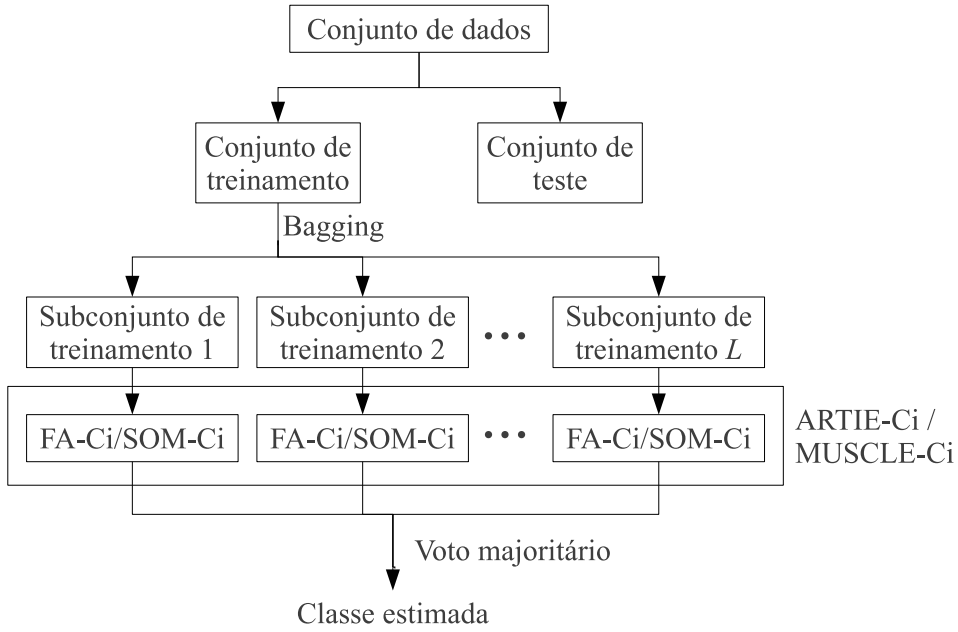


Figura 4.4: Arquiteturas dos modelos ARTIE e MUSCLE. Note que o comitê de L classificadores é obtido com o uso de Bagging no conjunto de treinamento original.

- **ARTIE-C**3: comitê de classificadores baseados na rede Fuzzy ART com rotulação auto-supervisionada.

A Figura 4.4 mostra a arquitetura do modelo ARTIE.

A vantagem imediata de usar classificadores baseados na Fuzzy ART no lugar de classificadores Fuzzy ARTMAP está no fato desta apresentar maior complexidade que aquela, dado que na rede Fuzzy ARTMAP são usadas duas redes Fuzzy ART, envolvendo maior custo computacional. Além disso, a rede Fuzzy ART não necessita da matriz Inter-Map $\mathbf{M}$, o que economiza memória.

É importante ressaltar que os modelos ARTIE podem ser homogêneos ou heterogêneos. No primeiro caso, somente são usados classificadores baseados na rede Fuzzy ART obtidos através do mesmo método, ou seja, classificadores Fuzzy ART-Ci para i igual a 1, 2 ou 3. Já um modelo ARTIE heterogêneo é composto por classificadores baseados na rede Fuzzy ART obtidos por diferentes métodos, ou seja, classificadores Fuzzy ART-Ci, $i \in \{1, 2, 3\}$.

Alguns parâmetros das redes Fuzzy ART que compõem o modelo ARTIE precisam ser determinados *a priori*; mais especificamente, o parâmetro de vigilância ρ, o parâmetro de escolha β e o passo de aprendizagem η. Nesta dissertação opta-se

por uma abordagem metaheurística em busca de valores ótimos para tais parâmetros, que são específicos de cada problema de classificação. A estratégia aplicada envolve um novo algoritmo PSO híbrido a ser descrito em detalhes na Seção 5.5.

4.3 Arquitetura MUSCLE: *Multiple* SOM *Classifiers in Ensembles*

Como mencionado anteriormente, comitês de classificadores formados a partir de redes SOM estão disponíveis na literatura já há algum tempo (PETRIKIEVA; FYFE, 2002; CORCHADO; BARUQUE; YIN, 2007). Nestes trabalhos, apenas a estratégia C1 foi utilizada para viabilizar a aprendizagem supervisionada em redes SOM. Entretanto, as abordagens C2 e C3 também podem ser escolhidas para o mesmo propósito. Nesta dissertação um estudo mais amplo sobre comitês de redes SOM é feito, explorando as três técnicas descritas na Seção 4.1. Essa ideia leva à proposição do modelo MUSCLE com as seguintes variantes:

- **MUSCLE-C**1: comitê de classificadores baseados na rede SOM com rotulação *a posteriori* por voto majoritário;
- **MUSCLE-C**2: comitê de classificadores baseados na rede SOM com rotulação *a priori* por redes individuais;
- **MUSCLE-C**3: comitê de classificadores baseados na rede SOM com rotulação auto-supervisionada.

A arquitetura do modelo MUSCLE é semelhante à do modelo ARTIE, substituindo as redes Fuzzy ART pelas redes SOM, como apresentado na Figura 4.4.

Como no modelo ARTIE, o modelo MUSCLE também pode ser homogêneo ou heterogêneo. No primeiro caso tem-se o uso exclusivo de redes SOM-Ci para i igual a 1, 2 ou 3, enquanto no segundo caso tem-se classificadores SOM-Ci, $i \in \{1, 2, 3\}$.

Em Petrikieva e Fyfe (2002) é reportado um estudo sobre a combinação de resultados de diferentes redes SOM treinadas independentemente. Os autores ressaltam a dificuldade encontrada ao se comparar múltiplos mapas em aplicações de quantização vetorial ou análise de agrupamento, pois estes podem apresentar topologias finais diferentes. Entretanto, no mesmo trabalho conclui-se que tal dificuldade não existe quando as redes SOM são usadas para classificação, pois

a combinação das classes preditas pelos mapas pode ser realizada sem maiores dificuldades, por exemplo, por voto majoritário simples. Os autores constatam experimentalmente no mesmo artigo que o uso de Bagging em redes SOM diminui o erro de generalização.

Assim como o modelo ARTIE, o modelo MUSCLE também possui parâmetros a serem especificados *a priori*. Os principais parâmetros são as dimensões dos mapas (P_1 e P_2), os valores inicial e final do passo de aprendizagem (η_0 e η_f) e os valores inicial e final do parâmetro de espalhamento (σ_0 e σ_f). Esses parâmetros são específicos para cada problema de classificação e, nesta dissertação, serão determinados através de um algoritmo PSO híbrido a ser apresentado na Seção 5.5.

4.4 Conclusões

Neste capítulo foram apresentadas três estratégias que permitem algoritmos originalmente não-supervisionados serem usados em problemas de classificação de padrões. Mais especificamente, foram detalhados os métodos de rotulação por voto majoritário, rotulação por redes individuais e rotulação auto-supervisionada.

A aplicação dessas três abordagens nos algoritmos não-supervisionados Fuzzy ART e SOM, seguida pela utilização dos classificadores resultantes em comitês, levam, respectivamente, à proposição dos modelos ARTIE e MUSCLE.

No Capítulo 5 será apresentado um algoritmo metaheurístico que será utilizado para encontrar parâmetros que conduzam ao melhor classificador base possível em termos de taxa de acerto.

Capítulo 5

Otimização Metaheurística: Fundamentos e um Novo Algoritmo

Métodos de otimização são aplicados em todas as áreas da Engenharia (BELEGUNDU; CHANDRUPATLA, 2011). A resolução de muitos problemas reais envolve a escolha de um conjunto de parâmetros que permita a obtenção de uma resposta desejada para um sistema em estudo. Entretanto, pelo nível de complexidade muitas vezes observado, soluções analíticas não são sempre possíveis de serem obtidas. Uma alternativa viável consiste em usar métodos de busca por soluções aproximadas, como métodos de otimização estocástica.

Nesta dissertação métodos metaheurísticos (inclusos na categoria de métodos de otimização estocástica) são aplicados para seleção de parâmetros ótimos para os classificadores base dos comitês construídos e para seleção de atributos. Mais especificamente, a solução do problema envolve um modelo contínuo, para sintonia dos parâmetros dos classificadores, e um modelo binário, para seleção de atributos. Como será apresentado a seguir, estes dois modelos podem ser otimizados simultaneamente.

Este capítulo descreve uma técnica de otimização metaheurística híbrida capaz de obter soluções para os problemas mencionados.

5.1 Definição do Problema de Otimização

Seja $\Theta \subseteq \mathbb{R}^D$ um domínio de valores possíveis para o vetor $\mathbf{x} \in \mathbb{D}$. O objetivo de um problema de otimização consiste em encontrar valores para $\mathbf{x} \in \Theta$ que minimizem

uma determinada função escalar $f(\mathbf{x})$, chamada de função de perdas ou função de avaliação, ou ainda, de função-objetivo. Formalmente, problemas de otimização podem ser descritos pela expressão abaixo (GENTLE; HÄRDLE; MORI, 2004):

$$\Theta^* \equiv \arg\min_{\mathbf{x}\in\Theta} f(\mathbf{x}) = \{\mathbf{x}^* \in \Theta : f(\mathbf{x}^*), \forall \mathbf{x} \in \Theta\}, \tag{5.1}$$

em que Θ^* é o conjunto de soluções que minimiza a função $f(\mathbf{x})$ para $\mathbf{x} = \mathbf{x}^*$.

Uma das principais dificuldades encontradas durante a resolução de um problema de otimização envolve da definição da função-objetivo, pois esta pode ser não-linear, não-diferenciável e depender de parâmetros de dimensão elevada. Para esta classe de problemas a aplicação de métodos de otimização clássicos determinísticos pode ser inadequada. Técnicas estocásticas, por outro lado, não possuem essas restrições.

5.2 Otimização estocástica

Uma estratégia para a solução de problemas de otimização consiste em construir um espaço de soluções a partir das variáveis em estudo. Considerando que esse espaço apresente soluções vizinhas similares, ou seja, boas soluções estão agrupadas e situam-se longe de soluções ruins, pode-se reduzir consideravelmente o custo da otimização através de métodos de busca. Tais métodos podem ser de natureza determinística ou estocástica (LØVBJERG, 2002).

Algoritmos de otimização estocástica têm sido muito utilizados na resolução de problemas (BELEGUNDU; CHANDRUPATLA, 2011) que apresentam muitas variáveis, funções objetivo não diferenciáveis ou fracamente definidas[1]. Problemas com estas características são comuns em situações reais e frequentemente não podem ser resolvidos de maneira satisfatória (compromisso entre qualidade da solução e tempo de processamento) por métodos determinísticos. É importante perceber que nem sempre a solução ótima exata é obtida em técnicas estocásticas, mas sim uma solução subótima, normalmente próxima da ideal e que possa ser obtida em tempo hábil.

Em suma, métodos de busca estocástica possuem vantagens sobre métodos determinísticos exatos. Em primeiro lugar, a abordagem estocástica permite resolver problemas complexos a partir de pouca informação *a priori* sobre o problema. Além

[1]Por *fracamente definida* entende-se que a função objetivo fora obtida empiricamente ou que explique somente em parte a operação do sistema a ser otimizado.

disso, é possível a obtenção de resultados parciais a cada passo de execução do algoritmo, tornando possível a realização de uma troca entre a qualidade da solução e o tempo de processamento (LØVBJERG, 2002).

De maneira geral, algoritmos de otimização estocástica são métodos de otimização que apresentam comportamento probabilístico na geração de soluções para o problema ou no processo de busca em si (SPALL, 2003). No caso de interesse, as variáveis do problema e a função objetivo que se deseja otimizar são determinísticas, enquanto as regras que regem a busca no espaço de soluções são probabilísticas.

5.3 Métodos metaheurísticos

A palavra *heurística* é originada da palavra grega *heuriskein*, que significa *a arte de descobrir novas estratégias para resolver problemas*. Já o prefixo *meta*, também de origem grega, significa *em um nível superior* (TALBI, 2009). Em Glover (1986), a expressão *técnicas de busca metaheurísticas* é introduzida e definida como sendo *metodologias gerais, em um nível mais elevado de abstração, capazes de guiar a modelagem de solução de problemas de otimização.*

Comumente, metaheurísticas são desenvolvidas a partir da observação da natureza. Fenômenos naturais mostram que é possível resolver problemas difíceis a partir de interações aleatórias locais. Como exemplo, pode-se citar o trabalho conjunto de uma colônia de formigas em busca da melhor rota até fontes de alimento. São exemplos de metaheurísticas as seguintes técnicas:

- **Otimização por Enxame de Partículas (PSO, *Particles Swarm Optimization*)** (KENNEDY; EBERHART, 1995);
- **Algoritmos Genéticos (AG)** (HOLLAND, 1975);
- **Recozimento Simulado (SA, *Simulated Annealing*)** (KIRKPATRICK *et al.*, 1983);
- **Otimização por Colônia de Formigas (ACO, *Ant Colony Optimization*)** (DORIGO, 1992).

Algoritmos como PSO, AG e ACO são baseados em populações de soluções, ou seja, a cada iteração um conjunto de possíveis soluções são

testadas e possivelmente aprimoradas. Enquanto isso, técnicas orientadas a trajetória, como o método SA, determinam uma única solução a cada iteração (ANGHINOLFI; PAOLUCCI, 2008).

5.4 Otimização por Enxame de Partículas

Nesta seção analisa-se a técnica PSO, uma das mais populares metaheurísticas para otimização de funções. Diversos conceitos e operações desse algoritmo serão descritos, enquanto maiores detalhes podem ser conferidos nas referências citadas.

5.4.1 PSO original

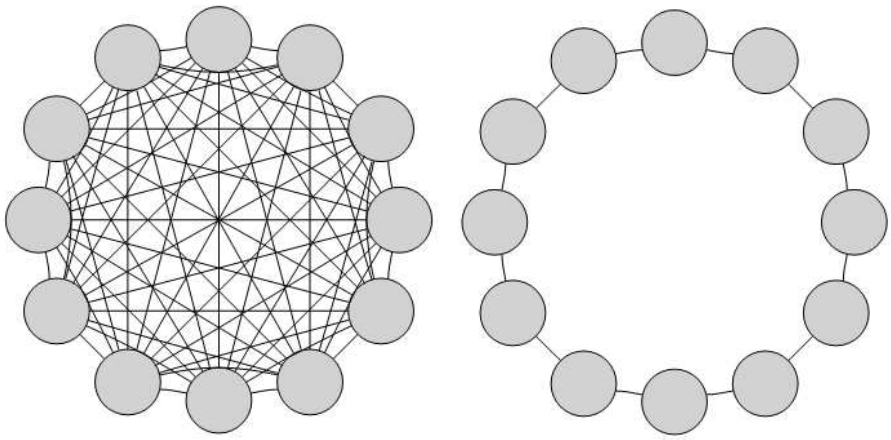
Proposto por Kennedy e Eberhart (1995), a técnica PSO é inspirada no comportamento social e na auto-organização de grupos de pássaros migratórios e cardumes de peixes. O comportamento social, demonstrado a partir da troca de informação entre os elementos da população, gera a exploração por melhores soluções, enquanto o aprendizado individual corresponde à componente de explotação[2]. Esse método tem se mostrado eficiente e de simples execução para a resolução de vários problemas reais de otimização.

A versão original da técnica PSO envolve a consideração de um enxame de partículas distribuídas em um espaço de soluções, sendo a posição de uma determinada partícula correspondente a uma possível solução. Cada partícula possui ainda uma velocidade associada que indica a sua tendência de movimento pelo espaço. As informações obtidas por cada partícula são utilizadas na busca por boas soluções a partir de uma organização em que todas as partículas têm conhecimento da melhor posição alcançada pelas demais. Essa forma de organização caracteriza uma topologia global de enxame.

5.4.2 PSO padrão 2007

Em Bratton e Kennedy (2007), sendo o último um dos criadores do algoritmo PSO original, analisa-se as diversas variações propostas por pesquisadores desde a proposição da primeira versão do algoritmo. Neste mesmo trabalho é apresentada uma sugestão de padronização para métodos baseados em PSO, reunindo várias características que melhoram o desempenho da técnica PSO original. Essa versão

[2]Explotação é um neologismo criado a partir da palavra *"exploitation"*, em inglês. Nesta dissertação, o termo explotação refere-se ao processo de exploração do espaço de busca considerando as informações das regiões anteriormente visitadas



(a) Topologia global de enxame. (b) Topologia local de enxame.

Figura 5.1: Topologias de enxame mais comuns na aplicação do algoritmo PSO (EBERHART; KENNEDY, 1995).

do algoritmo tem sido chamada desde então de PSO padrão 2007 (*Standard PSO 2007*).

A variação mais importante envolve alteração da topologia global do enxame de partículas para uma topologia local. Nessa abordagem, apresentada pela primeira vez em Eberhart e Kennedy (1995), cada partícula só é capaz de conhecer a melhor solução dentro de sua vizinhança, ou seja, a melhor solução dentro de um subconjunto de partículas. No caso em que uma partícula se comunica somente com as duas partículas adjacentes a ela, tem-se uma topologia de enxame com formato de anel. Nas Figuras 5.1(a) e 5.1(b) estão mostradas as duas topologias, global e local.

A topologia local tende a apresentar uma convergência mais lenta que a topologia global, pois a troca de informação ocorre em pequenos grupos de partículas, ocasionando a formação de vários grupos de busca inicialmente separados. Entretanto, essa característica permite à versão local do PSO evitar a convergência prematura em um subótimo local indesejável.

As demais características do método PSO padrão 2007 serão apresentadas na descrição do seu algoritmo, na seção seguinte.

5.4.2.1 Algoritmo PSO padrão 2007

Sejam $\mathbf{x}_i \in \mathbb{R}^D$ e $\mathbf{v}_i \in \mathbb{R}^D$, respectivamente, os vetores de posição e de velocidade do i-ésimo elemento de um enxame de partículas D-dimensionais, em que D é o número de variáveis das soluções. Sejam ainda $\mathbf{p}_i \in \mathbb{R}^D$ e $\mathbf{pl}_k \in \mathbb{R}^D$, respectivamente, os vetores de melhor posição histórica individual da partícula i e melhor posição histórica da vizinhança k. Abaixo tem-se a definição desses vetores:

$$\begin{aligned}
\mathbf{x}_i &= [x_{i1}\ x_{i2}\ \cdots\ x_{iD}]^T, \\
\mathbf{v}_i &= [v_{i1}\ v_{i2}\ \cdots\ v_{iD}]^T, \\
\mathbf{p}_i &= [p_{i1}\ p_{i2}\ \cdots\ p_{iD}]^T, \\
\mathbf{pl}_k &= [pl_{k1}\ pl_{k2}\ \cdots\ pl_{kD}]^T.
\end{aligned}$$

Considerando-se a função objetivo $f(\cdot)$ e um total de L_{PSO} gerações (iterações), as etapas do algoritmo de otimização PSO padrão 2007 são apresentadas a seguir.

Inicialização. Iniciam-se as variáveis do enxame de partículas com os seguintes valores:

$$\begin{aligned}
\mathbf{x}_i(0) &= \mathbf{x}_{min} + (\mathbf{x}_{max} - \mathbf{x}_{min})\mathbf{u}, \\
\mathbf{v}_i(0) &= (\mathbf{x}_{max} - \mathbf{x}_{min})\mathbf{u} - \mathbf{x}_i(0), \\
\mathbf{p}_i(0) &= \mathbf{0}^D, \\
\mathbf{pl}_k(0) &= \mathbf{0}^D,
\end{aligned}$$

em que $\mathbf{u} \in \mathbb{R}^D$ é um vetor de números aleatórios uniformemente distribuídos no intervalo $[0, 1]$, $\mathbf{0}^D$ é o vetor nulo de dimensão D e $\mathbf{x}_{min} \in \mathbb{R}^D$ e $\mathbf{x}_{max} \in \mathbb{R}^D$ são respectivamente os menores e maiores valores das variáveis que compõem uma possível solução.

Inicia-se então a primeira geração do algoritmo, fazendo $n = 1$.

