

DISSERTAÇÃO DE MESTRADO Nº 725

**UMA NOVA ABORDAGEM BASEADA
EM MARGEM PARA SELEÇÃO DE
MODELOS NEURAIIS**

Luiz Carlos Bambirra Torres

DATA DA DEFESA: 24/02/2012

Universidade Federal de Minas Gerais

Escola de Engenharia

Programa de Pós-Graduação em Engenharia Elétrica

**UMA NOVA ABORDAGEM BASEADA EM MARGEM PARA
SELEÇÃO DE MODELOS NEURAIIS**

Luiz Carlos Bambirra Torres

Dissertação de Mestrado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do Título de Mestre em Engenharia Elétrica.

Orientador: Prof. Antônio de Pádua Braga

Belo Horizonte - MG

Fevereiro de 2012


"Uma Nova Abordagem Baseada em Margem para Seleção de Modelos Neurais"

Luiz Carlos Bambirra Torres

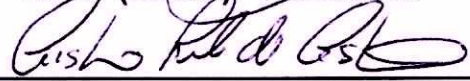
Dissertação de Mestrado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do grau de Mestre em Engenharia Elétrica.

Aprovada em 24 de fevereiro de 2012.

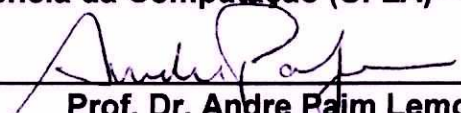
Por:



Prof. Dr. Antônio de Pádua Braga
DELT (UFMG) - Orientador



Prof. Dr. Cristiano Leite de Castro
Dep. Ciência da Computação (UFLA) - Co-Orientador



Prof. Dr. Andre Paim Lemos
DELT (UFMG)



Prof. Dr. Rogério Martins Gomes
DECOM (CEFET-MG)

*Não há limites, exceto aqueles que
impomos a nós mesmos.*

Walter Bishop

Agradecimentos

À Deus por caminhar sempre ao meu lado em todos os momentos.

Aos meus pais, Rosana e Ricardo, que estão sempre presentes me dando todo apoio sempre que eu preciso, e a minha avó Glória por estar sempre presente.

Em memória dos meus avós e da saudosa Kikinha, que estão olhando por mim do andar de cima.

Ao Prof. Antônio de Pádua Braga, por ter me recebido no LITC, pelos ensinamentos, orientação, discussões, e principalmente pela paciência que teve comigo durante o desenvolvimento desse trabalho, meus profundos agradecimentos.

Ao Prof. Cristiano Leite de Castro, por ter me apresentado o LITC, pela amizade, orientação e dedicação na qual me ajudou a conduzir esse trabalho, muito obrigado.

Aos colegas do LITC, pelo apoio e amizade.

A CEMIG pelo apoio financeiro.

Resumo

Este trabalho apresenta uma nova estratégia de decisão para o aprendizado multiobjetivo de redes neurais artificiais. O objetivo é encontrar no conjunto pareto-ótimo, a solução que fornece a melhor capacidade de generalização. A abordagem proposta para a tomada de decisão é baseada em uma estimativa geométrica para a margem (distância) máxima de separação entre as classes, que é obtida através das seguintes etapas: modelagem dos padrões de entrada com o grafo de gabriel, detecção das bordas de separação das classes e síntese de padrões junto à região de margem máxima. Essa metodologia permite que modelos suaves (que ignoram ruído) e bem ajustados sejam selecionados de forma transparente para o usuário, ou seja, sem a necessidade da definição de parâmetros ou do uso de um conjunto representativo de validação. Resultados com *benchmarks* conhecidos na literatura mostraram que o decisor proposto, aliado ao treinamento multiobjetivo, foi eficiente no controle da generalização de modelos neurais.

PALAVRAS-CHAVE: tomada de decisão, aprendizado de máquina multiobjetivo, grafo de gabriel, classificação.

Abstract

This work presents a new decision-making strategy to the multiobjective learning of artificial neural networks. The objective is to find the solution within the pareto-optimal set that has the best generalization performance. The proposed decision-making approach is based on a geometric approximation to the maximum margin (distance) of class separation, which is estimated through the following steps: modeling of input patterns using the gabriel graph, detection of class separation borders and synthesis of patterns along the maximum margin region. This methodology allows the selection of smooth (that ignore noise) and well-fitting models in a straightforward manner, i.e., without the need of the tuning of parameters by the user or the use of a representative validation data set. Results on benchmarks in literature showed that our decision-making method, combined with multiobjective training, was efficient to control the generalization of neural models.

KEY-WORDS: decision-making, multiobjective machine learning, gabriel graph, classification.

Lista de Figuras

1.1	(a) Pareto-ótimo. (b) Metodologia para construção do Decisor.	p. 17
2.1	Margem de separação entre as classes.	p. 20
2.2	Elementos dentro da margem.	p. 21
2.3	Pareto-ótimo	p. 26
3.1	Forma geométrica de um Grafo.	p. 30
3.2	(a) Diagrama de <i>Voronoi</i> . (b) Grafo Dual do Diagrama de <i>Voronoi</i> . (c) Triangulação de <i>Delaunay</i> resultante. (d) Par de pontos de <i>Voronoi</i> que possuem uma aresta em comum representado em uma Triangulação de <i>Delaunay</i>	p. 31
3.3	Construção do grafo de Gabriel.	p. 32
3.4	(a) Conjunto de padrões de entrada modelado com o Grafo de Gabriel. (b) Conjunto resultante após a eliminação dos dados ruidosos. (c) Detecção da borda das classes.	p. 34
3.5	(a) Pontos médios relativo às bordas das classes. (b) Superfícies de decisão geradas a partir de soluções de PO.	p. 35
3.6	(a) Grid. (b) Grid de uma superfície linear. (c) Grid de uma superfície não linear.	p. 36
3.7	(a) Superfície de decisão mais próxima dos pontos médios. (b) Conjunto Pareto-Ótimo.	p. 36
3.8	Função de transferência tangente hiperbólica e seu ponto de inflexão igual a zero.	p. 38
3.9	(a) Duas soluções (superfícies) próximas aos pontos médios (triângulos). (b) Distância dos pontos médios da Figura 3.9(a) para o ponto de inflexão da função tangente hiperbólica mostrado na Figura 3.8. . .	p. 38
3.10	Conjunto Pareto-ótimo e os diferentes mapeamentos das soluções. .	p. 39

3.11	Soluções do conjunto <i>PO</i>	p. 40
3.12	(a) Soluções do PO que não classificam todos os vértices da margem. (b) Um dos motivos para o deslocamento das soluções.	p. 40
3.13	Soluções mais próximas dos pontos médios.	p. 41
4.1	Problema do tabuleiro de xadrez.	p. 43
4.2	Modelagem do problema com o Grafo de Gabriel. O ruído dos dados foi eliminado e as bordas entre as classes foi detectada.	p. 44
4.3	Solução (superfície de decisão) escolhida pelo decisor baseado em margem.	p. 45
4.4	(a) Problema das duas luas. (b) Modelagem do problema “duas luas” com o Grafo de Gabriel. (c) Solução encontrada pelo decisor baseado em margem. (d) decisor por validação (linha contínua) vs. decisor baseado em margem (linha pontilhada).	p. 46

Lista de Tabelas

4.1	Características das Bases de Dados	p.47
4.2	Valores de parâmetros para o kernel RBF	p.48
4.3	Resultados	p.48

Lista de Símbolos

x	Vetor de entrada
y	vetor de saída desejado
T	Conjunto de Dados
w	vetor de pesos
b	Bias
p	Margem da SVM
C	Regulador de Complexidade da SVM
ξ	Variável Slack
φ	Função de Kernel
γ	Parâmetro de regularização
R_{emp}	Risco Empírico
N	Numero de amostras
G	Grafo
G_p	Grafo Planar
G_G	Grafo de Gabriel
E	Conjunto de Arestas
V	Conjunto de Vértices
v	Vértice
$\delta(.)$	Distância euclidiana entre dois pontos
B_r	Borda das classes
P_M	Pontos Médios
θ	Função de transferência
$N_{Tr/Vc}$	Quantidade de dados utilizados para treinamento ou validação cruzada
N_{teste}	Numero de observações no conjunto de teste
N_{num}	Numero de atributos numéricos
N_{cat}	Numero de atributos categóricos
n	Numero total de atributos

Lista de Abreviaturas

MOBJ Algoritmo Multiobjetivo

RNAs Redes Neurais Artificiais

MLP Multi Layer Perceptron

SVM Support Vector Machine

RBF Radial-Basis Function Networks

VC Vapnik-Chervonenkis

QP Programação Quadrática

KKT Karush-Kuhn-Tucker

LS-SVM Least Squares Support Vector Machine Classifiers

TD Triangulação de Delaunay

Vor Diagrama de Voronoi

KNN K-Nearest Neighbors

acr Stalog Australian Credit

gcr Stalog German Credit

hea Stalog heart disease

ion Johns Hopkins university ionosphere

pid Pima Indians diabetes

snr The sonar

wbc Wisconsin breast cancer

Sumário

1	Introdução	p. 15
1.1	Organização do Trabalho	p. 17
2	Fundamentação Teórica	p. 18
2.1	Revisão de algoritmos para controle da capacidade de generalização	p. 18
2.1.1	Máquinas de Vetores de Suporte	p. 18
2.1.2	Hiperplano de margem rígida	p. 19
2.1.3	Hiperplano de margem flexível	p. 20
2.1.4	Least Squares Support Vector Machine Classifiers	p. 21
2.1.5	Weight Decay	p. 23
2.2	Fundamentos do Aprendizado Multiobjetivo	p. 24
2.2.1	Minimização Estrutural do Risco	p. 24
2.2.2	Algoritmo Multiobjetivo	p. 25
2.2.3	Problema de Decisão Multiobjetivo	p. 27
2.3	Conclusões do Capítulo	p. 27
3	Abordagem Proposta	p. 29
3.1	Teoria dos Grafos	p. 29
3.2	Diagrama de <i>Voronoi</i>	p. 30
3.3	Triangulação de <i>Delaunay</i>	p. 30
3.4	Grafo de Gabriel	p. 31
3.5	Algoritmo de Decisão Baseado em Margem Para Problemas de 2-Dimensões	p. 32

3.6	Algoritmo de Decisão Baseado em Margem Para Problemas de n-dimensões	p. 37
3.6.1	Metodologia	p. 37
3.7	Conclusões do Capítulo	p. 41
4	Resultados	p. 42
4.1	Resultados Para Problemas de 2-Dimensões	p. 42
4.1.1	Tabuleiro de Xadrez	p. 42
4.1.2	Duas Luas	p. 45
4.2	Resultados Para Problemas de N-Dimensões	p. 47
4.2.1	Algoritmos utilizados nos resultados	p. 47
4.2.2	Análise do Resultados	p. 48
4.2.3	Conclusões do Capítulo	p. 49
5	Conclusões e Trabalhos Futuros	p. 50
	Referências Bibliográficas	p. 51

1 *Introdução*

O problema da generalização em Redes Neurais Artificiais (RNAs) foi analisado formalmente por Vladimir Vapnik [Vapnik 1995, Vapnik 1998], que demonstrou, com o princípio indutivo de minimização estrutural do risco (MER), que para se obter uma solução eficiente para o problema do aprendizado é necessário minimizar dois objetivos conflitantes: o erro de treinamento e a capacidade (ou complexidade) da classe de funções fornecida pela máquina de aprendizagem. Tais objetivos não devem ser apenas minimizados, mas também equilibrados; caso contrário, o modelo resultante não generalizará bem. Essa idéia é análoga ao conhecido dilema entre a polarização e a variância, descrito inicialmente em [Geman et al. 1992].

Grande parte das técnicas para controle de generalização, tais como as redes de regularização [Girosi et al. 1995], o método *weight decay* [Hinton 1989] e os algoritmos de poda (*pruning*) [Reed 1993], minimizam ambos o erro e a complexidade implicitamente através de um único funcional custo, determinando seu equilíbrio através do ajuste de um ou vários parâmetros (como, por exemplo, o parâmetro que controla a regularização).

Não obstante, a formulação multiobjetivo do aprendizado (MOBJ) fornece uma abordagem alternativa para a implementação do princípio MER através da minimização explícita (separada) do erro de treinamento e de uma medida que reflete a complexidade da rede [Jin e Sendhoff 2009, Teixeira et al. 2000, Costa et al. 2007, Kokshenev e Braga 2010]. Sabe-se porém, que na abordagem MOBJ não é possível minimizar esses objetivos simultaneamente, pois o ótimo de um funcional raramente corresponde ao ótimo do outro. Assim, não existe um único ótimo, mas um conjunto deles, que formam o conjunto Pareto-Ótimo (*PO*). A formulação MOBJ para o treinamento de RNAs resulta então em um conjunto de soluções (*PO*) que corresponde aos melhores compromissos entre os funcionais erro e complexidade, como é mostrado na Figura 1.1(a).

