

TESE DE DOUTORADO Nº 223

**CLASSIFICADOR POR ARESTAS DE SUPORTE (CLAS): MÉTODOS DE
APRENDIZADO BASEADOS EM GRAFOS DE GABRIEL**

Luiz Carlos Bambirra Torres

DATA DA DEFESA: 25/02/2016

Universidade Federal de Minas Gerais

Escola de Engenharia

Programa de Pós-Graduação em Engenharia Elétrica

**CLASSIFICADOR POR ARESTAS DE SUPORTE (CLAS):
MÉTODOS DE APRENDIZADO BASEADOS EM GRAFOS DE
GABRIEL**

Luiz Carlos Bambirra Torres

Tese de Doutorado 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 Doutor em Engenharia Elétrica.

Orientador: Prof. Antônio de Pádua Braga

Coorientador: Prof. Cristiano Leite de Castro

Belo Horizonte - MG

Fevereiro de 2016

T693c

Torres, Luiz Carlos Bambilra.

Classificador por arestas de suporte (CLAS) [manuscrito] : métodos de aprendizado baseados em Grafos de Gabriel / Luiz Carlos Bambilra Torres. - 2016.

xvi, 88 f., enc.: il.

Orientador: Antônio de Pádua Braga.

Coorientador: Cristiano Leite de Castro.

Tese (doutorado) Universidade Federal de Minas Gerais, Escola de Engenharia.

Bibliografia: f. 81-88.

1. Engenharia elétrica - Teses. 2. Teoria dos grafos - Teses.
3. Processo decisório - Teses. I. Braga, Antônio de Pádua. II. Castro, Cristiano Leite de. III. Universidade Federal de Minas Gerais. Escola de Engenharia. IV. Título.

CDU: 621.3(043)

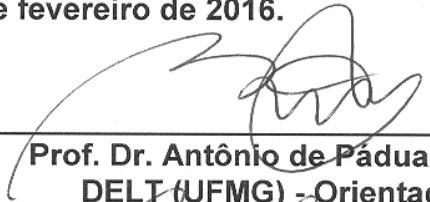
**"Classificador por Arestas de Suporte (clas): Métodos de
Aprendizado Baseados em Grafos de Gabriel"**

Luiz Carlos Bambirra Torres

Tese de Doutorado 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 Doutor em Engenharia Elétrica.

Aprovada em 25 de fevereiro de 2016.


Por:



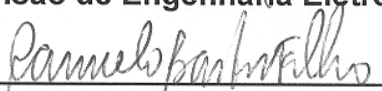
Prof. Dr. Antônio de Pádua Braga
DELT (UFMG) - Orientador



Prof. Dr. Cristiano Leite de Castro
DEE (UFMG) - Coorientador



Prof. Dr. Takashi Yoneyama
Divisão de Engenharia Eletrônica (ITA)



Prof. Dr. Carmelo Bastos Fiho
Departamento de Sistemas Computacionais (UPE)



Prof. Dr. Renato Cardoso Mesquita
DEE (UFMG)



Prof. Dr. André Paim Lemos
DELT (UFMG)

Dedico esta tese aos meus pais e ao meu irmão.

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 aponho sempre que eu preciso.

Ao Prof. Antônio de Pádua Braga, por ter me recebido no LITC, pelos ensinamentos, orientação, discussões, oportunidades 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, disponibilidade e dedicação na qual me ajudou a conduzir esse trabalho, muito obrigado.

Ao meu irmão Fernando e meu primo Wesley por estarem sempre presentes.

Aos colegas do LITC, pelo apoio e amizade.

A CAPES pelo apoio e suporte financeiro.

*O que sabemos é uma gota; o
que ignoramos é um oceano.*

Isaac Newton

Resumo

Este trabalho apresenta uma metodologia direcionada para problemas de classificação de padrões. O objetivo é projetar classificadores de margem larga, onde as informações necessárias para o projeto do classificador são obtidas a partir da estrutura geométrica dos dados. Através do grafo de Gabriel, o conjunto de dados é transformado em um grafo planar, onde as arestas deste grafo que possuem vértices com rótulos de classes distintas coincidem com as amostras que estão na margem de separação entre as classes. Estas arestas são denominadas arestas de suporte e formam a base para o desenvolvimento de uma família de métodos, tais como um decisor para o aprendizado multiobjetivo de redes neurais; uma estratégia para seleção de parâmetros em redes neurais RBF; por fim, a concepção de novos classificadores de margem larga. Resultados com *benchmarks* conhecidos na literatura mostram que essas abordagens maximizam a margem e aumentam a capacidade de generalização dos classificadores.

Abstract

This work presents a methodology directed to pattern classification problems. The goal is to design large margin classifiers where the information necessary is obtained from the geometric structure of data. Through the Gabriel graph, the data set is turned into a planar graph, where the edges with vertices of distinct labels corresponds to the samples which are on the margin of separation between the classes. These edge set is named as support edges and forms the basis for the development of a family of methods, such as a decision-maker for multi-objective learning of neural networks; a strategy for selecting parameters in RBF neural networks. Finally, the design of new large margin classifiers. Results with benchmarks known in the literature show that our approaches maximize the margin and increase the classifier generalization ability.

Publicações

Durante o desenvolvimento desta tese, os seguintes trabalhos foram publicados:

- TORRES, L.C.B.; Castro, C.L.; COELHO, F. ; SILL TORRES, F. ; BRAGA, A.P.. **Distance-based large margin classifier suitable for integrated circuit implementation.** Electronics Letters, v. 51, p. 1967-1969, 2015.
- TORRES, L. C. B.; Castro, C.L. ; BRAGA, A. P.. **Gabriel Graph for Dataset Structure and Large Margin Classification: A Bayesian Approach.** In: Proceedings of the European Symposium on Neural Networks, 2015, Bruges. ESANN 2015, 2015. p. 237-242.
- TORRES, LUIZ C.B.; Castro, Cristiano L. ; BRAGA, ANTONIO P.. **A parameterless mixture model for large margin classification.** In: 2015 International Joint Conference on Neural Networks (IJCNN), 2015, Killarney. p. 1-6.
- TORRES, L. C. B.; COELHO, Frederico ; CASTRO, C. L. ; BRAGA, A. P. . **A Graph of Gabriel Approach for Large Margin Classifiers.** In: LA-CCI - The Latin American Congress on Computational Intelligence Co-located with ARGENCON, 2014, San Carlos de Bariloche. Proceedings LA-CCI 2014, 2014. v. 1. p. 25-29.
- Torres, Luiz C. B.; Lemos, André P. ; Castro, Cristiano L. ; Braga, Antônio P.. **A Geometrical Approach for Parameter Selection of Radial Basis Functions Networks.** Lecture Notes in Computer Science. 1ed.: Springer International Publishing, 2014, v. 8681, p. 531-538.
- TORRES, L. C. B.; LEMOS, A. P. ; CASTRO, C. L. ; BRAGA, A. P.. **Projeto de redes RBF baseado na estrutura dos dados e em informações de margem.** In: 1st BRICS Countries Congress

(BRICS-CCI) and 11th Brazilian Congress (CBIC) on Computational Intelligence, 2013, Porto de Galinhas. Proceedings of (BRICS-CCI) & CBIC, 2013. p. 1-7.

- Torres, Luiz C. B.; Castro, Cristiano L. ; Braga, Antônio P.. **A Computational Geometry Approach for Pareto-Optimal Selection of Neural Networks.** Lecture Notes in Computer Science. 1ed.Berlin: Springer Berlin Heidelberg, 2012, v. 7553, p. 100-107.

Sumário

Lista de Símbolos	xi
Lista de Abreviaturas	xii
Lista de Figuras	xv
Lista de Tabelas	xvi
1 Introdução	1
1.1 Um decisor de margem larga para o método MOBJ	3
1.2 Uma Metodologia Para Encontrar Parâmetros do <i>Kernel</i> Gaussiano de Uma Rede Neural RBF	5
1.3 Classificador Geométrico de Margem Larga	5
1.4 Classificador Para Sistemas Embarcados	6
1.5 Organização do trabalho	6
2 Referencial Teórico	8
2.1 Revisão de métodos para controle da capacidade de generalização	8
2.1.1 Máquinas de Vetores de Suporte	8
2.1.1.1 Hiperplano de margem rígida	9
2.1.1.2 Hiperplano de margem flexível	11
2.1.2 Aprendizado Multiobjetivo de Redes Neurais	13
2.1.2.1 Minimização Estrutural do Risco	13
2.1.2.2 Método Multiobjetivo	13
2.1.2.3 Problema de Decisão Multiobjetivo	15
2.2 Revisão de Métodos Utilizados na Geometria Computacional . . .	16
2.2.1 Teoria dos Grafos	16
2.2.2 Diagrama de <i>Voronoi</i>	17
2.2.3 Triangulação de <i>Delaunay</i>	18

2.2.4	Grafo de Gabriel	19
3	Arestas de Suporte	21
3.1	Informação estrutural extraída a partir de um grafo planar	21
3.2	Lidando com bases de dados com sobreposição	22
3.2.1	Metodologia para eliminação de sobreposição entre classes	24
3.3	Algoritmo para concepção do conjunto de arestas de suporte e do conjunto de pontos médios	25
3.4	Maximização da Margem Utilizando Arestas de Suporte	27
3.4.1	Hiperplano de margem máxima	27
3.4.2	Maximização da Margem baseada no Grafo de Gabriel . .	29
4	Abordagens Propostas	32
4.1	Um decisor de margem larga para o método MOBJ	32
4.1.1	Decisor Proposto	34
4.1.2	Experimentos com base de dados sintéticas	37
4.1.3	Metodologia para experimento em base de dados reais . . .	44
4.1.4	Resultados	44
4.1.5	Teste de significância	46
4.2	Encontrando parâmetros do <i>Kernel</i> Gaussiano de uma rede RBF	48
4.2.1	Rede Neural RBF	50
4.2.2	Metodologia	51
4.2.3	Resultados	53
4.2.3.1	Experimento I	53
4.2.3.2	Experimento II	55
4.2.4	Teste de significância para o experimento I	56
4.2.5	Teste de significância para o experimento II	58
4.3	Classificador geométrico de margem larga	59
4.3.1	Extração de parâmetros	60
4.3.2	Metodologia para construção do classificador	62
4.3.3	Espaço de Verossimilhanças	64
4.3.4	Analogia com as SVMs	68
4.3.5	Resultados	69
4.3.6	Teste de significância	70

4.4	Classificador para Sistemas Embarcados	71
4.4.1	Combinação de Classificadores de Margem Larga	72
4.4.2	Mistura Hierárquica de Especialistas	72
4.4.3	Resultados	75
5	Conclusões e Propostas de Continuidade	77
5.1	Propostas de Continuidade	79
	Referências	88

Lista de Símbolos

\mathcal{PM}	Conjunto de pontos médios
\mathcal{AS}	Conjunto de arestas de suporte.
\mathcal{PO}	Conjunto Pareto-Ótimo
\mathcal{ES}	Conjunto de hiperesferas
\mathcal{N}	Conjunto de distribuições normais multivariadas
p	Margem de separação
\mathbf{x}	Vetor
\mathbf{w}	Vetor de pesos
A	Matriz
I_d	Matriz identidade de tamanho d
$\ \cdot\ $	Norma Euclidiana
d	Número de dimensões
$sign(\cdot)$	Função Sinal
$\Phi(\cdot)$	Função de mapeamento
Σ	Matriz de Covariância
μ	Média
$N(\mu, \Sigma)$	Distribuição normal multivariada
σ^2	Variância
$\Delta \ \mathbf{w}\ $	Distância Euclidiana entre a norma de cada solução
$\mathfrak{B}(\cdot)$	Bissetriz
$\mathfrak{D}(\cdot)$	Diagrama de Voronoi de um conjunto de dados
$\delta(\cdot)$	Fornece a distância Euclidiana entre dois pontos
G	Grafo
\tilde{G}	Grafo Planar
\ddot{G}	Grafo de Gabriel
\mathcal{H}	Hiperplano
$\varphi(\cdot)$	Função Gaussiana
$\mathcal{P}(\mu, \Sigma)$	Função de densidade de uma distribuição normal multivariada
$f_{\mathcal{P}}(\mathbf{x} \theta)$	Função de mistura de densidades

Lista de Abreviaturas

AS	Aresta de Suporte
AUC	<i>Area Under the Curve</i>
CD	<i>Critical Difference</i>
FDM	Função de Distribuição Normal Multivariada
GG	Grafo de Gabriel
LS-SVM	Least Squares Support Vector Machine
MOBJ	Algoritmo Multiobjetivo
MER	Minimização Estrutural do Risco
MLP	<i>Multilayer Perceptron</i>
MMG	Modelo de Mistura Gaussiana
PQ	Programação Quadrática
RBF	<i>Radial Basis Function</i>
RNAS	Redes Neurais Artificiais
SVM	<i>Support Vector Machine</i>
TSQ	Total da Soma dos Quadrados
TD	Triangulação de Delaunay
VC	Vapnik-Chervonenkis

Lista de Figuras

1.1	Separador Geométrico.	2
1.2	(a) Conjunto de dados no espaço de entrada. (b) Conjunto de padrões de entrada modelado com o grafo de Gabriel. (c) Arestas de suporte.	3
1.3	Conjunto Pareto-ótimo e os diferentes mapeamentos das soluções.	4
1.4	(a) Espaço de verossimilhanças. (b) Solução final dada pelo classificador.	6
2.1	Margem máxima de separação entre as classes. Os padrões que intersectam a margem são chamados de vetores de suporte. . . .	11
2.2	Exemplo em que aparecem padrões dentro da margem.	12
2.3	Conjunto Pareto-ótimo resultante do método MOBJ, onde não é possível minimizar os funcionais MSE e $\ \mathbf{W}\ $ ao mesmo tempo.	15
2.4	Forma geométrica de um Grafo.	17
2.5	(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>	18
2.6	Construção do grafo de Gabriel.	20
3.1	Arestas de suporte \mathcal{AS} indicadas pelos quadrados.	22
3.2	Base de dados com sobreposição entre as classes.	23
3.3	Conjunto de dados após a remoção da sobreposição.	25
3.4	Cada aresta de suporte possui um ponto médio indicado por um asterisco.	26

LISTA DE FIGURAS

3.5	Hiperplano encontrado a partir do método SVM.	28
3.6	Hiperplano gerado através dos dois pontos mais próximos de dois conjuntos de fecho convexo.	29
3.7	Comparação entre o classificador Geométrico gerado a partir da aresta de suporte e o classificador obtido pela SVM.	31
4.1	(a) Conjunto de dados do <i>benchmark Corners</i> . (b) Conjunto de soluções. (c) Solução escolhida vista com as hiperesferas. (d) Solução final.	38
4.2	(a) Conjunto de dados do <i>benchmark Full Moons</i> . (b) Conjunto de soluções. (c) Solução escolhida vista com as hiperesferas. (d) Solução final.	39
4.3	(a) Conjunto de dados do <i>benchmark Cluster</i> . (b) Conjunto de soluções. (c) Solução escolhida vista com as hiperesferas. (d) Solução final.	40
4.4	(a) Conjunto de dados do <i>benchmark Half Kernel</i> . (b) Conjunto de soluções. (c) Solução escolhida vista com as hiperesferas. (d) Solução final.	41
4.5	(a) Conjunto Pareto do <i>benchmark Corners</i> . (b) Conjunto Pareto do <i>benchmark Full Moons</i> . (c) Conjunto Pareto do <i>benchmark Cluster</i> . (d) Conjunto Pareto do <i>benchmark Half Kernel</i>	42
4.6	(a) SVM aplicada ao <i>benchmark Corners</i> . (b) SVM aplicada ao <i>benchmark Full Moons</i> . (c) SVM aplicada ao <i>benchmark Clusters</i> . (d) SVM aplicada ao <i>benchmark Half Kernel</i>	43
4.7	Diagrama de diferença crítica do teste <i>post-hoc</i> . Os métodos representados pelas linhas horizontais pontilhadas são estatisticamente diferentes do método MOBJ-clas.	48
4.8	Topologia da Rede RBF.	49
4.9	Centros e raios das funções de ativação dos neurônios da camada intermediária.	52
4.10	Metodologia proposta aplicada ao <i>benchmark Two Moons</i>	53