Avaliação do enxame. Todas as soluções do exame da geração n são avaliadas, ou seja, calcula-se $f(\mathbf{x}_i(n))$ para cada partícula $\mathbf{x}_i(n)$. Caso seja a primeira geração, i.e. $n = 1$, $\mathbf{p}_i(1)$ recebe a posição atual $\mathbf{x}_i(1)$ e $\mathbf{pl}_k(1)$ recebe a melhor posição entre as partículas da vizinhança k.

Caso não seja a primeira geração, os vetores $\mathbf{p}_i(n)$ e $\mathbf{pl}_k(n)$ recebem os seguintes

valores:

$$\begin{aligned} \text{se } f(\mathbf{x}_i(n)) &> f(\mathbf{p}_i(n)), \qquad \mathbf{p}_i(n) = \mathbf{x}_i(n),\ \forall i & (5.2)\\ \mathbf{p}_{k_{max}}(n) &= \arg\max_{\forall i \in \mathcal{N}_k} \{f(\mathbf{p}_i(n))\},\ \forall k & (5.3)\\ \text{se } f(\mathbf{p}_{k_{max}}(n)) &> f(\mathbf{pl}_k(n)), \qquad \mathbf{pl}_k(n) = \mathbf{p}_{k_{max}}(n),\ \forall k, & (5.4) \end{aligned}$$

em que $\mathcal{N}_k$ é o conjunto formado pelas partículas da k-ésima vizinhança.

Atualização do enxame. Calculam-se as novas velocidades $\mathbf{v}_i(n+1)$ e posições $\mathbf{x}_i(n+1)$ das partículas do enxame por meio das equações abaixo:

$$\begin{aligned} \mathbf{v}_i(n+1) &= \chi\{\mathbf{v}_i(n) + c_1 r_1[\mathbf{p}_i(n) - \mathbf{x}_i(n)] + c_2 r_2[\mathbf{pl}_k(n) - \mathbf{x}_i(n)]\}, & (5.5)\\ \mathbf{x}_i(n+1) &= \mathbf{x}_i(n) + \mathbf{v}_i(n+1), & (5.6) \end{aligned}$$

em que c_1 e c_2 são constantes positivas chamadas de coeficientes de aceleração e r_1 e r_2 são números aleatórios independentes uniformemente distribuídos no intervalo $[0, 1]$. O parâmetro χ é denominado fator de constrição e é dado por

$$\chi = \frac{2}{\left|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}\right|}, \qquad \text{em que } \varphi = c_1 + c_2. \tag{5.7}$$

O fator de constrição, proposto inicialmente em Clerc e Kennedy (2002), constitui outra incorporação do algoritmo padrão 2007 em relação ao algoritmo original. Este fator concede mais estabilidade ao algoritmo, proporcionando um equilíbrio entre a busca local de cada partícula e a comunicação com as partículas vizinhas.

O termo $c_2 r_2[\mathbf{pl}_k(n) - \mathbf{x}_i(n)]$ da Equação (5.5) é relativo à troca de informação entre as partículas de uma mesma vizinhança, constituindo a componente de cooperação do algoritmo. Já o termo $\mathbf{v}_i(n) + c_1 r_1[\mathbf{p}_i(n) - \mathbf{x}_i(n)]$ da Equação (5.5) indica a busca local motivada pelo histórico experimentado por cada partícula.

Após a etapa de atualização das partículas a geração atual é incrementada ($n \leftarrow n+1$) e os passos de avaliação e atualização do enxame são repetidos por um total de L_{PSO} gerações. A solução final obtida pelo algoritmo corresponde a

$$\mathbf{p}^* = \arg\max_{\forall k} \{f(\mathbf{pl}_k(L_{PSO}))\}. \tag{5.8}$$

5.4.3 Algoritmo PSO binário

O método PSO original considera variáveis reais contínuas durante o processo de otimização, ou seja, para uma solução representada pelo vetor D-dimensional $\mathbf{x} = [x_1\ x_2\ \cdots\ x_D]^T$, tem-se que $x_j \in \mathbb{R}, \forall j$.

Entretanto, existem problemas cujo modelo envolve variáveis binárias, em domínio discreto. O problema da seleção dos atributos a serem usados por um classificador é um exemplo em que essa modelagem binária pode ser feita: para cada atributo associa-se uma variável binária em que o valor 0 indica que o atributo não é usado e o valor 1 indica seu uso.

Kennedy e Eberhart (1997), propuseram uma versão PSO binária, capaz de operar variáveis binárias em domínios discretos. Em Khanesar, Teshnehlab e Shoorehdeli (2007) é proposta uma reformulação dessa versão binária do algoritmo. Devido aos bons resultados experimentais obtidos em Khanesar, Teshnehlab e Shoorehdeli (2007), esta será a versão PSO binária usada nesta dissertação.

A solução passa a ser composta por variáveis binárias, i.e. a posição da partícula i é representada pelo vetor $\mathbf{x}_i \in$ com D componentes dadas por $x_{ij} \in \{0, 1\}, j = 1, 2, \cdots, D$. As partículas são atualizadas da seguinte maneira:

(i) Calculam-se as seguintes velocidades referentes à j-ésima componente da i-ésima partícula na iteração n do algoritmo:

$$v^1_{ij}(n+1) = \chi\{v^1_{ij}(n) + d^1_{ij,1} + d^1_{ij,2}\}, \tag{5.9}$$

$$v^0_{ij}(n+1) = \chi\{v^0_{ij}(n) + d^0_{ij,1} + d^0_{ij,2}\}, \tag{5.10}$$

em que os valores parciais são obtidos pelas equações abaixo:

$$\text{Se } p_{ij}(n) = 1, \quad \text{Então} \quad d^1_{ij,1} = c_1 r_1, \text{ e } d^0_{ij,1} = -c_1 r_1, \tag{5.11}$$

$$\text{Se } p_{ij}(n) = 0, \quad \text{Então} \quad d^0_{ij,1} = c_1 r_1, \text{ e } d^1_{ij,1} = -c_1 r_1, \tag{5.12}$$

$$\text{Se } pl_{kj}(n) = 1, \quad \text{Então} \quad d^1_{ij,2} = c_2 r_2, \text{ e } d^0_{ij,2} = -c_2 r_2, \tag{5.13}$$

$$\text{Se } pl_{kj}(n) = 0, \quad \text{Então} \quad d^0_{ij,2} = c_2 r_2, \text{ e } d^1_{ij,2} = -c_2 r_2. \tag{5.14}$$

Note que $p_{ij}(n)$ e $pl_{kj}(n)$ são, respectivamente, componentes dos vetores $\mathbf{p}_i(n)$ e $\mathbf{pl}_k(n)$, definidos na Seção 5.4.2 para o método PSO padrão 2007. Note ainda que a vizinhança k é aquela referente à partícula i. Os coeficientes

de aceleração c_1 e c_2, assim como o fator de constrição χ, são os mesmos da Equação (5.5), enquanto r_1 e r_2 são números aleatórios independentes uniformemente distribuídos no intervalo $[0, 1]$.

A velocidade $v_{ij}^1(n+1)$ representa a chance da componente x_{ij} mudar de 0 para 1, enquanto a velocidade $v_{ij}^0(n+1)$ pode ser vista como a chance da transição no sentido oposto. Dessa maneira, somente uma delas é considerada para cada componente:

$$\text{Se } x_{ij}(n) = 0, \quad \text{Então} \quad v_{ij}(n+1) = v_{ij}^1(n+1), \tag{5.15}$$

$$\text{Se } x_{ij}(n) = 1, \quad \text{Então} \quad v_{ij}(n+1) = v_{ij}^0(n+1), \tag{5.16}$$

(ii) Atualizam-se as componentes da solução $\mathbf{x}_i$:

$$\text{Se } s(v_{ij}(n+1)) \geq U(0,1), \tag{5.17}$$

$$\text{Então } x_{ij}(n+1) = 1 - x_{ij}(n), \tag{5.18}$$

$$\text{Senão } x_{ij}(n+1) = x_{ij}(n), \tag{5.19}$$

em que $U(0, 1)$ é um número aleatório uniformemente distribuído no intervalo $[0, 1]$ e $s(\cdot)$ é uma função sigmoidal, comumente definida por

$$s(v_{ij}) = \frac{1}{1 + \exp(-v_{ij})}. \tag{5.20}$$

Note que somente após a aplicação da função $s(\cdot)$ a velocidade v_{ij} passa a representar uma probabilidade, pois o valor $s(v_{ij})$ encontra-se no intervalo $[0, 1]$.

É interessante perceber que a maneira como foram definidas as versões contínuas e discretas do algoritmo PSO permite que as duas abordagens sejam usadas simultaneamente, no caso de problemas que envolvam variáveis reais e binárias. Esse procedimento é realizado em Huang e Dun (2008), Yao, Cai e Zhang (2009), Guo (2009)

Por exemplo, para uma solução composta por duas variáveis reais ($x_{i1}(n)$ e $x_{i2}(n)$) e três variáveis binárias ($x_{i3}(n)$, $x_{i4}(n)$ e $x_{i5}(n)$), tem-se o seguinte vetor de

posição para a i-ésima partícula:

$$\mathbf{x}_i(n) = [x_{i1}(n)\ x_{i2}(n)\ x_{i3}(n)\ x_{i4}(n)\ x_{i5}(n)]^T. \tag{5.21}$$

Nesse caso os passos do algoritmo PSO descrito na Seção 5.4.2 seriam seguidos normalmente. Porém, na etapa de atualização das posições da partícula i, o segmento $x_i^r(n) = [x_{i1}(n)\ x_{i2}(n)]^T$ seria atualizado pela Equação (5.6), enquanto o segmento $x_i^b(n) = [x_{i3}(n)\ x_{i4}(n)\ x_{i5}(n)]^T$ seria atualizado de acordo com as Equações (5.18) ou (5.19).

5.5 Uma Versão Híbrida Melhorada do Algoritmo PSO

Com o crescimento da utilização da técnica PSO, várias formas de hibridização foram propostas, buscando reduzir o efeito de características indesejáveis do algoritmo original, como a alta dependência de parâmetros reguláveis e a possibilidade de convergência prematura a partir da supervalorização de uma solução específica. Como exemplos podem ser citados a utilização de PSO com AG em Kim, Abraham e Hirota (2007), com elementos de Lógica Fuzzy em Liu e Abraham (2007), com ACO em Holden e Freitas (2008) e com busca caótica em Liu *et al.* (2005).

Comumente técnicas híbridas apresentam estrutura geral determinada por uma metaheurística específica enquanto utilizam outros algoritmos, também metaheurísticos, para realizar procedimentos de busca local e garantir um equilíbrio entre exploração e explotação.

Nesta seção é apresentada uma técnica híbrida entre os métodos PSO e SA. Antes, porém, uma rápida descrição do algoritmo SA é realizada.

5.5.1 Recozimento Simulado

O algoritmo de otimização SA, proposto por Kirkpatrick *et al.* (1983), é uma abstração computacional do processo de recozimento utilizado na metalurgia, em que um sólido é inicialmente fundido a uma alta temperatura para em seguida passar por uma lenta etapa de resfriamento que volta a solidificar o material. No primeiro momento os átomos do sólido recebem energia suficiente para se movimentarem mais livremente, enquanto o resfriamento lento resulta na diminuição da movimentação desses átomos que, ao final de todo o processo, passam a ocupar posições com energia mínima.

Na otimização feita por SA, assim como no conhecido algoritmo *Hill Climbing* (RUSSEL; NORVIG, 1996), novas soluções são geradas aleatoriamente e a solução atual é trocada pela nova caso esta seja melhor. A diferença está na existência de uma probabilidade de aceitação de uma solução pior que a atual. Essa probabilidade é regida por uma temperatura controlada de forma a causar uma chance cada vez menor de se escolher soluções piores. Assim, no início do algoritmo a troca de soluções é freqüente, enquanto que ao final torna-se mais difícil de ocorrer. Essa técnica evita a escolha de uma solução sub-ótima logo no início da otimização, permitindo uma melhoria na qualidade da solução final.

5.5.2 Algoritmo I-HPSO (Improved Hybrid PSO)

A estagnação do processo de busca em ótimos locais é um fenômeno frequente em algoritmos com populações de soluções. Em contrapartida, tais técnicas costumam apresentar vasta exploração no espaço de soluções, principalmente no início de sua execução, além de se beneficiarem da troca de informação entre elementos da população. A partir dessas ideias e da tendência de hibridização comentada anteriormente, percebe-se a vantagem em incorporar o sistema de busca do método SA em um algoritmo de populações, buscando somar as qualidades do algoritmo SA e reduzir as deficiências do algoritmo PSO.

He e Wang (2007) propuseram uma variação do algoritmo PSO original chamada HPSO (*Hybrid* PSO) ao adicionar regras para tratar problemas de otimização com restrições e uma etapa de busca local baseada em SA. Em diversos experimentos de otimização de funções realizados em He e Wang (2007), essa nova técnica se mostrou superior ao método PSO original.

Nesta dissertação é proposto o algoritmo I-HPSO, que consiste em aplicar a etapa de busca local via SA, presente no método HPSO, no algoritmo PSO padrão 2007, definido na Seção 5.4.2.1. Dessa maneira, as principais diferenças entre os métodos I-HPSO e HPSO são: (*i*) o uso da topologia local para o enxame de partículas, visando evitar a convergência prematura do algoritmo. (*ii*) uso do fator de constrição, como apresentado na Equação (5.7), o que promove maior estabilidade ao processo de otimização.

É preciso mencionar ainda que, considerando a aplicação desejada nesta dissertação (seleção de parâmetros e atributos do classificador base), não serão usadas funções objetivo com restrições. Logo, não serão usadas as regras para tratar

restrições propostas em He e Wang (2007).

O algoritmo I-HPSO apresenta os mesmos passos descritos para o PSO padrão 2007 na Seção 5.4.2. Entretanto, após a etapa de atualização do enxame, segue a etapa de busca local. Considerando um total de L_{SA} iterações nessa etapa, as operações a seguir são executadas na iteração m.

Busca local via SA. Seleciona-se a melhor solução histórica encontrada por todas as vizinhanças na geração n do algoritmo I-HPSO

$$\mathbf{pl}'(m) = \arg\max_{\forall k}\{f(\mathbf{pl}_k(n))\}. \tag{5.22}$$

Em seguida, gera-se uma nova solução a partir de pequenas alterações aleatórias no vetor $\mathbf{pl}_{max}$:

$$\mathbf{x}'(m) = \mathbf{pl}'(m) + \eta_{SA}(\mathbf{x}_{max} - \mathbf{x}_{min})\mathbf{g}(\mathbf{0}, \mathbf{I}), \tag{5.23}$$

em que η_{SA} é um passo de incremento e $\mathbf{g}(\mathbf{0}, \mathbf{I})$ é um vetor aleatório D-dimensional de distribuição gaussiana com média zero e matriz de covariância igual à matriz identidade. Calcula-se então a probabilidade de aceitação da nova solução gerada:

$$P_a = \min\left\{1, \exp\left[\frac{f(\mathbf{pl}'(m)) - f(\mathbf{x}'(m))}{t(n)}\right]\right\}, \tag{5.24}$$

em que $t(n)$ constitui o valor do parâmetro de temperatura durante a geração n do algoritmo I-HPSO.

Caso $P_a \geq u(0,1)$, em que $u(0,1)$ é um número aleatório uniformemente distribuído no intervalo $[0,1]$, a próxima iteração de busca local será feita a partir da solução $\mathbf{x}'(m)$, o que equivale a fazer $\mathbf{pl}'(m+1) = \mathbf{x}'(m)$. Caso contrário, mantém-se a busca local na solução anterior, i.e. $\mathbf{pl}'(m+1) = \mathbf{pl}'(m)$. A iteração da busca local é incrementada ($m \leftarrow m+1$) e os passos dessa etapa são repetidos até $m = L_{SA}$.

O parâmetro de temperatura $t(n)$ determina uma maior ou menor probabilidade de o algoritmo aceitar uma solução inferior a atual. Por esse motivo a temperatura deve ser reduzida ao longo das n gerações do algoritmo I-HPSO. O processo de

redução de temperatura escolhido é o exponencial, ou seja, $t(n+1) = \lambda t(n)$, em que a taxa de recozimento λ satisfaz $0 < \lambda < 1$.

Para a execução do algoritmo, He e Wang (2007) sugerem o seguinte valor empírico para a temperatura inicial:

$$t_0 = -\frac{f_{max} - f_{min}}{ln(0.1)}, \tag{5.25}$$

em que f_{max} e f_{min} são o maior e o menor valor da função objetivo encontrados no enxame inicial de partículas.

O Algoritmo 5.1 resume os passos do método I-HPSO, enquanto o Algoritmo 5.2 apresenta as operações da etapa de busca local.

5.6 Conclusões

Neste capítulo foi feita uma breve revisão sobre otimização metaheurística com ênfase no algoritmo PSO e algumas de suas modificações. Foi detalhada ainda uma nova técnica híbrida, o algoritmo I-HPSO, a partir de melhorias implementadas no método HPSO, proposto em He e Wang (2007).

Para os algoritmos de aprendizado apresentados no Capítulo 3 é praticamente impossível determinar por tentativa e erro os parâmetros ótimos a serem usados para cada conjunto de dados. Um dos motivos desta dificuldade é a alta dimensão do espaço de busca. No caso dos parâmetros dos classificadores base, em que os valores são contínuos (números reais), a busca exaustiva se torna ainda mais difícil.

Como foi apresentado neste capítulo, métodos metaheurísticos apresentam-se como alternativas viáveis ao realizar buscas por soluções aceitáveis a partir de uma função objetivo que explique, pelo menos em parte, a operação do sistema.

No caso dos classificadores de padrões estudados nesta dissertação, uma função objetivo válida é a taxa de acerto do classificador dado um conjunto de parâmetros a serem otimizados pelo algoritmo I-HPSO.

Além disso, conforme discutido na Seção 5.4.3, é possível a partir da versão binária do método PSO selecionar os atributos mais relevantes em um determinado conjunto de dados. Nessa mesma seção foi apresentado a possibilidade dessa escolha ser feita ao mesmo tempo em que se realiza a otimização contínua de outras variáveis da solução.

Algoritmo 5.1 Algoritmo I-HPSO.

Constantes

N_p: número de partículas do enxame

L_{PSO}: número máximo de gerações do algoritmo

c_1 e c_2: coeficientes de aceleração das partículas

Entradas

$\mathbf{x}_{min}, \mathbf{x}_{max}$: valores mínimos e máximos para as variáveis da solução, dimensão D

$f(\cdot)$: função objetivo que se deseja otimizar

Algoritmo

1. Inicialização ($n = 0$)

Criar e inicializar as partículas do enxame ($i = 1, 2, \ldots, N_p$)

$$\mathbf{x}_i(0) = \mathbf{x}_{min} + (\mathbf{x}_{max} - \mathbf{x}_{min})\mathbf{u}$$
$$\mathbf{v}_i(0) = (\mathbf{x}_{max} - \mathbf{x}_{min})\mathbf{u} - \mathbf{x}_i(0)$$
$$\mathbf{p}_i(0) = \mathbf{0}^D, \quad \mathbf{pl}_k(0) = \mathbf{0}^D$$

2. Laço temporal ($n = 1, 2, \ldots, L_{PSO}$)

2.1 Avaliar o enxame de partículas, i.e. $\forall i$ calcular $f(\mathbf{x}_i(n))$

SE $f(\mathbf{x}_i(n)) > f(\mathbf{p}_i(n))$, FAZER $\mathbf{p}_i(n) = \mathbf{x}_i(n)$

$\forall k$ calcular $\mathbf{p}_{k_{max}}(n) = \arg\max_{\forall i \in \mathcal{N}_k}\{f(\mathbf{p}_i(n))\}$

SE $f(\mathbf{p}_{k_{max}}(n)) > f(\mathbf{pl}_k(n))$, FAZER $\mathbf{pl}_k(n) = \mathbf{p}_{k_{max}}(n)$

2.2 Atualizar as velocidades das partículas

$$\mathbf{v}_i(n+1) = \chi\{\mathbf{v}_i(n) + c_1 r_1[\mathbf{p}_i(n) - \mathbf{x}_i(n)] + c_2 r_2[\mathbf{pl}_k(n) - \mathbf{x}_i(n)]\}$$

2.2 Atualizar as posições das partículas

$$\mathbf{x}_i(n+1) = \mathbf{x}_i(n) + \mathbf{v}_i(n+1)$$

2.3 Iniciar o passo de Busca Local descrito no Algoritmo 5.2

Saídas ou variáveis de interesse

$\mathbf{p}^* = \arg\max_{\forall k}\{f(\mathbf{pl}_k(L_{PSO}))\}$: melhor solução encontrada pelo algoritmo

Algoritmo 5.2 Busca Local do algoritmo I-HPSO.