Uma vez obtido o conjunto PO , a escolha da solução final (através de um decisor) constitui a etapa mais crítica do algoritmo MOBJ. De acordo com o princípio MER, a solução escolhida deve fornecer um equilíbrio adequado entre os funcionais minimizados, para evitar (sub)sobreajuste do modelo aos dados de treinamento. Estratégias para seleção da solução final no aprendizado MOBJ têm sido propostas, tais como o decisor por mínimo erro de validação [Teixeira et al. 2000] e o decisor baseado em conhecimento prévio [Medeiros et al. 2009]. Cabe ressaltar, no entanto, que a eficiência desses decisores pode ficar limitada em situações em que o conjunto de dados é muito pequeno e informação a priori sobre o processo de amostragem dos dados não se encontra disponível. Infelizmente, essas características são frequentes em problemas reais de aprendizado.

Para superar essas dificuldades, esse trabalho apresenta uma nova estratégia de decisão direcionada a problemas de classificação de padrões. Usando ferramentas da Geometria Computacional [Berg et al. 2000] foi desenvolvido um método de um decisor que busca pela solução do conjunto PO que possui a maior margem (ou distância) de separação entre as classes. Na abordagem proposta o espaço de entrada é modelado através de um grafo planar chamado de grafo de Gabriel. A partir do grafo foi possível encontrar a margem de separação entre as classes além de eliminar elementos ruidos na distribuição (vide Figura 1.1(b)). Para encontrar a solução de maior separação entre as classes foi proposto a inserção de novas amostras na distribuição no espaço de entrada. As novas amostras foram inseridas nos pontos médios definidos a partir da distância euclidiana entre as amostras da margem. Esse conjunto de novas amostras foi chamado de conjunto de pontos médios (P_M), e a solução mais próxima desse conjunto é a escolhida pelo decisor. Foram propostos dois algoritmos para encontrar a solução mais próxima do conjunto P_M . O primeiro encontra a solução de menor distância euclidiana entre os pontos médios. No segundo algoritmo, busca-se encontrar no espaço de características as soluções que correspondem as superfícies mais próximas da margem.

Através do ponto de inflexão da função de transferência da camada escondida da rede neural foi possível encontrar dentre as soluções pré-selecionadas, a mais próxima dos pontos médios no espaço de entrada. A abordagem se mostrou eficiente ao selecionar modelos com elevadas capacidades de generalização.

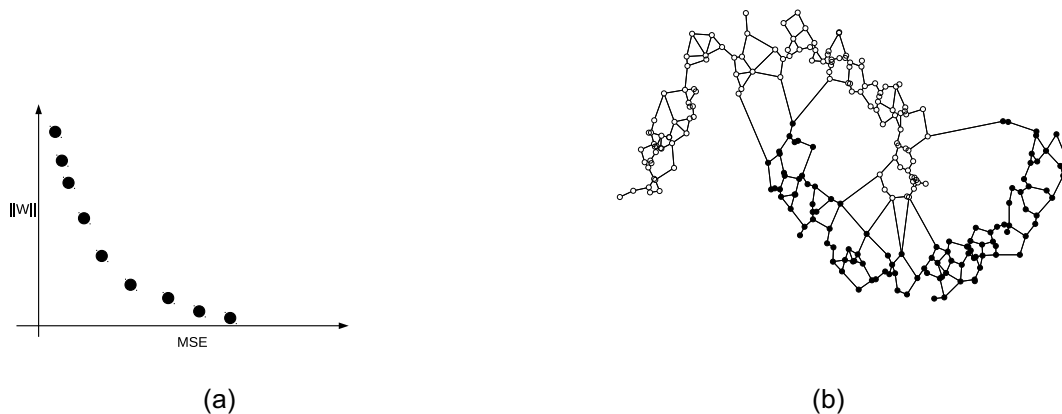


Figura 1.1: (a) Pareto-ótimo. (b) Metodologia para construção do Decisor.

1.1 Organização do Trabalho

O trabalho encontra-se organizado da seguinte forma. O capítulo 2 descreve os conceitos teóricos que fundamentam o problema da seleção de modelos no treinamento multiobjetivo de RNAs: Minimização Estrutural do Risco (MER), Aprendizado Multiobjetivo e o Problema de Decisão Multiobjetivo. É feita também uma revisão de alguns algoritmos conhecidos na literatura por encontrarem soluções de alta capacidade de generalização como as *Support Vector Machines*, *LS-Support Vector Machines*, *Cross-Validation* e o *Weight Decay*. Em seguida, no capítulo 3, a estratégia de decisão baseada em margem é apresentada. No capítulo 4, são descritos a metodologia adotada na condução dos experimentos e os resultados obtidos com a aplicação do novo decisor. Finalmente, o capítulo 5 traz as conclusões.

2 *Fundamentação Teórica*

No presente capítulo são definidas as bases sobre as quais a teoria do treinamento multiobjetivo de redes neurais é construída. Neste capítulo também são apresentados os principais métodos do aprendizado de máquina, como o Weight Decay, Ls-SVM e as SVM. Esses métodos se destacam por apresentarem soluções com alta capacidade de generalização.

2.1 Revisão de algoritmos para controle da capacidade de generalização

2.1.1 Máquinas de Vetores de Suporte

As Máquinas de Vetores de Suporte (SVM) são utilizadas para solucionar problemas de classificação de padrões e regressão. Foram propostas inicialmente por V. Vapnik [Vapnik 1995, Vapnik 1998].

A SVM se baseia no princípio de minimização estrutural do risco, que se origina na teoria do aprendizado estatístico. Essa teoria diz que o erro do algoritmo de aprendizagem junto aos dados de validação (erro de generalização), é limitado pelo erro de treinamento mais um termo que depende da dimensão Vapnik-Chervonenkis (VC), que é uma medida da capacidade de expressão de uma família de funções.

O objetivo é construir um conjunto de hiperplanos variando a dimensão VC, fazendo com que o risco empírico, também conhecido como erro de treinamento e a dimensão VC sejam minimizados ao mesmo tempo. Através de um kernel a SVM faz o mapeamento dos dados no espaço de entrada para um espaço de alta dimensão, chamado de espaço de características, em que um problema de natureza não-linear pode tornar linearmente separável. Nesse espaço, um hiperplano ótimo é construído para separar os dados em duas classes. Quando os dados de treinamento são se-

paráveis, o hiperplano ótimo no espaço de características apresenta uma máxima margem de separação. Se houver sobreposição, ou seja, dados não separáveis, é utilizado uma generalização do conceito. Segundo [Haykin 2009], a SVM é treinada por um algoritmo de otimização quadrático (QP) que garante a convergência para um mínimo global da superfície de erro. O algoritmo de otimização transforma o problema de otimização primal em sua representação dual permitindo que o problema de dimensionalidade não seja mais uma dificuldade. Logo, o número de parâmetros ajustados não dependerá mais da dimensão do espaço a que pertencem os dados de treinamento.

2.1.2 Hiperplano de margem rígida

Considere um conjunto de dados $T = \{\mathbf{x}_i, y_i \mid i = 1 \dots N\}$, onde x_i é o i -ésimo dado de entrada e y_i representa o i -ésimo elemento de saída desejável. Assume-se que todos os elementos $y = -1$ representam a classe 1 e o subconjunto $y = +1$ representa a classe 2. Parte-se da premissa que os dados são linearmente separáveis. De acordo com [Haykin 2009], a equação do hiperplano de separação é dada pela Equação (2.1),

$$\mathbf{w}^T \mathbf{x} + b = 0 \quad (2.1)$$

onde \mathbf{w} é o vetor normal, \mathbf{x} é o dado de entrada e b é o bias. O hiperplano descrito pela Equação (2.1) divide os espaço em dois sub-espacos, um para cada classe. A margem de separação p é definida como sendo a distância entre o hiperplano de separação determinado por \mathbf{w} e b e o padrão de entrada \mathbf{x} mais próximo. O objetivo de uma SVM é encontrar a separação ótima, ou seja, determinar o vetor \mathbf{w} de pesos da rede e o bias b de forma que p seja máximo, conforme descrito pela Figura 2.1. De acordo com [Haykin 2009], a classificação dos padrões é dada por sua posição em relação ao hiperplano, ou seja, em relação as margens de separação como é mostrado na Equação 2.2,

$$\begin{cases} \mathbf{w}^T \mathbf{x}_i + b \geq 0 & \text{para } d_i = +1 \\ \mathbf{w}^T \mathbf{x}_i + b < 0 & \text{para } d_i = -1 \end{cases} \quad (2.2)$$

Os pontos em que a primeira e a segunda expressão da Equação (2.2) são satisfeitos com uma igualdade são chamados de vetores de suporte. A margem de separação p entre as classe é dada pela Equação (2.3). A partir dessa equação é possível observar que maximizar a margem p , significa minimizar a norma do vetor \mathbf{w} ,

no espaço de características.

$$p = \frac{2}{\|w\|} \quad (2.3)$$

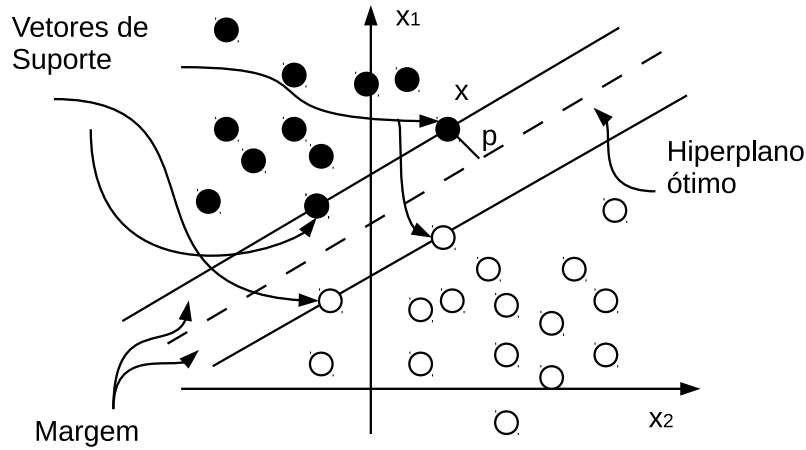


Figura 2.1: Margem de separação entre as classes.

2.1.3 Hiperplano de margem flexível

O conceito de Hiperplano de margem rígida impõe restrições que limitam bastante a sua aplicação, uma vez que a maioria dos problemas possuem ruídos ou sobreposição entre as classes. Para dar uma maior flexibilidade à SVM, foi desenvolvido o conceito de SVM de margem flexível. Nessa abordagem, é necessário introduzir, um conjunto de variáveis escalares não negativas $(\xi)_{i=1}^N$, onde N é o número de padrões de entrada.

$$d_i(w^T x_i + b) \geq 1 - \xi_i, \quad i = 1, 2, \dots, N \quad (2.4)$$

As variáveis ξ são conhecidas como variáveis *slack* e medem o grau de desvio de um padrão de entrada em relação ao hiperplano de separação ótimo. Para $0 < \xi \leq 1$ o padrão em questão está dentro da faixa da margem, mas do lado correto da separação, se $\xi > 1$ então ele está classificado incorretamente, e se $\xi = 0$ o padrão está sobre a margem. Logo, se estes padrões forem deixados de fora do treinamento, os vetores de suporte não mudarão (vide Figura 2.2). Isto mostra que os vetores de suporte são definidos da mesma maneira, seja para o caso linearmente separável ou não. Tendo

em vista as restrições impostas, o objetivo do treinamento é encontrar um hiperplano que tenha o menor erro de classificação dos dados de entrada, isso pode ser feito através da minimização do funcional ϕ mostrado na Equação 2.5

$$\phi(w, \xi) = \frac{1}{2} w^T w + C \sum_{i=1}^N \xi_i \quad (2.5)$$

sujeito a,

$$\begin{cases} y_i [w^T \varphi(x_i) + b] \geq 1 - \xi_i, \\ \xi_i \geq 0 \end{cases} \quad (2.6)$$

onde φ é a função de kernel e o parâmetro C controla o impasse (*tradeoff*) entre a complexidade da máquina e o número de pontos que podem infringir a restrição imposta na Equação 2.6.

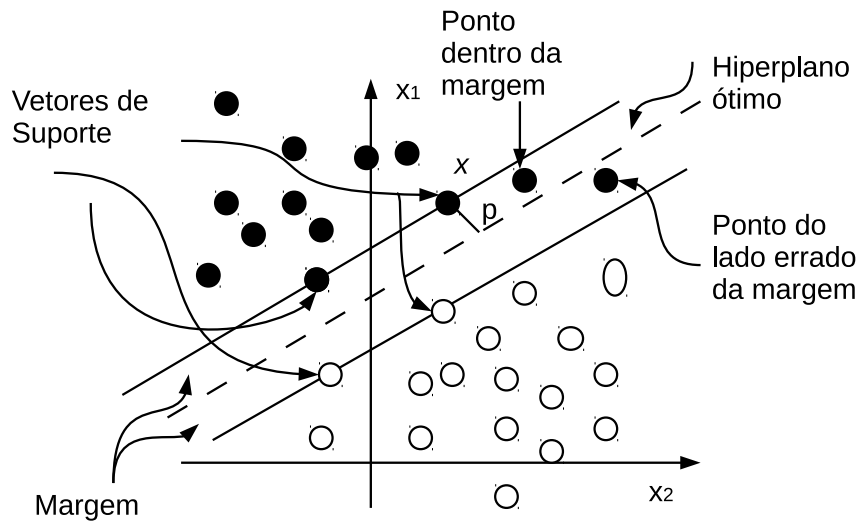


Figura 2.2: Elementos dentro da margem.