4.11	Diagrama de diferença crítica do teste <i>post-hoc</i> . Os métodos representados pelas linhas horizontais pontilhadas são estatisticamente diferentes do método RBF-clas.	57
4.12	Diagrama de diferença crítica do teste <i>post-hoc</i> . Os métodos representados pelas linhas horizontais pontilhadas são estatisticamente diferentes do método RBF-clas	58
4.13	Diagrama de diferença crítica do teste <i>post-hoc</i> . Os métodos representados pelas linhas horizontais pontilhadas são estatisticamente diferentes do método RBF-clas. Neste caso, todos os métodos são equivalente.	59
4.14	Superfície de separação de duas gaussianas, onde o hiperplano intersecta o ponto médio entre um par de vértices de uma aresta de suporte.	60
4.15	(a) Mistura de densidades de $f_{\mathcal{P}}(\mathbf{x} \theta_{\mathcal{A}})$. (b) Mistura de densidades de $f_{\mathcal{P}}(\mathbf{x} \theta_{\mathcal{B}})$	63
4.16	Hiperplano separador obtido através do nosso método.	64
4.17	(a) Projeção no espaço de verossimilhanças de duas distribuições Gaussianas. (b) Hiperplano separador no espaço de entrada maximizando a margem entre as duas classes.	65
4.18	(a) Projeção no espaço de verossimilhanças utilizando σ aleatório (b) Hiperplano separador no espaço de entrada fazendo σ aleatório.	66
4.19	(a) Projeção no espaço de verossimilhanças utilizando nossa abordagem (b) Hiperplano separador resultante da nossa abordagem.	67
4.20	Diagrama de diferença crítica do teste <i>post-hoc</i>	71
4.21	(a) Conjunto de padrões no espaço de entrada. (b) Conjunto de padrões no espaço de entrada modelado com o Grafo de Gabriel. (c) Conjunto de arestas de suporte. (d) Conjunto de pontos médios representado por asteriscos. (e) Classificadores gerados através de cada aresta de suporte. (f) Superfície de separação gerada pelo classificador.	73
4.22	Estrutura do modelo de Mistura Hierárquica de Especialistas.	74

Lista de Tabelas

4.1	Resultados do decisor para o método MOBJ: média da AUC e características das bases de dados. Os melhores valores encontram-se em negrito.	46
4.2	Representação dos nomes utilizados em cada metodologia para seleção e número K de centros para serem utilizados no projeto da rede RBF.	53
4.3	Resultados para o método RBF-clas: média da AUC e características das bases de dados. Foi utilizada a Equação (4.25) para encontrar o valor do σ	55
4.4	Resultados para o método RBF-clas: média da AUC e características das bases de dados. Foi utilizada a metodologia para encontrar o σ de acordo com a Equação (4.26).	55
4.5	Resultados do método RBF-clas: média da AUC e características das bases de dados.	56
4.6	Resultados do classificador Gaussiano: média da AUC e características das bases de dados.	70
4.7	Resultados: AUC média e desvio padrão	76

Capítulo 1

Introdução

Classificadores de margem larga se tornaram populares nas últimas duas décadas, logo após a publicação das *Support Vector Machines* (SVM) (Boser *et al.*, 1992). A ideia de maximização da margem na região de separação entre as classes também tem sido explorada por outras abordagens, tais como *Boosting* (Freund & Schapire, 1996; Freund *et al.*, 1996; Schapire, 1990), Modelos de Mistura Gaussiana (Sha & Saul, 2006) e *Direct Parallel Perceptrons* (Fernandez-Delgado *et al.*, 2011). Alguns classificadores geométricos de margem larga também têm sido encontrados na literatura, como em Cevikalp *et al.* (2010), onde é projetado um classificador de margem larga baseado em *affine hulls*. Nos trabalhos de Peng & Wang (2012); Peng & Xu (2013) também é explorado o conceito de *affine hulls* para encontrar o classificador de margem larga. Já no trabalho de Cevikalp & Triggs (2013), em vez de *affine hulls* são utilizados *Hyperdisks* para encontrar o classificador. Embora estes métodos possam se diferenciar na maneira como a margem é maximizada, a ideia geral é selecionar um classificador que se distancie igualmente das amostras da margem de separação, como é esquematicamente mostrado na Figura 1.1. Para problemas linearmente separáveis, a margem da SVM pode ser controlada pela norma dos pesos $\|\mathbf{w}\|$, que conduz a um problema de Programação Quadrática (PQ), onde os vetores de suporte (VS) representam a solução do problema de PQ.

Os vetores de suporte podem ser considerados propriedades geométricas de um problema de classificação, de modo que a solução de margem máxima pode ser concebida explorando as propriedades estatísticas e geométricas do conjunto

de treinamento. Esta afirmação pode ser corroborada no trabalho de [Bennett & Bredensteiner \(2000\)](#), onde é demonstrado que a solução da SVM pode ser aproximada geometricamente, encontrando-se o separador de margem máxima em relação a dois conjuntos de fecho convexo. A partir desse princípio, este trabalho visa explorar a estrutura geométrica dos dados, a fim de obter informações que possam ser utilizadas pelos métodos de classificação. Tais informações podem ser transformadas em parâmetros de ajustes, critérios de decisão e até mesmo um classificador propriamente dito.

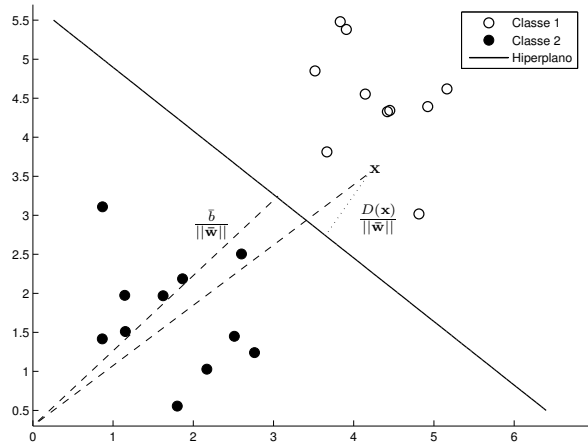


Figura 1.1: Separador Geométrico.

Ao modelar o conjunto de dados através de um grafo planar chamado grafo de Gabriel ([Gabriel & Sokal, 1969](#)) (vide Figuras 1.2(a) e 1.2(b)), é possível selecionar um subconjunto de vértices do grafo que são análogos aos vetores de suporte da SVM. Esses vértices formam uma ou mais arestas denominadas de Arestas de Suporte (AS) ([Torres et al., 2015a](#)). Essas arestas podem ser observadas na Figura 1.2(c). As Arestas de Suporte formam as bases teóricas que fundamentam as contribuições deste trabalho, que estão listadas abaixo:

- Um decisor de margem larga para o método multiobjetivo de treinamento de redes neurais (MOBJ) (baseado em [Torres et al. \(2012\)](#));
- Uma metodologia para encontrar os parâmetros do Kernel gaussiano de uma rede RBF (*Radial Basis Function*) ([Torres et al., 2014b](#));

1.1 Um decisor de margem larga para o método MOBJ

- Um classificador geométrico de margem larga (baseado em [Torres *et al.* \(2014a, 2015b\)](#));
- Um classificador para sistemas embarcados ([Torres *et al.*, 2015a](#)).

Nas Seções que se seguem deste Capítulo, são apresentadas resumidamente cada abordagem.

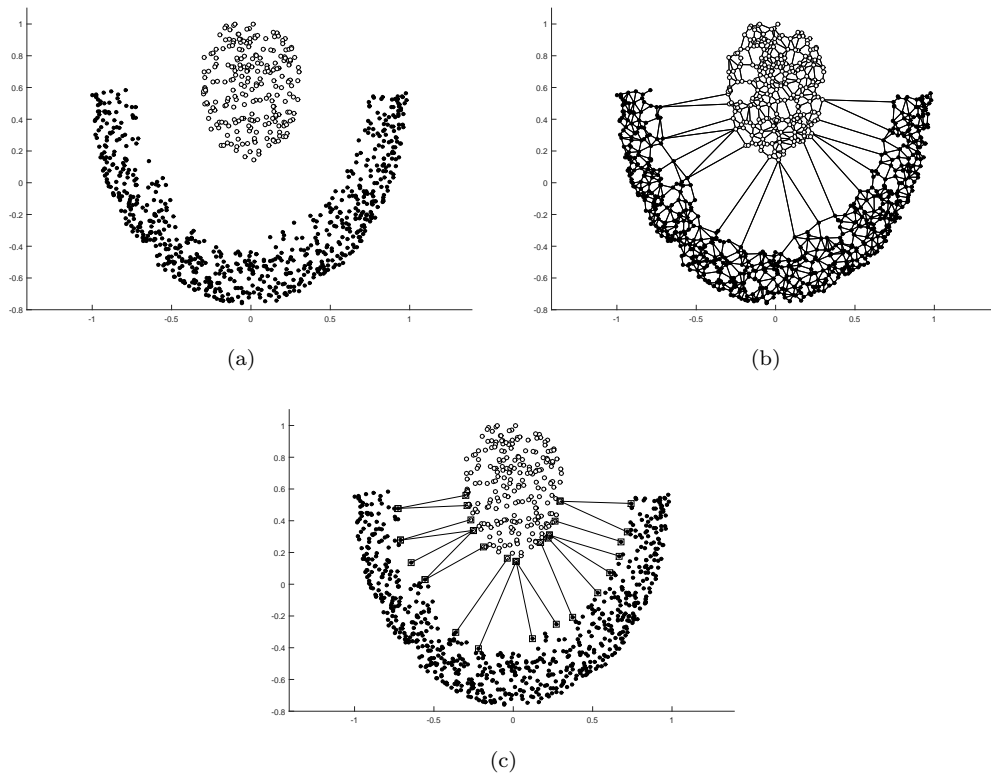


Figura 1.2: (a) Conjunto de dados no espaço de entrada. (b) Conjunto de padrões de entrada modelado com o grafo de Gabriel. (c) Arestas de suporte.

1.1 Um decisor de margem larga para o método MOBJ

O método multiobjetivo para o treinamento de redes neurais (MOBJ) gera como resultado um conjunto Pareto-Ótimo, \mathcal{PO} que corresponde aos melhores compro-

1.1 Um decisor de margem larga para o método MOBJ

missos entre os funcionais erro de treinamento e complexidade¹ (Teixeira *et al.*, 2000) (Vide Figura 1.3). A partir das arestas de suporte (AS), foi desenvolvido um método decisor que busca pela solução do conjunto \mathcal{PO} que possui a maior margem de separação entre as classes. Para encontrar a solução de maior distância (margem) entre as classes, foi proposta a inserção de novas amostras no espaço de entrada. As novas amostras foram inseridas nos pontos médios, definidos a partir da distância euclidiana entre os pares das AS. Este novo grupo de exemplos é chamado de conjunto de pontos médios \mathcal{PM} e a solução mais próxima deste conjunto é a escolhida pelo decisor.

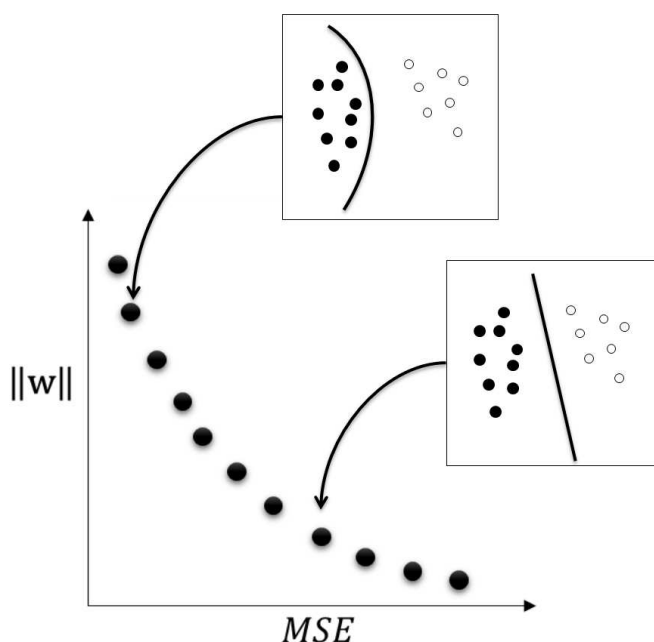


Figura 1.3: Conjunto Pareto-ótimo e os diferentes mapeamentos das soluções.

¹Norma Euclidiana dos pesos da rede.

1.2 Uma Metodologia Para Encontrar Parâmetros do *Kernel* Gaussiano de Uma Rede Neural RBF

Visto que o grafo de Gabriel permite encontrar a margem de separação entre as classes (conjunto \mathcal{AS}) sem o uso de parâmetros, o conjunto \mathcal{AS} foi usado no projeto de uma rede neural *Radial Basis Function* (RBF). O primeiro passo para construção da rede RBF é encontrar o número de neurônios da camada escondida, que correspondem à quantidade de funções base $\varphi(\cdot)$. Normalmente, $\varphi(\cdot)$ é descrita como uma função gaussiana $\varphi(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{c}_i\|^2}{2\sigma_i^2}\right)$. Portanto, é necessário definir os valores do centro \mathbf{c}_i e do raio σ_i para cada (i -ésimo) neurônio oculto. Utilizando-se os vetores geométricos, os valores ótimos de \mathbf{c}_i e σ_i são encontrados sem a necessidade de qualquer informação a priori.

Em síntese, os exemplos do conjunto \mathcal{AS} são assinalados como centros das funções base $\varphi(\cdot)$ da rede RBF. Além de não necessitar de parâmetros adicionais para a obtenção dos centros, o método conta com as coordenadas geométricas de cada exemplo da margem para encontrar o respectivo σ_i de cada neurônio.

1.3 Classificador Geométrico de Margem Larga

As informações fornecidas pela estrutura dos dados no projeto da rede neural RBF, descrita na Seção anterior, foram utilizadas para projetar um classificador geométrico. Nesta abordagem é utilizado o conjunto \mathcal{AS} , onde cada amostra representa a média μ de uma distribuição normal multivariada $N(\mu, \Sigma)$, e a complexidade do classificador é encontrada de modo determinístico e um novo espaço chamado de “verossimilhanças”, que pode ser observado na Figura 1.4(a). Para a classificação de uma dada amostra \mathbf{x} , é utilizado todo o conjunto \mathcal{AS} como componentes de duas funções de misturas de densidades $p(\mathbf{x}, \theta_1|C_1)$ e $p(\mathbf{x}, \theta_2|C_2)$. Uma para a classe positiva (+1) e a outra para negativa (-1). Por fim, o exemplo \mathbf{x} é rotulado de acordo com a regra de decisão de Bayes (Duda *et al.*, 2012). A Figura 1.4(b) mostra a separação proporcionada por este classificador.