Constantes

λ: taxa de recozimento

L_{SA}: número de iterações da etapa de busca local

η_{SA}: passo de incremento

Entradas

$\mathbf{x}_{min}, \mathbf{x}_{max}$: valores mínimos e máximos para as variáveis da solução, dimensão D

$f(\cdot)$: função objetivo que se deseja otimizar

n: geração atual do algoritmo I-HPSO

Algoritmo

1. Inicialização $(m = 0)$

Encontrar a melhor posição histórica encontrada por todas as vizinhanças

$\mathbf{pl}'(m) = \arg\max_{\forall k}\{f(\mathbf{pl}_k(n))\}$

2. Laço temporal $(m = 1, 2, \ldots, L_{SA})$

2.1 Gerar uma nova solução a partir de $\mathbf{pl}'(m)$

$\mathbf{x}'(m) = \mathbf{pl}'(m) + \eta_{SA}(\mathbf{x}_{max} - \mathbf{x}_{min})\mathbf{g}(\mathbf{0}, \mathbf{I})$

2.2 Calcular a probabilidade de aceitação da nova solução gerada

$P_a = \min\left\{1, \exp\left[\frac{f(\mathbf{pl}'(m)) - f(\mathbf{x}'(m))}{t(n)}\right]\right\}$

SE $P_a \geq U(0,1)$, FAZER $\mathbf{pl}'(m+1) = \mathbf{x}'(m)$

SENÃO FAZER $\mathbf{pl}'(m+1) = \mathbf{pl}'(m)$

3. Atualizar o parâmetro de temperatura

$t(n+1) = \lambda t(n)$

Saídas ou variáveis de interesse

$\mathbf{pl}'(L_{SA})$: solução encontrada pela etapa de busca local

Considerando ainda a aplicação de reconhecimento de padrões, pode-se realizar simultaneamente a otimização de parâmetros de um classificador e a seleção de atributos usados de maneira a melhorar sua taxa de acerto, i.e. sua capacidade de generalização.

O Capítulo 6 detalha a metodologia de construção de comitês a partir de classificadores base otimizados com técnicas metaheurísticas.

Capítulo 6

Metodologia de Projeto e Comparação

Como o objetivo geral desta dissertação é projetar (construir) e avaliar o desempenho de comitês de classificadores baseados nas redes neurais competitivas SOM, Fuzzy ART, Fuzzy ARTMAP e LVQ, é preciso definir uma metodologia de trabalho e de testes antes da realização das simulações computacionais.

Neste capítulo serão detalhados os procedimentos adotados na presente dissertação para a obtenção de comitês de classificadores, assim como os métodos escolhidos para compará-los entre si.

6.1 Construção dos Comitês de Classificadores

Como comentado no Capítulo 2, três componentes devem ser definidos durante o projeto de um comitê de classificadores de padrões: (*i*) a maneira como as diferentes predições serão combinadas, (*ii*) as técnicas de promoção de diversidade e (*iii*) a escolha dos classificadores base.

A seguir são listadas as componentes da metodologia de construção de um comitê de classificadores que será seguida nesta dissertação.

- O método de combinação será o voto majoritário simples, conforme descrito na Seção 2.1.
- A diversidade entre os classificadores base será promovida por meio dos seguintes procedimentos:

  - Utilização de Bagging, como descrito na Seção 2.2, com a intenção de criar subconjuntos de treinamento diferentes para cada classificador base.
  - Uso de condições iniciais aleatórias para os classificadores base. Redes neurais, de forma geral, apresentam pesos sinápticos com valores aleatórios no início processo de aprendizado.
  - Apresentação em ordem aleatória dos exemplos de treinamento para cada classificador base. Por se tratarem de processos iterativos, os algoritmos de aprendizagem apresentados no Capítulo 3 são influenciados pela ordem de apresentação das amostras de treinamento. Esta característica será explorada como mais uma maneira de geração de diversidade, exceto para comitês de redes ELM (*Extreme Learning Machine*) (HUANG; ZHU; SIEW, 2004).

- Os comitês analisados são homogêneos, ou seja, todos os seus classificadores base são construídos a partir de uma mesma arquitetura de redes neurais. Serão usados comitês das arquiteturas ARTIE e MUSCLE (ambas apresentadas no Capítulo 4), e comitês de redes Fuzzy ARTMAP, LVQ e ELM. A rede ELM é uma rede neural supervisionada cujos detalhes de operação podem ser conferidos no Apêndice A desta dissertação.

- Os classificadores base de um comitê são otimizados antes de serem combinados. Este processo envolve tanto a seleção dos parâmetros de operação do classificador, quanto a seleção de atributos usando o algoritmo I-HPSO (incorporando etapas do PSO binário) detalhado no Capítulo 5. O processo de otimização é realizado somente para um classificador base, cujos parâmetros e atributos escolhidos são replicados para todos os outros classificadores base que comporão um determinado comitê.

Na Figura 6.1 é apresentado um fluxograma do processo de construção e avaliação de um comitê de classificadores. O comitê é testado através da técnica de *validação cruzada de k partições* (STONE, 1974). De acordo com este método de avaliação, o conjunto de dados disponíveis é dividido em k partições sem interseções. São realizadas então repetidos ciclos de treinamento e teste em que cada uma das partições é escolhida como conjunto de teste e as demais formam o conjunto de treinamento. As estatísticas de acerto são calculadas para as taxas obtidas para

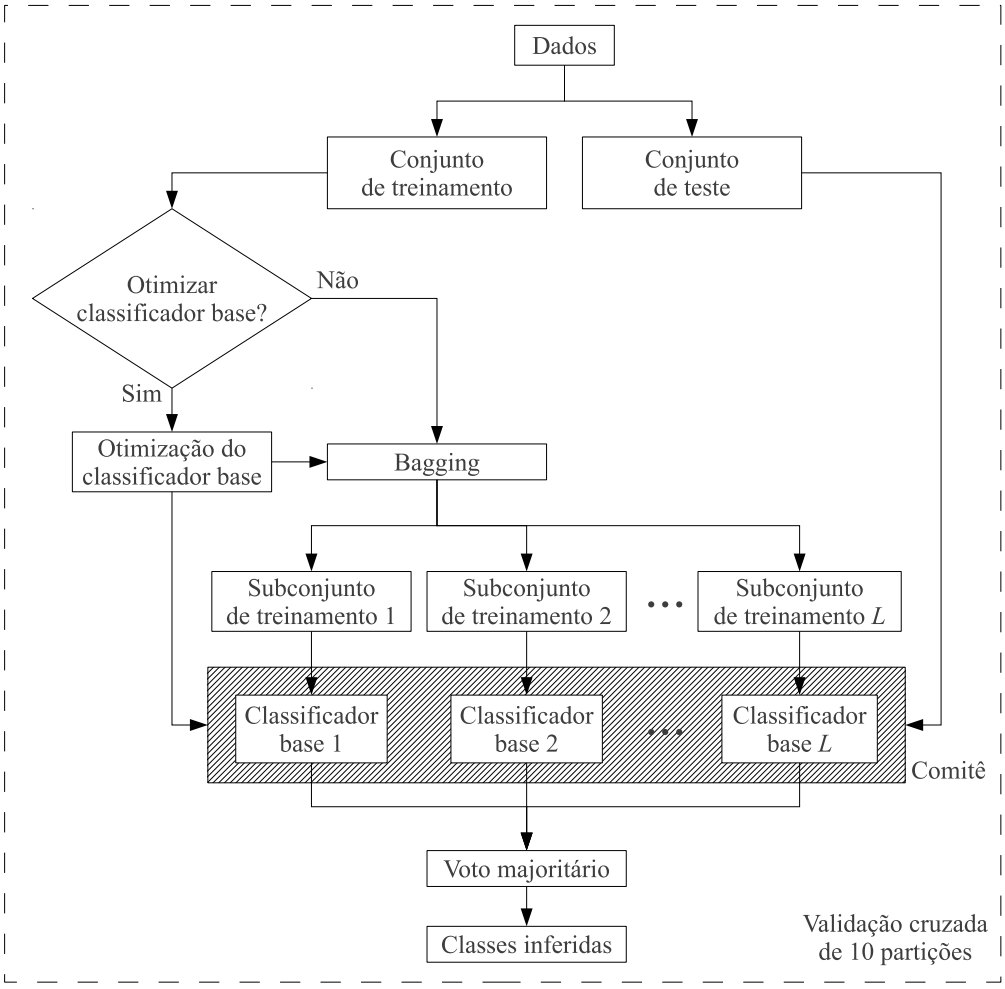


Figura 6.1: Fluxograma da metodologia de projeto e avaliação dos comitês de classificadores.

os k ciclos. Nesta dissertação optou-se por usar $k = 10$ para o teste das diversas arquiteturas de comitês.

6.2 Otimização Metaheurística dos Classificadores Base

É importante perceber que a arquitetura ilustrada na Figura 6.1 é independente de como o bloco de "Otimização do classificador base" é realizado. Como mencionado antes, nesta dissertação a otimização será feita através de uma abordagem metaheurística via algoritmo I-HPSO. A Figura 6.2 detalha melhor a etapa de otimização de um classificador base.

A função objetivo $f(\boldsymbol{\theta})$ do algoritmo metaheurístico será dada pela taxa de

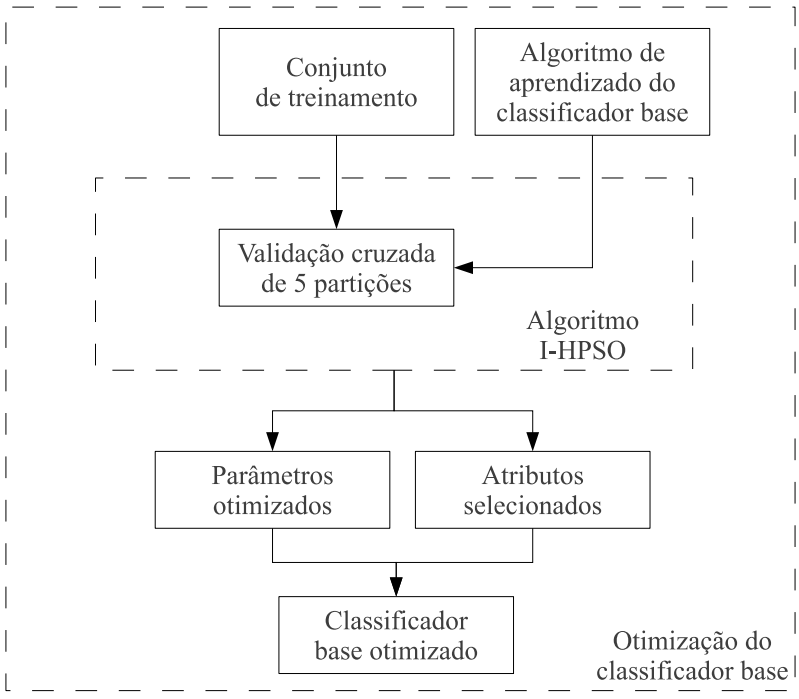


Figura 6.2: Diagrama de blocos do processo de otimização do classificador base.

acerto média de uma validação cruzada de 5 partições, ou seja,

$$f(\boldsymbol{\theta}) = \frac{1}{5}\sum_{k=1}^{5} acuracia_{\psi}(\boldsymbol{\theta}, k), \tag{6.1}$$

em que $\boldsymbol{\theta}$ é o vetor de parâmetros que representa a solução avaliada e a função $acuracia_{\psi}(\boldsymbol{\theta}, k)$ é a taxa de acerto obtida com esse vetor, considerando-se o algoritmo de aprendizado ψ, na k-ésima partição da validação cruzada.

A metodologia de otimização dos parâmetros e seleção de atributos envolve, portanto, seguidas iterações de avaliação da Equação (6.1) ao longo das iterações do algoritmo I-HPSO. Note ainda que todo o processo de otimização resumido na Figura 6.2 é repetido para cada iteração do processo de validação cruzada de 10 partições realizado nos testes representados na Figura 6.1. São encontrados portanto diferentes conjuntos de parâmetro para cada partição diferente usado como conjunto de treinamento. Essa metodologia é semelhante à proposta em (HUANG; DUN, 2008).

Os elementos do vetor $\boldsymbol{\theta}$ mudam de acordo com o classificador base ψ a ser otimizado. Na Tabela 6.1 apresenta as diferentes formatações de solução usadas. Note que todas as soluções apresentam um segmento formado por variáveis binárias

Constantes
R: número de atributos disponíveis
d_i: Variável binária indicando se o i-ésimo atributo é usado ($d_i = 1$) ou não ($d_i = 0$).
Redes FAM e Fuzzy ART-Ci, $i \in \{1, 2, 3\}$
$\boldsymbol{\theta} = [\alpha \;\; \rho \;\; \eta \;\mid\; d_1 \;\; d_2 \;\cdots\; d_R]^T$, $\boldsymbol{\theta} \in \mathbb{R}^{3+R}$
$\alpha \in [0, 1]$: parâmetro de escolha
$\rho \in [0, 0{,}999]$: parâmetro de vigilância
$\eta \in [0, 1]$: passo de aprendizagem
Redes SOM-Ci, $i \in \{1, 2, 3\}$
$\boldsymbol{\theta} = [P_1 \;\; P_2 \;\; \eta_0 \;\; \eta_f \;\; \sigma_0 \;\; \sigma_f \;\mid\; d_1 \;\; d_2 \;\cdots\; d_R]^T$, $\boldsymbol{\theta} \in \mathbb{R}^{6+R}$
$P_1 \in [1, 10]$ e $P_2 \in [1, 10]$: dimensões da rede $P_1 \times P_2$
$\eta_0 \in [0, 1]$ e $\eta_f \in [0, 1]$: valores inicial e final do passo de aprendizagem
$\sigma_0 \in [0, 10]$ e $\sigma_f \in [0, 10]$: valores inicial e final do parâmetro de espalhamento
Rede LVQ
$\boldsymbol{\theta} = [N_c \;\; \eta_0 \;\mid\; d_1 \;\; d_2 \;\cdots\; d_R]^T$, $\boldsymbol{\theta} \in \mathbb{R}^{2+R}$
$N_c \in [1, 10]$: número de neurônios por classe
$\eta_0 \in [0, 1]$: passo de aprendizagem inicial
Rede ELM
$\boldsymbol{\theta} = [N_w \;\mid\; d_1 \;\; d_2 \;\cdots\; d_R]^T$, $\boldsymbol{\theta} \in \mathbb{R}^{1+R}$
$N_w \in [1, 100]$: número de neurônios ocultos

Tabela 6.1: Vetores de soluções usados na otimização metaheurística dos classificadores base dos comitês avaliados.

referentes à seleção dos atributos usados, conforme discutido na Seção 5.4.3. São apresentados ainda na Tabela 6.1 os valores limites escolhidos para os parâmetros dos classificadores. Apesar de algumas redes apresentarem os mesmos parâmetros a serem ajustados, como as redes da família ART, a otimização é feita separadamente para cada classificador e para cada conjunto de treinamento.

6.3 Comparação de Desempenho via Teste de Hipótese

Ao comparar métodos de classificação distintos, se faz necessário determinar uma metodologia de avaliação objetiva. Vários autores advogam o uso sistemático de testes estatísticos para a avaliação de algoritmos de classificação, especialmente redes neurais (FLEXER, 1996; SALZBERG, 1997; DIETTERICH, 1998; DEMŠAR,

Tabela 6.2: Situações possíveis na aplicação do teste t-pareado.

	H_0 é verdadeira	H_1 é verdadeira
Aceita-se H_0	Decisão correta	Erro do Tipo II (β)
Rejeita-se H_0	Erro do Tipo I (α)	Decisão correta

2006).

Entretanto, é importante ressaltar que testes estatísticos assumem características que muitas vezes não são atendidas, como certas considerações de independência (DIETTERICH, 1998). Dessa maneira, os testes usados nesta dissertação devem ser vistos como aproximações heurísticas.

A seguir são detalhados os testes estatísticos usados nesta dissertação: teste t-pareado e teste de Wilcoxon. Os valores de média e variância usados nos testes a seguir são estimados a partir dos resultados obtidos com o teste de validação cruzada de 10 partições descrito na Seção 6.2.

6.3.1 Teste t-Pareado

No teste t-pareado deseja-se comparar dois algoritmos a partir da definição de duas hipóteses (MONTEIRO, 2009):

Hipótese nula (H_0). Não existe diferença significativa entre os desempenhos dos métodos comparados.

Hipótese alternativa (H_1). Existe diferença significativa entre os desempenhos dos algoritmos.

Por serem hipóteses mutuamente exclusivas, a não-rejeição de uma implica na rejeição da outra.

A Tabela B.1 mostra todas as situações possíveis ao se aplicar o teste t-pareado (MONTEIRO, 2009). Nesta dissertação somente o erro do tipo I é levado em consideração, ou seja, somente o erro de rejeitar a hipótese nula mesmo ela sendo a verdadeira é verificado.

Sejam os dois classificadores a serem comparados, denotados por A e B. Seja ainda M_A e M_B as taxas médias de acerto obtidas respectivamente pelos algoritmos A e B durante R_A e R_B execuções da fase de teste. Seja também σ_A^2 e σ_B^2 as variâncias das taxas de acerto calculadas nessas diferentes execuções. O teste t

de Student consiste em calcular a seguinte estatística (BOSLAUGH; WATTERS, 2008):

$$t = \frac{M_A - M_B}{\sqrt{\frac{\sigma_A^2}{R_A - 1} + \frac{\sigma_B^2}{R_B - 1}}}. \tag{6.2}$$

Para $M = M_A - M_B$, $\sigma^2 = \sigma_A^2 + \sigma_B^2$ e $R = R_A = R_B$, tem-se:

$$t = \frac{M}{\sqrt{\frac{\sigma^2}{R-1}}}. \tag{6.3}$$

O valor da estatística t deve ser comparado com o valor tabelado $t_{\frac{\alpha}{2},R-1}$ (tabela de valores disponível no Apêndice B), em que α refere-se à tolerância de erro do tipo I aceitável. A comparação deve ser feita da seguinte maneira:

- Se $-t_{\frac{\alpha}{2},R-1} \leq t \leq t_{\frac{\alpha}{2},R-1}$, aceita-se a hipótese nula (H_0).
- Caso contrário, rejeita-se a hipótese nula (H_0).

Nesta dissertação adotar-se-á uma tolerância $\alpha = 0{,}05$ (5%). Como nos testes usa-se validação cruzada de 10 partições, correspondendo a $R = 10$ execuções, tem-se o valor $t_{\frac{\alpha}{2},R-1} = t_{0{,}025;9} = 2{,}685$.

6.3.2 Teste de Wilcoxon

O teste de Wilcoxon constitui uma alternativa não paramétrica a comparações estatísticas como o teste t-pareado. Em suma, este teste ranqueia as diferenças de desempenhos entre dois algoritmos para diferentes conjuntos de dados, ignorando os sinais, e compara os ranques referentes às diferenças positivas e negativas (DEMŠAR, 2006).