2.1.4 Least Squares Support Vector Machine Classifiers

Como mostrado na seção 2.1.1, a SVM resolve problemas de classificação através de um algoritmo de programação quadrática (QP). O algoritmo Least Squares Support Vector Machine Classifiers (LS-SVM) é uma modificação da SVM, que ao invés do QP, utiliza o método dos mínimos quadrados para resolver um sistema de equações

lineares. Outra característica da LS-SVM é que o método utiliza igualdades para tratar as restrições, ao contrário da SVM que usa desigualdades.

De acordo com o princípio de minimização estrutural do risco [Vapnik 1995] o funcional ϕ descrito na Equação (2.5), pode ser minimizado através do algoritmo *QP*. Segundo [Suykens e Vandewalle 1999] na formulação da LS-SVM há uma pequena diferença que pode ser observada nas Equações (2.7) e (2.8)

$$\zeta(w, e, b) = \frac{1}{2} w^T w + \frac{1}{2} v \sum_{i=1}^N e_i^2 \quad (2.7)$$

sujeito a restrição de igualdade,

$$y_i[w^T \phi(x_i) + b] = 1 - e_i, \quad i = 1, \dots, N. \quad (2.8)$$

onde N representa o numero de padrões de entrada, w a matriz de pesos, e e_i o erro do i -ésimo padrão de treinamento.

Ainda de acordo com [Suykens e Vandewalle 1999] a solução pode ser obtida a partir do método de Lagrange como mostra a Equação (2.9)

$$\ell(w, b, e, \alpha) = \zeta(w, e, b) - \sum_{i=1}^N \alpha_i \{y_i[w^T \phi(x_i) + b] - 1 + e_i\} \quad (2.9)$$

Onde α_i são os multiplicadores de Lagrange. A partir das condições de otimalidade, obtém-se o sistema de Karush-Kuhn-Tucker (KKT) mostrado na Equação (2.10)

$$\begin{cases} \frac{\partial \ell}{\partial w} = 0 \rightarrow w = \sum_{i=1}^N \alpha_i y_i \phi(x_i) \\ \frac{\partial \ell}{\partial b} = 0 \rightarrow \sum_{i=1}^N \alpha_i y_i = 0 \\ \frac{\partial \ell}{\partial e_i} = 0 \rightarrow \alpha_i = v e_i, \quad i = 1, \dots, N \\ \frac{\partial \ell}{\partial \alpha_i} = 0 \rightarrow y_i[w^T \phi(x_i) + b] - 1 + e_i = 0, \quad i = 1, \dots, N. \end{cases} \quad (2.10)$$

que segundo [Gestel et al. 2004] pode ser escrita como um conjunto de equações lineares descrito na Equação (2.11)

$$\left[\begin{array}{ccc|c} I & 0 & 0 & -Z^T \\ 0 & 0 & 0 & -Y^T \\ 0 & 0 & vI & -I \\ \hline Z & Y & I & 0 \end{array} \right] \begin{bmatrix} w \\ b \\ e \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vec{1} \end{bmatrix} \quad (2.11)$$

onde $Z = [\varphi(x_1)^T y_1; \dots; \varphi(x_N)^T y_N]$, $Y = [y_1; \dots; y_N]$, $[\vec{1} = 1; \dots; 1]$, $e = [e_1; \dots; e_N]$, $\alpha = [\alpha_1; \dots; \alpha_N]$ e I é matriz identidade. A solução simplificada é dada pela Equação (2.12)

$$\left[\begin{array}{c|c} 0 & y^T \\ \hline y & ZZ^T + v^{-1}I \end{array} \right] \left[\begin{array}{c} b \\ \alpha \end{array} \right] = \left[\begin{array}{c} 0 \\ \vec{1} \end{array} \right] \quad (2.12)$$

aplicando a condição de Mercer's na matriz $\Omega = ZZ^T$, onde

$$\Omega_{ij} = y_i y_j \alpha(x_i)^T \varphi(x_j) = y_i y_j K(x_i, x_j) \quad (2.13)$$

assim o classificador LS-SVM da Equação (2.14) pode ser resolvido a partir das Equações (2.12) e (2.13) ao invés do QP [Gestel et al. 2004].

$$y(x) = \text{sign} \left[\sum_{i=1}^N \alpha_i \varphi(x, x_i) + b \right] \quad (2.14)$$

2.1.5 Weight Decay

O algoritmo *Weight Decay* é um método de *prunning* que adiciona termos de penalidade a função objetivo, proporcionando um controle de complexidade do modelo, ou seja, o método modifica a função de custo penalizando as soluções com normas elevadas. De acordo com [Hinton 1989] a modificação consiste na introdução de um termo de regularização que penaliza os pesos com grandes magnitudes como mostra a Equação (2.15)

$$J(w) = e(w) + \frac{\lambda}{2} \|w\|^2 \quad (2.15)$$

onde λ é a importância relativa entre o termo de complexidade e o termo somatório dos erros quadráticos representado por $e(w)$, como é mostrado na Equação (2.16). De acordo com [Friedman et al. 2008], a escolha de grandes valores para λ tende a diminuir os valores dos pesos próximo a zero, tipicamente o método *cross-validation* é usado para estimar o valor de λ .

$$e(w) = \sum_{i=1}^N (d_i - y_i)^2 \quad (2.16)$$

A Equação (2.17) mostra a nova regra para o ajuste de pesos.

$$\Delta w_k = -\eta \frac{\partial e}{\partial w_k} - \rho \lambda w_k \quad (2.17)$$

onde η é a taxa de aprendizado e ρ é um parâmetro de regularização, normalmente $0 \leq \rho < 1$.

Segundo [Friedman et al. 2008], existe outra maneira de se adicionar penalidade mas de modo que a magnitude dos pesos seja regulada. Esse método é conhecido como *weights eliminations penalty*. A eliminação dos pesos se da pela minimização da Equação (2.18)

$$J(w) = e(w) + \frac{\lambda}{2} \sum_k \frac{w_k^2}{w_0^2} \left(1 + \frac{w_k^2}{w_0^2}\right)^{-1} \quad (2.18)$$

onde $w_0 > 0$, é um parâmetro que pode ser determinado utilizando o método *cross-validation*.

2.2 Fundamentos do Aprendizado Multiobjetivo

2.2.1 Minimização Estrutural do Risco

A teoria do aprendizado estatístico, através do princípio indutivo de minimização estrutural do risco (MER) estabelece condições matemáticas que permitem definir, com probabilidade de pelo menos $1 - \varepsilon$ sobre N (tamanho do conjunto de treinamento), um limite superior para o risco esperado (ou erro de generalização) de uma máquina de aprendizagem [Vapnik 1995, Vapnik 1998],

$$R \leq R_{emp} + \sqrt{\frac{h \left(\ln \frac{2N}{h} + 1 \right) - \ln \left(\frac{\varepsilon}{4} \right)}{N}} \quad (2.19)$$

onde ε é um parâmetro livre.

Analisando a Equação (2.19), observa-se que o limite superior de R é uma função inversa de N e direta de dois termos: o primeiro, denominado risco empírico (R_{emp}), representa o erro de treinamento e o segundo, conhecido como capacidade (Ω), depende da complexidade h da classe de funções implementada pela máquina

de aprendizagem. A minimização do limite superior de R pode então ser obtida através do aumento do número de exemplos N e/ou do decréscimo simultâneo de: R_{emp} e Ω .

2.2.2 Algoritmo Multiobjetivo

O princípio de minimização estrutural do risco (MER) pode ser interpretado como um problema de otimização multiobjetivo que busca encontrar o melhor compromisso entre dois objetivos conflitantes,

$$(MER) : \min \begin{cases} J_1 = R_{emp} \\ J_2 = \Omega \end{cases} \quad (2.20)$$

onde R_{emp} corresponde a uma estimativa para o erro de treinamento e Ω é uma medida de complexidade da máquina de aprendizagem. No caso particular de redes *MultiLayer Perceptron* (MLP) [Haykin 2009], uma medida de complexidade (Ω) comumente usada é a norma euclidiana do vetor de pesos da rede [Hinton 1989, Teixeira et al. 2000, Costa et al. 2007].

Dentre os algoritmos de treinamento multiobjetivo para redes MLP, destaca-se o método MOBJ que foi projetado para resolver o problema MER descrito na Equação (2.20) [Teixeira et al. 2000]. De acordo com [Teixeira et al. 2000] o método MOBJ consiste em controlar a complexidade das redes através da minimização simultânea do erro para os padrões de treinamento e da norma do vetor de pesos. Dado o conjunto de dados $T = \{\mathbf{x}_i, y_i \mid i = 1 \dots N\}$, a Equação (2.21) a seguir, fornece a formulação biobjetivo, proposta em [Teixeira et al. 2000], para o aprendizado de redes MLP,

$$\min \begin{cases} J_1(\mathbf{w}) = \sum_{i=1}^N (y_i - \hat{f}(\mathbf{x}_i, \mathbf{w}))^2 \\ J_2(\mathbf{w}) = \|\mathbf{w}\| \end{cases} \quad (2.21)$$

onde $\hat{f}(\mathbf{x}_i, \mathbf{w})$ é a saída estimada pela rede para o i -ésimo padrão de entrada, y_i é a saída esperada (rótulo), \mathbf{w} é o vetor que armazena todos os pesos da rede e, $\|\cdot\|$ é o operador que fornece a norma euclidiana de um vetor.

Ao final do aprendizado, o algoritmo MOBJ gera uma estimativa para o conjunto de soluções não dominadas¹ denominado conjunto Pareto-ótimo (PO), onde a distância

¹Em um problema de otimização multiobjetivo, uma solução é dita ser não dominada, se não existe nenhuma solução com desempenho superior a ela em todos os objetivos.

euclidiana entre a norma de cada solução é dada por $\Delta\|w\|$, como é mostrado na Figura 2.3. Tais soluções representam o *trade-off* entre o erro de treinamento e a complexidade da rede, ou seja, essas soluções são aquelas as quais não há mais como melhorar um dos objetivos sem que haja uma perda do outro. O conjunto (*PO*) possui dois conjuntos de soluções que podem ser classificadas como sub-ajustadas e super-ajustadas. Soluções super-ajustadas são aquelas com alta complexidade e baixo erro para o conjunto de treinamento que podem gerar *overfitting*. Enquanto as sub-ajustadas apresentam erros grandes para os padrões de treinamento. Ainda de acordo com [Teixeira et al. 2000], o método possui as seguintes vantagens: O compromisso entre o erro e a complexidade, expressa através da norma, fica explícito. Para cada solução pertencente ao conjunto Pareto-Ótimo, existirá um valor para erro e para norma específico. Logo existirá no conjunto de soluções (*PO*) aquela com a complexidade efetiva adequada, ou seja, dentre as soluções de norma (complexidade) máxima e norma mínima, poderá existir uma solução que possui a norma adequada, com um erro menor possível para a mesma.

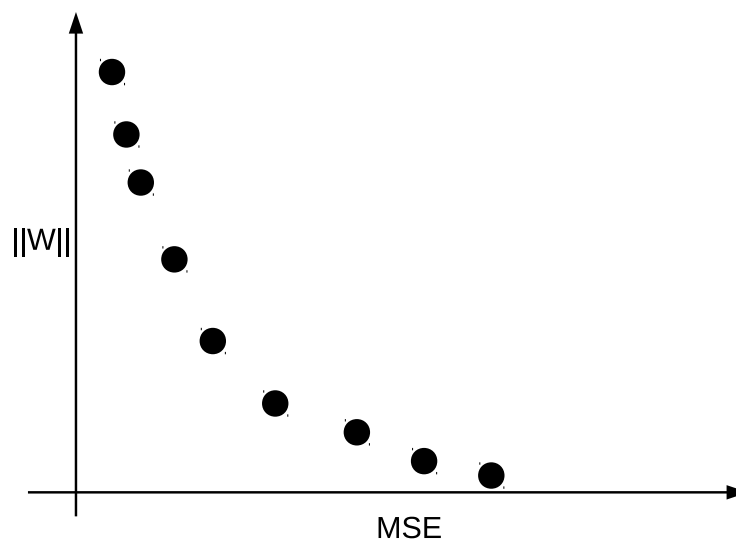


Figura 2.3: Pareto-ótimo

Na ausência de qualquer informação a priori referente aos objetivos $J_1(w)$ e $J_2(w)$, todas as soluções pertencentes ao conjunto *PO* são candidatas à solução do problema descrito na Equação (2.21). Uma etapa de decisão (decisor) é então necessária para a escolha da solução que fornece o melhor compromisso entre o erro de treinamento e a complexidade da rede. Tal solução constitui a melhor aproximação para o mínimo

absoluto do funcional risco esperado R (ou erro de generalização).