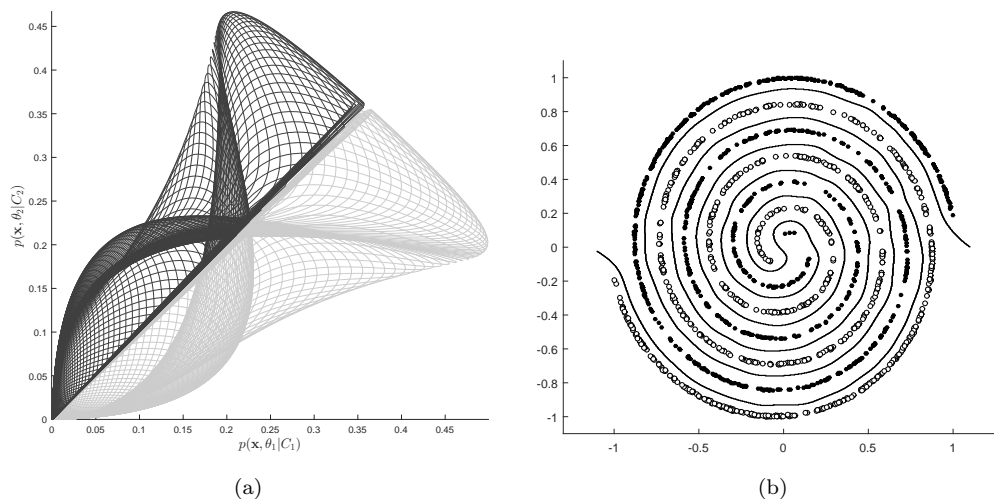


Figura 1.4: (a) Espaço de verossimilhanças. (b) Solução final dada pelo classificador.

1.4 Classificador Para Sistemas Embarcados

Inspirado nos métodos apresentados nas Seções anteriores, foi projetado um novo método de aprendizagem para problemas de classificação que é adequado para a implementação em circuitos integrados. Este método de classificação, que é estatisticamente equivalente a muitas abordagens conhecidas como estado da arte, baseia-se numa descrição estrutural do conjunto de treinamento representado pelo grafo de Gabriel. Assim, a função de classificação final é composta por uma mistura hierárquica de especialistas, onde cada especialista é representado por um classificador de margem larga local. Dessa forma, a abordagem pode ser facilmente embarcada em *Hardware*, uma vez que o método é projetado com base na função de distância Euclidiana, que é facilmente construída através de operações matemáticas básicas, e ainda pode ser otimizada utilizando o paralelismo proporcionado por FPGAs e GPUs.

1.5 Organização do trabalho

O restante do trabalho está organizado da seguinte forma:

No Capítulo 2 são apresentados os principais métodos do aprendizado de máquina e também são definidas as bases sobre as quais a teoria do treinamento multiobjetivo de redes neurais é construída. Nesse Capítulo também são mostrados alguns conceitos de métodos utilizados na geometria computacional. Já no Capítulo 3, é mostrado como o conjunto de arestas de suporte \mathcal{AS} e o conjunto de pontos médios \mathcal{PM} são encontrados, também é descrito como a sobreposição entre as classes é encontrada e eliminada. No final do Capítulo 3 é mostrada matematicamente a similaridade dos vértices das arestas de suporte com os VS da SVM. Por sua vez, o Capítulo 4 apresenta as contribuições da tese, onde são apresentados a metodologia e os resultados para cada abordagem. Finalmente, as conclusões e propostas de continuidade do trabalho são apresentadas no Capítulo 5.

Capítulo 2

Referencial Teórico

2.1 Revisão de métodos para controle da capacidade de generalização

Neste capítulo são apresentados os principais princípios das máquinas de vetores de suporte e também são definidas as bases sobre as quais a teoria do treinamento multiobjetivo de redes neurais é construída. Por fim, são mostrados alguns conceitos de métodos utilizados na geometria computacional.

2.1.1 Máquinas de Vetores de Suporte

As Máquinas de Vetores de Suporte ou *Support Vector Machines* (SVM), são utilizadas para solucionar problemas de classificação de padrões e regressão. Foram propostas inicialmente por Vapnik (1995, 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

2.1 Revisão de métodos para controle da capacidade de generalização

dimensão, chamado de espaço de características, em que um problema de natureza não-linear pode tornar-se linearmente separável. Nesse espaço, um hiperplano ótimo é construído para separar os dados em duas classes. Quando os dados de treinamento são separáveis, o hiperplano ótimo no espaço de características apresenta uma margem de separação máxima. 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 (PQ), 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, fazendo com que o problema de dimensionalidade deixe de ser uma dificuldade. Logo, o número de parâmetros ajustados não dependerá mais da dimensão do espaço a que pertencem os dados de treinamento.

2.1.1.1 Hiperplano de margem rígida

Considere-se um conjunto de dados $T = \{\mathbf{x}_i, \mathbf{y}_i \mid i = 1 \dots n\}$, onde \mathbf{x}_i é o i -ésimo dado de entrada, onde $\mathbf{x} \in \mathbb{R}^d$, e \mathbf{y}_i representa o i -ésimo elemento de saída desejável. Assume-se que todos os elementos $\mathbf{y} = -1$ representam a classe 1 e o subconjunto $\mathbf{y} = +1$ 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 $b \in \mathbb{R}^d$ 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. A Figura 2.1 ilustra a superfície de um hiperplano ótimo separando um espaço \mathbb{R}^2 em duas partes. 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):

2.1 Revisão de métodos para controle da capacidade de generalização

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

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. Fazendo b^* e \mathbf{w}^* definidos como parâmetros ótimos para o hiperplano de margem máxima (hiperplano ótimo \mathcal{H}^*), a Equação (2.1) pode ser reescrita por uma função discriminante

$$g(\mathbf{x}) = \mathbf{w}^{*T} \mathbf{x} + b^*, \quad (2.3)$$

está função corresponde a uma medida algébrica de distância entre o padrão \mathbf{x} e o hiperplano ótimo \mathcal{H}^* . Esta medida também pode ser expressa como

$$\mathbf{x} = \mathbf{x}_p + r \frac{\mathbf{w}^*}{\|\mathbf{w}^*\|}, \quad (2.4)$$

onde \mathbf{x}_p é a projeção de \mathbf{x} em \mathcal{H}^* , tal que $g(\mathbf{x}_p) = 0$. O parâmetro r representa a distância algébrica, onde r sendo positivo significa que o padrão \mathbf{x} encontra-se no lado positivo do hiperplano, caso contrário, \mathbf{x} está no lado negativo. De acordo com Haykin (1999), segue que

$$g(\mathbf{x}) = \mathbf{w}^{*T} \mathbf{x} + b^* = r \|\mathbf{w}^*\|, \quad (2.5)$$

ou

$$r = \frac{g(\mathbf{x})}{\|\mathbf{w}^*\|}. \quad (2.6)$$

Definindo um vetor de suporte como \mathbf{x}^* e sua saída desejada $d = \pm 1$, a distância algébrica de um vetor de suporte para um hiperplano ótimo \mathcal{H}^* pode ser dada como

$$r = \frac{g(\mathbf{x}^*)}{\|\mathbf{w}^*\|} = \begin{cases} \frac{1}{\|\mathbf{w}^*\|} & \text{se } d^* = +1 \\ -\frac{1}{\|\mathbf{w}^*\|} & \text{se } d^* = -1 \end{cases}. \quad (2.7)$$

Denotando por p^* o valor de margem ótima de separação entre duas classes, verifica-se por intermédio da Equação (2.7) que

2.1 Revisão de métodos para controle da capacidade de generalização

$$p^* = 2r = \frac{2}{\|\mathbf{w}\|}. \quad (2.8)$$

A partir da Equação (2.8), é possível observar que maximizar a margem p^* significa minimizar a norma do vetor \mathbf{w} no espaço de características.

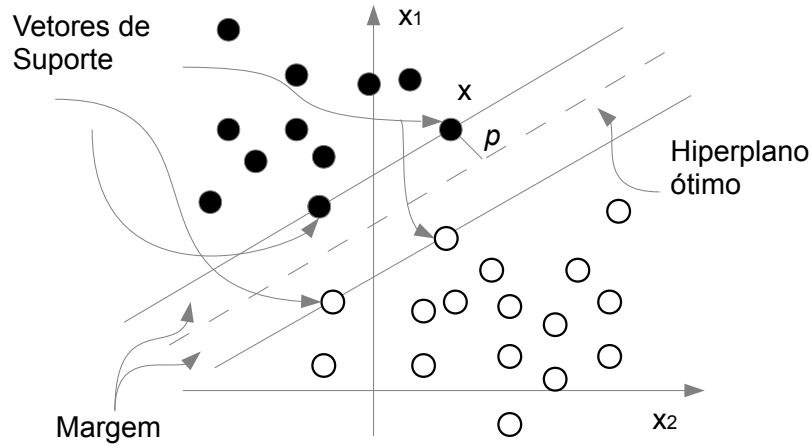


Figura 2.1: Margem máxima de separação entre as classes. Os padrões que intersectam a margem são chamados de vetores de suporte.

2.1.1.2 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. Segundo Haykin (2009), a definição do hiperplano de margem flexível é dada como

$$\mathbf{y}_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1 - \xi_i, \quad i = \{1, 2, \dots, n\}, \quad (2.9)$$

onde ξ são conhecidas como variáveis de folga e medem o grau de desvio de um padrão de entrada em relação ao hiperplano de separação ótimo. Para $0 < \xi \leq 1$

2.1 Revisão de métodos para controle da capacidade de generalização

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 (2.10).

$$\phi(\mathbf{w}, \xi) = \frac{1}{2} \mathbf{w}^T \mathbf{w} + \mathbf{C} \sum_{i=1}^n \xi_i \quad (2.10)$$

sujeito a,

$$\begin{cases} \mathbf{y}_i [\mathbf{w}^T \Phi(\mathbf{x}_i) + b] & \geq 1 - \xi_i, \\ \xi_i & \geq 0 \end{cases} \quad (2.11)$$

onde Φ é a função de mapeamento (kernel) e o parâmetro \mathbf{C} controla o impasse (*tradeoff*) entre a complexidade da máquina e o número de pontos que podem infringir a restrição imposta na Equação (2.11).

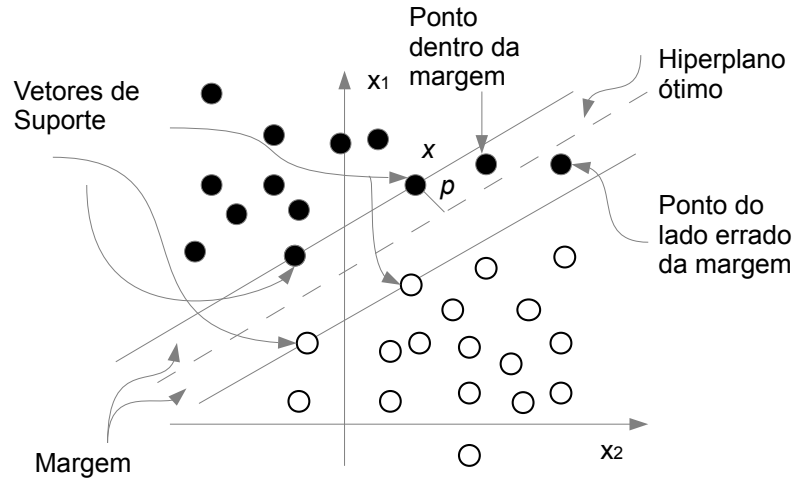


Figura 2.2: Exemplo em que aparecem padrões dentro da margem.

2.1 Revisão de métodos para controle da capacidade de generalização

2.1.2 Aprendizado Multiobjetivo de Redes Neurais

2.1.2.1 Minimização Estrutural do Risco

A teoria do aprendizado estatístico (Vapnik, 1995), através do princípio indutivo de minimização estrutural do risco (MER), estabelece condições matemáticas que permitem definir, com probabilidade de pelo menos $(\frac{1-\epsilon}{n})$, um limite superior para o risco esperado (ou erro de generalização) de uma máquina de aprendizagem (Vapnik, 1995, 1998)

$$\mathbf{R} \leq \mathbf{R}_{emp} + \sqrt{\frac{h \left(\ln \frac{2n}{h} + 1 \right) - \ln \left(\frac{\epsilon}{4} \right)}{n}}, \quad (2.12)$$

onde ϵ é um parâmetro livre e n é o tamanho do conjunto de treinamento.

Analisando a Equação (2.12), observa-se que o limite superior de \mathbf{R} é uma função inversa de n e direta de dois termos: o primeiro, denominado risco empírico (\mathbf{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 \mathbf{R} pode então ser obtida através do aumento do número de exemplos n e/ou do decréscimo simultâneo de \mathbf{R}_{emp} e Ω .

2.1.2.2 Método 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, ou seja, as funções são conflitantes em uma determinada faixa. Por exemplo, pode ocorrer o crescimento de uma função que estima o erro da rede ao mesmo tempo que uma função de complexidade da rede é minimizada, ou o contrário. O MER pode ser formalmente definido como

$$(MER) : \min \begin{cases} \mathbf{J}_1 = \mathbf{R}_{emp} \\ \mathbf{J}_2 = \Omega \end{cases} \quad (2.13)$$

onde \mathbf{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

2.1 Revisão de métodos para controle da capacidade de generalização

(Ω) comumente usada é a norma euclidiana do vetor de pesos da rede (Costa *et al.*, 2007; Hinton, 1989; Teixeira *et al.*, 2000).

Dentre os algoritmos de treinamento multiobjetivo para redes MLP, destaca-se o método MOBJ que foi projetado para resolver o problema de MER descrito na Equação (2.13) (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 $\mathcal{S} = \{\mathbf{x}_i, \mathbf{y}_i \mid i = 1 \cdots n\}$, a Equação (2.14) fornece a formulação biobjetivo, proposta em Teixeira *et al.* (2000), para o aprendizado de redes MLP,

$$\min \begin{cases} \mathbf{J}_1(\mathbf{w}) = \sum_{i=1}^N \left(\mathbf{y}_i - \hat{f}(\mathbf{x}_i, \mathbf{w}) \right)^2 \\ \mathbf{J}_2(\mathbf{w}) = \|\mathbf{w}\| \end{cases} \quad (2.14)$$

onde $\hat{f}(\mathbf{x}_i, \mathbf{w})$ é a saída estimada pela rede para o i -ésimo padrão de entrada, \mathbf{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 dominadas², denominado conjunto Pareto-ótimo \mathcal{PO} , onde a distância euclidiana entre a norma de cada solução é dada por $\Delta\|\mathbf{w}\|$. 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 piora do outro. O conjunto \mathcal{PO} possui dois conjuntos de soluções que podem ser classificadas como sub-ajustadas e super-ajustadas, como é mostrado na Figura 2.3. 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 a vantagem de que 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 específico para o erro e a norma. Logo, existirá no conjunto \mathcal{PO} uma solução com complexidade efetiva adequada, ou seja, dentre

²Em um problema de otimização multiobjetivo, uma solução é dita ser não dominada, se não existe nenhuma solução com desempenho superior em um objetivo não sendo pior nos outros objetivos.

2.1 Revisão de métodos para controle da capacidade de generalização

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.

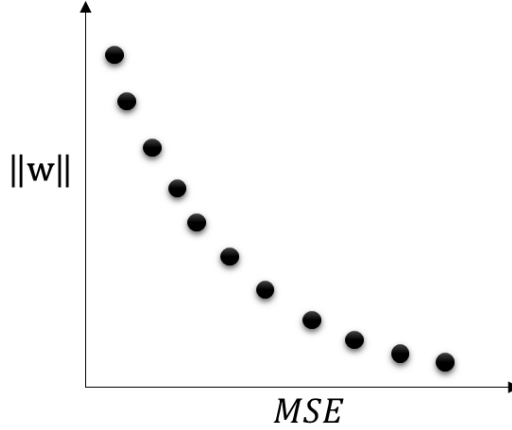


Figura 2.3: Conjunto Pareto-ótimo resultante do método MOBJ, onde não é possível minimizar os funcionais MSE e $\|\mathbf{W}\|$ ao mesmo tempo.