Os seguintes passos devem ser seguidos para a realização desse teste:

- A partir de N_D conjuntos de dados diferentes, calcular todas as diferenças de desempenho $d_i = d_i^A - d_i^B, i = 1, \ldots, N_D$, entre os algoritmos A e B.
- Ranquear as diferenças d_i de acordo com seus valores absolutos, atribuindo postos S_i. No caso de empates, são determinados postos médios.
- Calcular S^+ e S^- como sendo a soma dos ranques correspondentes às diferenças positivas e negativas, respectivamente. Em caso de haver diferenças nulas, seus

ranques são divididos igualmente entre S^+ e S^-. Este passo é resumido pelas equações abaixo.

$$S^+ = \sum_{d_i>0} \text{ranque}(d_i) + \frac{1}{2}\sum_{d_i=0} \text{ranque}(d_i), \quad (6.4)$$

$$S^- = \sum_{d_i<0} \text{ranque}(d_i) + \frac{1}{2}\sum_{d_i=0} \text{ranque}(d_i). \quad (6.5)$$

- Calcular $S = \min(S^+, S^-)$ e comparar o resultado com os valores de uma tabela de valores críticos para o teste de Wilcoxon (tabela de valores disponível no Apêndice C). Normalmente essas tabelas possuem valores críticos até $N_D = 25$ e variam de acordo com a tolerância de erro escolhida[1]. Por exemplo, para $N_D = 10$ e uma tolerância de erro de $\alpha = 0{,}05$ (5%), tem-se o valor crítico 8. Caso $S \leq 8$, a hipótese nula é rejeitada. Caso contrário, a mesma é aceita e os métodos são considerados estatisticamente semelhantes.

Nota-se, portanto, que o teste de Wilcoxon permite comparar simultaneamente o desempenho de dois métodos para vários conjuntos de dados.

6.4 Conclusões

Neste capítulo foram detalhadas as metodologias de construção e avaliação dos comitês de classificadores de padrões, foco principal do estudo desta dissertação. Foi discutido ainda como usar o algoritmo de otimização metaheurística I-HPSO para buscar parâmetros adequados e selecionar atributos para os classificadores base dos comitês.

Foram apresentadas ainda duas técnicas estatísticas para comparação de algoritmos de aprendizado, sendo uma paramétrica, o teste t-pareado, e outra não paramétrica, o teste de Wilcoxon. A aplicação dessas técnicas conjuntamente com as estatísticas tradicionais (média e variância) proporcionará uma análise correta dos classificadores desenvolvidos.

No Capítulo 7 serão apresentados e discutidos os resultados obtidos a partir

[1]No caso de $N_D > 25$, pode-se usar uma aproximação gaussiana com média $\mu_S = \frac{N_D(N_D+1)}{4}$ e desvio-padrão $\sigma_S = \sqrt{\frac{N_D(N_D+1)(2N_D+1)}{24}}$. Dessa maneira, deve-se comparar o valor $Z = \frac{S-\mu_S}{\sigma_S}$ com o valor percentil desejado da distribuição gaussiana. Nesta dissertação essa aproximação não é usada, pois serão avaliados menos de 25 conjuntos de dados.

de simulações computacionais dos comitês de classificadores avaliados para vários conjuntos de dados reais.

Capítulo 7

Resultados Experimentais

Neste capítulo são apresentados e discutidos os resultados obtidos a partir de simulações computacionais desenvolvidas. Os testes de classificação de padrões envolvem doze conjuntos de dados reais do banco UCI (FRANK; ASUNCION, 2010). A Tabela 7.1 resume as características dos bancos de dados usados.

Tabela 7.1: Resumo dos conjuntos de dados usados nos testes.

Conjunto de dados	**Amostras**	**Atributos**	**Classes**
Breast-w	683	9	2
Car	1728	6	4
Vertebral Column	310	6	3
Credit	653	15	2
Dermatology	358	34	6
Glass	214	9	6
Haberman	306	3	2
Heart	270	13	2
Ionosphere	351	34	2
Sonar	208	60	2
Votes	435	16	2
Wall-Following	5456	2	4

Todas as N amostras disponíveis nos conjuntos de dados da Tabela 7.1 foram normalizadas no intervalo $[0, 1]$ antes das etapas de treinamento e teste pela equação

abaixo:

$$a_j(n) = \frac{a_j(n) - a_j^{min}}{a_j^{max} - a_j^{min}}, \forall n, \tag{7.1}$$

em que $a_j(n)$ é a j-ésima componente do n-ésimo padrão de entrada e a_j^{max} e a_j^{min} são dados por

$$a_j^{max} = \max_{n=1,\cdots,N}\{a_j(n)\} \qquad \text{e} \qquad a_j^{min} = \min_{n=1,\cdots,N}\{a_j(n)\}. \tag{7.2}$$

As simulações computacionais foram desenvolvidas e executadas em ambiente Ubuntu Linux 10.10 com linguagem de programação C++ e pacote de bibliotecas de manipulação matemática IT++ versão 4.0.7, disponível em `http://sourceforge.net/apps/wordpress/itpp/`.

7.1 Experimentos de otimização dos classificadores base

Uma etapa importante da obtenção dos comitês a serem analisados é a otimização dos classificadores base via algoritmo I-HPSO. Para aplicação deste é necessário a especificação de seus parâmetros de funcionamento. Tais parâmetros são resumidos na Tabela 7.2.

Tabela 7.2: Parâmetros do algoritmo I-HPSO durante a otimização dos classificadores base para os conjuntos de dados avaliados.

Parâmetro	**Valor usado**
Número de partículas (N_p)	20
Número de gerações (L_{PSO})	50
Coeficientes de aceleração (c_1 e c_2)	$c_1 = c_2 = 2{,}05$
Parâmetro de constrição (χ)	0,72984
Número de iterações da etapa de SA (L_{SA})	10
Parâmetro de recozimento (λ)	0,94
Passo de incremento (η_{SA})	0,01

Os valores dos parâmetros c_1, c_2 e χ foram tomados de Bratton e Kennedy (2007), enquanto o valor de λ é o mesmo usado em He e Wang (2007). Os demais valores foram obtidos experimentalmente. Os valores da Tabela 7.2 serão os mesmos usados na otimização de todas as redes neurais e bancos de dados.

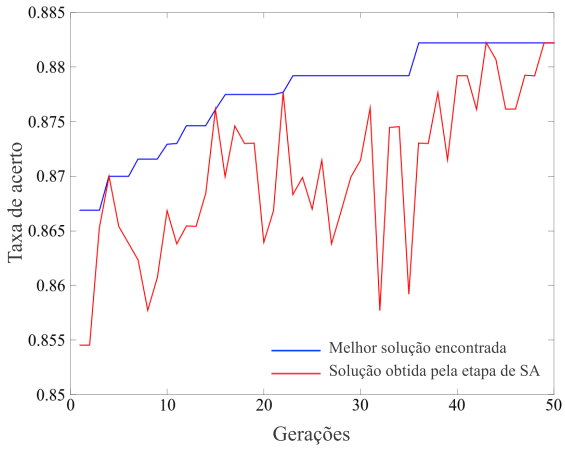
É interessante verificar a maneira como o algoritmo I-HPSO realiza a otimização dos classificadores base. As Figuras 7.1 e 7.2 ilustram esse processo de otimização para os nove diferentes classificadores base estudados, considerando parâmetros e atributos inicialmente aleatórios e o banco de dados *Credit*. A linha em azul representa a evolução da função objetivo (taxa de acerto resultante da validação cruzada de 5 partições no conjunto de treinamento), enquanto a linha vermelha indica as oscilações provocadas pela etapa de SA do algoritmo I-HPSO. Nota-se que a grande variação da linha vermelha, principalmente durante as primeiras gerações, é importante para evitar que o enxame convirja prematuramente para uma solução não desejável, permitindo uma melhor exploração do espaço de soluções.

Outra observação relevante é perceber que de fato o ajuste dos parâmetros dos algoritmos de classificação, assim como a escolha dos atributos mais apropriados, incrementa as taxas de acerto obtidas, revelando o quanto as técnicas usadas são sensíveis a essas escolhas.

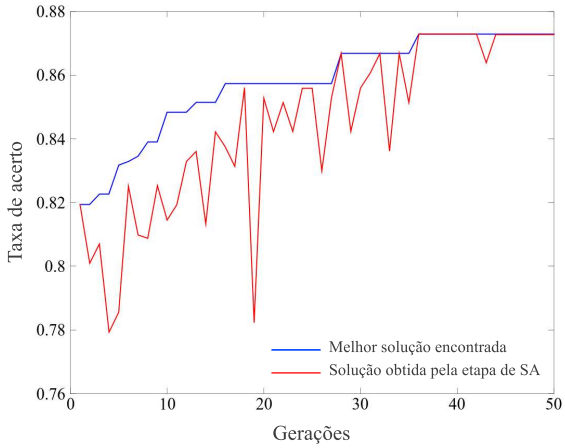
A Tabela 7.3 apresenta os valores médios encontrados pelo algoritmo I-HPSO durante o processo de otimização dos classificadores base para o banco de dados *Heart*. Observa-se que os valores médios dos classificadores da família ART são consideravelmente diferentes. Já os valores médios dos parâmetros das três variantes do modelo MUSCLE variam menos. Este comportamento sugere que os classificadores baseados em redes da família ART são mais sensíveis à escolha da estratégia de aprendizagem supervisionada (C1, C2 ou C3), pelo menos em relação aos valores dos seus parâmetros. É importante lembrar no entanto que ao longo das várias otimizações, referentes a cada um das 10 partições, são encontrados 10 diferentes conjuntos de parâmetros diferentes. Por este motivo o uso dos valores da Tabela 7.3 em um dado conjunto de teste não garante boas taxas de acerto médias.

Como o processo de otimização dos classificadores base é repetido para cada conjunto de treinamento diferente, em um ciclo de testes com validação cruzada de 10 partições, 10 escolhas de parâmetros e atributos são feitas. Reunindo os atributos selecionados a cada teste, é possível fazer um histograma a partir da frequência que cada atributo é selecionado. As Figuras 7.3 e 7.4 trazem histogramas para os nove comitês estudados, considerando o conjunto de dados *Heart*, que possui ao todo 13 atributos.

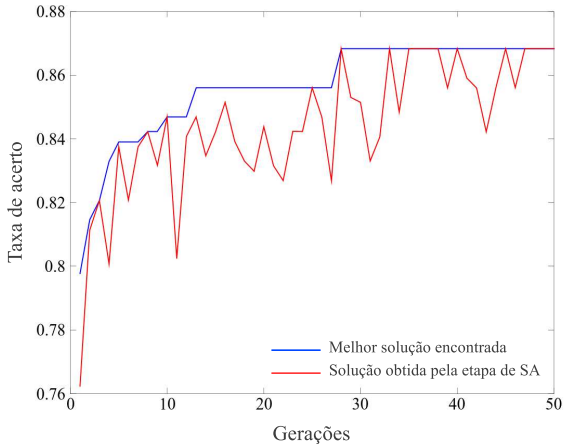
É interessante perceber que alguns atributos são selecionados na maioria das vezes para todos os classificadores, como os atributos representados pelos índices 3,



(a) ELM

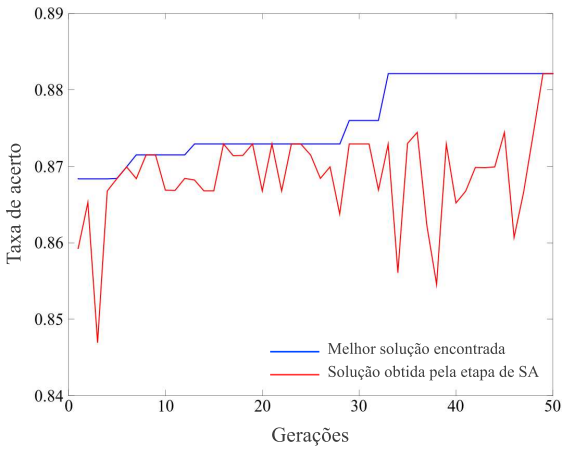


(b) FAM

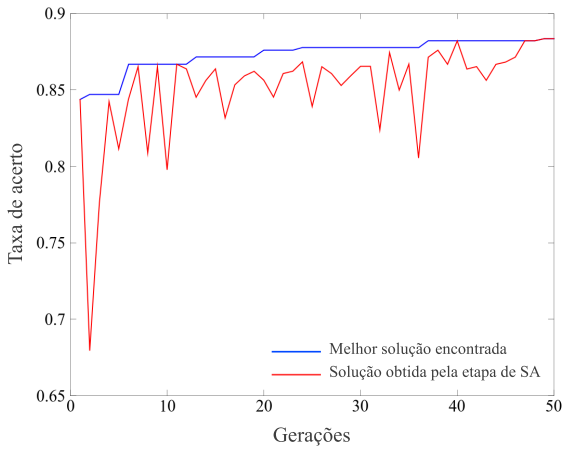


(c) LVQ

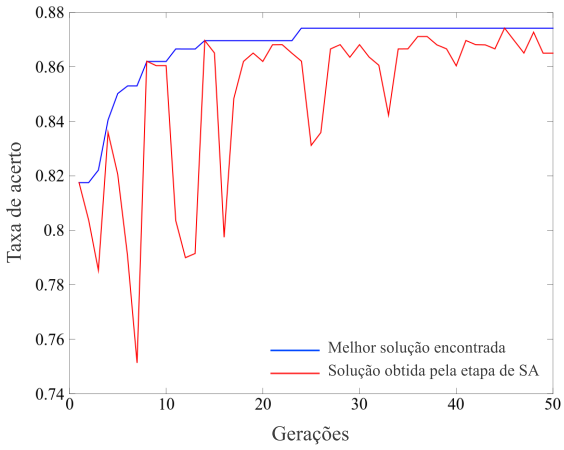
Figura 7.1: Processo de otimização dos classificadores base ELM, FAM e LVQ via algoritmo I-HPSO. O conjunto de dados usado para este experimento é o *Credit*.



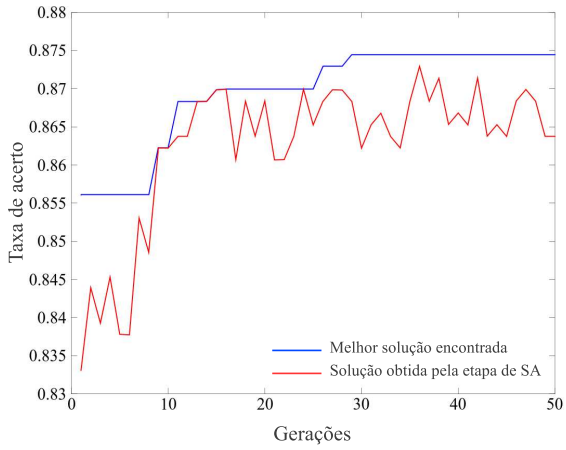
(a) Fuzzy ART-C1



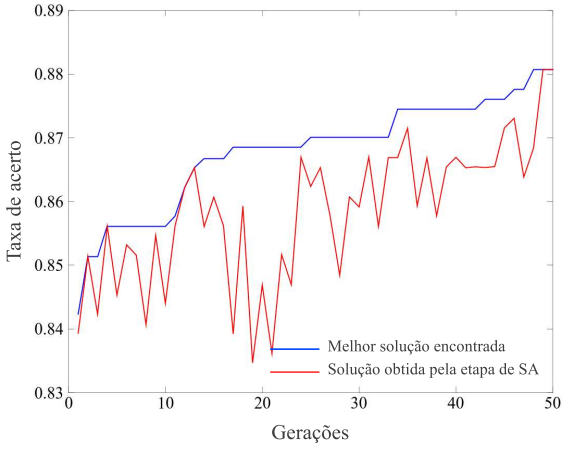
(b) Fuzzy ART-C2



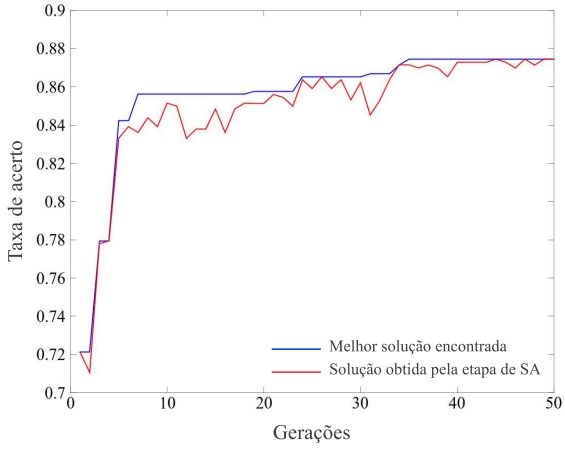
(c) Fuzzy ART-C3



(d) SOM-C1



(e) SOM-C2



(f) SOM-C3

Figura 7.2: Processo de otimização dos classificadores base Fuzzy ART e SOM via algoritmo I-HPSO. O conjunto de dados usado para este experimento é o *Credit*.

Tabela 7.3: Valores médios para os parâmetros otimizados via I-HPSO para o conjunto *Heart*.

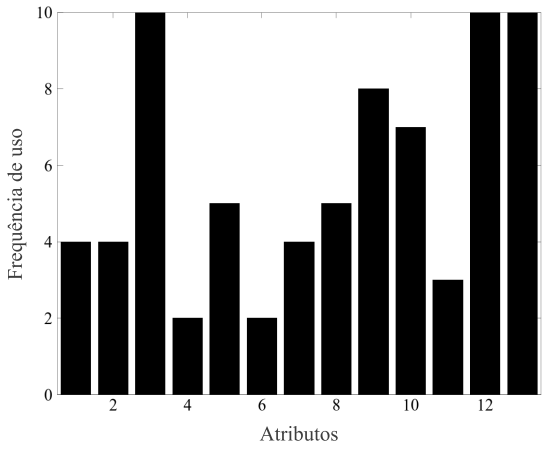
Classificador Base	Valores médios dos parâmetros
Fuzzy ART-C1	$\rho = 0{,}8603$, $\alpha = 0{,}4663$, $\eta = 0{,}4647$
Fuzzy ART-C2	$\rho = 0{,}3167$, $\alpha = 0{,}3326$, $\eta = 0{,}1356$
Fuzzy ART-C3	$\rho = 0{,}2576$, $\alpha = 0{,}1853$, $\eta = 0{,}05407$
MUSCLE-C1	$L_1 = 7{,}7$, $L_2 = 7{,}2$, $\alpha_0 = 0{,}5273$, $\alpha_T = 0{,}0635$, $\sigma_0 = 4{,}8092$, $\sigma_T = 0{,}4922$
MUSCLE-C2	$L_1 = 6{,}5$, $L_2 = 6{,}6$, $\alpha_0 = 0{,}5168$, $\alpha_T = 0{,}0495$, $\sigma_0 = 5{,}2810$, $\sigma_T = 0{,}4049$
MUSCLE-C3	$L_1 = 8{,}1$, $L_2 = 6{,}9$, $\alpha_0 = 0{,}4818$, $\alpha_T = 0{,}0642$, $\sigma_0 = 5{,}6742$, $\sigma_T = 0{,}3971$
FAM	$\rho = 0{,}4649$, $\alpha = 0{,}0236$, $\eta = 0{,}2901$
LVQ	7,7 neurônios por classe, $\eta_0 = 0{,}5377$
ELM	55,8 neurônios ocultos

12 e 13, enquanto outros são usados menos vezes, como os referentes aos índices 4, 5 e 8, sugerindo diferentes relevâncias entre os atributos disponíveis. Esses últimos não foram selecionados sequer uma vez pelos classificadores LVQ, SOM-C2 e SOM-C3 e somente uma vez pelo classificador SOM-C1.

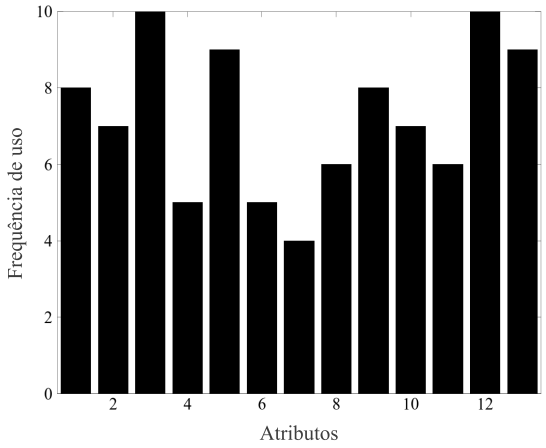
A Figura 7.5 mostra o número médio de atributos usados pelos classificadores após a etapa de otimização, ainda para o conjunto *Heart*. Verifica-se que para este conjunto o número médio de atributos encontra-se entre 6 e 9, correspondendo a uma redução entre 31% e 54% em relação aos 13 atributos originais. Nota-se ainda que os classificadores baseados em SOM e LVQ foram os que usaram menos atributos, em média, após o processo de otimização.