2.2.3 Problema de Decisão Multiobjetivo

O problema geral de decisão multiobjetivo deve obedecer ao esquema dado pela Equação (2.22) [Medeiros et al. 2009],

$$\mathbf{w}^* = \arg \max_{\mathbf{w} \in \mathcal{W}} f_e \quad (2.22)$$

onde f_e é um funcional capaz de classificar as soluções componentes do conjunto Pareto-Ótimo segundo um critério especificado; f_e deverá fornecer uma medida de “qualidade” entre uma dada solução (função) particular, $\hat{f}(\mathbf{x}, \mathbf{w})$, obtida com o aprendizado multiobjetivo (MOBJ) e a função desconhecida que representa o mínimo absoluto do risco esperado, $f_0(\mathbf{x})$.

Dentre as estratégias de decisão multiobjetivo propostas na literatura destacam-se:

- O decisor por mínimo erro de validação [Teixeira et al. 2000], onde a tomada de decisão é feita através de um conjunto de validação apresentado a todas as soluções pertencentes ao conjunto PO . A rede que apresentar o menor erro aos padrões de validação é escolhida como solução final. A desvantagem desse método é a necessidade de se separar parte do conjunto de dados para posterior validação dos modelos. Isso representa um problema para tarefas de aprendizado baseadas em um número muito limitado de exemplos.
- O decisor com conhecimento prévio [Medeiros et al. 2009] que realiza um teste estático para quantificar a probabilidade de um modelo de classificação ser o melhor comparado com os outros do conjunto PO . A formulação desse decisor depende de uma informação a priori sobre a distribuição do ruído presente nos dados. Na maioria dos problemas reais de aprendizado, no entanto, essa informação não se encontra disponível.

2.3 Conclusões do Capítulo

Neste capítulo foram analisados os principais métodos de classificação presentes na literatura. Os algoritmos descritos na seção 2.1 têm como objetivo obter modelos

com boa capacidade de generalização. Mas para alcançar tal capacidade é necessário estimar parâmetros. Como é o caso do parâmetro λ de regularização do método *weight decay*. E os parâmetro de kernel φ e de regularização γ da SVM e LS-SVM. Estimar esses parâmetros é um processo caro e demorado.

Enquanto que o algoritmo MOBJ, através de uma rede superdimensionada, é capaz de encontrar uma boa solução mesmo que a complexidade da rede neural seja elevada em relação ao problema de aprendizagem. O MOBJ depende de um decisor que seja capaz de encontrar dentre as soluções do Pareto Ótimo a solução de maior capacidade de generalização.

No capítulo seguinte será abordado um novo decisor para o algoritmo MOBJ, essa abordagem se difere por não necessitar de nenhum parâmetro fornecido pelo usuário.

3 *Abordagem Proposta*

O decisor proposto nesse capítulo foi projetado para problemas de classificação. Nesse contexto, parte-se da premissa que, dentre as soluções do conjunto Pareto Ótimo, deve-se selecionar aquela que ignora o ruído presente nos dados e maximiza a margem (ou distância) de separação entre as classes.

A implementação dessa estratégia de decisão é baseada nas seguintes etapas: eliminação de dados ruidosos, detecção da borda das classes e síntese de padrões junto à região que representa a margem de separação. Isso é obtido modelando-se os padrões de entrada como um grafo de proximidade, conhecido como Grafo de Gabriel [Berg et al. 2000]. Antes de detalhar o algoritmo de decisão proposto, é necessário definir o Grafo de Gabriel com base nos fundamentos da Geometria Computacional [Berg et al. 2000].

3.1 Teoria dos Grafos

As definições aqui apresentadas foram extraídas de [Barroso 2007, Christofides 1975]. Um grafo $G(V, E)$ é uma estrutura matemática constituída de dois conjuntos: um finito e não vazio, de n vértices V , e outro E , de m arestas, que são pares não ordenados de elementos de V . Na representação geométrica de um grafo os pontos estão associados aos vértices e, as arestas, correspondem a linhas arbitrárias que ligam os vértices (pontos) que as definem. A Figura 3.1 mostra a representação geométrica do grafo $G(V, E)$, onde $V = 1, 2, 3, 4$ e $E = (1, 3), (1, 4), (2, 3), (3, 4)$. Diz-se que dois vértices de um grafo são adjacentes quando definem uma aresta. Um grafo é caracterizado como planar quando suas arestas nunca se intersectam.

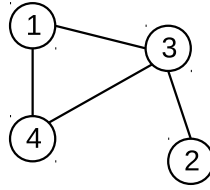


Figura 3.1: Forma geométrica de um Grafo.

3.2 Diagrama de Voronoi

O Diagrama de Voronoi representa a decomposição de um espaço em regiões de acordo com a distância entre determinados pontos [Aurenhammer e Klein 1990]. Dados dois pontos $p, q \in S$, onde S representa um conjunto de pontos em um plano, a bissetriz $B(p, q)$ corresponde a uma reta perpendicular que atravessa o centro do segmento de reta \overline{pq} ; $\delta(\cdot)$ é o operador que fornece a distância euclidiana entre dois pontos (vetores). O diagrama de Voronoi ($Vor(S)$) pode ser considerado como a divisão do plano em m polígonos convexos P [Figueiredo 1991, Berg et al. 2000]. A Figura 3.2(a) ilustra um exemplo. Um polígono $P(x_i)$ é chamado de polígono de Voronoi relativo a x_i e é formado através da intersecção do conjunto das bissetrizes $B(x_i, P(x_i))$. Um ponto $p \in S$ pertence a $P(x_i)$ se e somente se a seguinte desigualdade for satisfeita,

$$\delta(p, x_i) \leq \delta(p, x_j), x_i, x_j \in S, j \neq i \quad (3.1)$$

3.3 Triangulação de Delaunay

No diagrama de Voronoi $Vor(S)$ cada elemento $v \in S$ está associado a um polígono de $Vor(S)$. O grafo dual de $Vor(S)$ tem por vértices os elementos de S e por arestas os pares de elementos de S cujos polígonos de $Vor(S)$ são vizinhos [Li e Kuo 1998, Zhang e He 2006]. Tal diagrama resultante é chamado de Diagrama de Delaunay [Figueiredo 1991, Berg et al. 2000], conforme ilustrado na Figura 3.2(b). A Triangulação de Delaunay (TD) é uma subdivisão de um objeto geométrico em triângulos. A TD pode ser modelada através de um grafo planar G_p composto por um conjunto de vértices V e um conjunto de arestas E não ordenadas, como pode ser visto na Figura 3.2(c). Uma aresta $(v_i, v_j) \in G_p$ é definida se e somente se existir um círculo contendo o par (v_i, v_j) e

que todos os outros vértices de V sejam exteriores a este círculo. Essa característica é ilustrada na Figura 3.2(d).

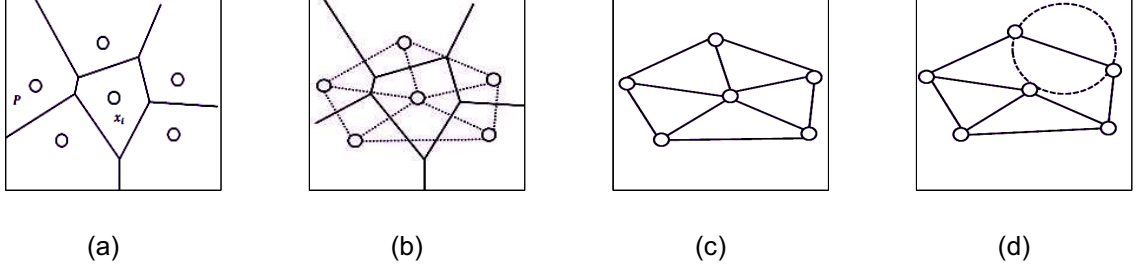


Figura 3.2: (a) Diagrama de *Voronoi*. (b) Grafo Dual do Diagrama de *Voronoi*. (c) Triangulação de *Delaunay* resultante. (d) Par de pontos de *Voronoi* que possuem uma aresta em comum representado em uma Triangulação de *Delaunay*.

3.4 Grafo de Gabriel

O grafo de Gabriel G_G é um subconjunto de pontos do Diagrama de *Voronoi* e também um subgrafo da Triangulação de *Delaunay* [Zhang e King 2002], ou seja, $G_G \subseteq T_D \subseteq Vor$. Segundo [Berg et al. 2000], o Grafo de Gabriel G_G de um conjunto de pontos S , é um grafo cujo conjunto de vértices $V = S$ e seu conjunto de arestas E deve obedecer à seguinte definição,

$$(v_i, v_j) \in E \leftrightarrow \delta^2(v_i, v_j) \leq [\delta^2(v_i, z) + \delta^2(v_j, z)] \quad \forall z \in V, v_i, v_j \neq z \quad (3.2)$$

o que implica que para (v_i, v_j) constituir uma aresta de G_G , não pode haver nenhum outro vértice dentro do círculo cujo o diâmetro é a distância euclidiana entre v_i e v_j . As Figuras 3.3(a), 3.3(b), 3.3(c), 3.3(d), 3.3(e), 3.3(f), 3.3(g), 3.3(h) e 3.3(i) mostram a construção do grafo de Gabriel de forma detalhada. É possível observar através da Figura 3.3(f) que a escolha dos dois vértices em questão (círculo pontilhado) não satisfazem Equação (3.2), logo não possuem uma aresta.

O KNN (*K-Nearest Neighbors*) é um método de classificação cuja idéia principal é classificar um elemento no espaço n -dimensional de acordo com a classe dominante de seus K vizinhos mas próximos obtidos no conjunto de treinamento. Um dos problemas encontrados no método é encontrar o melhor valor para o parâmetro K . O Grafo de Gabriel e o Diagrama de *Voronoi* também podem ser usados para esse propósito, mas diferente do KNN não necessitam do parâmetro K . Como mencionado na

Seção 3.2, um ponto no Diagrama de *Voronoi* representa um polígono convexo cujas fronteiras representam seus vizinhos mais próximos. Assim, a fronteira que separa dois pontos de diferentes classes pode ser usada como uma fronteira de decisão para classificação.

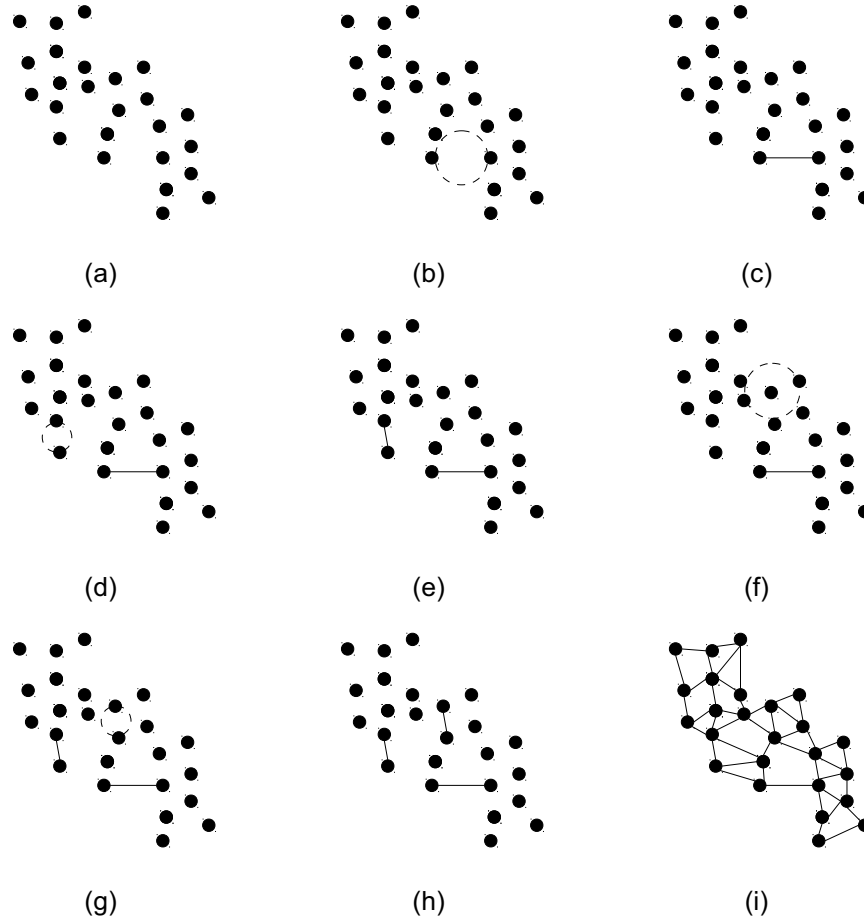


Figura 3.3: Construção do grafo de Gabriel.