Na ausência de qualquer informação *a priori* referente aos objetivos $\mathbf{J}_1(\mathbf{w})$ e $\mathbf{J}_2(\mathbf{w})$, todas as soluções pertencentes ao conjunto \mathcal{PO} são candidatas à solução do problema descrito na Equação (2.14). Uma etapa de decisão 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 \mathbf{R} (ou erro de generalização).

2.1.2.3 Problema de Decisão Multiobjetivo

O problema geral de decisão multiobjetivo deve obedecer ao esquema dado pela Equação (2.15) (Medeiros *et al.*, 2009),

$$\mathbf{w}^* = \arg \max_{\mathbf{w} \in \mathcal{W}} f_e \quad (2.15)$$

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

2.2 Revisão de Métodos Utilizados na Geometria Computacional

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 \mathcal{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 \mathcal{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.

2.2 Revisão de Métodos Utilizados na Geometria Computacional

2.2.1 Teoria dos Grafos

As definições aqui apresentadas foram extraídas de Barroso (2007); Christofides (1975). Um grafo $G(\mathcal{V}, \mathcal{E})$ é uma estrutura matemática constituída de dois conjuntos: um finito e não vazio, de n vértices \mathcal{V} , e outro \mathcal{E} , de m arestas, que são pares não ordenados de elementos de \mathcal{V} . Na representação geométrica de um grafo, os pontos estão associados aos vértices e as arestas correspondem a linhas arbitrárias que ligam os vértices que as definem. A Figura 2.4 mostra a representação geométrica do grafo $G(\mathcal{V}, \mathcal{E})$, onde $\mathcal{V} = \{1, 2, 3, 4\}$ e

2.2 Revisão de Métodos Utilizados na Geometria Computacional

$\mathcal{E} = \{(1, 3), (1, 4), (2, 3), (3, 4)\}$. Diz-se que dois vértices de um grafo são adjacentes quando definem uma aresta. Um grafo é caracterizado como planar quando pelo menos uma representação gráfica no plano pode ser obtida, tal que as arestas não-se interceptam.

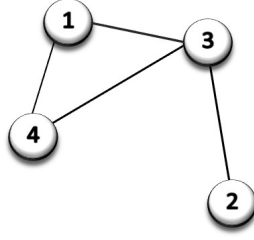


Figura 2.4: Forma geométrica de um Grafo.

2.2.2 Diagrama de *Voronoi*

O Diagrama de *Voronoi* representa a decomposição de um espaço em regiões de acordo com a distância entre determinados pontos (Aurenhammer & Klein, 1990). Dados dois pontos $\mathbf{p}, \mathbf{q} \in \mathcal{S}$, onde \mathcal{S} representa um conjunto de pontos em um plano, a bissetriz $\mathfrak{B}(\mathbf{p}, \mathbf{q})$ corresponde a uma reta perpendicular que atravessa o centro do segmento de reta $\overline{\mathbf{p}\mathbf{q}}$; $\delta(\cdot)$ é o operador que fornece a distância euclidiana entre dois pontos (vetores). O diagrama de *Voronoi* $\mathfrak{D}(\mathcal{S})$ pode ser considerado como a divisão do hiperplano em m polígonos convexos \mathfrak{L} (Berg et al., 2008; Figueiredo, 1991). A Figura 2.5(a) ilustra um exemplo de um diagrama de *Voronoi*. Um polígono $\mathfrak{L}(\mathbf{x}_i)$ é chamado de polígono de *Voronoi* relativo a \mathbf{x}_i e é formado através da interseção do conjunto das bissetrizes $\mathfrak{B}(\mathbf{x}_i, \mathfrak{L}(\mathbf{x}_i))$. Um ponto $\mathbf{p} \in \mathcal{S}$ pertence a $\mathfrak{L}(\mathbf{x}_i)$ se, e somente se, a seguinte desigualdade for satisfeita:

$$\delta(\mathbf{p}, \mathbf{x}_i) \leq \delta(\mathbf{p}, \mathbf{x}_j), \quad \mathbf{x}_i, \mathbf{x}_j \in \mathcal{S}, \quad \forall j \neq i. \quad (2.16)$$

De uma forma geral, uma região de *Voronoi* V_i associada a um sítio $\mathbf{v}_i \in \mathcal{S}$ é definida como

$$V_i = \{\mathbf{v} \in \mathbb{R}^d | \forall \mathbf{v}_j \in \mathcal{S}, \delta(\mathbf{v}, \mathbf{v}_i) \leq \delta(\mathbf{v}, \mathbf{v}_j)\}. \quad (2.17)$$

2.2.3 Triangulação de *Delaunay*

No diagrama de Voronoi $\mathfrak{D}(\mathcal{S})$, cada elemento $\mathbf{v} \in \mathcal{S}$ está associado a um poliedro de $\mathfrak{D}(\mathcal{S})$. O grafo dual de $\mathfrak{D}(\mathcal{S})$ tem por vértices os elementos de \mathcal{S} e por arestas os pares de elementos de \mathcal{S} , cujos polígonos de $\mathfrak{D}(\mathcal{S})$ são vizinhos (Li & Kuo, 1998; Zhang & He, 2006). Tal diagrama resultante é chamado de Diagrama de *Delaunay* (Berg *et al.*, 2008; Figueiredo, 1991), conforme ilustrado na Figura 2.5(b). Seja $K \in \mathbb{R}^d$ um conjunto de pontos no espaço Euclidiano, a TD é uma triangulação $\mathcal{TD}(K)$ tal que nenhum ponto em K esta contido em uma hiperesfera de um simplexo em $\mathcal{TD}(K)$. Em duas dimensões, a TD pode ser modelada através de um grafo planar $\tilde{G}(\mathcal{V}, \mathcal{E})$, composto por um conjunto de vértices \mathcal{V} e um conjunto de arestas \mathcal{E} não ordenadas, como pode ser visto na Figura 2.5(c). Uma aresta $(\mathbf{v}_i, \mathbf{v}_j) \in G$ é definida se, e somente se, existir uma hiperesfera contendo o par $(\mathbf{v}_i, \mathbf{v}_j)$ e se todos os outros vértices de \mathcal{V} forem exteriores a esta hiperesfera. Essa característica é ilustrada na Figura 2.5(d).

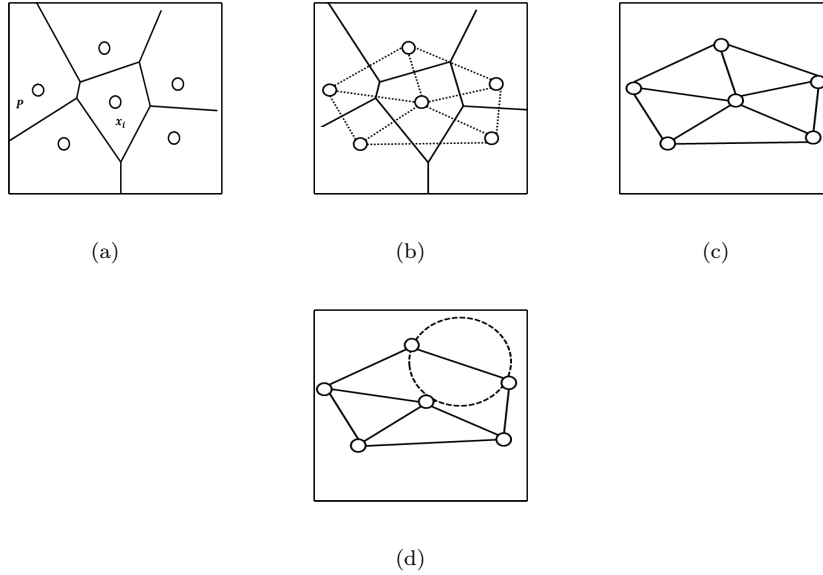


Figura 2.5: (a) Diagrama de *Voronoi*. (b) Grafo Dual do Diagrama de *Voronoi*. (c) Triangulação de *Delaunay* resultante. (d) Par de pontos de *Voronoi* que possuem uma aresta em comum representado em uma Triangulação de *Delaunay*.

2.2.4 Grafo de Gabriel

O grafo de Gabriel \ddot{G} é um subconjunto de pontos do Diagrama de *Voronoi* e também um subgrafo da Triangulação de *Delaunay* (Zhang & King, 2002), ou seja, $\ddot{G} \subseteq \tilde{G}$. Segundo (Berg *et al.*, 2008), o grafo de Gabriel \ddot{G} de um conjunto de pontos \mathcal{S} é um grafo cujo conjunto de vértices $\mathcal{V} = \mathcal{S}$ e seu conjunto de arestas \mathcal{E} deve obedecer à seguinte definição:

$$(\mathbf{v}_i, \mathbf{v}_j) \in \mathcal{E} \leftrightarrow \delta^2(\mathbf{v}_i, \mathbf{v}_j) \leq [\delta^2(\mathbf{v}_i, z) + \delta^2(\mathbf{v}_j, z)] \quad \forall z \in \mathcal{V}, \mathbf{v}_i, \mathbf{v}_j \neq z \quad (2.18)$$

o que implica que, para $(\mathbf{v}_i, \mathbf{v}_j)$ constituir uma aresta de \ddot{G} , não pode haver nenhum outro vértice dentro da hipersfera cujo o diâmetro é a distância euclidiana entre \mathbf{v}_i e \mathbf{v}_j . As Figuras 2.6(a) a 2.6(h) mostram a construção do grafo de Gabriel de forma detalhada. É possível observar na Figura 2.6(f) que a escolha dos dois vértices em questão (círculo pontilhado) não satisfazem Equação (2.18). Logo, eles não possuem uma aresta.

A respeito da ordem de complexidade, é sabido que o algoritmo intuitivo de construção do grafo de Gabriel tem complexidade $O(n^3)$ (Zhang & King, 2002). Entretanto, se o grafo de Gabriel for construído utilizando a estrutura da triangulação de *Delaunay*, cuja complexidade dado o pior caso é $O(n \log n)$ a ordem de complexidade de construção do grafo se torna $O(n)$ (Toussaint, 1980).

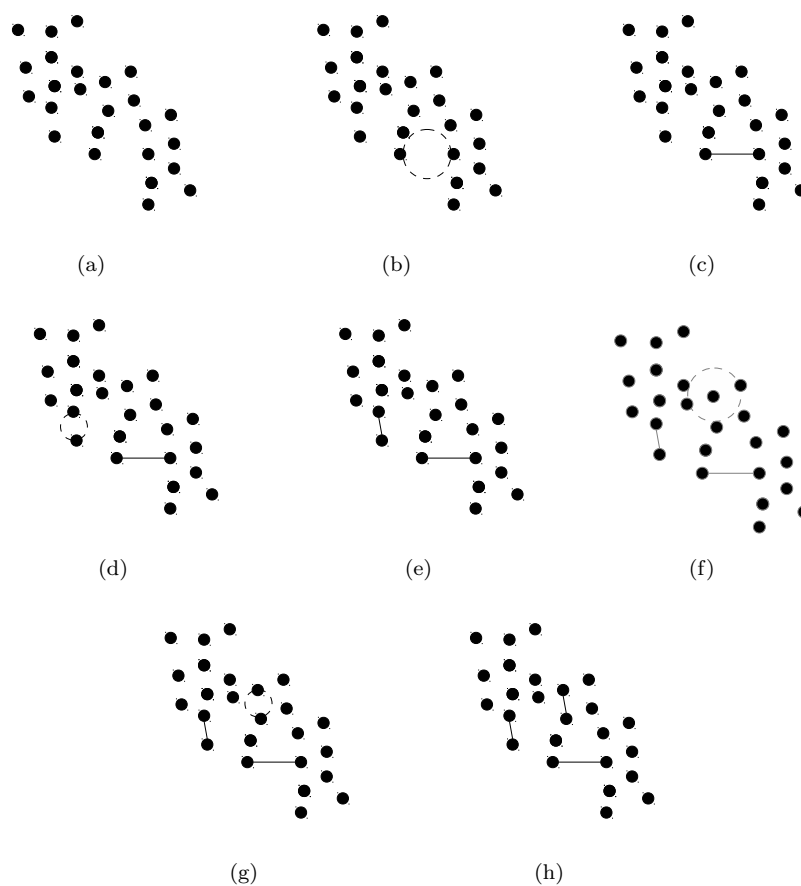


Figura 2.6: Construção do grafo de Gabriel.

Capítulo 3

Arestas de Suporte

O conjunto de arestas de suporte \mathcal{AS} e pontos médios \mathcal{PM} utilizados neste trabalho são encontrados a partir dos exemplos que compõem a margem de separação entre as classes. A implementação do método se dá por intermédio dos seguintes passos: construção do grafo de Gabriel, eliminação de sobreposição entre as classes e detecção da borda das classes. A ordem de sequencia destes passos é importante, uma vez que diminui a possibilidade de padrões serem escolhidos erroneamente para os conjuntos \mathcal{AS} e \mathcal{PM} , assim, aumentando a eficácia do método onde serão utilizados. Não obstante, será mostrado neste capítulo a similaridade dos vértices do conjunto \mathcal{AS} com os vetores de suporte da SVM.

3.1 Informação estrutural extraída a partir de um grafo planar

A informação estrutural é obtida através do grafo de Gabriel \tilde{G} , que depende somente da distância entre os padrões do conjunto de treinamento. Considerando o conjunto de dados $\mathcal{S} = \{(\mathbf{x}_i, \mathbf{y}_i) \mid i = 1 \dots N\}$, onde $\mathbf{y}_i \in \{+1, -1\}$ e $\mathbf{x}_i \in \mathbb{R}^n$. O grafo $\tilde{G} = \{\mathcal{V}, \mathcal{E}\}$ de \mathcal{S} possui um conjunto de Arestas de Suporte \mathcal{AS} , que representa todas arestas em \mathcal{E} que possuem um par de vértices $(\mathbf{x}_i, \mathbf{x}_j)$ de classes diferentes. Não havendo sobreposição entre os dados, podemos dizer que os vértices de \mathcal{AS} estão localizados na margem de separação entre as classes, como mostra a Figura 3.1. Assim, o hiperplano \mathcal{H}_l , que passa pelo ponto médio de um

3.2 Lidando com bases de dados com sobreposição

par pertencente a \mathcal{AS} , corresponde a um classificador de margem máxima em relação aos vértices \mathbf{x}_i e \mathbf{x}_j .

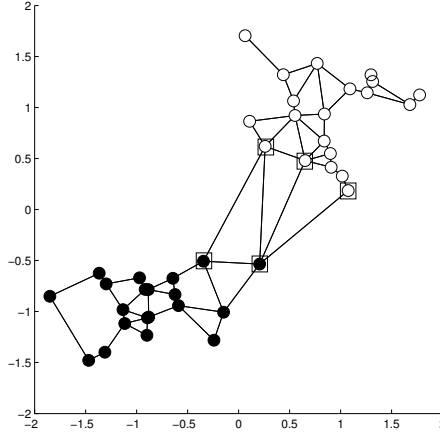


Figura 3.1: Arestas de suporte \mathcal{AS} indicadas pelos quadrados.