7.2 Resultados de classificação

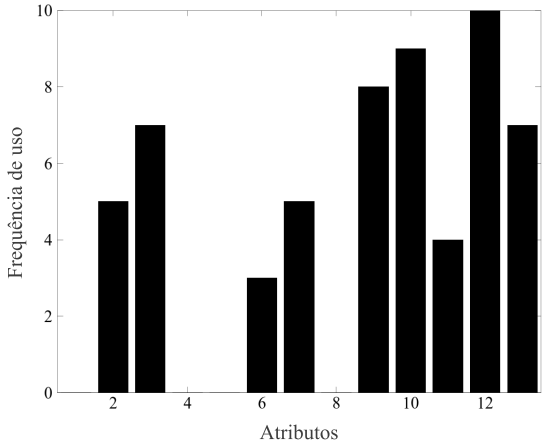
As Tabelas 7.4 e 7.5 apresentam os resultados de classificação obtidos para os comitês avaliados. Nestas são mostradas as taxas médias de acerto, os desvios padrões obtidos por cada método e a proporção média de atributos usados na



(a) ELM

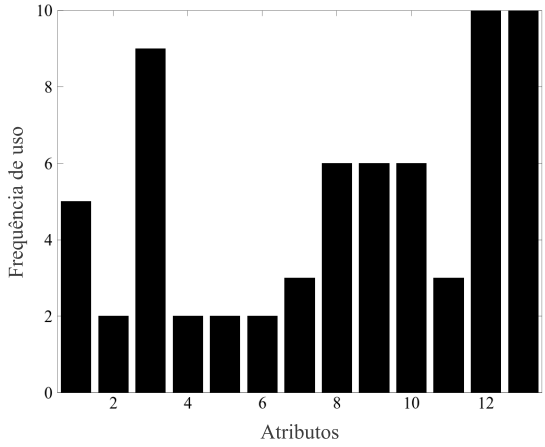


(b) FAM

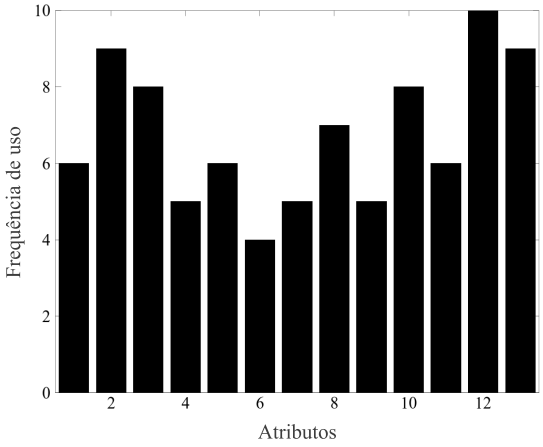


(c) LVQ

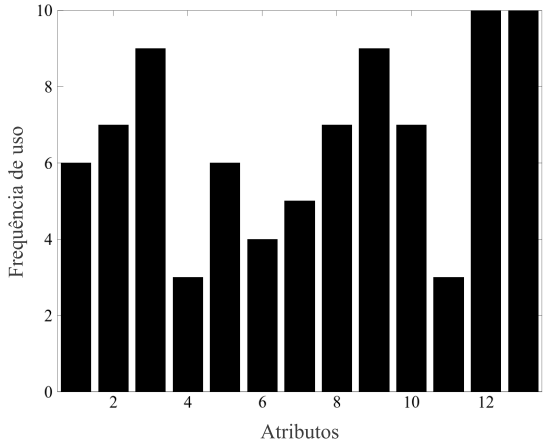
Figura 7.3: Histogramas dos atributos selecionados para classificadores base ELM, FAM e LVQ via algoritmo I-HPSO. O conjunto de dados usado para este experimento é o *Heart*.



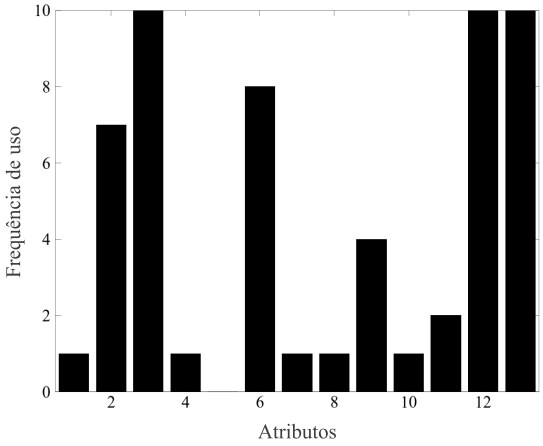
(a) Fuzzy ART-C1



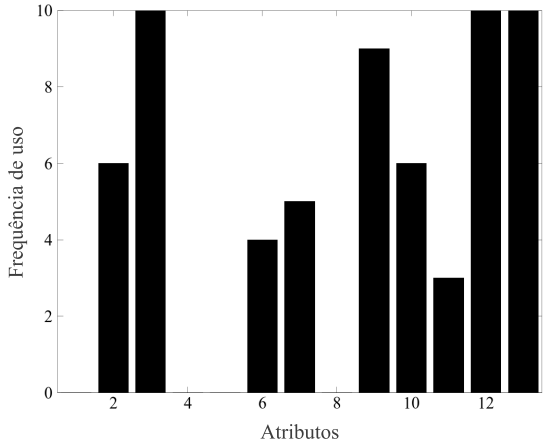
(b) Fuzzy ART-C2



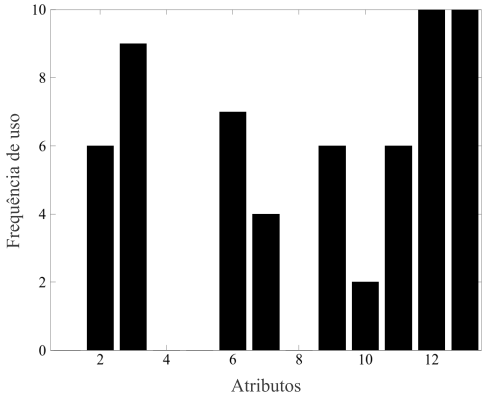
(c) Fuzzy ART-C3



(d) SOM-C1



(e) SOM-C2



(f) SOM-C3

Figura 7.4: Histogramas dos atributos selecionados para os classificadores base Fuzzy ART e SOM via algoritmo I-HPSO. O conjunto de dados usado para este experimento é o *Heart*.

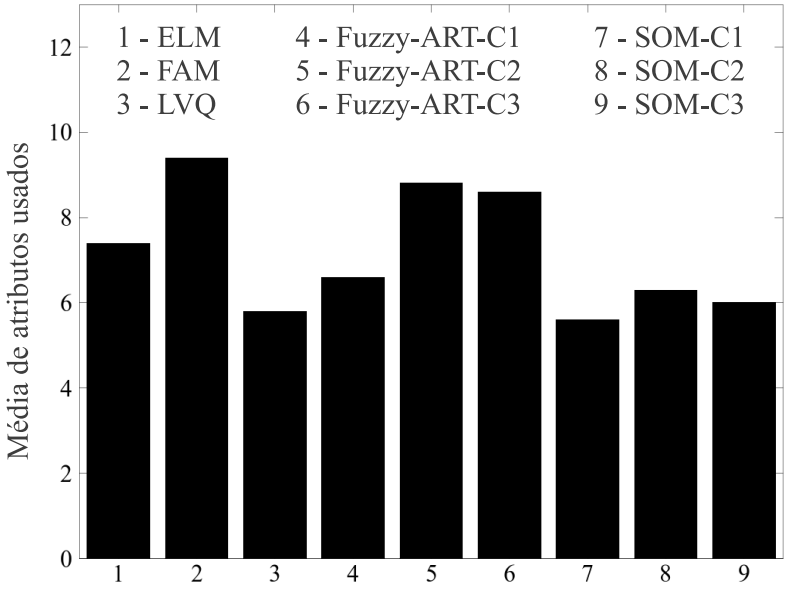


Figura 7.5: Média de atributos usados pelos classificadores base otimizados para o conjunto de dados *Heart*.

validação cruzada, dada por

$$\bar{n}^{atr} = \left\lceil \sum_{k=1}^{10} n_k^{atr}/10 \right\rceil, \tag{7.3}$$

em que n_k^{atr} é o número de atributos usados na k-ésima partição e $\lceil z \rceil$ indica a operação de escolher o menor inteiro maior que z. Os valores em negrito indicam o classificador que obteve melhor taxa média de acerto.

Pode-se perceber que os comitês ARTIE, em geral, apresentaram melhores desempenhos, com a variação ARTIE-C3 sendo a melhor em 4 dos 12 conjuntos de dados avaliados e o comitê ARTIE-C2 obtendo maiores taxas médias de acerto em 3 casos. Dentre os comitês MUSCLE, ambas as variante MUSCLE-C2 e MUSCLE-C3 apresentaram cada uma melhores resultados médios em 2 conjuntos, sendo que em um deles, *Heart*, apresentaram a mesma taxa de acerto. Os comitês de classificadores FAM, LVQ e ELM apresentaram resultados bons em vários testes, mas sempre inferiores a pelo menos uma variante dos comitês propostos.

Comparando os comitês da família ART, percebe-se que em todos os testes pelo menos uma variante do modelo ARTIE foi superior ao comitê de redes Fuzzy

Tabela 7.4: Resultados obtidos nos problemas de classificação (Parte 1). Os dois primeiros valores de cada teste são, respectivamente, a taxa média de acerto e o desvio padrão, ambos em porcentagem. O terceiro valor revela a proporção de atributos usados em média ao longo das execuções. Os campos em negrito realçam o classificador com maior taxa média de acerto para o conjunto de dados correspondente.

	Breast-w	**Car**	**Column**	**Credit**	**Dermatology**	**Glass**
ARTIE-C1	96,20 ±2,32 7/9	93,76 ±2,05 5/6	76,45 ±7,76 3/6	86,37 ±3,20 7/15	96,00 ±3,07 20/34	74,74 ±10,54 7/9
ARTIE-C2	96,78 ±1,35 7/9	97,50 ±0,95 6/6	80,00 ±4,76 5/6	**86,67** ±3,43 10/15	96,34 ±1,43 22/34	75,07 ±7,80 7/9
ARTIE-C3	97,07 ±2,40 7/9	**98,03** ±0,68 6/6	79.68 ±6,98 5/6	85,60 ±2,94 8/15	**98,29** ±2,41 20/34	**77,05** ±11,37 7/9
MUSCLE-C1	**97,08** ±1,95 7/9	92,43 ±2,10 5/6	83.23 ±6,94 5/6	84,38 ±2,40 5/15	95,48 ±2,80 21/34	71,41 ±7,76 5/9
MUSCLE-C2	96,04 ±3,54 7/9	95,16 ±1,86 5/6	**86,13** ±5,28 4/6	85,61 ±2,30 5/15	96,39 ±2,27 19/34	71,41 ±11,22 6/9
MUSCLE-C3	96,64 ±1,96 7/9	93,53 ±2,38 5/6	85,48 ±6,13 5/6	85,61 ±2,05 5/15	97,48 ±2,50 19/34	69,43 ±7,36 7/9
FAM comitê	95,90 ±2,28 8/9	97,11 ±0,87 6/6	78,39 ±5,90 4/6	85,45 ±1,64 11/15	96,91 ±3,15 21/34	74,67 ±7,38 7/9
LVQ comitê	96,64 ±2,59 7/9	91,67 ±2,93 5/6	83,55 ±5,98 5/6	85,15 ±4,01 5/15	96,34 ±2,38 20/34	70,93 ±9,37 6/9
ELM comitê	96,34 ±2,32 7/9	94,39 ±2,10 5/6	83,23 ±5,65 4/6	84,85 ±3,51 9/15	96,39 ±2,27 18/34	65,37 ±8,15 6/9

Tabela 7.5: Resultados obtidos nos problemas de classificação (Parte 2). Os dois primeiros valores de cada teste são, respectivamente, a taxa média de acerto e o desvio padrão, ambos em porcentagem. O terceiro valor revela a proporção de atributos usados em média ao longo das execuções. Os campos em negrito realçam o classificador com maior taxa média de acerto para o conjunto de dados correspondente.

	Haberman	**Heart**	**Ionosphere**	**Sonar**	**Votes**	**Wall-Following**
ARTIE-C1	73,50 ±8,62 3/3	81,11 ±10,25 7/13	**93,73** ±2,95 14/34	83,79 ±7,80 30/60	93,56 ±1,46 6/16	98,70 ±0,46 2/2
ARTIE-C2	72,67 ±7,67 2/3	77,78 ±77,78 9/13	90,87 ±4,83 14/34	85,14 ±6,50 33/60	**95,68** ±2,61 8/16	**99,94** ±0,09 2/2
ARTIE-C3	74,06 ±10,57 2/3	82,59 ±9,25 9/13	90,87 ±5,53 16/34	**87,29** ±5,10 32/60	94,26 ±2,93 8/16	99,93 ±0,13 2/2
MUSCLE-C1	74,22 ±7,06 2/3	82,96 ±9,27 6/13	90,31 ±3,36 17/34	84,57 ±5,51 32/60	93,56 ±2,40 4/16	95,51 ±1,20 2/2
MUSCLE-C2	73,89 ±4,95 2/3	**83,33** ±8,42 7/13	90,31 ±5,08 18/34	87,00 ±6,32 32/60	93,84 ±2,45 5/16	97,86 ±0,66 2/2
MUSCLE-C3	**75,44** ±6,74 2/3	**83,33** ±5,59 6/13	87,48 ±4,25 18/34	83,93 ±12,17 31/60	94,03 ±3,31 5/16	95,80 ±1,23 2/2
FAM comitê	68,83 ±7,54 3/3	77,41 ±8,45 10/13	91,73 ±3,69 16/34	83,29 ±6,30 31/60	93,59 ±2,54 8/16	99,78 ±0,14 2/2
LVQ comitê	72,17 ±7,46 2/3	79,26 ±8,94 6/13	86,88 ±5,44 19/34	77,43 ±7,78 30/60	94,96 ±3,02 4/16	93,79 ±1,47 2/2
ELM comitê	74,44 ±6,35 3/3	81,11 ±7,70 8/13	85,17 ±4,84 11/34	79,29 ±7,53 25/60	93,59 ±2,77 7/16	93,61 ±1,16 2/2

ARTMAP. O número de atributos selecionados pelos classificadores da família ART foram semelhantes na maioria dos casos. É importante lembrar que a rede Fuzzy ARTMAP é originalmente supervisionada, enquanto a rede Fuzzy ART somente foi capaz de realizar aprendizagem supervisionada após o uso das estratégias C1, C2 ou C3.

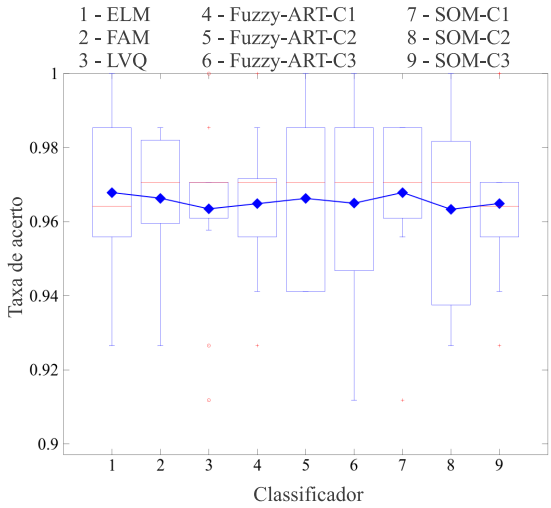
É de interesse analisar maiores detalhes dos testes de classificação realizados, além de compará-los com as taxas obtidas pelos classificadores base quando usados isoladamente. Nas Figuras 7.6, 7.7, 7.8 e 7.9 são mostrados os diagramas de caixa (*boxplot*, em inglês) dos resultados obtidos pelos comitês de classificadores, comparando com os resultados de classificadores individuais. Esses gráficos possuem as seguintes características (FREIXA *et al.*, 1992):

- O retângulo (caixa) tem início na posição do primeiro quartil e fim no terceiro quartil, reunindo portanto 50% dos valores analisados.
- A linha vermelha dentro da caixa indica a mediana das taxas de acerto obtidas pelo comitê em questão.
- Um segmento de reta é desenhando do primeiro quartil ao valor adjacente inferior, enquanto outro segmento vai do terceiro quartil ao valor adjacente superior. Esses segmentos são denominados bigodes.
- Os valores abaixo e acima dos bigodes são considerados atípicos (*outliers*) e representados por pequenas marcações em cruz.

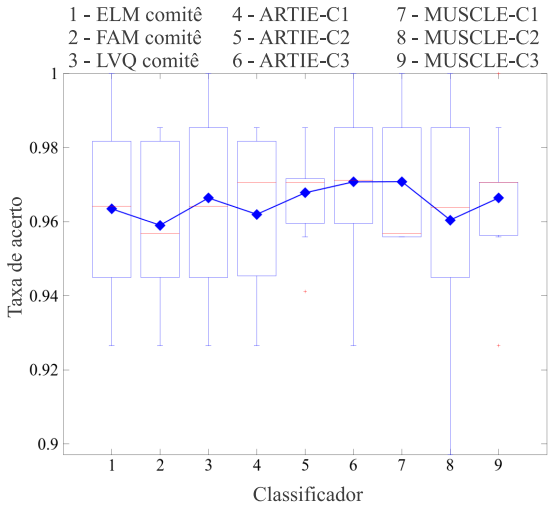
Foram sobrepostos aos *boxplots* os valores médios obtidos, representados pelas marcações azuis.

Pode-se perceber que, para alguns conjuntos de dados diferentes, classificadores se beneficiam mais que os outros quando agrupados em comitês. Na Figura 7.6, por exemplo, no conjunto *Breast* somente os comitês ARTIE-C2 e ARTIE-C3 apresentaram resultados melhores que os classificadores individuais, considerando tanto a acurácia média superior quanto a menor variabilidade ao longo dos testes (altura da caixa no *boxplot*). No entanto, na mesma figura observa-se casos como o do conjunto *Car*, em que todos os classificadores obtiveram melhores taxas de acerto quando reunidos em comitês.