3.5 Algoritmo de Decisão Baseado em Margem Para Problemas de 2-Dimensões

O decisor proposto nesse trabalho busca pela solução \mathbf{w}^* do conjunto PO mais próxima a pontos distribuídos dentro da margem de separação entre as classes. Seu algoritmo pode ser formulado de acordo com os seguintes passos:

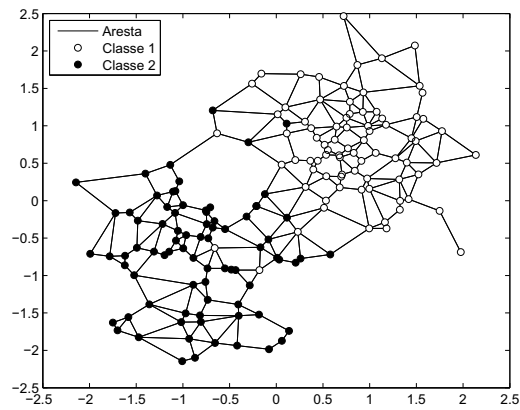
1. A partir de um conjunto de dados $T = \{\mathbf{x}_i, y_i \mid i = 1 \dots N\}$, é realizado o treinamento multiobjetivo (MOBJ) da rede MLP, conforme descrito na Seção 2.2.2, e o conjunto PO é obtido.

2. Obtém-se o Grafo de Gabriel G_G com o conjunto de vértices sendo formado por todos os padrões de entrada, *i.e.*, $V = \{\mathbf{x}_i \mid i = 1 \dots N\}$, e o conjunto de arestas E satisfazendo a condição estabelecida na Equação (3.2). Veja exemplo na Figura 3.4(a).
3. Este passo do algoritmo é responsável por detectar e remover o ruído dos dados. Para todo $\mathbf{x}_i \in V$, analisa-se o subgrafo induzido pelo vértice \mathbf{x}_i , ou seja, o subgrafo formado pelas arestas que possuem \mathbf{x}_i como uma das extremidades. Se a maioria dos vizinhos (vértices adjacentes de \mathbf{x}_i) possui rótulo diferente de y_i , então \mathbf{x}_i é considerado como ruído e eliminado de V . Esse passo deve ser repetido até que não haja mais exclusão. A Figura 3.4(b) ilustra o conjunto de dados após a eliminação dos dados ruidosos.
4. A borda B_r das classes é encontrada da seguinte forma: seja $\mathbf{x}_i \in V$ e, $\forall \mathbf{x}_j \in V$ com $j \neq i$, caso a aresta $(\mathbf{x}_i, \mathbf{x}_j)$ for formada por vértices de classes distintas (com rótulos diferentes), então ela é incluída no conjunto B_r , conforme ilustrado pela Figura 3.4(c).
5. Neste passo são calculados os pontos médios entre os exemplos que compõem as bordas das classes. Para cada aresta $(\mathbf{x}_i, \mathbf{x}_j)$ pertencente ao conjunto B_r , calcula-se o ponto médio entre os vértices \mathbf{x}_i e \mathbf{x}_j de acordo com a seguinte expressão,

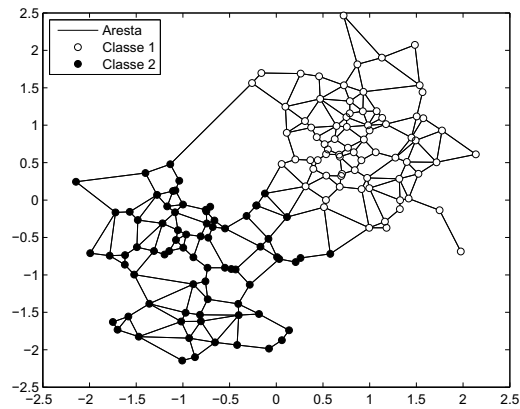
$$\bar{\mathbf{x}}_{ij} = \sum_{t=1}^n \mu(\mathbf{x}_i(t), \mathbf{x}_j(t)) \quad (3.3)$$

onde n é número de atributos (características) dos padrões de entrada e $\mu(\cdot)$ é o operador que calcula a média para o t -ésimo atributo. Após o cálculo dos pontos médios para todas as arestas de B_r , obtém-se um conjunto de ponto médios P_M relativo à bordas das classes, conforme ilustrado pela Figura 3.5(a).

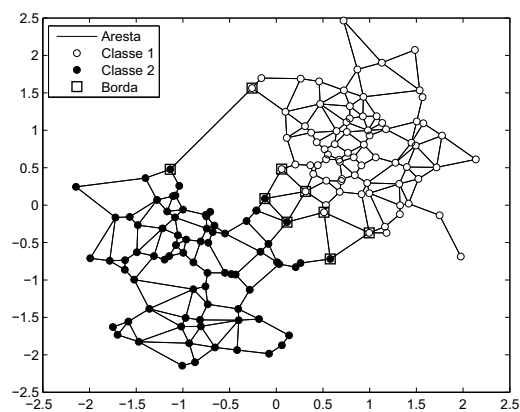
6. Seja $W = \{\mathbf{w}_k \mid k = 1 \dots L\}$ o conjunto de soluções do conjunto PO . Para se obter a solução que maximiza a margem de separação entre as classes, o decisor deve escolher a solução $\mathbf{w}_k \in W$ mais próxima dos pontos médios (P_M) calculados no passo anterior. Isso é obtido calculando-se, para cada \mathbf{w}_k , a distância entre os vetores de entrada que se encontram exatamente na superfície de decisão s_k calculado através do grid de soluções mostrado nas Figuras 3.6(a), 3.6(b), 3.6(c) (estimada a partir de \mathbf{w}_k) e os vetores pertencentes ao conjunto P_M , conforme as expressões a seguir,



(a)



(b)



(c)

Figura 3.4: (a) Conjunto de padrões de entrada modelado com o Grafo de Gabriel. (b) Conjunto resultante após a eliminação dos dados ruidosos. (c) Detecção da borda das classes.

$$d_k(j) = \min \delta(s_k, P_M(j)), \quad \forall j \in P_M \quad (3.4)$$

$$D_k = \sum_{j=1}^{|P_M|} d_k(j) \quad (3.5)$$

onde $d_k(j)$ fornece a distância entre o j -ésimo ponto médio ($P_M(j)$) e o ponto pertencente à superfície de decisão s_k que se encontra mais próximo de $P_M(j)$; $|P_M|$ é o número de pontos médios nas bordas das classes e $\delta(\cdot)$ é o operador que fornece a distancia euclidiana entre dois vetores. Finalmente, a solução escolhida pelo decisor proposto deve ser,

$$\mathbf{w}^* = \min D_k \quad (3.6)$$

onde \mathbf{w}^* é a solução mais próxima dos pontos médios, ou seja, a solução que possui a maior margem de separação entre as classes. A Figura 3.5(b) ilustra algumas superfícies de decisões geradas a partir de diferentes soluções do conjunto PO. A superfície de decisão mais próxima dos pontos médios é mostrada pela Figura 3.7(a). A solução correspondente no conjunto PO é marcada na Figura 3.7(b).

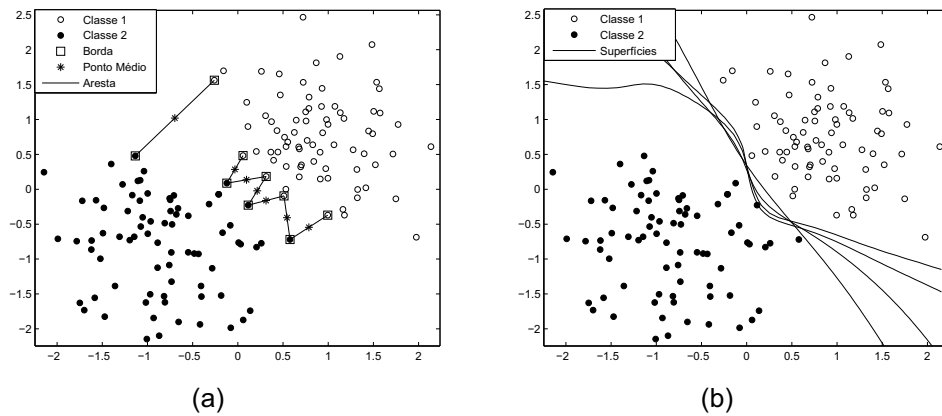


Figura 3.5: (a) Pontos médios relativo às bordas das classes. (b) Superfícies de decisão geradas a partir de soluções de PO.

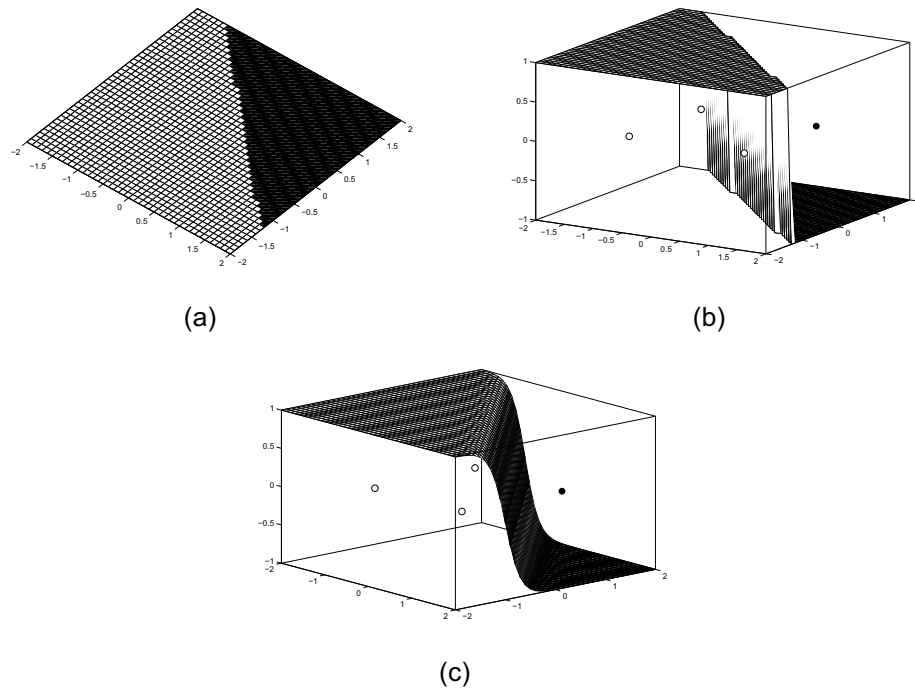


Figura 3.6: (a) Grid. (b) Grid de uma superfície linear. (c) Grid de uma superfície não linear.

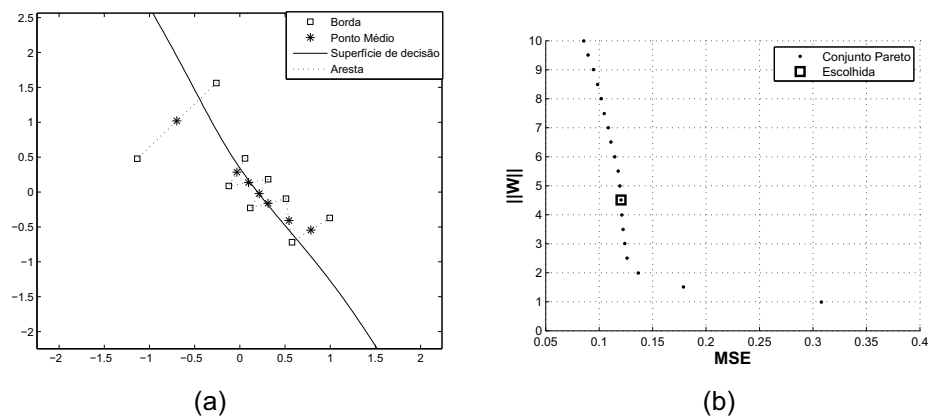


Figura 3.7: (a) Superfície de decisão mais próxima dos pontos médios. (b) Conjunto Pareto-Ótimo.

3.6 Algoritmo de Decisão Baseado em Margem Para Problemas de n -dimensões

É sabido que o cálculo do grid de soluções do algoritmo da seção 3.5 exige um processamento computacional alto, e para problemas com vários atributos se torna praticamente inviável. Pensado nisso foi proposto outro algoritmo para resolver problemas de várias dimensões sem a necessidade da construção do grid. Esse algoritmo busca encontrar no espaço de características as soluções que estão mais próximas dos pontos médios no espaço de entrada.

3.6.1 Metodologia

Nessa metodologia, parte-se da mesma premissa do algoritmo descrito na seção 3.5, o decisor busca selecionar dentre as soluções do conjunto PO a superfície mais próxima dos pontos médios.