3.2 Lidando com bases de dados com sobreposição

Em alguns tipos de problemas, podem haver ruídos e sobreposição entre as classes de dados, como ilustrado na Figura 3.2. Sendo assim, a metodologia para encontrar o conjunto \mathcal{AS} pode retornar vértices que estão distante da margem. Uma solução encontrada para resolver este problema é utilizar operações da própria teoria dos Grafos, como o grau do vértice. Segundo a definição em [Christofides \(1975\)](#), o grau de um vértice representa o número de arestas conectadas a ele. No trabalho de [\(Garcia et al., 2015\)](#), é proposta uma técnica de filtragem de ruído, onde o grau de cada vértice informa se ele é um ruído ou não. Ainda de acordo com [\(Garcia et al., 2015\)](#), os vértices de grau com baixo valor podem ser rotulados como um possível ruído. Baseando-se no grau do vértice, uma medida de qualidade para cada vértice de um grafo foi proposta em [Aupetit & Catz \(2005\)](#):

$$q(\mathbf{x}_i) = \frac{\hat{\mathcal{A}}(\mathbf{x}_i)}{\mathcal{A}(\mathbf{x}_i)}, \quad (3.1)$$

$$\hat{\mathcal{A}}(\mathbf{x}_i) = \{\forall \mathbf{x}_j \in \mathcal{A}(\mathbf{x}_i) | \mathbf{y}_j = \mathbf{y}_i\},$$

3.2 Lidando com bases de dados com sobreposição

onde $\mathcal{A}(\mathbf{x}_i)$ representa o grau do vértice \mathbf{x}_i e $\hat{\mathcal{A}}(\mathbf{x}_i)$ descreve o grau de \mathbf{x}_i menos os vértices de classe diferente de \mathbf{x}_i . O valor de saída de $q(\mathbf{x}_i)$ pode se dividido em 3 tipos:

1. $q(\mathbf{x}_i) = 0$: representa um ruído, ou seja, todos os vértices que compartilham uma aresta com $q(\mathbf{x}_i)$ são de classe diferente de $q(\mathbf{x}_i)$.
2. $q(\mathbf{x}_i) = 1$: todos os vértices que compartilham uma aresta com $q(\mathbf{x}_i)$ são da mesma classe de $q(\mathbf{x}_i)$.
3. $0 < q(\mathbf{x}_i) < 1$: \mathbf{x}_i compartilha arestas com vértices de ambas as classes. Nesse caso, \mathbf{x}_i pode ser um candidato ao conjunto de arestas de suporte \mathcal{AS} .

Para exemplificar, a Figura 3.2 ilustra cada caso apresentado acima através de 3 vértices ocupando diferentes posições no espaço de entrada. A medida de qualidade calculada para cada um destes vértices foram: $q(\mathbf{x}_1) = 0$, $q(\mathbf{x}_2) = 1$ e $q(\mathbf{x}_3) = 2/3$.

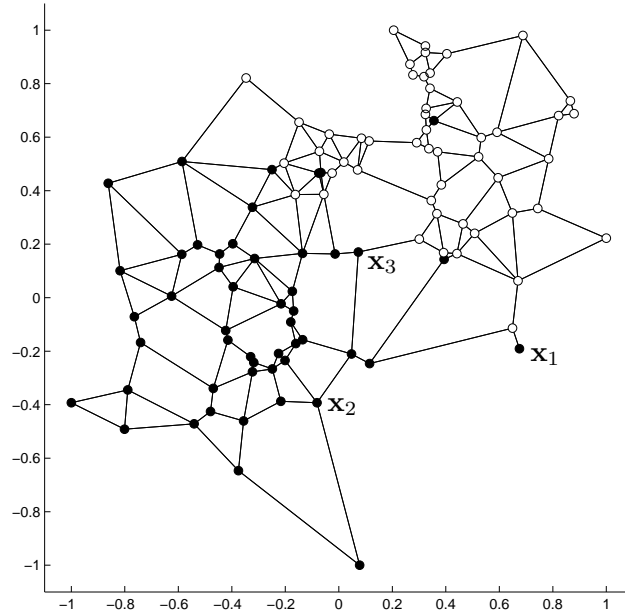


Figura 3.2: Base de dados com sobreposição entre as classes.

3.2.1 Metodologia para eliminação de sobreposição entre classes

Baseando-se nas definições de medida de qualidade $q(\cdot)$ retirada de [Aupetit & Catz \(2005\)](#), foi concebida uma metodologia para eliminar a sobreposição dos dados entre as classes. O método é fundamentado em 3 passos que se seguem:

1. Para todo $\mathbf{x}_i \in \ddot{G}$, calcule $q(\mathbf{x}_i)$ de acordo com a Equação (3.1).
2. Agrupar $q(\mathbf{x}_i)$ por classe, tal que \mathcal{Q}^+ e \mathcal{Q}^- representam a medida de qualidade para os padrões com os rótulos $+1$ e -1 , respectivamente.
3. Calcular o valor do limiar t^+ e t^- de cada classe como a média da medida de qualidade pertencendo à \mathcal{Q}^+ e \mathcal{Q}^- , onde

$$t^+ = \frac{\sum_{q(\mathbf{x}_i) \in \mathcal{Q}^+} q(\mathbf{x}_i)}{|\mathcal{Q}^+|}, \quad t^- = \frac{\sum_{q(\mathbf{x}_i) \in \mathcal{Q}^-} q(\mathbf{x}_i)}{|\mathcal{Q}^-|}.$$

4. Remover de \ddot{G} todos os vértices cuja $q(\mathbf{x}_i)$ é menor que t^+ e t^- .

O procedimento acima permite obter o conjunto \mathcal{AS} de um problema com sobreposição entre as classes, como mostrado na Figura 3.2. O resultado após a remoção da sobreposição pode ser visto na Figura 3.3, onde os vértices pertencentes a \mathcal{AS} são marcados no formato de quadrados.

3.3 Algoritmo para concepção do conjunto de arestas de suporte e do conjunto de pontos médios

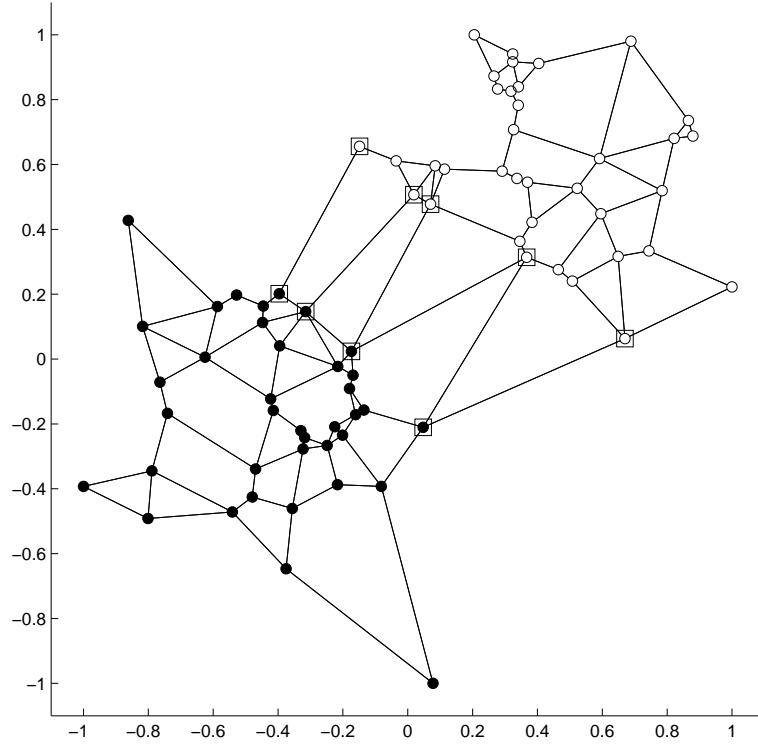


Figura 3.3: Conjunto de dados após a remoção da sobreposição.

3.3 Algoritmo para concepção do conjunto de arestas de suporte e do conjunto de pontos médios

1. A partir de um conjunto de dados $\mathcal{S} = \{(\mathbf{x}_i, \mathbf{y}_i) \mid i = 1 \dots N\}$, onde $\mathbf{y}_i \in \{+1, -1\}$ e $\mathbf{x}_i \in \mathbb{R}^n$, obtém-se o Grafo de Gabriel \tilde{G} com o conjunto de vértices sendo formado por todos os padrões de entrada, ou seja, $\mathcal{V} = \{\mathbf{x}_i \mid i = 1 \dots n\}$ e o conjunto de arestas \mathcal{E} satisfazendo a condição estabelecida na Equação (2.18).
2. Este passo do algoritmo é responsável por detectar e remover a sobreposição dos dados. Para todo $\mathbf{x}_i \in \mathcal{V}$, 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 $\{q(\mathbf{x}_i) < t^+ \mid y_i = 1\}$ ou $\{q(\mathbf{x}_i) < t^- \mid y_i = -1\}$, então \mathbf{x}_i é considerado como ruído e eliminado de \mathcal{V} .

3.3 Algoritmo para concepção do conjunto de arestas de suporte e do conjunto de pontos médios

3. O conjunto de vértices de \mathcal{AS} são os padrões encontrados na borda entre as classes. O conjunto \mathcal{AS} é encontrado da seguinte forma: seja $\mathbf{x}_i \in \mathcal{V}$ e, $\forall \mathbf{x}_j \in \mathcal{V}$ com $j \neq i$, caso a aresta $(\mathbf{x}_i, \mathbf{x}_j)$ for formada por vértices de classes distintas, então ela é incluída no conjunto \mathcal{AS} , conforme ilustrado pela Figura 3.3.
4. 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 \mathcal{AS} , calcula-se o ponto médio entre os vértices \mathbf{x}_i e \mathbf{x}_j de acordo com a seguinte expressão:

$$\bar{\mathbf{x}}_{ij} = \sum_{t=1}^n \mu(\mathbf{x}_i(t), \mathbf{x}_j(t)), \quad (3.2)$$

onde n é número de características (atributos) 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 \mathcal{AS} , obtém-se um conjunto de pontos médios \mathcal{PM} relativo às bordas das classes, conforme ilustrado pela Figura 3.4.

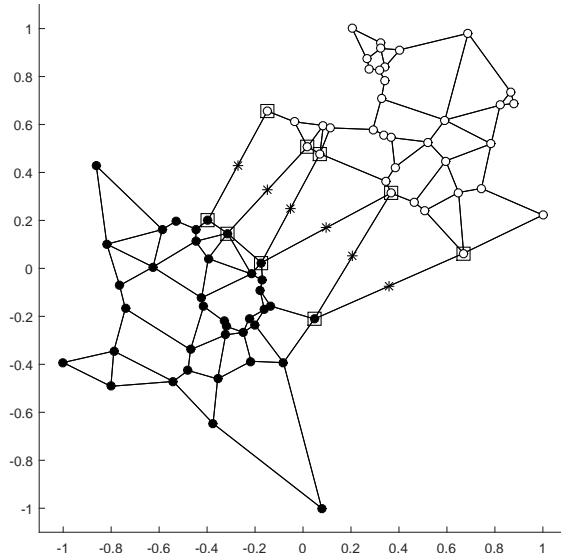


Figura 3.4: Cada aresta de suporte possui um ponto médio indicado por um asterisco.

3.4 Maximização da Margem Utilizando Arestas de Suporte

Para mostrar que os vértices relativos ao conjunto de arestas de suporte \mathcal{AS} podem ser utilizados como ferramentas para maximizar a margem de separação entre as classes, optou-se por demonstrar a similaridade entre as \mathcal{AS} e os vetores de suporte (VS) da SVM, uma vez que os VS possuem uma profunda fundamentação matemática (Cortes & Vapnik, 1995; Vapnik, 1995), e formam uma base para construção de um hiperplano de margem máxima.

3.4.1 Hiperplano de margem máxima

Dado o conjunto de dados $\mathcal{S} = \{(\mathbf{x}_i, \mathbf{y}_i) \mid i = 1 \dots n\}$, sendo $\mathbf{y}_i \in \{+1, -1\}$ denotando o rótulo de cada padrão $\mathbf{x}_i \in \mathbb{R}^d$, um hiperplano ótimo pode ser definido como uma função linear $f(\mathbf{x}) = \text{sgn}(\mathbf{w}^T \mathbf{x} + b)$ de parâmetros $\mathbf{w} \in \mathbb{R}^d$ e $b \in \mathbb{R}$, que gera a maior margem entre as classes (Boser *et al.*, 1992; Cortes & Vapnik, 1995).

Na formulação da SVM, o hiperplano é considerado em sua forma canônica, o que significa que os parâmetros (\mathbf{w}, b) são normalizados de forma que os padrões de treinamento satisfazem $|f(\mathbf{x})| = 1$ e, conseqüentemente, a margem é dada por $\frac{1}{\|\mathbf{w}\|}$. Assim, o problema de aprendizado pode ser expresso como se segue: encontrar os parâmetros \mathbf{w} e b que maximizam a margem, enquanto garantem que todos os padrões de treinamento sejam classificados corretamente. Ou seja:

$$\begin{aligned} \min_{(\mathbf{w}, b)} \quad & \frac{1}{2} \|\mathbf{w}\| \\ \text{S.t.} \quad & (\mathbf{w}^T \Phi(\mathbf{x}_i) + b) \geq 1, \quad \forall \mathbf{x}_i \in \mathcal{A} \\ & (\mathbf{w}^T \Phi(\mathbf{x}_j) + b) \leq 1, \quad \forall \mathbf{x}_j \in \mathcal{B} \end{aligned} \tag{3.3}$$

onde $\Phi(\cdot)$ é uma função de mapeamento. Nos exemplos a seguir, será utilizado como uma função de *Kernel* linear $\Phi(\mathbf{x}) = \mathbf{x} \cdot \mathbf{x}^T$; \mathcal{A} e \mathcal{B} são dois conjuntos de dados linearmente separáveis (classes). A Figura 3.5 mostra um hiperplano ótimo encontrado a partir do método SVM.

3.4 Maximização da Margem Utilizando Arestas de Suporte

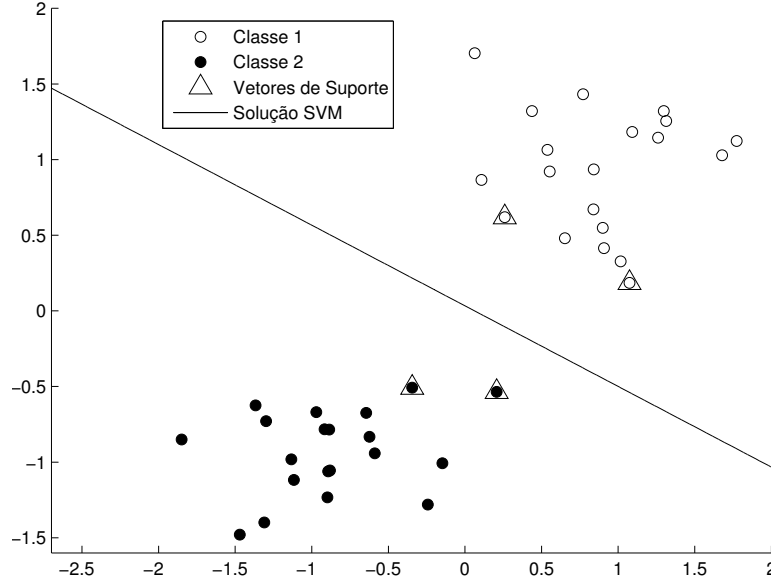


Figura 3.5: Hiperplano encontrado a partir do método SVM.

Uma interpretação do problema de encontrar o hiperplano ótimo para separar dois conjuntos de dados é encontrada no trabalho de [Bennett & Bredensteiner \(2000\)](#). Nesse trabalho, uma forma de encontrar o hiperplano de separação é construir o fecho convexo de cada conjunto (classe) e encontrar os pontos \mathbf{c} e \mathbf{d} mais próximos dos dois conjuntos de fecho convexo (vide Figura 3.6). É construído um segmento de reta através destes dois pontos, onde um plano ortogonal que corta o segmento de reta em seu ponto mediano é escolhido como o hiperplano de margem máxima. Uma representação esquemática é mostrada na Figura 3.6.