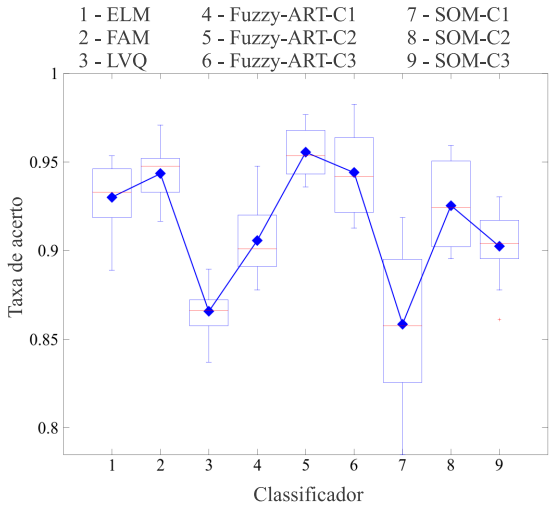
Nos gráficos referentes aos conjuntos *Credit* e *Dermatology*, apresentados na Figura 7.7, pode-se constatar a capacidade da técnica de aprendizado em comitês



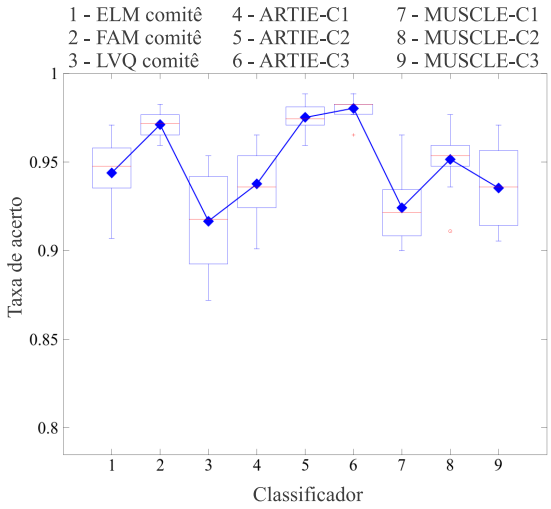
(a) Breast - um classificador



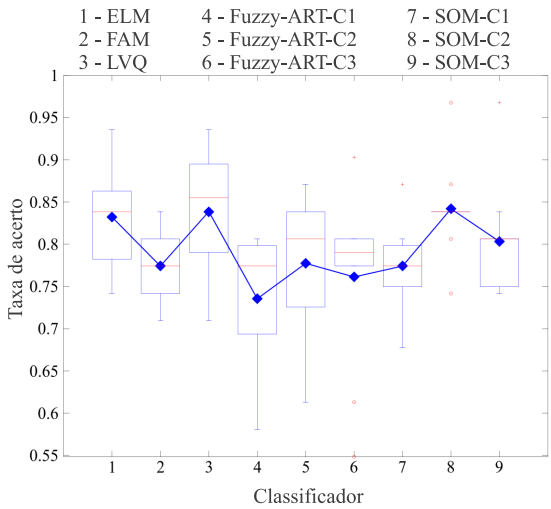
(b) Breast - comitê de classificadores



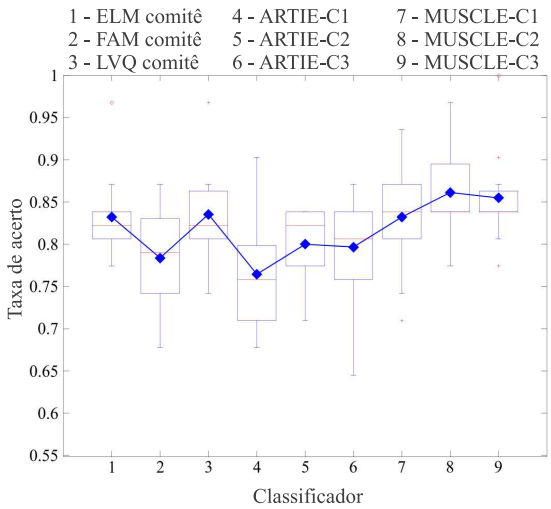
(c) Car - um classificador



(d) Car - comitê de classificadores

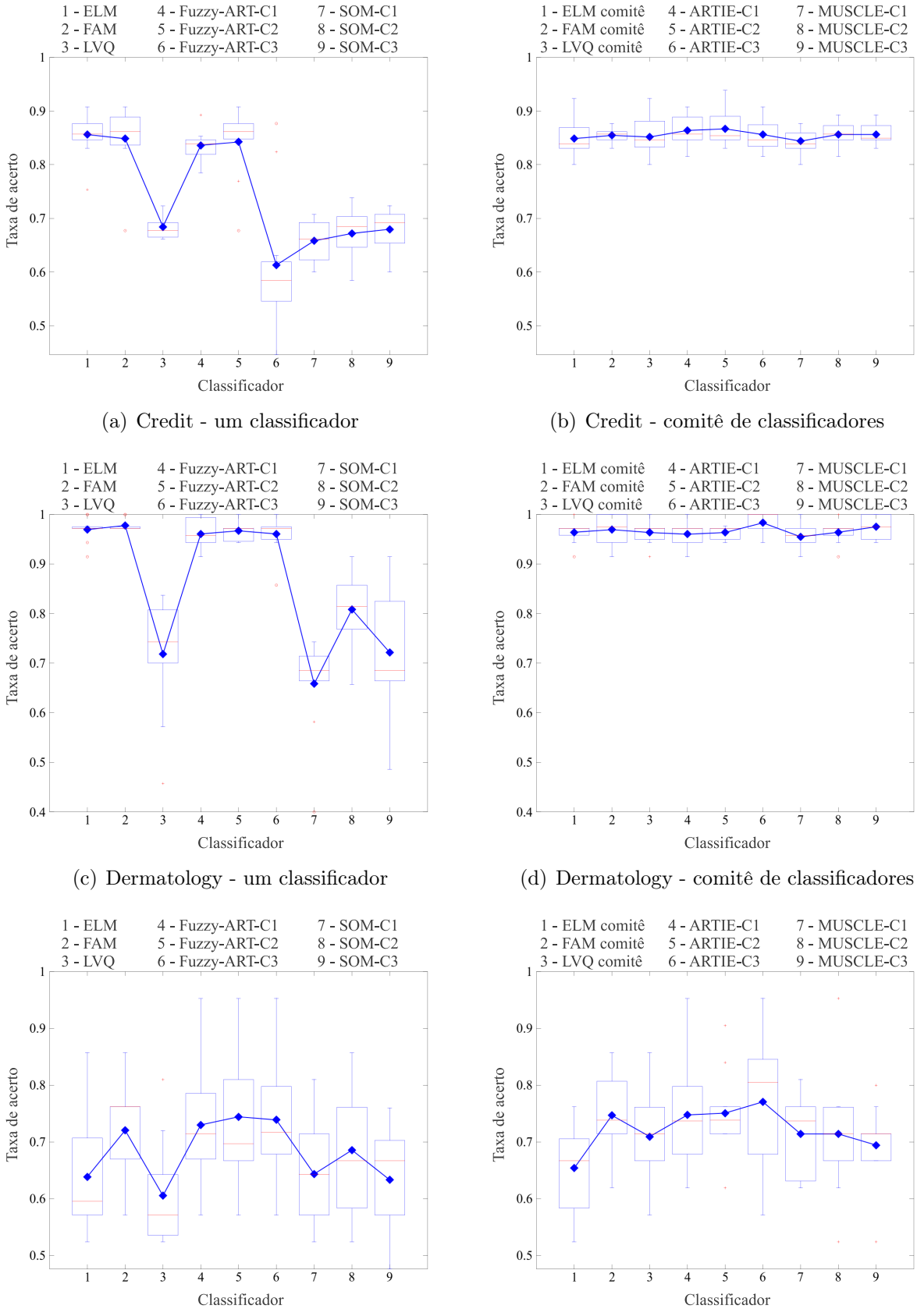


(e) Vertebral Column - um classificador



(f) Vertebral Column - comitê de classificadores

Figura 7.6: Gráficos de caixa dos classificadores e comitês de classificadores avaliados (Parte 1).



(a) Credit - um classificador

(b) Credit - comitê de classificadores

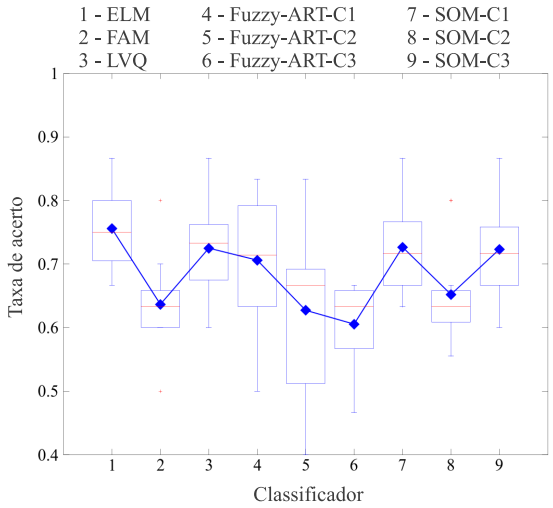
(c) Dermatology - um classificador

(d) Dermatology - comitê de classificadores

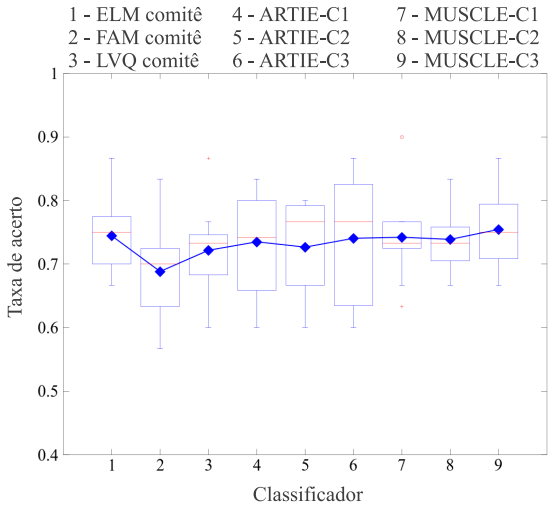
(e) Glass - um classificador

(f) Glass - comitê de classificadores

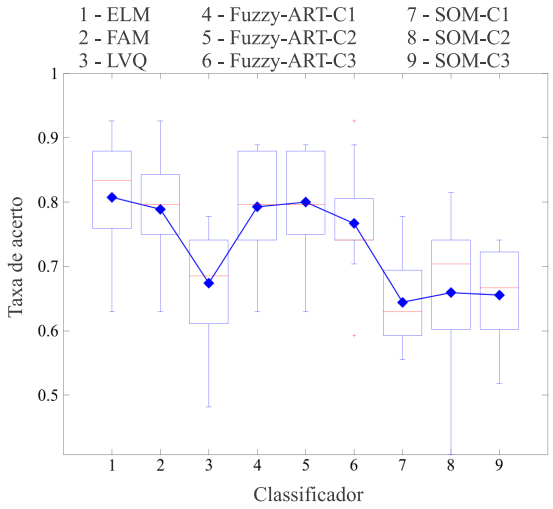
Figura 7.7: Gráficos de caixa dos classificadores e comitês de classificadores avaliados (Parte 2).



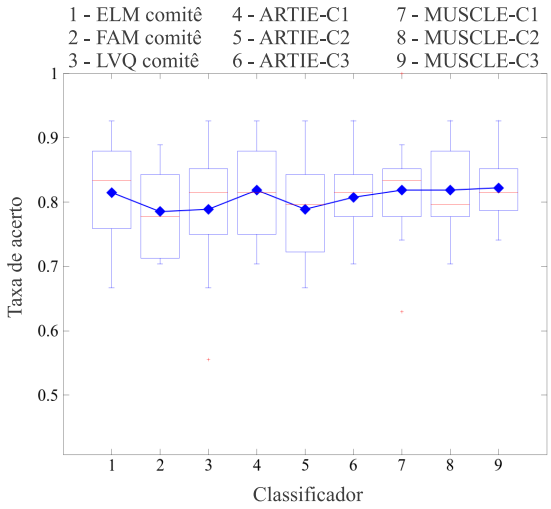
(a) Haberman - um classificador



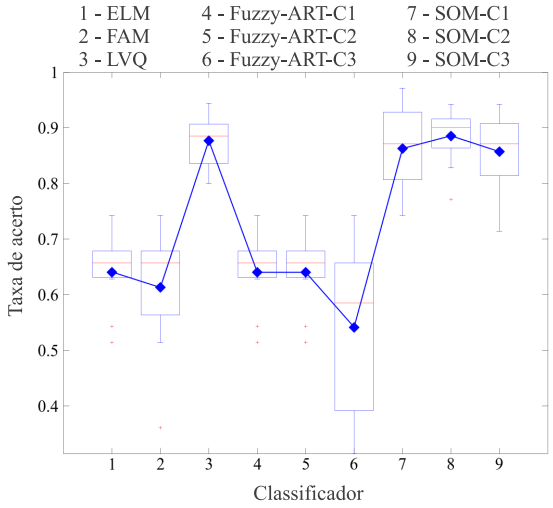
(b) Haberman - comitê de classificadores



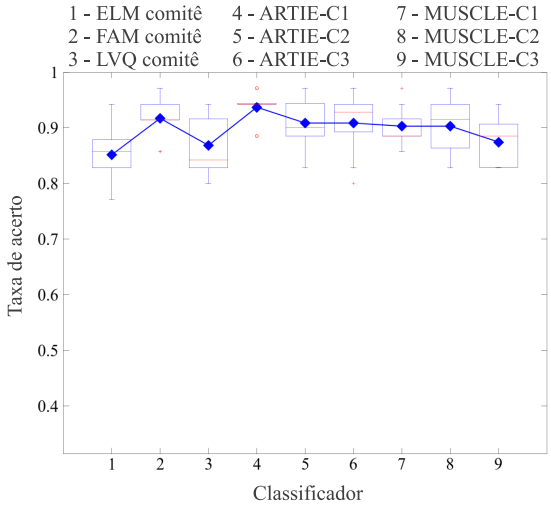
(c) Heart - um classificador



(d) Heart - comitê de classificadores

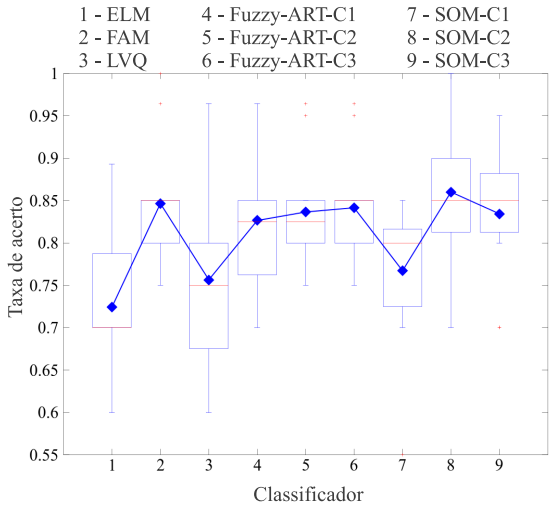


(e) Ionosphere - um classificador

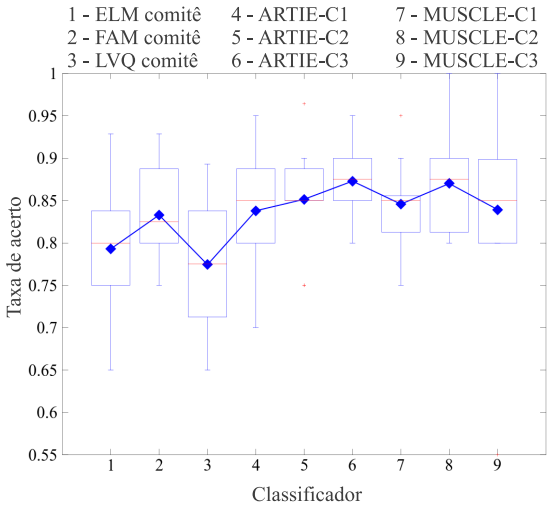


(f) Ionosphere - comitê de classificadores

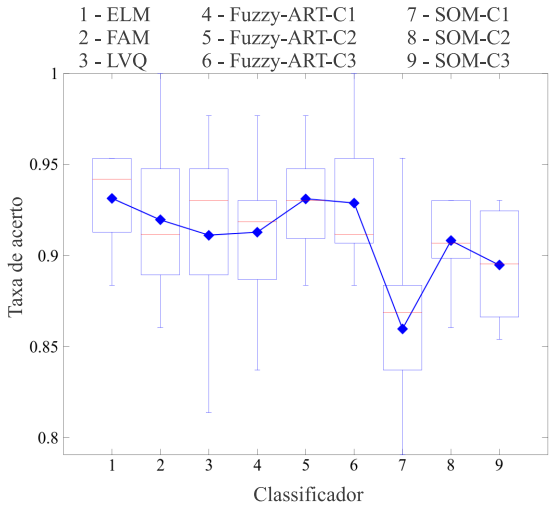
Figura 7.8: Gráficos de caixa dos classificadores e comitês de classificadores avaliados (Parte 3).



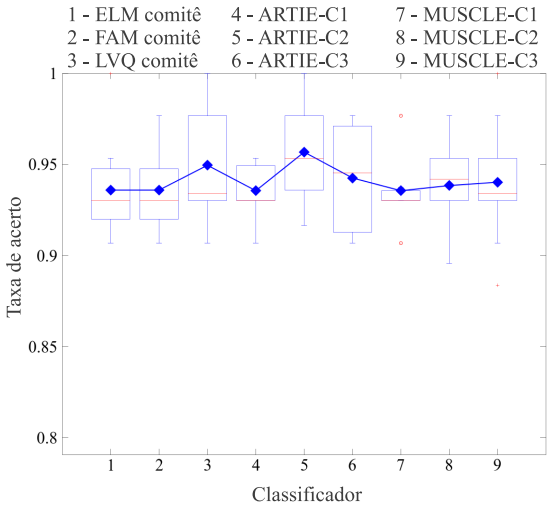
(a) Sonar - um classificador



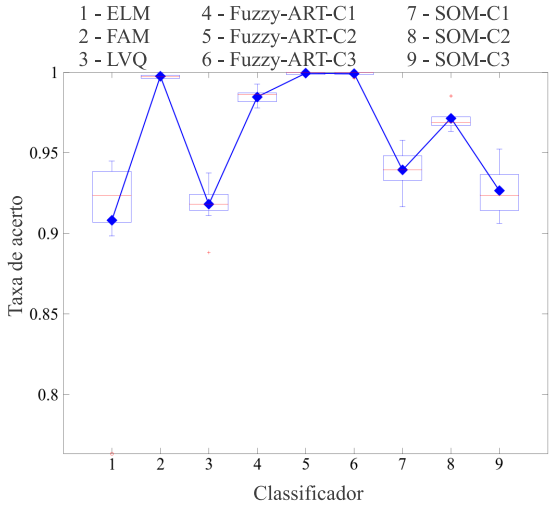
(b) Sonar - comitê de classificadores



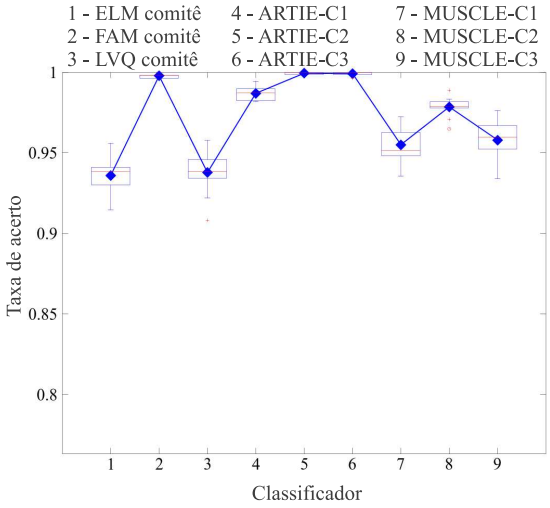
(c) Votes - um classificador



(d) Votes - comitê de classificadores



(e) Wall-Following - um classificador



(f) Wall-Following - comitê de classificadores

Figura 7.9: Gráficos de caixa dos classificadores e comitês de classificadores avaliados (Parte 4).

em agrupar classificadores com fraca capacidade de generalização e obter um classificador capaz de obter altas taxas de acerto e com reduzido desvio padrão.

Comportamento semelhante observa-se nos *boxplots* para o conjunto *Ionosphere*, na Figura 7.8. É interessante analisar os gráficos dos conjuntos *Haberman* e *Heart*, em que percebe-se que alguns métodos apresentaram melhora mais intensas quando agrupados, especificamente os comitês ARTIE-C2, ARTIE-C3 e MUSCLE-C2, no primeiro caso, e os comitês de LVQ e os modelos MUSCLE-C1, MUSCLE-C2 e MUSCLE-C3 no segundo caso. Esse comportamento diferenciado entre as técnicas avaliadas também pode ser notado nos gráficos da Figura 7.9.

Outra observação importante é que o comitê com maior capacidade de generalização não necessariamente é aquele obtido pelo melhor classificador base quando analisado individualmente. Essa característica pode ser notada para 8 conjuntos: *Car*, *Credit*, *Dermatology*, *Glass*, *Haberman*, *Heart*, *Ionosphere* e *Sonar*.

7.3 Testes estatísticos

Visando uma comparação formal dos resultados apresentados, serão aplicados os testes estatísticos detalhados na Seção 6.3.

A Tabela 7.6 resume a aplicação do teste t-pareado entre os modelos propostos (ARTIE e MUSCLE) e os três comitês de referência (FAM, LVQ e ELM). O critério para determinar que um classificador é estatisticamente superior ao outro depende do resultado da Equação (6.3), considerando um valor crítico $t_{0,025;9} \geq 2{,}685$, referente a uma tolerância de erro de 5% e 10 testes independentes (ver Tabela B.1). Note que os valores usados para calcular a Equação (6.3) são os mesmo apresentados anteriormente nas Tabelas 7.4 e 7.5.

Pelos valores da Tabela 7.6 pode-se perceber que na maioria dos casos as variantes dos comitês propostos foram estatisticamente melhores que os comitês de classificadores mais tradicionais. Em relação ao comitê de FAM, os classificadores ARTIE-C2, ARTIE-C3 e MUSCLE-C2 mostraram-se expressivamente superiores, enquanto as variantes ARTIE-C1 e MUSCLE-C3 apresentaram resultados semelhantes estatisticamente e somente o modelo MUSCLE-C1 revelou-se inferior. Quando comparados com o comitê de LVQ, todos os modelos propostos foram melhores. O mesmo pode ser verificado em relação ao comitê de ELM.