Partiu-se da busca pela solução através do espaço de características. A "métrica" ψ utilizada no espaço de características, foi a diferença entre o valor de saída da camada escondida de uma amostra x , e o ponto de inflexão da função de transferência da rede. Nesse caso foi utilizada a função tangente hiperbólica (vide Equação (3.7)), sendo seu ponto de inflexão igual a zero, como mostrado na Figura 3.8. (As Figuras 3.9(a) e 3.9(b) ilustram esse procedimento). A solução que apresentasse a menor diferença para as amostras do conjunto de pontos médios P_M seria escolhida. Entretanto, é sabido que o mapeamento dos dados no espaço de entrada para o espaço de características é diferente para cada solução do conjunto PO , como é possível observar na Figura 3.10. A solução de superfície mais próxima dos pontos médios no espaço de entrada, nem sempre é a solução mais próxima do hiperplano no espaço de características. E na maioria das vezes a solução escolhida é uma das superfícies com formato de reta, como mostrado na Figura 3.11.

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (3.7)$$

Buscou-se então encontrar uma característica de similaridade entre o espaço de entrada e o espaço de características. As soluções que classificam todos os dados da borda (descrita no algoritmo da seção 3.5) tem um mapeamento com características semelhantes. Uma delas é que as superfícies dessas soluções no espaço de entrada

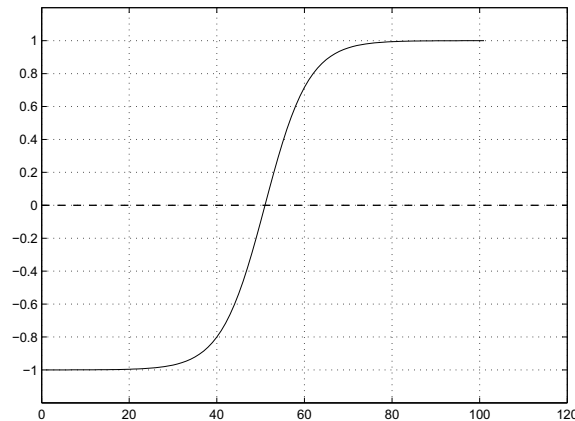


Figura 3.8: Função de transferência tangente hiperbólica e seu ponto de inflexão igual a zero.

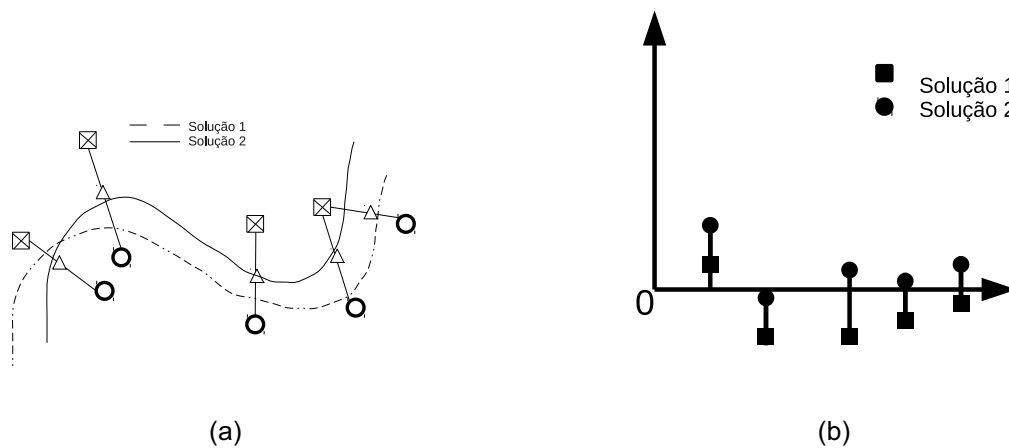


Figura 3.9: (a) Duas soluções (superfícies) próximas aos pontos médios (triângulos). (b) Distância dos pontos médios da Figura 3.9(a) para o ponto de inflexão da função tangente hiperbólica mostrado na Figura 3.8.

estão dentro da margem, e quando é aplicada a "métrica" ψ somente nessas soluções, a solução com a superfície mais próxima dos pontos médios no espaço de entrada é encontrada.

Infelizmente esse método não pôde ser aplicado, devido a possibilidade de não se encontrar nenhuma solução do conjunto PO que seja capaz de classificar corretamente todos os vértices da borda, como mostra a Figura 3.12(a). É possível que a classificação não ocorra pelo fato de haver ruído no conjunto de treinamento. A Figura 3.12(b) mostra os ruídos do conjunto de treinamento dentro dos círculos, que pode ser o motivo do deslocamento da solução para fora da última borda. Um relaxamento

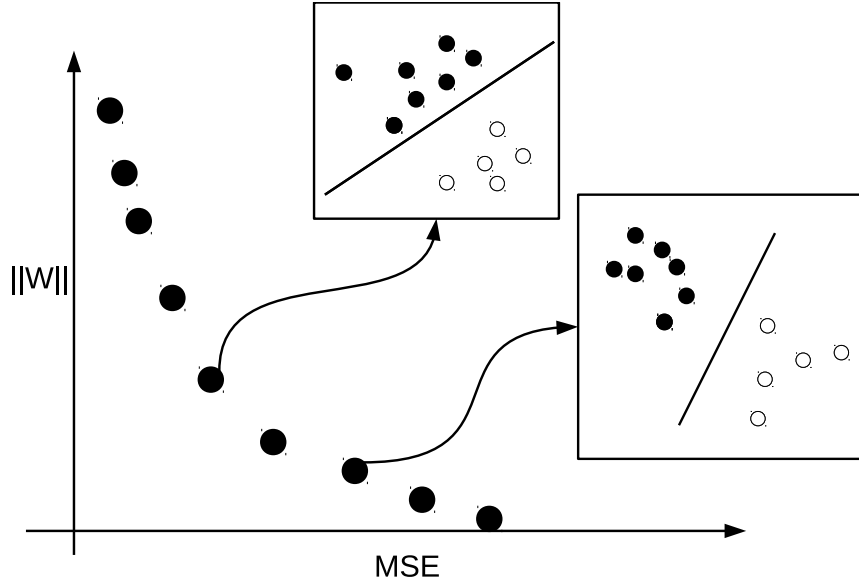


Figura 3.10: Conjunto Pareto-ótimo e os diferentes mapeamentos das soluções.

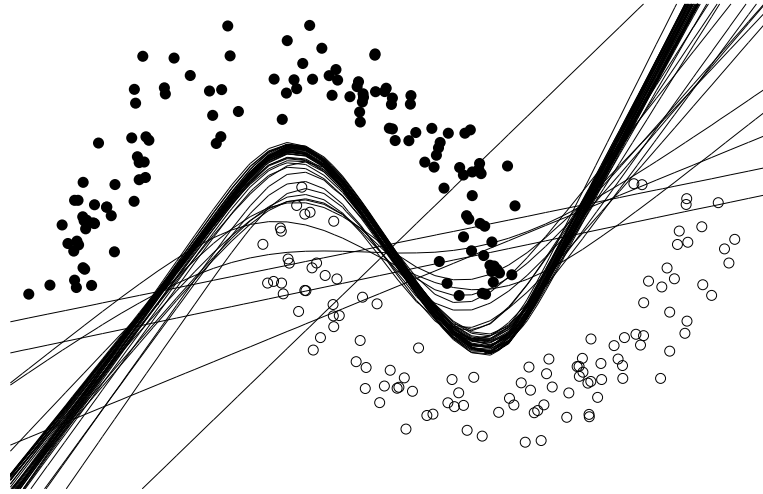
dessa metodologia poderia ser criado, ou seja, selecionar as soluções que classificam apenas k bordas. Foi pensado também em pré-selecionar as soluções com uma taxa c de coeficiente de variação. Mas essas abordagens tirariam a principal característica do decisor, a de não precisar de parâmetros fornecidos pelo usuário.

Por fim, uma última metodologia para uma pré-seleção foi desenvolvida. O conjunto de soluções ρ que será utilizado pelo decisor, contém as soluções onde os pontos médios são mais próximos do hiperplano no espaço de características do que outras amostras. A diferença desse método é que não são comparados somente o conjunto dos $|P_M|$ pontos médios, mas também as $|P_M|$ amostras mais próximas do hiperplano, o resultado é mostrado na Figura 3.13. Logo, uma solução α é candidata, se ela satisfaz a desigualdade da Equação (3.8)

$$\rho(\alpha) = \sum_{j=1}^{|P_M|} \psi(w(\alpha), P_M(j)) < F_k(w(\alpha), T) \quad (3.8)$$

$$\psi(w, x) = \theta^* - \varphi(x, w) \quad (3.9)$$

onde P_M é a quantidade de pontos médios, T o conjunto de amostras no espaço de entrada, x é uma amostra no espaço de entrada, $w(\alpha)$ representa a α -ésima solução do conjunto PO , $\psi(\cdot)$ é a função que encontra a diferença entre o funcional de saída da

Figura 3.11: Soluções do conjunto PO

camada escondida da rede $\varphi(\cdot)$ e o ponto de inflexão da função de transferência θ^* , e $F_k(\cdot)$ é uma função que retorna a soma das diferenças das $|P_M|$ amostras do conjunto T mais próximas do ponto de inflexão, ou seja, as $|P_M|$ amostras com menor diferença de acordo com a Equação (3.9).

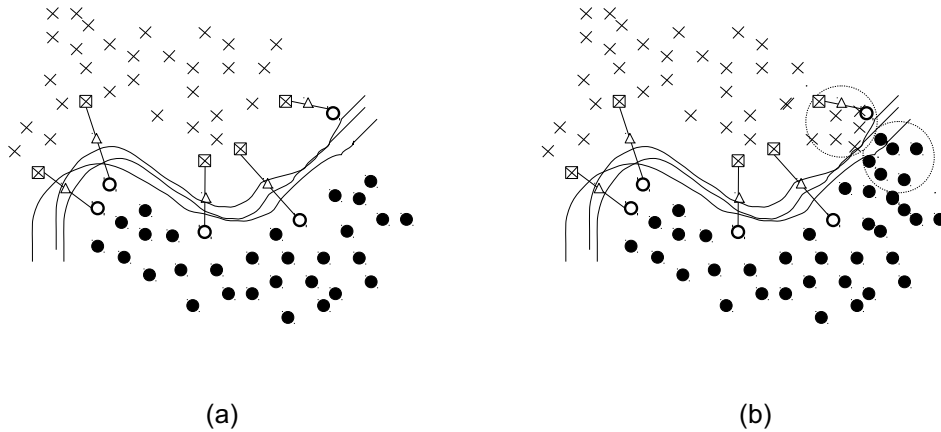


Figura 3.12: (a) Soluções do PO que não classificam todos os vértices da margem. (b) Um dos motivos para o deslocamento das soluções.

Após a pré-seleção para construção do conjunto ρ , é preciso tomar a decisão final. Para encontrar a solução mais próxima dos pontos médios calcula-se para cada solução $w \in \rho$ a distância dos pontos médios para o ponto de inflexão por meio da Equação (3.9). A solução w^* retornada pela função formulada pela Equação (3.10) é a escolhida como sendo a mais próxima dos pontos médios no espaço de entrada.

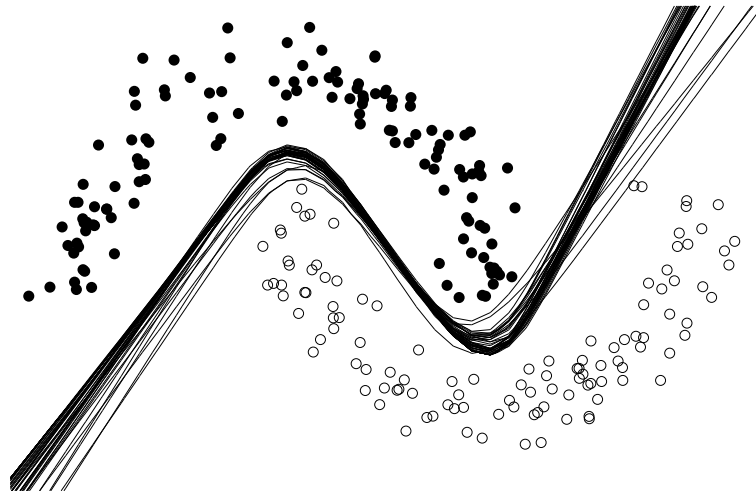


Figura 3.13: Soluções mais próximas dos pontos médios.

$$\mathbf{w}^* = \min \rho \quad (3.10)$$

3.7 Conclusões do Capítulo

Neste capítulo foi apresentado a nova abordagem baseada em margem para tomada de decisão no aprendizado multiobjetivo. A metodologia foi apresentada em dois algoritmos, sendo que o primeiro aborda problemas bidimensionais e o segundo trata problemas de várias dimensões. Além disso, é mostrado o progresso da construção do decisor desde a sua fase inicial. A metodologia apresentada mostra que é preciso pré-selecionar soluções do conjunto PO para depois tomar a decisão final. Essa pré-seleção é feita através de um algoritmo que encontra as soluções que estão dentro da margem de separação entre as classes, essa concepção é feita através de uma relação entre o espaço de características e o espaço de entrada.