O conjunto de fecho convexo constitui todos os pontos que podem ser escritos como uma combinação convexa de pontos do conjunto de dados original ([Berg et al., 2008](#)). Uma combinação convexa de pontos é uma soma ponderada de pesos cuja soma é igual a um. Assim, fazendo as coordenadas dos padrões da classe \mathcal{A} ser dado por uma matriz $A^{n_A \times d}$ com n_A linhas e d colunas. Uma combinação convexa de pontos em \mathcal{A} é definida matematicamente como $u_1 A_1, u_2 A_2, \dots, u_{n_A} A_{n_A} = \mathbf{u}^T A$, com A_i denotando a i -ésima linha (padrão) de \mathcal{A} e $\mathbf{u} \in \mathbb{R}^d$. A classe \mathcal{B} é definida de forma análoga.

Dada estas definições, o problema de encontrar os pontos mais próximos entre dois conjuntos de fecho convexo $\mathbf{u}^T A$ e $\mathbf{v}^T B$, pode ser escrito como se segue:

3.4 Maximização da Margem Utilizando Arestas de Suporte

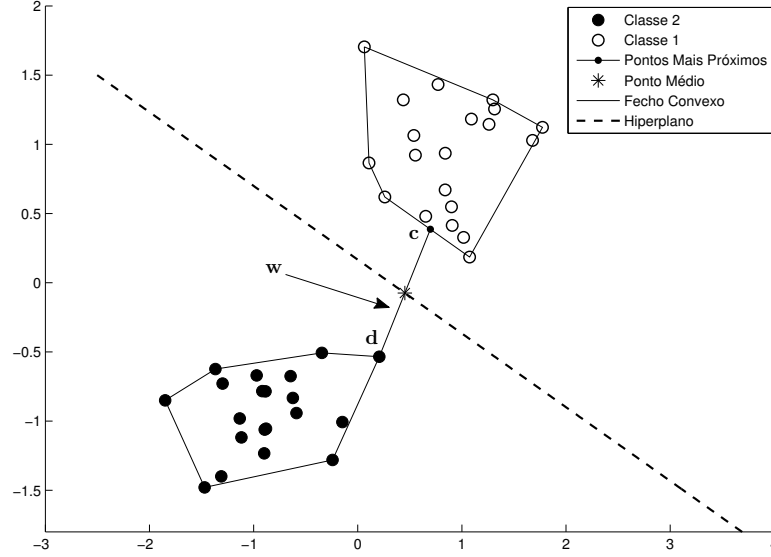


Figura 3.6: Hiperplano gerado através dos dois pontos mais próximos de dois conjuntos de fecho convexo.

$$\begin{aligned} \min_{(\mathbf{u}, \mathbf{v})} \quad & \frac{1}{2} \|\mathbf{u}^T A - \mathbf{v}^T B\| \\ \text{S.t.} \quad & \mathbf{e}^T \mathbf{u} = 1, \quad \mathbf{e}^T \mathbf{v} = 1, \quad \mathbf{u} \geq 0, \quad \mathbf{v} \geq 0 \end{aligned} \quad (3.4)$$

Onde \mathbf{e} é dado por $\mathbf{e} = \{\mathbf{e}_i = 1 | i = 1 \dots n\}$. Fazendo $\bar{\mathbf{u}}$ e $\bar{\mathbf{v}}$ ser uma solução ótima de (3.4). O vetor normal ao hiperplano é a diferença entre os pontos mais próximos $\mathbf{c} = \bar{\mathbf{u}}^T A$ e $\mathbf{d} = \bar{\mathbf{v}}^T B$. Assim, $\mathbf{w} = \mathbf{c} - \mathbf{d} = \bar{\mathbf{u}}^T A - \bar{\mathbf{v}}^T B$. O bias, b , é a distância da origem até o ponto mediano entre \mathbf{c} e \mathbf{d} ao longo do vetor normal \mathbf{w} , ou seja, $b = \frac{(\mathbf{c} + \mathbf{d})^T}{2} \mathbf{w}$.

3.4.2 Maximização da Margem baseada no Grafo de Gabriel

Considerem-se dois conjuntos de dados linearmente separáveis \mathcal{A} e \mathcal{B} de diferentes classes. Os padrões que serão utilizados para encontrar o hiperplano separador podem ser selecionados também através de um método de otimização (Torres *et al.*, 2014a). Fundamentado por intermédio da função (3.4) e na definição do

3.4 Maximização da Margem Utilizando Arestas de Suporte

grafo de Gabriel apresentada na Seção 2.2.4, o algoritmo para encontrar um par de arestas de suporte \mathcal{AS} definido na Seção 3.3 pode ser escrito como

$$f(A, B) = \min_{(\mathbf{u}, \mathbf{v})} \frac{1}{2} \|\mathbf{u}^T A - \mathbf{v}^T B\| \quad (3.5)$$

S.t.

$$\delta^2(\mathbf{u}^T A, \mathbf{v}^T B) \leq [\delta^2(\mathbf{u}^T A, z) + \delta^2(\mathbf{v}^T B, z)]$$

$$\mathbf{e}^T \mathbf{u} = 1, \quad \mathbf{e}^T \mathbf{v} = 1, \quad \mathbf{u} \geq 0, \quad \mathbf{v} \geq 0,$$

$\forall z, \mathbf{u}, \mathbf{v} \in \{(A \cup B) \setminus \Psi^{t-1}\}$, para $\{\mathbf{u}^T A, \mathbf{v}^T B\} \neq z$, $\mathbf{u} \in \{0, 1\}$, $\mathbf{v} \in \{0, 1\}$, onde Ψ^{t-1} é o conjunto de pares de padrões selecionados $\{\mathbf{u}, \mathbf{v}\}$ até a iteração $t - 1$. Note-se que, a cada iteração, o método seleciona o par de padrões mais próximos entre as duas classes. O tamanho da margem ρ é avaliado como

$$\rho = \sum_{\{(\psi_{\mathcal{A}}, \psi_{\mathcal{B}}) \in \Psi\}} \frac{1}{2} \|\psi_{\mathcal{A}}^T A - \psi_{\mathcal{B}}^T B\|^2 \quad (3.6)$$

onde $\psi_{\mathcal{A}}$ é o conjunto de padrões selecionados na borda do conjunto \mathcal{A} . De forma análoga, $\psi_{\mathcal{B}}$ são os padrões selecionados em \mathcal{B} . O conjunto de arestas de suporte \mathcal{AS} é formado como $\mathcal{AS} = \{\psi_{\mathcal{A}} \cup \psi_{\mathcal{B}}\}$.

Uma vez selecionado o conjunto \mathcal{AS} , o hiperplano \mathcal{H} pode ser encontrado utilizando-se a pseudo-inversa linear *Moore-Penrose* (Albert, 1972) de Ψ :

$$\mathbf{w} = \Psi^\dagger \mathbf{y} \quad (3.7)$$

onde \mathbf{w} é o vetor de pesos que define \mathcal{H} , Ψ^\dagger é a pseudo-inversa e Ψ é dado por:

$$\Psi = \begin{bmatrix} \psi_{11} & \psi_{12} & \cdots & \psi_{1d} & 1 \\ \psi_{21} & \psi_{22} & \cdots & \psi_{2d} & 1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \psi_{k1} & \psi_{k2} & \cdots & \psi_{kd} & 1 \end{bmatrix} \quad (3.8)$$

sendo k a cardinalidade de Ψ e d , o número de dimensões dos padrões.

A solução encontrada utilizando a pseudo-inversa minimiza a norma de separação do hiperplano, se os padrões selecionados forem vetores de suporte (Guyon,

3.4 Maximização da Margem Utilizando Arestas de Suporte

2006). Como o método proposto visa encontrar os padrões da borda, que são os mais próximos entre as classes, é claro que Ψ aproxima um vetor de dados gerado por um classificador SVM. Como exemplo, a Figura 3.7 mostra dois hiperplanos, um deles encontrado por um classificador SVM e o outro pelo método proposto. Pode ser visto que as duas soluções são muito próximas.

Observe-se que a primeira restrição da Função (3.5) é na verdade a definição do grafo de Gabriel apresentado na Equação (2.18). Como mostrado na função de minimização (3.4), quando $\mathbf{u}^T A - \mathbf{v}^T B$ é minimizado a margem é maximizada (Bennett & Bredensteiner, 2000). Todavia, neste método proposto, não é necessário encontrar o fecho convexo, porque os padrões selecionados pertencem ao conjunto de dados.

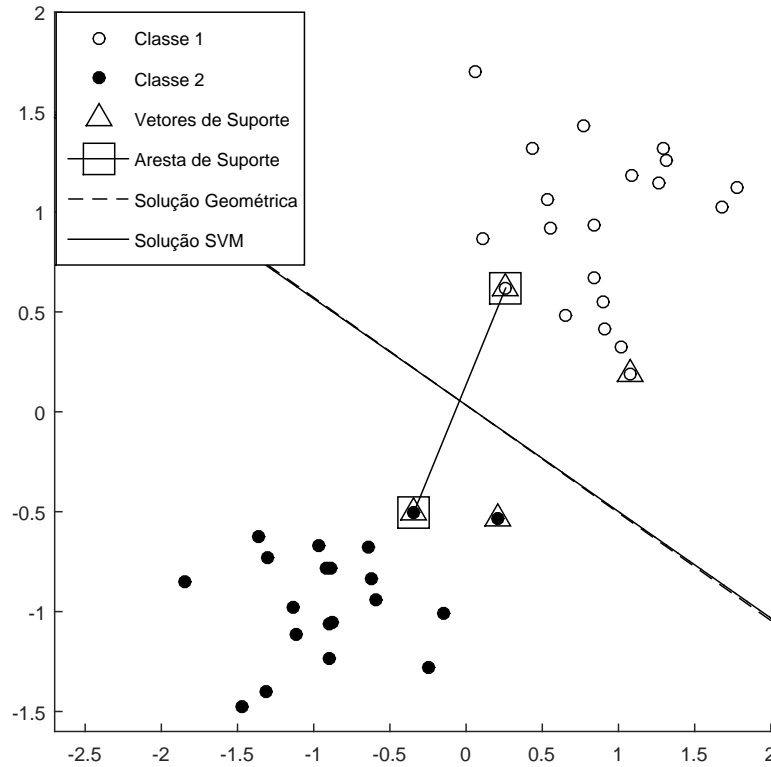


Figura 3.7: Comparação entre o classificador Geométrico gerado a partir da aresta de suporte e o classificador obtido pela SVM.

Capítulo 4

Abordagens Propostas

Neste Capítulo são mostradas as contribuições da tese. A ordem de apresentação segue a ordem em que os métodos foram originados. Assim, é possível observar a evolução desde a primeira abordagem, que foi um decisor para o método MOBJ até chegar no desenvolvimento de um novo método de classificação.

4.1 Um decisor de margem larga para o método MOBJ

O problema da generalização em Redes Neurais Artificiais (RNAs) foi analisado formalmente por V. Vapnik ([Vapnik, 1995, 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 ideia 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

4.1 Um decisor de margem larga para o método MOBJ

seu equilíbrio através do ajuste de um ou vários parâmetros (como, por exemplo, o parâmetro que controla a regularização).

Além disto, a formulação multiobjetivo do aprendizado (MOBJ) fornece uma abordagem alternativa para a implementação do princípio de MER através da minimização explícita (separada) do erro de treinamento e de uma medida que reflete a complexidade da rede (Costa *et al.*, 2007; Jin & Sendhoff, 2009; Kokshe-
nev & Braga, 2010; Teixeira *et al.*, 2000). 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 \mathcal{PO} . A formulação MOBJ para o treinamento de RNAs resulta então em um conjunto de soluções \mathcal{PO} que corresponde aos melhores compromissos entre os funcionais erro e complexidade, como é mostrado na Figura 2.3.

Uma vez obtido o conjunto \mathcal{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 o sub ou o 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.

Tendo em vista essas dificuldades, é apresentada uma nova estratégia de decisão direcionada a problemas de classificação de padrões. Usando ferramentas da Geometria Computacional (Berg *et al.*, 2008), foi desenvolvido um método de um decisor que busca pela solução do conjunto \mathcal{PO} que possui a maior margem (ou distância) de separação entre as classes. Na abordagem proposta, o espaço de entrada é modelado através do grafo de Gabriel. A partir do grafo foi possível encontrar a margem de separação entre as classes, além de eliminar elementos ruidosos na distribuição, como mostrado na Seção 3.3. Para encontrar a solução de maior separação entre as classes, foi proposta a inserção de novas amostras

advindas de uma distribuição normal multivariada. Estes exemplos foram associados ao conjunto de hiperesferas \mathcal{ES} , onde cada centro é uma coordenada de um ponto médio pertencente ao conjunto \mathcal{PM} . Para encontrar o classificador de maior margem, o decisor seleciona a solução do conjunto \mathcal{PO} que classifica as amostras relativas a todas hiperesferas do conjunto \mathcal{ES} com a menor variância.

4.1.1 Decisor Proposto

A partir de um conjunto de dados $\mathcal{S} = \{(\mathbf{x}, \mathbf{y}) \mid i = 1 \dots n\}$, sendo $\mathbf{x}_i \in \mathbb{R}^d$, $\mathbf{y}_i = \{-1, 1\}$ e n representando o tamanho de \mathcal{S} . O conjunto \mathcal{S} é transformado em um grafo de Gabriel, conforme definido na Seção 2.2.4 e editado na Seção 3.3, onde são obtidos também os conjuntos \mathcal{AS} e \mathcal{PM} . \mathcal{PM} representa o conjunto de pontos médios de \mathcal{S} , e $\mathcal{AS} \subset \mathcal{S}$ é o conjunto de arestas de suporte \mathcal{AS} .

A partir do conjunto de pontos médios \mathcal{PM} , é gerado um outro conjunto \mathcal{ES} de $n_{\mathcal{ES}}$ hiperesferas. Para cada hiperesfera \mathcal{ES}_i , é associada uma distribuição normal multivariada da forma $N(\boldsymbol{\mu}_i = \mathcal{PM}_i, \Sigma_i)$, onde $\Sigma_i = I_d \cdot \sigma_i^2$, ou seja, o centro de cada uma dessas hiperesferas é associado à coordenada de um ponto médio. E o σ_i é calculado como ± 3 desvios padrões da média como

$$\mathcal{R}_i = 3\sigma_i, \quad (4.1)$$

$$\sigma_i = \frac{\mathcal{R}_i}{3}, \quad (4.2)$$

onde o raio

$$\mathcal{R}_i = \frac{1}{2}(\mathbf{c}_i + \mathbf{d}_i), \quad \forall i (\mathbf{c}_i, \mathbf{d}_i) \in \mathcal{AS}_i, \quad (4.3)$$

sendo $(\mathbf{c}_i, \mathbf{d}_i) \in \mathcal{AS}_i$, um par de vértices de uma aresta de suporte. Assim, garantindo estatisticamente que 99,7% dos valores encontram-se a uma distância da média inferior a 3 vezes o desvio padrão.

O conjunto de distribuições é definido como $\mathcal{N} = \{N(\boldsymbol{\mu}_i, \Sigma_i) \mid i = 1 \dots n_{\mathcal{ES}}\}$, sendo a distribuição \mathcal{N}_i expressa na forma $\mathcal{N}_i = \{(\mathbf{x}_i, \mathbf{y}_i) \mid i = 1 \dots n_{\mu}\}$, em que $\mathbf{x}_i \in \mathbb{R}^d$, $\mathbf{y}_i = \{-1, 1\}$ e n_{μ} representando o tamanho de \mathcal{N}_i . Os padrões pertencentes a \mathcal{N}_i são rotulados na mesma proporção numérica para cada classe $\{-1, 1\}$.