Diferentemente do teste t-pareado, que compara resultados por conjunto de dados, o teste não-paramétrico de Wilcoxon compara dois classificadores considerando as taxas médias de acerto em todos os testes simultaneamente. A Tabela 7.7 apresenta os resultados do teste de Wilcoxon, com valores calculados de acordo com o procedimento detalhado na Seção 6.3.2 e valor crítico $S \leq 14$ (para 12 bancos de dados e 5% de tolerância de erro), de acordo com a Tabela C.1.

Pelos resultados da Tabela 7.7, tem-se que, em relação ao comitê de FAM, somente os modelos ARTIE-C2 e ARTIE-C3 foram superiores estatisticamente pelo critério do teste de Wilcoxon. Para o comitê de LVQ, foram superiores ARTIE-C2, ARTIE-C3, MUSCLE-C2 e MUSCLE-C3. Já em relação ao comitê de ELM, foram expressivamente melhores os comitês ARTIE-C3, MUSCLE-C2 e MUSCLE-C3. Todas as outras combinações são consideradas estatisticamente equivalentes pelo teste.

Tabela 7.6: Resultados do teste t-pareado para os 12 conjuntos de dados usados. Os classificadores destacados são aqueles que mais vezes foram escolhidos como estatisticamente superiores.

Comitê A	**Comitê B**	**A ≈ B**	**A > B**	**A < B**
FAM comitê	ARTIE-C1	8	2	2
FAM comitê	**ARTIE-C2**	6	-	6
FAM comitê	**ARTIE-C3**	5	-	7
FAM comitê	MUSCLE-C1	3	5	4
FAM comitê	**MUSCLE-C2**	7	1	4
FAM comitê	MUSCLE-C3	6	3	3
LVQ comitê	**ARTIE-C1**	6	2	4
LVQ comitê	**ARTIE-C2**	6	1	5
LVQ comitê	**ARTIE-C3**	5	1	6
LVQ comitê	**MUSCLE-C1**	8	1	3
LVQ comitê	**MUSCLE-C2**	6	-	6
LVQ comitê	**MUSCLE-C3**	6	-	6
ELM comitê	**ARTIE-C1**	6	1	5
ELM comitê	**ARTIE-C2**	3	2	7
ELM comitê	**ARTIE-C3**	5	1	6
ELM comitê	**MUSCLE-C1**	7	1	4
ELM comitê	**MUSCLE-C2**	7	-	5
ELM comitê	**MUSCLE-C3**	7	-	5

Tabela 7.7: Resultados do teste de Wilcoxon para os 12 conjuntos de dados usados. Os classificadores destacados são aqueles considerados estatisticamente superiores.

Comitê A	**Comitê B**	**Valor crítico** $S \leq 14$
FAM comitê	ARTIE-C1	31
FAM comitê	**ARTIE-C2**	11
FAM comitê	**ARTIE-C3**	4
FAM comitê	MUSCLE-C1	38
FAM comitê	MUSCLE-C2	30
FAM comitê	MUSCLE-C3	30
LVQ comitê	ARTIE-C1	20
LVQ comitê	**ARTIE-C2**	10
LVQ comitê	**ARTIE-C3**	10
LVQ comitê	MUSCLE-C1	19
LVQ comitê	**MUSCLE-C2**	9
LVQ comitê	**MUSCLE-C3**	8
ELM comitê	ARTIE-C1	24
ELM comitê	ARTIE-C2	19
ELM comitê	**ARTIE-C3**	8
ELM comitê	MUSCLE-C1	19
ELM comitê	**MUSCLE-C2**	5
ELM comitê	**MUSCLE-C3**	4

7.4 Conclusões

Neste capítulo foram apresentados os resultados obtidos nas simulações computacionais realizadas com 9 comitês de classificadores em 12 bancos de dados reais.

Inicialmente, ilustrou-se o processo de otimização dos classificadores base a partir da evolução da função objetivo ao longo das iterações do algoritmo I-HPSO e de exemplos de seleção de atributos.

Os resultados de classificação foram reunidos e discutidos, verificando quais os melhores classificadores para cada um dos bancos avaliados. Os testes realizados ao longo da etapa de validação cruzada de 10 partições foram detalhados em gráficos de caixa, que permitiram ainda comprovar a eficiência dos comitês em relação ao uso de classificadores individuais.

Aplicou-se os testes estatísticos t-pareado e de Wilcoxon a fim de fazer uma comparação mais específica entre os modelos propostos e os comitês de referência. Analisando os resultados dos dois testes estatísticos escolhidos, é possível inferir que, dentre os 9 comitês avaliados. Os comitês ARTIE-C2, ARTIE-C3 e MUSCLE-C2 são aqueles mais capazes de obter resultados superiores em geral, sendo a variante ARTIE-C3 a que apresentou melhor desempenho de acordo com os testes usados. É importante ressaltar no entanto que todas as variantes propostas se sobressaíram em pelo menos um dos bancos de dados apresentados nas Tabelas 7.4 e 7.5.

Capítulo 8

Conclusões e Perspectivas

Esta dissertação estudou o uso de comitês de redes neurais competitivas em problemas de classificação de padrões. Para tanto, foram apresentadas técnicas que permitem algoritmos auto-organizáveis realizarem aprendizado supervisionado. Essas técnicas têm sido usadas na literatura em redes SOM, mas neste trabalho, pela primeira vez, elas também foram aplicadas à rede Fuzzy ART.

Um resumo dos conceitos referentes ao paradigma de aprendizado em comitês foi realizado. Em seguida, foram descritas as operações das redes competitivas, enfatizando as diferenças entre as redes não-supervisionadas (Fuzzy ART e SOM) e as redes supervisionadas (Fuzzy ARTMAP e LVQ). A descrição da rede ELM, também avaliada neste trabalho, encontra-se no Apêndice A.

Foram detalhados os modelos de comitês propostos, ARTIE e MUSCLE, cada um possuindo três variantes referentes ao método escolhido para tornar, respectivamente, as redes Fuzzy ART e SOM supervisionadas.

A otimização dos parâmetros de uma rede neural antes da etapa de treinamento comumente é realizada por métodos de busca exaustiva. O mesmo pode ser dito da seleção dos atributos usados por um classificador. No entanto, nesta dissertação optou-se por modelar ambos os problemas como um só problema de otimização da taxa de acerto média obtida na validação cruzada do conjunto de treinamento. A solução deste problema seguiu uma abordagem metaheurística a partir da proposição do algoritmo híbrido I-HPSO.

Após a descrição da metodologia de avaliação escolhida, apresentou-se e discutiu-se os resultados das simulações computacionais desenvolvidas. O processo

de otimização de parâmetros e seleção dos atributos foi exemplificado e os resultados de classificação para 12 bancos de dados reais foram listados. Avaliou-se os testes realizados através de *boxplots* que comprovaram a eficácia da metodologia de múltiplos classificadores adotada. Finalmente, os 9 comitês projetados foram comparados a partir de dois testes estatísticos, o teste t-pareado (paramétrico) e o teste de Wilcoxon (não paramétrico). Os testes indicaram o desempenho superior dos modelos propostos em relação aos comitês de referência, sobretudo as variantes ARTIE-C2, ARTIE-C3 e MUSCLE-C2.

Confirma-se com este trabalho a viabilidade de construção de comitês de classificadores a partir de algoritmos tradicionalmente auto-organizáveis. Os experimentos realizados sugerem que tais métodos, uma vez agrupados em comitês, podem ter desempenho superior a técnicas tradicionalmente supervisionadas. Além disso, foi validada a metodologia proposta baseada em escolha de parâmetros e seleção de atributos a partir do algoritmo metaheurístico I-HPSO.

8.1 Perspectivas para trabalhos futuros

Os modelos ARTIE e MUSCLE propostos são flexíveis o suficiente para permitir que diversas modificações sejam feitas com a intenção de aumentar ainda mais a capacidade de generalização dos comitês obtidos. Algumas ideias possíveis nesse sentido são listadas a seguir.

- Investigar métodos alternativos que permitam redes auto-organizáveis realizar aprendizado supervisionado. Nesta dissertação comprovou-se experimentalmente que a escolha de tal método reflete diretamente na qualidade do classificador obtido. A técnica C1, por exemplo, se mostrou inferior na maioria dos casos quando comparada às técnicas C2 e C3.

- Formar comitês mistos, i.e., heterogêneos, compostos por variantes de uma mesma rede (e.g. Fuzzy ART-C2 e Fuzzy ART-C3) ou mesmo de redes diferentes (e.g. Fuzzy ART-C2 e SOM-C2). Os resultados obtidos via ARTIE e MUSCLE apresentaram várias diferenças de precisão em alguns bancos de dados. Uma arquitetura mista poderia se aproveitar dessa característica para construir comitês mais eficientes.

- Aprofundar o caráter evolucionário dos modelos propostos. Em Yao (1999) é feita uma revisão sobre redes neurais evolucionárias em que são citadas

diversas abordagens possíveis: (*i*) evolução das arquiteturas das redes; (*ii*) evolução das funções de transferência dos neurônios; (*iii*) evolução dos pesos sinápticos; (*iv*) evolução das regras de aprendizado; (*v*) evolução dos parâmetros dos algoritmos; (*vi*) evolução dos atributos; etc. Além da otimização paramétrica e a seleção de atributos usadas nesta dissertação, outras abordagens evolucionárias podem ser de interesse.

- Ainda em relação à característica evolucionária dos modelos apresentados, existe a possibilidade de delegar a um método de otimização estocástica (o método I-HPSO proposto, por exemplo) a responsabilidade de promover diversidade entre os classificadores base que compõem o comitê. Algumas propostas para esta abordagem podem ser conferidas em Yao e Islam (2008).

Apêndice A

Redes ELM

A rede ELM (*Extreme Learning Machine*), proposta por Huang, Zhu e Siew (2004), é uma rede neural do tipo *feedforward* (sem realimentação) com uma única camada oculta. O principal conceito por trás dessa rede é realizar um mapeamento não linear aleatório na camada oculta e calcular diretamente os valores dos pesos da camada de saída. Por esse motivo, o treinamento da rede ELM é não-iterativo

Seja uma rede com p entradas, q neurônios ocultos e m saídas, a i-ésima saída no instante n é dada pela seguinte expressão:

$$o_i(n) = \mathbf{m}_i^T \mathbf{z}(n), \tag{A.1}$$

em que $\mathbf{m}_i \in \mathbb{R}^q, \forall i \in \{1, \ldots, C\}$, é o vetor de pesos que conecta os neurônios ocultos ao i-ésimo neurônio da camada de saída, e $\mathbf{z}(n) \in \mathbb{R}^q$ é o vetor de saídas da camada oculta para um padrão de entrada $\mathbf{a}(n) \in \mathbb{R}^p$. O vetor $\mathbf{z}(n)$ é definido por

$$\mathbf{z}(n) = [f(\mathbf{w}_1^T \mathbf{a}(n) + b_1) \ \cdots \ f(\mathbf{w}_q^T \mathbf{a}(n) + b_q)]^T, \tag{A.2}$$

em que b_l é o limiar do l-ésimo neurônio oculto, $\mathbf{w}_l \in \mathbb{R}^p$ é o vetor de pesos associado a esse neurônio e $f(\cdot)$ é uma função de ativação sigmoidal. Na rede ELM, os vetores de pesos $\mathbf{w}_l$ são escolhidos aleatoriamente a partir de uma distribuição uniforme ou normal.

Seja $\mathbf{Z} = [\mathbf{z}(1) \ \mathbf{z}(2) \ \cdots \ \mathbf{z}(N)]$ uma matriz $q \times N$ cujas N colunas são os vetores de saída da camada oculta, dados por $\mathbf{z}(n) \in \mathbb{R}^q$, $n = 1, ..., N$, em que N é o número de amostras disponíveis para treinamento. De forma similar, seja

$\mathbf{D} = [\mathbf{d}(1)\ \mathbf{d}(2)\ \cdots\ \mathbf{d}(N)]$ uma matriz $C \times N$ cuja n-ésima coluna é o vetor desejado (alvo) $\mathbf{d}(n) \in \mathbb{R}^C$ associado ao padrão de entrada $\mathbf{a}(n)$, $n = 1, \ldots, N$. Finalmente, seja $\mathbf{M} = [\mathbf{m}_1\ \mathbf{m}_2\ \cdots\ \mathbf{m}_C]$ uma matriz $q \times C$ cuja i-ésima coluna é dada pelo vetor de pesos da camada de saída, $\mathbf{m}_i \in \mathbb{R}^q$, $i = 1, ..., C$.

As três matrizes definidas anteriormente estão relacionadas pelo mapeamento linear $\mathbf{D} = \mathbf{M}^T\mathbf{Z}$. Enquanto as matrizes $\mathbf{D}$ e $\mathbf{Z}$ são conhecidas, a matriz de pesos da camada de saída $\mathbf{M}$ não o é. Entretanto, a matriz $\mathbf{M}$ pode ser calculada pelo método da pseudoinversa, de acordo com a equação a seguir:

$$\mathbf{M} = \left(\mathbf{Z}\mathbf{Z}^T\right)^{-1}\mathbf{Z}\mathbf{D}^T. \tag{A.3}$$

Assumindo que o número de neurônios de saída é igual ao número de classes, o índice da classe inferida i^* para um padrão de entrada desconhecido, apresentado na fase de teste, é dado pela equação a seguir:

$$i^* = \arg\max_{i=1,\ldots,C}\{o_i\}, \tag{A.4}$$

em que o_i é calculado pela Equação (A.1).

É importante perceber que o treinamento da rede ELM, pela sua característica não-iterativa, é muito mais rápido que o algoritmo *backpropagation*, por exemplo (HUYNH; YONGGWAN; KIM, 2008). Comitês de classificadores baseados de redes ELM já foram usados na literatura, como em Liu, Xu e Wang (2009) e Lan, Soh e Huang (2009).

Apêndice B

Tabela de Valores para o Teste t-Pareado

Tabela B.1: Tabela resumida de valores críticos para teste t-pareado.

α (1 cauda)	0,05	0,025	α (1 cauda)	0,05	0,025
α (2 caudas)	0,10	0,050	α (2 caudas)	0,10	0,050
Graus de liberdade			Graus de liberdade		
1	6,3138	12,707	11	1,7959	2,2010
2	2,9200	4,3026	12	1,7823	2,1788
3	2,3534	3,1824	13	1,7709	2,1604
4	2,1319	2,7764	14	1,7613	2,1448
5	2,0150	2,5706	15	1,7530	2,1314
6	1,9432	2,4469	16	1,7459	2,1199
7	1,8946	2,3646	17	1,7396	2,1098
8	1,8595	2,3060	18	1,7341	2,1009
9	1,8331	2,2621	19	1,7291	2,0930
10	1,8124	2,2282	20	1,7247	2,0860

Apêndice C

Tabela de Valores Críticos para o Teste de Wilcoxon

Tabela C.1: Tabela resumida de valores críticos para teste de Wilcoxon.

α (1 cauda)	0,01	0,025	α (1 cauda)	0,01	0,025
α (2 caudas)	0,02	0,050	α (2 caudas)	0,02	0,050
N			N		
6	0	-	16	30	24
7	2	0	17	35	28
8	4	2	18	40	33
9	6	3	19	46	38
10	8	5	20	52	43
11	11	7	21	59	49
12	14	10	22	66	56
13	17	13	23	73	62
14	21	16	24	81	69
15	25	20	25	89	77

Referências Bibliográficas

AGUAYO, L. *Redes Neurais Competitivas para Detecção de Novidades em Séries Temporais*. Tese (Doutorado) — Universidade Federal do Ceará, Brasil, 2008.

ANGHINOLFI, D.; PAOLUCCI, M. Simulated annealing as an intensification component in hybrid population-based metaheuristics. In: *Simulated Annealing*. Vienna, Austria: I-Tech Education and Publishing, 2008.

BARRETO, G. A.; AGUAYO, L. Time series clustering for anomaly detection using competitive neural networks. In: PRINCIPE, J. C.; MIIKKULAINEN, R. (Ed.). *Proceedings of the 7th International Workshop on Advances in Self-Organizing Maps (WSOM'09)*. St. Augustine, EUA: Springer, 2009. v. 5629, p. 28–36.

BARUQUE, B.; CORCHADO, E. A weighted voting summarization of SOM ensembles. *Data Mining and Knowledge Discovery*, Springer, p. 1–29, 2010.

BAUER, E.; KOHAVI, R. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, Springer, v. 36, n. 1, p. 105–139, 1999.

BELEGUNDU, A.; CHANDRUPATLA, T. *Optimization concepts and applications in engineering*. 2. ed. [S.l.]: Cambridge University Press, 2011.

BERMEJO, S.; CABESTANY, J. Local averaging of ensembles of LVQ-based nearest neighbor classifiers. *Applied Intelligence*, v. 20, n. 1, p. 47–58, 2004.

BIEBELMANN, E.; KÖPPEN, M.; NICKOLAY, B. Pratical aplications of neural networks in texture analysis. *Neurocomputing*, Elsevier, v. 13, n. 2–4, p. 261–279, 1996.

BOSLAUGH, S.; WATTERS, P. *Statistics in a Nutshell*. [S.l.]: O'Reilly Media, Inc., 2008.

BRATTON, D.; KENNEDY, J. Defining a standard for particle swarm optimization. In: *IEEE Swarm Intelligence Symposium*. Honolulu, Hawaii, USA: [s.n.], 2007. p. 120–127.

BREIMAN, L. Bagging predictors. *Machine Learning*, Springer, v. 24, n. 2, p. 123–140, 1996.

CARPENTER, G. Default ARTMAP. *CAS/CNS Technical Report Series*, n. 008, 2003.

CARPENTER, G.; GJAJA, M. Fuzzy ART choice functions. *CAS/CNS Technical Report Series*, n. 060, 1993.

CARPENTER, G.; GROSSBERG, S. ART 2: Stable self-organization of pattern recognition codes for analog input patterns. *Applied Optics*, Optical Society of America, v. 26, n. 23, p. 4919–4930, 1987.

CARPENTER, G.; GROSSBERG, S. A massively parallel architeture of a self-organizing neural pattern recogntion machine. *Computer Vision, Graphics and Image Processing*, Elsevier, v. 37, p. 54–115, 1987.

CARPENTER, G.; GROSSBERG, S. Stable self-organization of pattern recognition codes for analog input patterns. *Applied Optics*, Optical Society of America, v. 26, p. 4919–4930, 1987.

CARPENTER, G.; GROSSBERG, S.; MARKUZON, N.; REYNOLDS, J.; ROSEN, D. Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. *IEEE Transactions on Neural Networks*, IEEE, v. 3, n. 5, p. 698–713, 1992.

CARPENTER, G.; GROSSBERG, S.; REYNOLDS, J. H. ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network. *Neural Networks*, Elsevier, v. 4, n. 5, p. 565–588, 1991.

CARPENTER, G.; GROSSBERG, S.; ROSEN, D. ART 2-A: An adaptive resonance algorithm for rapid category learning and recognition. *Neural Networks*, Elsevier, v. 4, n. 4, p. 493–504, 1991.

CARPENTER, G. A.; GROSSBERG, S. The ART of adaptive pattern recognition by a self-organizing neural network. *IEEE Computer Magazine*, IEEE, v. 21, n. 3, p. 77–88, 1988.

CARPENTER, G. A.; GROSSBERG, S.; ROSEN, D. B. Fuzzy ART: Fast stable learning, categorization of analog patterns by an adaptive resonance system. *Neural Networks*, Elsevier, v. 4, n. 6, p. 759–771, 1991.

CHANG, Y.; LEE, D.; HONG, Y.; ARCHIBALD, J. Unsupervised video shot detection using clustering ensemble with a color global scale-invariant feature transform descriptor. *Journal on Image and Video Processing*, v. 2008, p. 9, 2008.

CHO, S. Self-organizing map with dynamical node splitting: Application to handwritten digit recognition*. *Neural Computation*, v. 9, n. 6, p. 1345–1355, 1997.