No capítulo seguinte são mostrados os resultados obtidos com o decisor aplicado a problemas de classificação bidimensionais, e também de várias dimensões.

4 *Resultados*

Neste capítulo são apresentados os resultados do método de decisão proposto nesse trabalho. O capítulo se divide em duas partes. A primeira parte aborda problemas de classificação bidimensionais. A segunda parte, por sua vez aborda os problemas de classificação de n-dimensões.

4.1 Resultados Para Problemas de 2-Dimensões

Com o objetivo de verificar a eficiência da estratégia de decisão proposta no aprendizado multiobjetivo de redes MLP, experimentos foram conduzidos com dois problemas (*benchmarks*) de classificação conhecidos na literatura: “tabuleiro de xadrez” e “duas luas”.

4.1.1 Tabuleiro de Xadrez

O “tabuleiro de xadrez” é um problema multimodal em que as classes (círculos preenchidos e não-preenchidos) encontram-se distribuídas na forma de um tabuleiro 4x4, apresentando sobreposição, conforme ilustrado pela Figura 4.1. O número total de padrões de entrada é igual a 1600.

O algoritmo de treinamento MOBJ [Teixeira et al. 2000] teve seus parâmetros ajustados da seguinte forma: o conjunto PO de soluções foi composto por 189 redes MLP com 30 neurônios na camada oculta. Foi utilizada a função de transferência tangente hiperbólica para a camada oculta e de saída. A diferença de norma euclidiana (complexidade) entre as soluções foi de $\Delta\|w\| = 0.5$. Esses valores de parâmetros foram escolhidos já que segundo [Teixeira et al. 2000] é característico do algoritmo MOBJ a geração de soluções desde sub-ajustadas até super-ajustadas.

O decisor baseado em margem foi aplicado para a escolha da solução PO . Tal

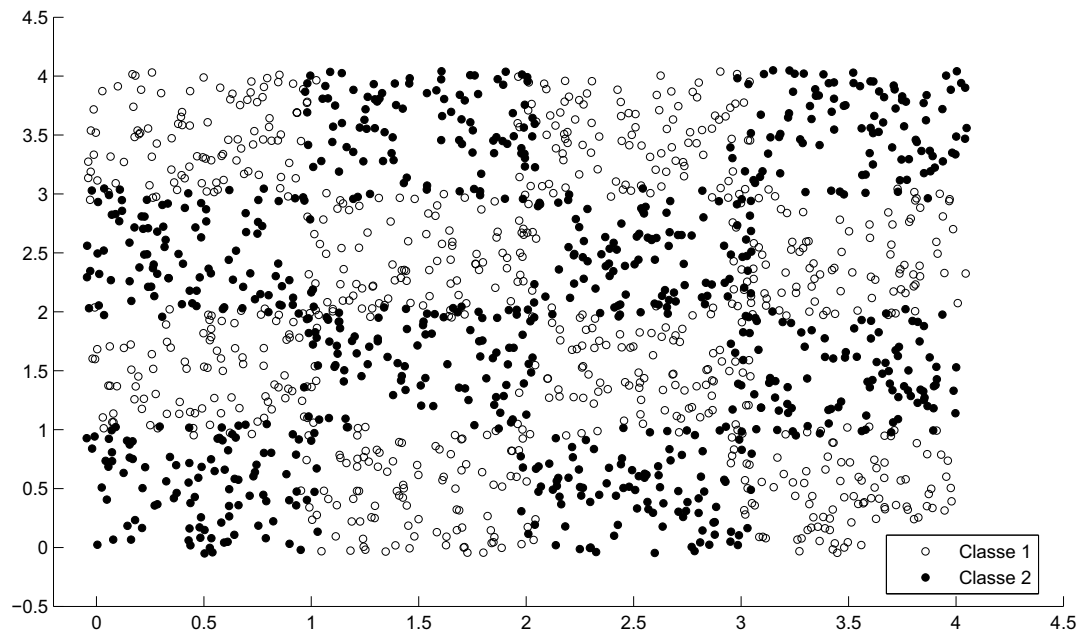


Figura 4.1: Problema do tabuleiro de xadrez.

decisor não possui parâmetros para ajuste e também não necessita do uso de um conjunto de novos exemplos para validação. Assim, todos os padrões puderam ser usados no treinamento MOBJ da rede MLP. A Figura 4.2 mostra a fase inicial do algoritmo decisor proposto com o conjunto de padrões de entrada modelado através do Grafo de Gabriel. O ruído foi eliminado e as bordas de separação entre as classes detectada. A Figura 4.3 mostra a superfície de decisão (estimada a partir de \mathbf{w}^*) selecionada pelo decisor baseado em margem. Note a partir da Figura 4.3, que apesar da dificuldade do problema do “tabuleiro de xadrez”, a superfície de decisão escolhida apresenta certa suavidade (complexidade baixa), sendo capaz de separar corretamente as classes. Além disso, não se observa comportamento de *under/overfitting* em relação aos dados de treinamento.

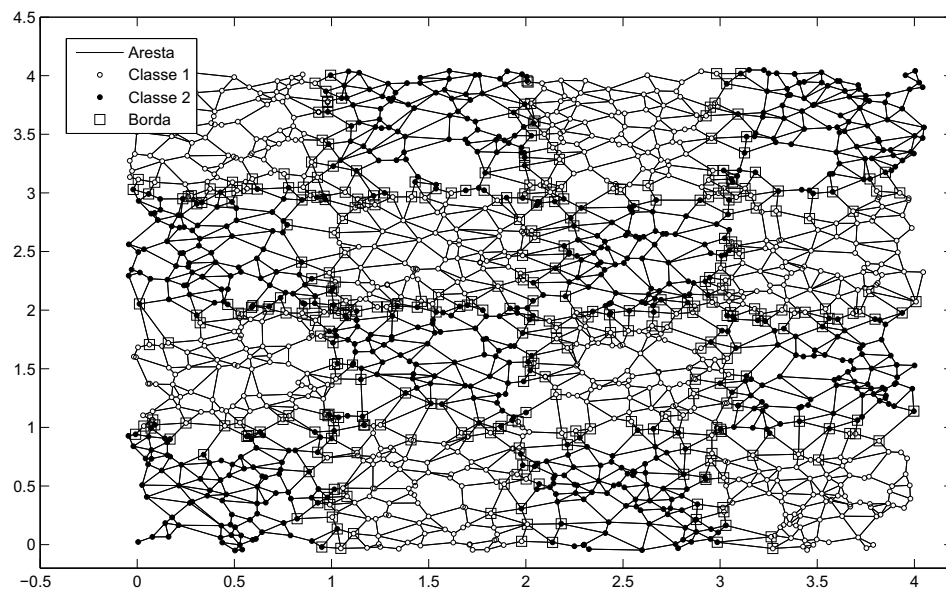


Figura 4.2: Modelagem do problema com o Grafo de Gabriel. O ruído dos dados foi eliminado e as bordas entre as classes foi detectada.

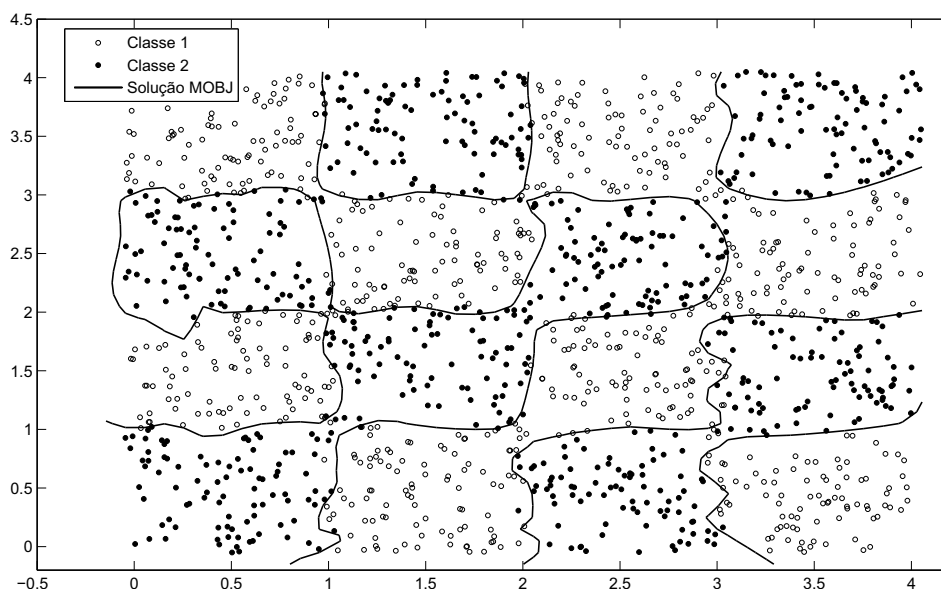


Figura 4.3: Solução (superfície de decisão) escolhida pelo decisor baseado em margem.

4.1.2 Duas Luas

A base de dados “duas luas” é não linearmente separável e possui 400 padrões de entrada divididos, igualmente, em duas classes (círculos preenchidos e não preenchidos), conforme ilustrado pela Figura 4.4(a).

Nesse experimento, os resultados obtidos com nossa estratégia de decisão foram confrontados com o decisor por mínimo erro de validação [Teixeira et al. 2000], comumente usado por algoritmos de aprendizado multiobjetivo. Para a aplicação desse decisor, o conjunto de dados original teve que ser particionado em 70% para treinamento e 30% para validação. O decisor baseado em margem pôde contar com 100% do conjunto de dados para treinamento da rede MLP.

No aprendizado com o algoritmo MOBJ, o conjunto PO de soluções foi composto por 19 redes MLP com 5 neurônios na camada oculta. Similarmente ao problema do “tabuleiro de xadrez”, foi utilizada a função de transferência tangente hiperbólica para a camada oculta e de saída. A diferença de norma euclidiana (complexidade) entre as soluções foi de $\Delta\|w\| = 0.5$.

A Figura 4.4(b) ilustra a aplicação do decisor proposto com os padrões entrada sendo modelados através do Grafo de Gabriel. A superfície de decisão selecionada

(linha pontilhada) é mostrada na Figura 4.4(c). A Figura 4.4(d) compara as superfícies de decisão obtidas, respectivamente, pelo decisor de validação (linha contínua) e pelo decisor baseado em margem (linha pontilhada). Nota-se, pela Figura 4.4(d), que a solução escolhida por nosso decisor baseado em margem apresenta melhor margem (distância) de separação entre as classes que a solução obtida por validação.

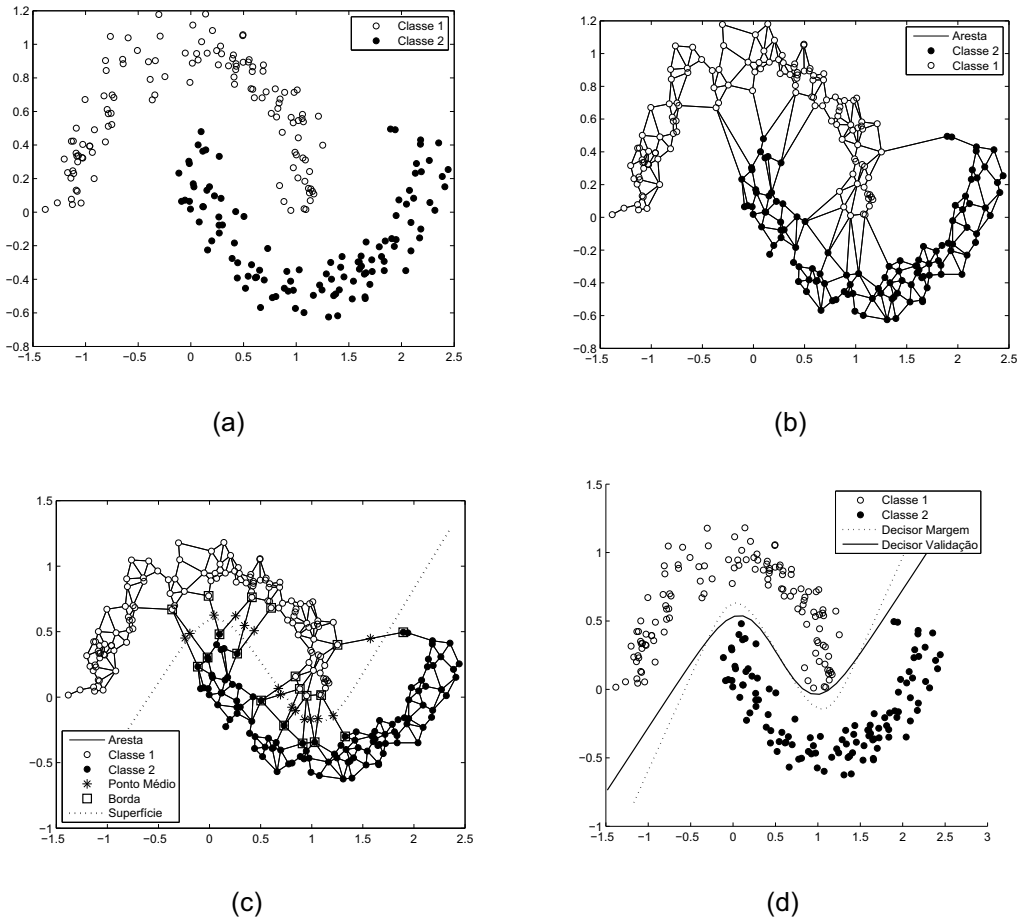


Figura 4.4: (a) Problema das duas luas. (b) Modelagem do problema “duas luas” com o Grafo de Gabriel. (c) Solução encontrada pelo decisor baseado em margem. (d) decisor por validação (linha contínua) vs. decisor baseado em margem (linha pontilhada).