4.1 Um decisor de margem larga para o método MOBJ

O propósito do decisor é selecionar, dentre as soluções do conjunto Pareto-Ótimo, a solução mais próxima dos pontos médios. Diferentemente da rede neural MLP, que tem como saída apenas uma solução (uma única rede treinada), o MOBJ tem como saída várias soluções para o mesmo problema, onde cada solução possui uma complexidade diferente. Seja cada uma delas um hiperplano $\mathcal{H}_i = \{\mathbf{w}, b\}$, onde \mathbf{w} é o vetor de pesos da rede neural e b o *bias*. O hiperplano escolhido pelo decisor deve ter complexidade suficiente para classificar os ruídos presentes no conjunto de hiperesferas.

Para um dado classificador \mathcal{H}_i , cada distribuição do conjunto \mathcal{N} é classificada da seguinte forma

$$f_{\mathcal{N}}(t) = \hat{f}_{\mathcal{H}}(\mathcal{N}_t, \mathcal{H}_i), \quad t = \{1 \dots n_{\varepsilon s}\} \quad (4.4)$$

onde

$$\hat{f}_{\mathcal{H}}(\mathcal{N}_t, \mathcal{H}_i) = \text{sgn}(\mathbf{w}^T \Phi(\mathbf{x}) + b), \quad \{w, b\} \in \mathcal{H}_i, \mathbf{x} \in \mathcal{N}_t, \quad (4.5)$$

$\text{sgn}(\cdot)$ é uma função sinal e Φ uma função de mapeamento. A proporção de classificação para cada classe dado um conjunto \mathcal{N}_i é expressa como

$$\mathbf{y}_i^+ = \frac{1}{n_{\varepsilon s}} \sum_{(\mathbf{y} \neq -1)} \mathbf{y}, \quad (4.6)$$

$$\mathbf{y}_i^- = \frac{1}{n_{\varepsilon s}} \sum_{(\mathbf{y} \neq 1)} \mathbf{y}, \quad (4.7)$$

onde $\mathbf{y} = f_{\mathcal{N}}(i)$, e \mathbf{y}_i^- é a porcentagem de elementos de um conjunto \mathcal{N}_i classificados como da classe negativa (-1) e \mathbf{y}_i^+ da classe positiva ($+1$). O intuito é que se possa classificar corretamente os padrões de cada classe, de forma que o hiperplano classificador passe pela média da distribuição, ou seja, no ponto médio. Assim, maximizando a margem de separação entre as classes.

Dadas as matrizes $\{M_i, Z\} \in \mathbb{R}^{n_{\varepsilon s} \times 2}$, para cada classificador \mathcal{H}_i existirá uma matriz M_i , tal que

4.1 Um decisor de margem larga para o método MOBJ

$$M_i = \begin{bmatrix} \mathbf{y}_1^+ & \mathbf{y}_1^- \\ \mathbf{y}_2^+ & \mathbf{y}_2^- \\ \vdots & \vdots \\ \mathbf{y}_{n_{\varepsilon S}}^+ & \mathbf{y}_{n_{\varepsilon S}}^- \end{bmatrix}, \quad (4.8)$$

que será comparada com a matriz

$$Z = \begin{bmatrix} z_1 & z_1 \\ z_2 & z_2 \\ \vdots & \vdots \\ z_{n_{\varepsilon S}} & z_{n_{\varepsilon S}} \end{bmatrix}, \quad (4.9)$$

onde z representa o valor esperado da porcentagem de cada classe, ou seja, $\forall i \ z_i = \frac{1}{2}$. O classificador escolhido será o que tiver a menor variância entre as matrizes M_i e Z ou, equivalentemente, aquele que mais se aproxima dos pontos médios³. A comparação entre as matrizes é calculada através da soma total de quadrados (STQ), disponível como uma das informações de saída do método estatístico *two-way analysis of variance* (ANOVA2) (Hogg & Ledolter, 1987) e calculado como

$$f_{\sigma}(M_i, Z) = \overbrace{\sum_{i=1}^N \sum_{j=1}^2 (X_{i,j})^2}^{\text{Soma Total de Quadrados}} - \left(\sum_{i=1}^2 \overline{\overline{X}}_i \right)^2, \quad (4.10)$$

onde

$$\overline{\overline{X}} = \overbrace{\begin{bmatrix} \frac{1}{2} \sum_{i=1}^2 \overline{X}_{i,1} & \frac{1}{2} \sum_{i=1}^2 \overline{X}_{i,2} \end{bmatrix}}^{\text{Média entre os grupos}} \in \mathbb{R}^{1 \times 2}, \quad (4.11)$$

$$\overline{X} = \overbrace{\begin{bmatrix} \frac{1}{n_{\varepsilon S}} \sum_{i=1}^{n_{\varepsilon S}} X_{i,1} & \frac{1}{n_{\varepsilon S}} \sum_{i=1}^{n_{\varepsilon S}} X_{i,2} \\ \frac{1}{n_{\varepsilon S}} \sum_{i=n_{\varepsilon S}+1}^{N/2} X_{i,1} & \frac{1}{n_{\varepsilon S}} \sum_{i=n_{\varepsilon S}+1}^{n_{\varepsilon S}} X_{i,2} \end{bmatrix}}^{\text{Média dentro dos grupos}} \in \mathbb{R}^{2 \times 2}, \quad (4.12)$$

$$X = \begin{bmatrix} M_i \\ Z \end{bmatrix} \in \mathbb{R}^{N \times 2}, \quad N = 2(n_{\varepsilon S}). \quad (4.13)$$

³Neste caso, a média μ da distribuição dos dados de cada hipersfera.

4.1 Um decisor de margem larga para o método MOBJ

Por fim, o critério de seleção é o classificador que minimiza

$$\mathcal{H}^* = \min f_{\sigma}(M_i, Z), \quad i = \{1 \dots n_{\mathcal{H}}\}, \quad (4.14)$$

onde $n_{\mathcal{H}}$ representa o número de classificadores, $f_{\sigma}(\cdot)$ é a Equação (4.10), que retorna o valor da STQ para o i -ésimo classificador de \mathcal{H} , e \mathcal{H}^* é o classificador escolhido como o de menor valor da STQ.

4.1.2 Experimentos com base de dados sintéticas

Com o objetivo de verificar o decisor, experimentos foram conduzidos com quatro *benchmarks*: *Corners*, *Full Moon*, *Cluster* e *Half Kernel*. O algoritmo de treinamento MOBJ (Teixeira *et al.*, 2000) teve seus parâmetros ajustados da seguinte forma. O conjunto \mathcal{PO} de soluções foi composto por 30 redes MLP com 15 neurônios na camada oculta. Foi utilizada a função de transferência sigmóide para a camada oculta e de saída, e a diferença de norma euclidiana (complexidade) entre as soluções foi de $\Delta \|\mathbf{w}\| = 0.5$ (Teixeira, 2001). As Figuras 4.1(a) a 4.4(d) apresentam os resultados, e as Figuras 4.5(a) a 4.5(d) apresentam o conjunto Pareto para cada *benchmark*, onde a solução escolhida é representada pelo círculo preenchido.

Para comparar os resultados dos *benchmarks* gerados através do método MOBJ-clas, foi utilizada uma SVM com função de *kernel* gaussiana de base radial, onde os parâmetros do *Kernel* e regularização para SVM foram encontrados através de validação cruzada com 10 partições e busca em *grid*. A implementação deste método se deu através dos pacotes *Kernlab* e *Caret* disponíveis para linguagem *R* (R Core Team, 2015). As Figuras 4.6(a) a 4.6(d) apresentam os resultados do método SVM aplicado aos *benchmarks*: *Corners*, *Full Moon*, *Cluster* e *Half Kernel*.

Benchmark Corners

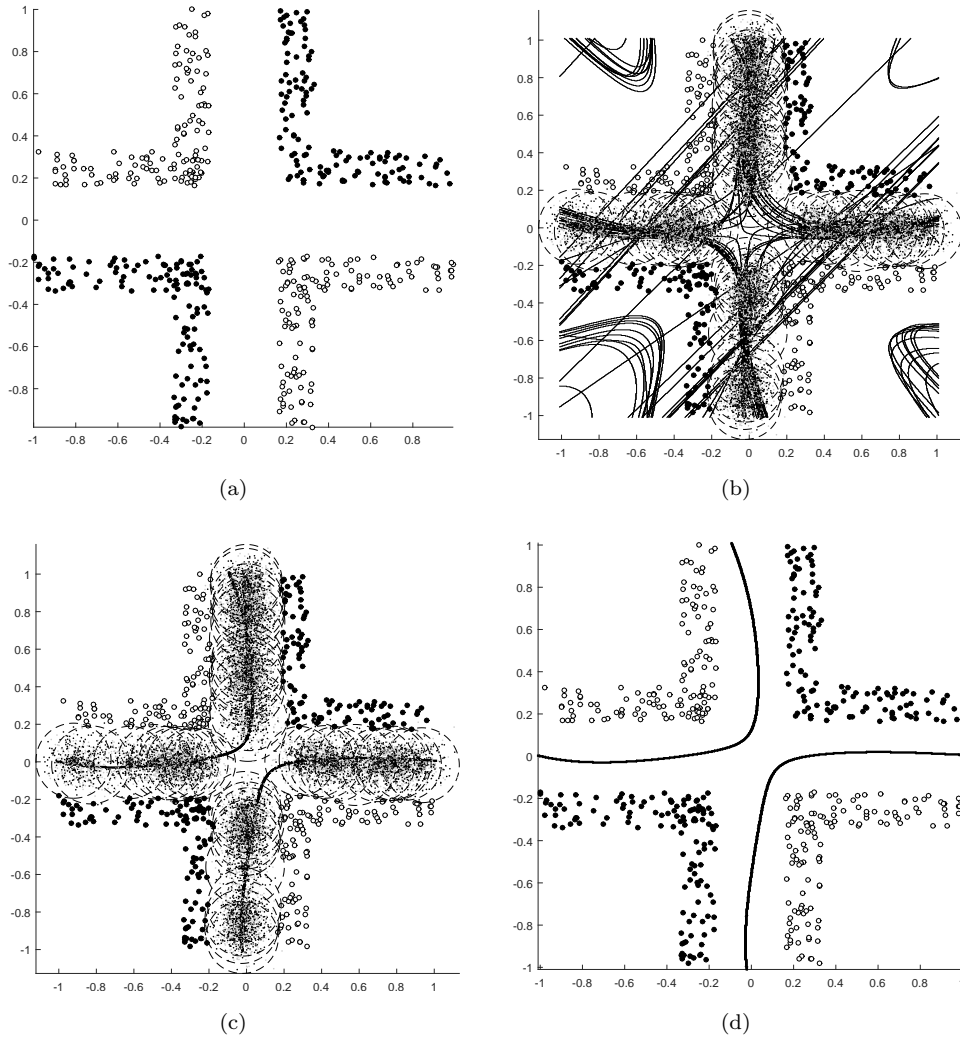


Figura 4.1: (a) Conjunto de dados do *benchmark Corners*. (b) Conjunto de soluções. (c) Solução escolhida vista com as hipersferas. (d) Solução final.

Benchmark Full Moons

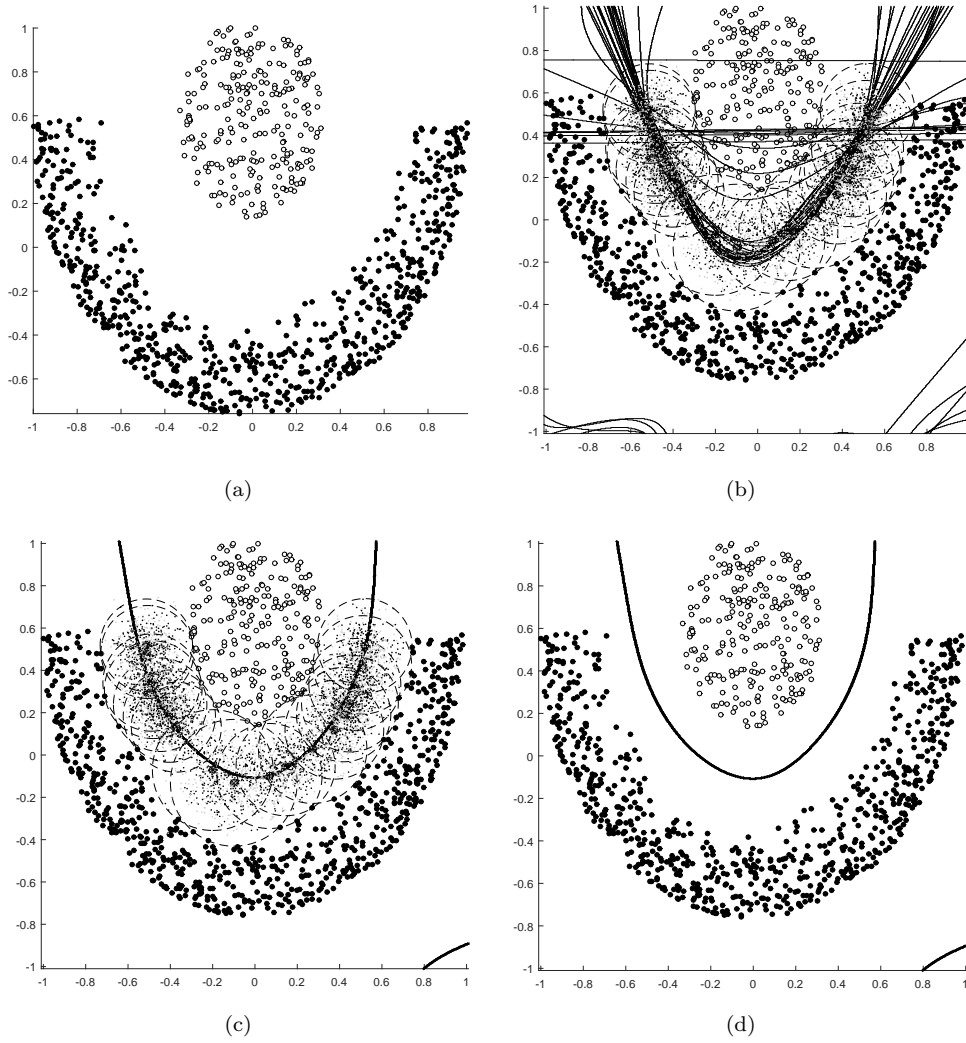


Figura 4.2: (a) Conjunto de dados do *benchmark Full Moons*. (b) Conjunto de soluções. (c) Solução escolhida vista com as hiperesferas. (d) Solução final.