CHRISTODOULOU, C. I.; MICHAELIDES, S. C.; PATTICHIS, C. S. Multifeature texture analysis for the classification of clouds in satellite imagery. *IEEE Transactions on Geoscience and Remote Sensing*, IEEE, v. 41, n. 11, p. 2662–2668, 2003.

CLERC, M.; KENNEDY, J. The particle swarm - explosion, stability, and convergence in a multidimensional complex space. In: *IEEE Transactions on Evolutionary Computation*. Piscataway, EUA: [s.n.], 2002. v. 6, n. 1, p. 58–73.

CORCHADO, E.; BARUQUE, B.; YIN, H. Boosting unsupervised competitive learning ensembles. In: de Sá, J. M.; ALEXANDRE, L. A.; DUCH, W.; MANDIC, D. P. (Ed.). *Proceedings of the 17th International Conference on Artificial Neural Networks (ICANN'07), Part I*. Porto, Portugal: Springer, 2007. LNCS 4668, p. 339–348.

CRUZ, R.; CAVALCANTI, G.; REN, T.; RECIFE, B. Handwritten digit recognition using multiple feature extraction techniques and classifier ensemble. In: *17th International Conference on Systems, Signals and Image Processing*. Rio de Janeiro, Brasil: [s.n.], 2010.

DAS, R.; SENGUR, A. Evaluation of ensemble methods for diagnosing of valvular heart disease. *Expert Systems with Applications*, Elsevier, v. 37, n. 7, p. 5110–5115, 2010.

DEMŠAR, J. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, MIT Press, v. 7, p. 1–30, 2006.

DENG, W.; ZHENG, Q.; LIAN, S.; CHEN, L. Ordinal extreme learning machine. *Neurocomputing*, Elsevier, 2010.

DIETTERICH, T. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, MIT Press, v. 10, n. 7, p. 1895–1923, 1998.

DIETTERICH, T. G. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, Springer, v. 40, n. 2, p. 139–157, 2000.

DORIGO, M. *Optimization, learning and natural algorithms*. Tese (Doutorado) — Politecnico di Milano, Milão, Itália, 1992.

DUBOIS, D.; PRADE, H. A review of fuzzy set aggregation connectives. *Information Sciences*, v. 36, n. 1-2, p. 85–121, 1985.

EBERHART, R.; KENNEDY, J. A new optimizer using particle swarm theory. In: *Proceedings of the Sixth International Symposium on Micro Machine and Human Science, 1995. MHS '95.* Piscataway, NJ, USA: [s.n.], 1995. p. 39–43.

FLEXER, A. Statistical evaluation of neural network experiments: Minimum requirements and current practice. *Cybernetics and Systems Research*, p. 1005–1008, 1996.

FRANK, A.; ASUNCION, A. *UCI Machine Learning Repository*. 2010. Disponível em: <http://archive.ics.uci.edu/ml>.

FREIXA, M.; SALAFRANCA, L.; GUARDIA, J.; FERRER, R.; TURBANY, J. Análisis exploratorio de datos: nuevas técnicas estadísticas. *Promociones y Publicaciones Universitarias SA Barcelona*, 1992.

FREUND, Y.; SCHAPIRE, R. A desicion-theoretic generalization of on-line learning and an application to boosting. In: *Computational Learning Theory*. [S.l.]: Springer, 1995. p. 23–37.

FREUND, Y.; SCHAPIRE, R. Experiments with a new boosting algorithm. In: *Machine Learning - International Workshop Then Conference*. [S.l.: s.n.], 1996. p. 148–156.

GAMA, J.; BRAZDIL, P. Cascade generalization. *Machine Learning*, Springer, v. 41, n. 3, p. 315–343, 2000.

GENTLE, J.; HÄRDLE, W.; MORI, Y. *Handbook of computational statistics*. Berlin, Alemanha: Springer Berlin, 2004.

GEORGAKIS, A.; LI, H.; GORDAN, M. An ensemble of som networks for document organization and retrieval. In: *International Conference on Adaptive Knowledge Representation and Reasoning 2005 (AKRR 2005)*. Espoo, Finlândia: [s.n.], 2005. p. 6.

GLOVER, F. Future paths for integer programming and links to artificial intelligence. *Computers and Operations Research*, v. 13, n. 5, p. 533–549, 1986.

GORGÔNIO, F.; COSTA, J. Parallel self-organizing maps with application in clustering distributed data. In: *IEEE International Joint Conference on Neural Networks 2008 (IJCNN 2008)*. Hong Kong, China: [s.n.], 2008. p. 3276–3283.

GUNES, V.; M., M.; PETITRENAUD, S. Multiple classifier systems: Tools and methods. In: CHEN, C. (Ed.). *Handbook of Pattern Recognition and Computer Vision*. [S.l.]: World Scientific, 2010. cap. 1.2.

GUO, Y. An integrated PSO for parameter determination and feature selection of svr and its application in stlf. In: IEEE. *International Conference on Machine Learning and Cybernetics, 2009*. Baoding, China, 2009. v. 1, p. 359–364.

HANSEN, L. K.; SALAMON, P. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE, v. 12, n. 10, p. 993–1001, 2002.

HAYKIN, S. *Neural Networks and Learning Machines*. 3rd. ed. Canada: Prentice Hall, 2008.

HE, Q.; WANG, L. A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization. *Applied Mathematics and Computation*, Amsterdam, Netherlands, v. 186, n. 2, p. 1407–1422, 2007.

HOLDEN, N.; FREITAS, A. A. A hybrid pso/aco algorithm for discovering classification rules in data mining. *Journal of Artificial Evolution and Applications*, v. 2008, 2008. Disponível em: <http://www.hindawi.com/journals/jaea/2008/316145.html>.

HOLLAND, J. Adaptation in natural and artificial systems. *Ann Arbor MI: University of Michigan Press*, 1975.

HOYO, R. del; BULDAIN, D.; MARCO, A. Supervised classification with associative SOM. In: *Proceedings of the 7th International Work-Conference on Artificial and Neural Networks, (IWANN)'03*. [S.l.: s.n.], 2003. p. 334–341.

HUANG, C.; DUN, J. A distributed pso-svm hybrid system with feature selection and parameter optimization. *Applied Soft Computing*, Elsevier, v. 8, n. 4, p. 1381–1391, 2008.

HUANG, G. B.; ZHU, Q. Y.; SIEW, C. K. Extreme learning machine: A new learning scheme of feedforward neural networks. In: IEEE. *Proceedings of the 2004 IEEE International Joint Conference on Neural Networks (IJCNN'2004)*. Budapeste, Hungria, 2004. p. 985–990.

HUYNH, H.; YONGGWAN, W.; KIM, J. An improvement of extreme learning machine for compact single-hidden-layer feedforward neural networks. *International Journal of Neural Systems*, v. 18, n. 5, p. 433–441, 2008.

JIANG, Y.; ZHOU, Z. Som ensemble-based image segmentation. *Neural Processing Letters*, v. 20, n. 3, p. 171–178, 2004.

KANGAS, J. A.; KOHONEN, T. K.; LAAKSONEN, J. T. Variants of self-organizing maps. *IEEE Transactions on Neural Networks*, v. 1, n. 1, p. 93–99, 1990.

KASUBA, T. Simplified fuzzy ARTMAP. *AI EXPERT*, v. 8, p. 18–25, 1993.

KENNEDY, J.; EBERHART, R. A discrete binary version of the particle swarm algorithm. In: IEEE. *IEEE International Conference on Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation, 1997*. Orlando, EUA, 1997. v. 5, p. 4104–4108.

KENNEDY, J.; EBERHART, R. C. Particle swarm optimization. *Proceedings of the IEEE International Conference on Neural Networks*, Piscataway, NJ, USA, v. 4, p. 1942–1948, 1995.

KESKIN, G. A.; ÖZKAN, C. An alternative evaluation of FMEA: Fuzzy art algorithm. *Quality and Reliability Engineering International*, v. 25, n. 6, p. 647–661, 2009.

KHANESAR, M.; TESHNEHLAB, M.; SHOOREHDELI, M. A novel binary particle swarm optimization. In: IEEE. *Mediterranean Conference on Control & Automation, 2007. MED'07.* Atenas, Grécia, 2007. p. 1–6.

KIM, D. H.; ABRAHAM, A.; HIROTA, K. Hybrid genetic: Particle swarm optimization algorithm. In: *Studies in Computational Intelligence*. Berlin, Germany: Springer Berlin / Heidelberg, 2007. p. 147–170.

KIRKPATRICK, S.; GELATT, C. D.; JR.; VECCHI, M. P. Optimization by simulated annealing. *Science*, USA, v. 220, p. 671–680, 1983.

KOHONEN, T. Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, Springer, v. 43, n. 1, p. 59–69, 1982.

KOHONEN, T. An introduction to neural computing. *Neural Networks*, v. 1, n. 1, p. 3–16, 1988.

KOHONEN, T. The 'neural' phonetic typewriter. *Computer*, v. 21, n. 3, p. 11–22, 1988.

KOHONEN, T. *Self-Organizing Maps*. 2nd extended. ed. Berlin, Alemanha: Springer-Verlag, 1997.

KOTSIANTIS, S.; ZAHARAKIS, I.; PINTELAS, P. Machine learning: a review of classification and combining techniques. *Artificial Intelligence Review*, Springer, v. 26, n. 3, p. 159–190, 2006.

KROGH, A.; VEDELSBY, J. Neural network ensembles, cross validation, active learning. *Advances in Neural Information Processing Systems*, MIT Press, p. 231–238, 1995.

KUMAR, P. A. R.; SELVAKUMAR, S. Distributed denial of service attack detection using an ensemble of neural classifier. *Computer Communications*, Elsevier, 2011.

KUNCHEVA, L.; JAIN, L. Designing classifier fusion systems by genetic algorithms. *Evolutionary Computation, IEEE Transactions on*, IEEE, v. 4, n. 4, p. 327–336, 2000.

KUSIAK, A.; LI, M.; ZHANG, Z. A data-driven approach for steam load prediction in buildings. *Applied Energy*, Elsevier, v. 87, n. 3, p. 925–933, 2010.

LAHA, A.; PAL, N. R. Some novel classifiers designed using prototypes extracted by a new scheme based on self-organizing feature map. *IEEE Transactions on Systems, Man, and Cybernetics*, IEEE, B-31, n. 6, p. 881–890, 2001.

LAN, Y.; SOH, Y.; HUANG, G. Ensemble of online sequential extreme learning machine. *Neurocomputing*, Elsevier, v. 72, n. 13-15, p. 3391–3395, 2009.

LIN, C.; CHANG, C.; HSU, C. A practical guide to support vector classification. *National Taiwan University*, 2004.

LIU, B.; WANG, L.; JIN, Y.-H.; TANG, F.; HUANG, D.-X. Improved particle swarm optimization combined with chaos. *Chaos, Solitons & Fractals*, Amsterdam, Netherlands, v. 25, n. 5, p. 1261–1271, 2005.

LIU, H.; ABRAHAM, A. An hybrid fuzzy variable neighborhood particle swarm optimization algorithm for solving quadratic assignment problems. *Journal of Universal Computer Science*, v. 13, n. 9, p. 1309–1331, 2007. Disponível em: <http://www.jucs.org/jucs_13_9/an_hybrid_fuzzy_variable>.

LIU, Y.; XU, X.; WANG, C. Simple ensemble of extreme learning machine. In: IEEE. *Image and Signal Processing, 2009. CISP'09. 2nd International Congress on*. Tianjin, China, 2009. p. 1–5.

LOO, C. K.; LAW, A.; LIM, W. S.; RAO, M. V. C. Probabilistic ensemble simplified fuzzy ARTMAP for sonar target differentiation. *Neural Computing & Applications*, Springer, v. 15, n. 1, p. 79–90, 2006.

LØVBJERG, M. *Improving Particle Swarm Optimization by Hybridization of Stochastic Search Heuristics and Self-Organized Criticality*. Dissertação (Mestrado) — Aarhus Universitet, Datalogisk Institut, Denmark, 2002.

MACLIN, R.; SHAVLIK, J. Combining the predictions of multiple classifiers: Using competitive learning to initialize neural networks. In: *International Joint Conference on Artificial Intelligence*. Montréal, Canadá: Morgan Kaufmann, 1995. v. 14, p. 524–531.

MADEO, R.; PERES, S. M.; BíSCARO, H. H.; DIAS, D. B.; BOSCARIOLI, C. A committee machine implementing the pattern recognition module for fingerspelling applications. In: *Proceedings of the 2010 ACM Symposium on Applied Computing (SAC'2010)*. [S.l.: s.n.], 2010. p. 954–958.

MONTEIRO, I.; QUEIROZ, S.; CARNEIRO, A.; SOUZA, L.; BARRETO, G. Face recognition independent of facial expression through som-based classifiers. In: IEEE. *Telecommunications Symposium, 2006 International*. Fortaleza, Brasil, 2006. p. 263–268.

MONTEIRO, I. Q. *Métodos de Aprendizado de Máquina Para Reconhecimento de Faces: Uma Comparação de Desempenho*. Dissertação (Mestrado) — Universidade Federal do Ceará, Brasil, 2009.

NIEMINEN, P.; KäRKKäINEN, T.; LUOSTARINEN, K.; MUHONEN, J. Neural prediction of product quality based on pilot paper machine process measurements. *Adaptive and Natural Computing Algorithms*, Springer, p. 240–249, 2011.

PALANIAPPAN, R.; ESWARAN, C. Using genetic algorithm to select the presentation order of training patterns that improves simplified fuzzy ARTMAP classification performance. *Applied Soft Computing*, v. 9, n. 1, p. 100–106, 2009.

PETRIKIEVA, L.; FYFE, C. Bagging and bumping self-organising maps. *Computing and Information Systems*, v. 9, n. 2, p. 69, 2002.

PILLAY, R. *Instantaneous intrusion detection system*. Tese (Doutorado) — OKLAHOMA STATE UNIVERSITY, 2011.

RAAFAT, H. M.; TOLBA, A. S.; ALY, A. M. A novel training weighted ensemble (TWE) with application to face recognition. *Applied Soft Computing*, p. 3608–3617, 2011.

RAJASEKARAN, S.; PAI, G. Simplified fuzzy ARTMAP as pattern recognizer. *Journal of computing in civil engineering*, v. 14, p. 92, 2000.

RITTER, H.; SCHULTEN, K. Convergence properties of Kohonen's topology conserving maps: Fluctuations, stability, and dimension selection. *Biological Cybernetics*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, v. 60, n. 1, p. 59–71, 1988.

ROCHA NETO, A. R.; BARRETO, G. A. On the application of ensembles of classifiers to the diagnosis of pathologies of the vertebral column: A comparative analysis. *IEEE Latin America Transactions*, IEEE, v. 7, n. 4, p. 487–496, 2009.

RUSSEL, S. J.; NORVIG, P. *Artificial Intelligence: A Modern Approach*. Englewood Cliffs, EUA: Prentice-Hall, 1996.

SALZBERG, S. On comparing classifiers: Pitfalls to avoid and a recommended approach. *Data Mining and Knowledge Discovery*, Springer, v. 1, n. 3, p. 317–328, 1997.

SANTOS, A.; CANUTO, A. Investigating the influence of RePART in ensemble systems designed by boosting. In: IEEE. *Neural Networks, 2008. IJCNN 2008.(IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on*. Hong Kong, China, 2008. p. 2907–2914.

SANTOS, A. M.; CANUTO, A. M. P. Using ARTMAP-based ensemble systems designed by three variants of boosting. In: *Proceedings of the International Conference on Artificial Neural Networks (ICANN'08)*. Praga, República Tcheca: Springer, 2008. p. 562–571.

SCHERBART, A.; NATTKEMPER, T. Looking inside self-organizing map ensembles with resampling and negative correlation learning. *Neural Networks*, Elsevier, 2010.

SCHWENK, H.; BENGIO, Y. Boosting neural networks. *Neural Computation*, MIT Press, v. 12, n. 8, p. 1869–1887, 2000.

SILVA, I.; SPATTI, D.; FLAUZINO, R. *Redes Neurais Artificiais Para Engenharia e Ciências Aplicadas*. Primeira edição. São Paulo: Artliber, 2010.

SOUZA JÚNIOR, A. H.; BARRETO, G. A.; VARELA, A. T. A speech recognition system for embedded applications using the SOM and TS-SOM networks. In: MWASIAGI, J. I. (Ed.). *Self-Organizing Maps - Applications and Novel Algorithm Design*. [S.l.]: InTech, 2011.

SPALL, J. C. *Introduction to Stochastic Search and Optimization*. New York, USA: Wiley, 2003.

STONE, M. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society. Series B (Methodological)*, JSTOR, v. 36, n. 2, p. 111–147, 1974.

SUGANTHAN, P. N. Hierarchical overlapped SOM's for pattern classification. *IEEE Transactions on Neural Networks*, IEEE, v. 10, n. 1, p. 193–196, 1999.

TALBI, E.-G. *Metaheuristics : from design to implementation*. EUA: John Wiley & Sons, 2009.

TIAN, J.; GU, H.; LIU, W. Imbalanced classification using support vector machine ensemble. *Neural Computing & Applications*, v. 20, n. 2, p. 203–209, 2011.

TODOROVSKI, L.; DŽEROSKI, S. Combining classifiers with meta decision trees. *Machine Learning*, Springer, v. 50, n. 3, p. 223–249, 2003.

TRAN, M.; LIM, C.; ABEYNAYAKE, C.; JAIN, L. Feature extraction and classification of metal detector signals using the wavelet transform and the fuzzy ARTMAP neural network. *Journal of Intelligent and Fuzzy Systems*, IOS Press, v. 21, n. 1, p. 89–99, 2010.

TSYMBAL, A.; PECHENIZKIY, M.; CUNNINGHAM, P. Diversity in search strategies for ensemble feature selection. *Information Fusion*, Elsevier, v. 6, n. 1, p. 83–98, 2005.

TURKY, A. M.; AHMAD, M. S. The use of SOM for fingerprint classification. In: *IEEE International Conference on Information Retrieval & Knowledge Management (CAMP'2010)*. Shah Alam, Malásia: [s.n.]. p. 287–290.

WANG, G.; LI, P. Evolutionary extreme learning machine based on dynamic adaboost ensemble. In: *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*. Xi'an, China: [s.n.], 2010. v. 7820, p. 65.

WINDEATT, T. Ensemble mlp classifier design. *Computational Intelligence Paradigms*, Springer, p. 133–147, 2008.

WYNS, B.; SETTE, S.; BOULLART, L.; BAETEN, D.; HOFFMAN, I. E. A.; De Keyser, F. Prediction of diagnosis in patients with early arthritis using a combined Kohonen mapping and instance-based evaluation criterion. *Artificial Intelligence in Medicine*, Elsevier, v. 31, n. 1, p. 45–55, 2004.

XIAO, Y.-D.; CLAUSET, A.; HARRIS, R.; BAYRAM, E.; SANTAGO, P.; SCHMITT, J. D. Supervised self-organizing maps in drug discovery. 1. robust behavior with overdetermined data sets. *Journal of Chemical Information and Modeling*, v. 45, n. 6, p. 1749–1758, 2005.

YAO, Q.; CAI, J.; ZHANG, J. Simultaneous feature selection and ls-svm parameters optimization algorithm based on pso. In: *Computer Science and Information Engineering, 2009 WRI World Congress on*. Los Angeles, EUA: [s.n.], 2009. v. 5, p. 723–727.

YAO, X. Evolving artificial neural networks. *Proceedings of the IEEE*, IEEE, v. 87, n. 9, p. 1423–1447, 1999.

YAO, X.; ISLAM, M. Evolving artificial neural network ensembles. *Computational Intelligence Magazine, IEEE*, IEEE, v. 3, n. 1, p. 31–42, 2008.

ZADEH, L. Fuzzy sets. *Information and Control*, v. 2, p. 338–353, 1965.

ZHOU, L.; LAI, K.; YU, L. Least squares support vector machines ensemble models for credit scoring. *Expert Systems with Applications*, v. 37, n. 1, p. 127–133, 2010.

ZHOU, Z.; WU, J.; TANG, W. Ensembling neural networks: Many could be better than all. *Artificial intelligence*, Elsevier, v. 137, n. 1-2, p. 239–263, 2002.