4.2 Resultados Para Problemas de N-Dimensões

As bases de dados foram obtidas através do repositório publico UCI [Blake e Merz 1998]. Todas as bases utilizadas no trabalho são binárias, são elas: the Stalog Australian Credit (acr), the Stalog German Credit (gcr), the Stalog heart disease (hea), the Johns Hopkins university ionosphere (ion), the Pima Indians diabetes (pid), the sonar (snr) e the Wisconsin breast cancer (wbc). A Tabela 4.1 mostra as características de cada base, em que $N_{Tr/Vc}$ é a quantidade de dados utilizados para treinamento ou validação cruzada, N_{teste} é o numero de observações no conjunto de teste e N é o numero total da base de dados. O numero de atributos numéricos e categóricos são denotados por n_{num} e n_{cat} respectivamente, e n é o numero total de atributos. Todas as bases foram normalizadas com média $\bar{x} = 0$ e desvio padrão $\sigma = 1$. As bases foram divididas em três partes, cada uma contendo 10 partições aleatórias. Os primeiros $2/3$ dos dados foram reservados para treinamento e/ou validação cruzada e o restante para teste. Para cada base de dados, como resultado, é apresentado a média e o desvio padrão da acurácia.

Tabela 4.1: Características das Bases de Dados

	acr	gcr	hea	ion	pid	snr	wbc
$N_{Tr/Vc}$	460	666	180	234	512	138	455
N_{teste}	230	334	90	117	256	70	228
N	690	1000	270	351	768	208	683
n_{num}	6	7	7	33	8	60	9
n_{cat}	8	13	6	0	0	0	0
n	14	20	13	33	8	60	9

4.2.1 Algoritmos utilizados nos resultados

Os resultados obtidos foram comparados com os Benchmarks do algoritmo LS-SVM apresentado no artigo [Gestel et al. 2004], e também de uma SVM através do toolbox libsvm [Chang e Lin 2011]. De acordo com [Gestel et al. 2004] o parâmetro de regularização γ e do kernel ϕ foram selecionados a partir de *10-fold cross-validation*. Ainda de acordo com [Gestel et al. 2004], o melhor kernel para o classificador LS-SVM para bases binárias foi o Kernel RBF. A Tabela 4.2 mostra os valores dos parâmetros para cada base de dados. Os parâmetros γ e ϕ da SVM foram encontrados a partir de *10-fold cross-validation*. O tipo de Kernel foi o mesmo utilizado na LS-SVM. Os

melhores parâmetros para cada base são mostrados na Tabela 4.2.

Tabela 4.2: Valores de parâmetros para o kernel RBF

	acr	gcr	hea	ion	pid	snr	wbc
$LS-SVM: \phi$	22.75	31.25	5.69	3.30	240.00	33.00	6.97
$LS-SVM: \log_{10}(\gamma)$	0.09	2.43	-0.76	0.63	3.04	0.86	-0.66
$SVM: \phi$	512	32768	8192	32768	2	32768	8192
$SVM: \log_{10}(\gamma)$	-2.70	2.30	-4.51	-0.9	-2.10	-1.5	-4.51

Para as bases de dados mostradas na Tabela 4.1 o algoritmo de treinamento MOBJ [Teixeira et al. 2000] teve seus parâmetros ajustados da seguinte forma: o conjunto PO de soluções foi composto por 37 redes MLP com 10 neurônios na camada oculta. Foi utilizada a função de transferência tangente hiperbólica para a camada oculta e de saída. A diferença de norma euclidiana (complexidade) entre as soluções foi de $\Delta\|w\| = 0.3$. O decisor apresentado na seção 3.6 foi utilizado para escolha da melhor solução do conjunto PO .

Tabela 4.3: Resultados

	acr	gcr	hea	ion	pid	snr	wbc
$LS-SVM(RBF)$	87.0(2.1)	76.3(1.4)	84.7(4.8)	96.0(2.1)	76.8(1.7)	73.1(4.2)	96.4(1.0)
$MOBJ(DecisorMargem)$	87.79(0.88)	78.13(0.61)	87.3(2.3)	88.12(2.39)	76.04(2.38)	76.28(1.11)	97.05(1.01)
SVM	86.24(0.88)	75.86(2.20)	83.08(3.10)	93.86(3.31)	<u>77.25(1.20)</u>	75.82(1.11)	97.02(1.55)

4.2.2 Análise do Resultados

A Tabela 4.3 mostra os resultados obtidos utilizando os algoritmos LS-SVM, MOBJ aliado ao decisor de margem proposto e SVM. São mostrados os resultados da média e desvio padrão para a métrica da acurácia e, os melhores resultados foram sublinhados. Das 7 bases utilizadas, o método com o decisor mostrou-se melhor em 5. O resultado obtido com a base Ionosphere apresentou grande diferença em relação aos outros dois algoritmos. Tal fato pode ser explicado porque dentre as soluções que poderiam ser escolhidas, nenhuma apresentava valores similares aos dos outros algoritmos, ou seja, o método MOBJ para essa base não forneceu resultados satisfatórios para o decisor. Por outro lado, dentre as soluções fornecidas, o decisor conseguiu escolher a segunda solução de maior acurácia entre 37 redes MLP. Esse resultado é importante e sugere que o decisor proposto, aliado ao algoritmo MOBJ, é eficiente no controle da capacidade de generalização da rede MLP.

4.2.3 Conclusões do Capítulo

Neste capítulo o decisor proposto foi aplicado a problemas de classificação com diferentes dimensões, conseguindo selecionar soluções com alta capacidade de generalização. Os algoritmos LS-SVM e SVM, tiveram seus resultados comparados com o resultado do decisor baseado em margem. Ao contrário da abordagem proposta, que não precisa de parâmetros fornecidos pelo usuário, os algoritmos confrontados necessitaram de ajustes de parâmetros. Na maioria das vezes esses ajustes são muito difíceis de serem feitos, exigindo a experiência do projetista ou uma busca exaustiva.

5 Conclusões e Trabalhos Futuros

Conforme argumentado nas seções iniciais dessa dissertação, o problema do aprendizado de RNAs e de máquinas de aprendizagem de uma maneira geral é intrinsecamente multiobjetivo. As funções custo relativas aos ajustes do modelo aos dados (erro) e da complexidade do modelo a uma dada tarefa de aprendizagem são conflitantes. Como não existe um único mínimo global que satisfaça à minimização simultânea dessas funções, um processo de decisão que determinará o equilíbrio entre o erro e a complexidade é necessário. Em outras máquinas de aprendizado como as *Support Vector Machines* (SVMs), por exemplo, a tomada de decisão ocorre quando o usuário define previamente os parâmetros de margem (regularização) e Kernel. A definição desses parâmetros por si só é considerado um processo de tomada de decisão multiobjetivo que, geralmente, é feito através de uma busca exaustiva com base em um conjunto representativo de validação (*crossvalidation*).

No caso particular desse trabalho, partiu-se do princípio de que a solução se encontra no conjunto Pareto-Ótimo (*PO*) e que uma delas deve então ser escolhida. O processo de escolha aqui proposto se baseia em uma estimativa geométrica para a margem de separação entre as classes. A solução que se encontra mais próxima da margem máxima é selecionada. Com essa metodologia, os processos de treinamento (geração do conjunto *PO*) e escolha da solução eficiente ocorrem de forma transparente para o usuário, o que contrasta com outros modelos de aprendizado, tais como as SVMs e as redes de regularização, em que a solução é obtida com base em parâmetros fornecidos pelo usuário.

Resultados com *benchmarks* bidimensionais e n-dimensionais mostraram que o método de decisão proposto, aliado ao algoritmo MOBJ, foi eficiente para a obtenção de modelos suaves (que ignoram ruído) e bem ajustados. Pretende-se no futuro formalizar essa metodologia para abrir caminho para uma concepção mais geral e única na construção de modelos neurais artificiais.

Referências Bibliográficas

- AURENHAMMER, F.; KLEIN, R. *Voronoi Diagrams: In Handbook of Computational Geometry*. [S.l.]: Elsevier, 1990. 152-159 p.
- BARROSO, M. M. A. Operações elementares em grafos. *EMARC, SBMAC*, v. 7, 2007.
- BERG, M. et al. *Computational Geometry: Algorithms and Applications*. second. [S.l.]: Springer-Verlag, 2000.
- BLAKE, C.; MERZ, C. *UCI Repository of machine learning databases*. Irvine, CA: University of California, Dept. of Information and Computer Science: [s.n.], 1998. Disponível em: <<http://www.ics.uci.edu/mlearn/MLRepository.html>>.
- CHANG, C.-C.; LIN, C.-J. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, v. 2, p. 27:1–27:27, 2011. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- CHRISTOFIDES, N. *Graph theory: An algorithmic approach*. [S.l.]: Academic press New York, 1975.
- COSTA, M. A.; BRAGA, A. P.; MENEZES, B. R. Improviing generalization of mlps witch multi-objective witch sliding mode control and the levenberg-maquardt algorithm. *Neurocomputing*, vol. 70, n. 7-9, p. 1342–1347, 2007.
- FIGUEIREDO, L. H. *Introdução a Geometria Computacional*. Rio de Janeiro: Impa, 1991.
- FRIEDMAN, J.; HASTIE, T.; TIBSHIRANI, R. *The Elements of Statistical Learning*. [S.l.]: Spinger, 2008.
- GEMAN, S.; BIENENSTOCK, E.; DOURSAT, R. Neural networks and the bias / variance dilemma. *Neural Computation*, Vol. 4, p. 1–58, 1992.
- GESTEL, T. et al. Benchmarking least squares support vector machine classifiers. *Machine Learning*, Springer, v. 54, n. 1, p. 5–32, 2004.
- GIROSI, F.; JONES, M.; POGGIO, T. Regularization theory and neural network architectures. *Neural computation*, vol. 7, p. 219–269, 1995.
- HAYKIN, S. *Neural networks and learning machines*. [S.l.]: Prentice Hall, 2009.
- HINTON, G. Connectionist learning procedures. *Artificial intelligence*, Elsevier, vol. 40, n. 1-3, p. 185–234, 1989.

- JIN, Y.; SENDHOFF, B. Pareto-based multiobjective machine learning: An overview and case studies. *IEEE Transactions on Systems Science and Cybernetics*, Vol. 39, n. 3, p. 373, 2009.
- KOKSHENEV, I.; BRAGA, A. P. An efficient multi-objective learning algorithm for rbf neural network. *Neurocomputing*, v. 37, n. 16-18, p. 2799–2808, 2010.
- LI, J.; KUO, C. A dual graph approach to 3d triangular mesh compression. *Image Processing ICIP*, vol. 2, p. 891–894, oct 1998.
- MEDEIROS, T. H.; TAKAHASHI, H. C. R.; BRAGA, A. A incorporação do conhecimento prévio na tomada de decisão do aprendizado multiobjetivo. *Congresso Brasileiro de Redes Neurais - Inteligência Computacional*, vol. 9, p. 25–28, 2009.
- REED, R. Pruning algorithms: A survey. v. 4, n. 5, p. 740–746, 1993.
- SUYKENS, J.; VANDEWALLE, J. Least squares support vector machine classifiers. *Neural processing letters*, Springer, v. 9, n. 3, p. 293–300, 1999.
- TEIXEIRA, R. A. et al. Improving generalization of mlps with multi-objective optimization. *Neurocomputing*, Vol. 35, n. 1, p. 189–194, 2000.
- VAPNIK, V. N. *The nature of statistical learning theory*. [S.l.]: Springer Verlag New York, 1995.
- VAPNIK, V. N. *Statistical Learning Theory*. [S.l.: s.n.], 1998.
- ZHANG, H.; HE, X. On simultaneous straight-line grid embedding of a planar graph and its dual. *Information Processing Letters*, vol. 99, n. 1, p. 1–6, 2006.
- ZHANG, W.; KING, I. A study of the relationship between support vector machine and gabriel graph. *Neural Networks, IJCNN*, vol. 1, p. 239–244, 2002. Proceedings of the 2002 International Joint Conference.