Benchmark Cluster

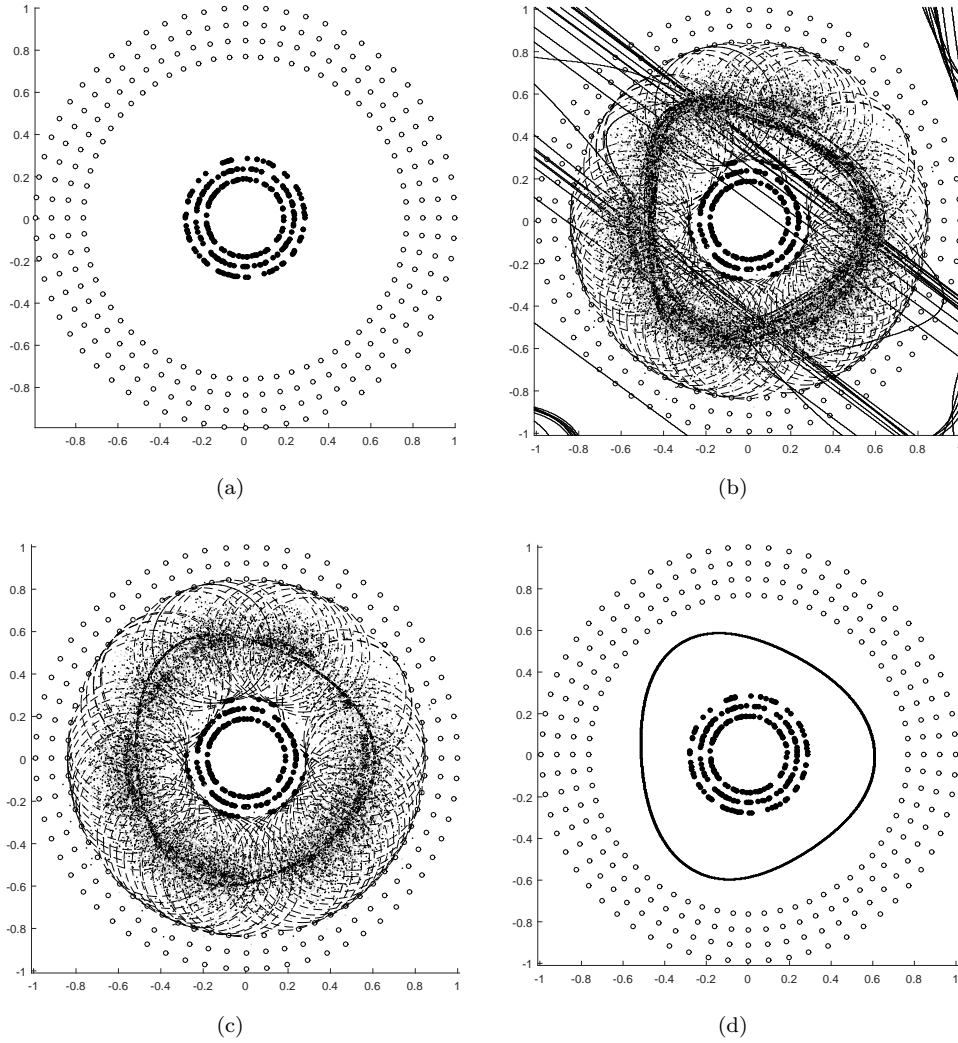


Figura 4.3: (a) Conjunto de dados do *benchmark Cluster*. (b) Conjunto de soluções. (c) Solução escolhida vista com as hiperesferas. (d) Solução final.

Benchmark Half Kernel

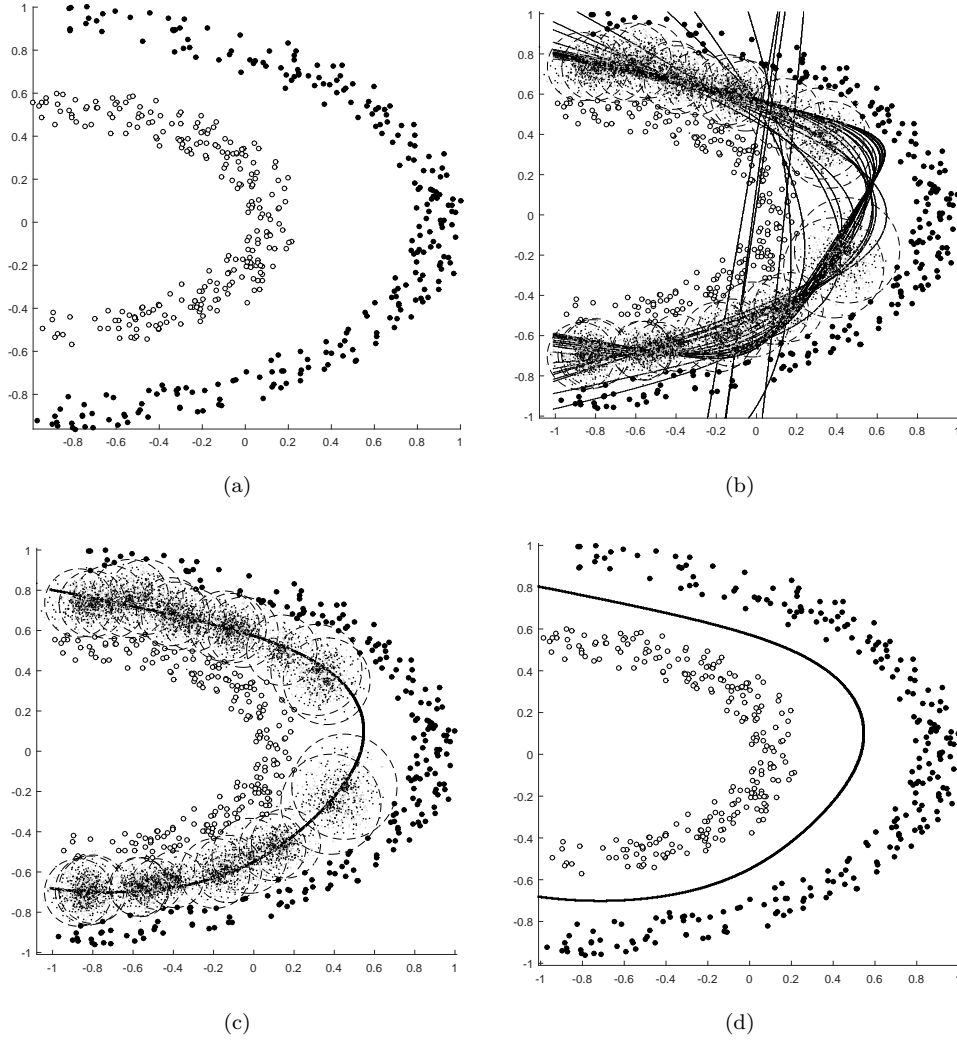


Figura 4.4: (a) Conjunto de dados do *benchmark Half Kernel*. (b) Conjunto de soluções. (c) Solução escolhida vista com as hipersferas. (d) Solução final.

Conjunto Pareto dos benchmarks

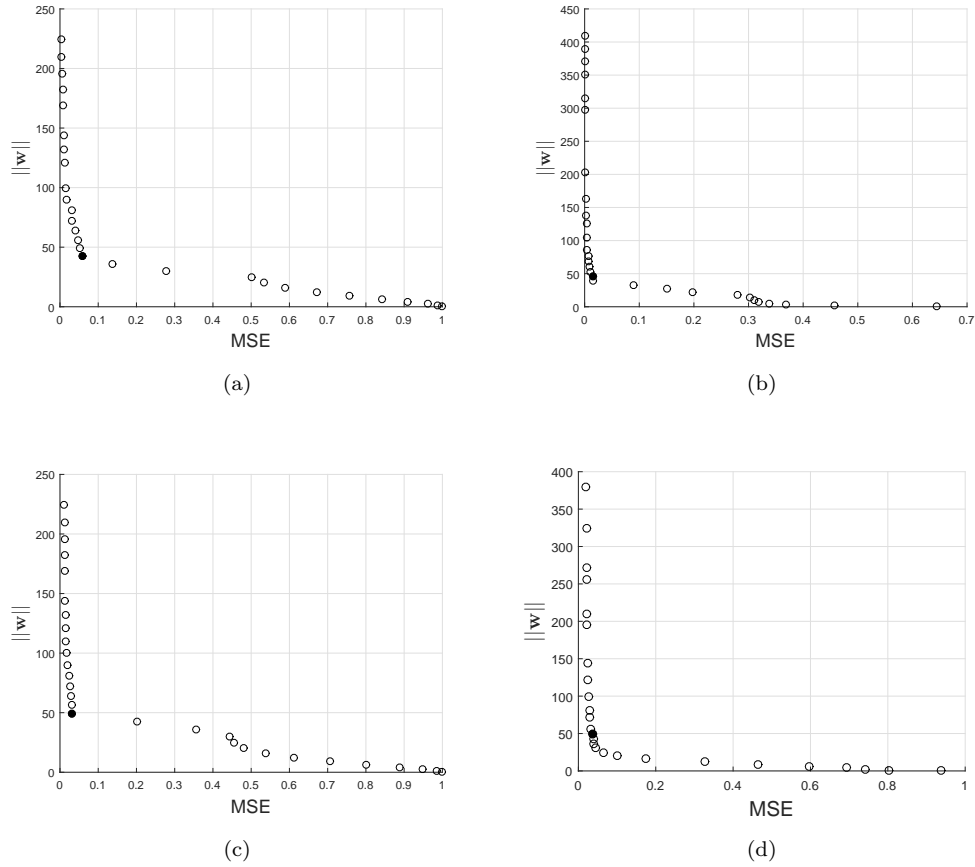


Figura 4.5: (a) Conjunto Pareto do *benchmark Corners*. (b) Conjunto Pareto do *benchmark Full Moons*. (c) Conjunto Pareto do *benchmark Cluster*. (d) Conjunto Pareto do *benchmark Half Kernel*.

Benchmark utilizando SVM

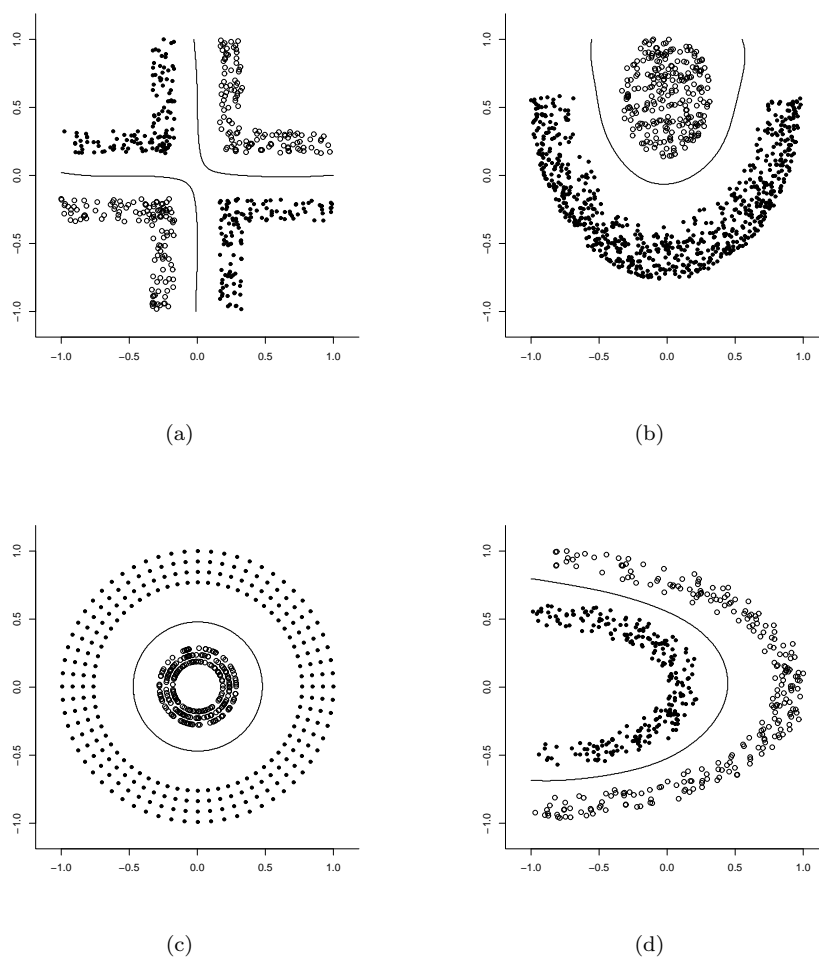


Figura 4.6: (a) SVM aplicada ao *benchmark Corners*. (b) SVM aplicada ao *benchmark Full Moons*. (c) SVM aplicada ao *benchmark Clusters*. (d) SVM aplicada ao *benchmark Half Kernel*.

4.1.3 Metodologia para experimento em base de dados reais

Os experimentos foram realizados com 17 bases de dados reais de n -dimensões e tamanhos diferentes. Sendo que 15 destas, foram obtidas do repositório UCI (Bache & Lichman, 2013). A base de dados “*Appendicitis data set*” foi obtida do repositório *Keel Data Set Repository* (Alcalá et al., 2010) e a base “*Breast Cancer Hess Probes*” em (Hess et al., 2006). Estas bases são referenciadas em inúmeros trabalhos na literatura, o que as torna um bom *benchmark* para este trabalho. Todas as bases utilizadas no trabalho são binárias, são elas: *Appendicitis data set*(*appendicitis*), *Statlog Australian Credit*(*australian*), *Banknote Authentication Data Set*(*Banknote*), *Blood Transfusion Service Center Data Set*(*blood*), *The Wisconsin Breast cancer*(*breastcancer*), *Breast Cancer Hess Probes*(*BcrHess*) *Bupa liver disorders*(*bupa*), *Climate Model Simulation Crashes Data Set*(*climate*), *Pima Indians Diabetes Data Set*(*diabetes*), *Fertility Data Set*(*fertility*), *Statlog German Credit Data Set*(*german*), *Glass Identification Data Set*(*glass*), *Haberman’s Survival Data Set*(*haberman*), *Statlog Heart Data Set*(*heart*), *Indian Liver Patient Dataset*(*ILPD*), *Parkinsons Data Set*(*parkinsons*) e *Breast Cancer Wisconsin Prognostic Data Set*(*wdbc*).

Todas estas bases passaram pelas seguintes etapas de pré-processamento: filtragem de ruído, remoção de amostras contendo atributos faltantes e normalização dos dados entre $\{-1, 1\}$. A base de dados *glass* foi transformada em um problema de classificação binária, uma vez que no seu formato original é composta por 7 classes. O procedimento de transformação pode ser visto em Castro & Braga (2013). Para garantir relevância estatística, o experimento foi repetido usando validação cruzada com 10 partições. A medida de desempenho foi feita através da média da AUC(*Area under the curve*) (Fawcett, 2006).

4.1.4 Resultados

Os métodos SVM e LSSVM-RBF (Suykens & Vandewalle, 1999) foram utilizados para comparar o nosso decisor (MOBJ-clas). Duas configurações de *Kernels* foram utilizadas. A primeira, SVM-RBF e LSSVM-RBF com função de bases

4.1 Um decisor de margem larga para o método MOBJ

radiais, e SVM-Poly com função polinomial. Os parâmetros de *Kernel* e regularização para SVM-RBF e SVM-Poly foram encontrados através de validação cruzada com 10 partições e busca em *grid*. A implementação destes métodos se deu através dos pacotes *Kernlab* e *Caret* disponíveis para linguagem *R* (R Core Team, 2015).

Para gerar as soluções necessárias para os decisores de validação (MOBJ-VAL) e MOBJ-clas, o algoritmo de treinamento MOBJ (Teixeira *et al.*, 2000) teve seus parâmetros ajustados da seguinte forma: o conjunto \mathcal{PO} de soluções foi composto por 150 redes MLP com 10 neurônios na camada oculta (Teixeira, 2001). 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 $\delta\|\mathbf{w}\| = 0.03$. Ao contrário do nosso decisor, que pode utilizar todo o conjunto de dados de treinamento para tomada de decisão, o decisor MOBJ-VAL depende de um subconjunto dos dados de treinamento, chamado de conjunto de validação. Este conjunto funciona como um conjunto de pré-teste para determinar entre as soluções do conjunto \mathcal{PO} qual será a escolhida. Para este experimento, o conjunto de validação foi composto por 20% dos dados de treinamento escolhidos aleatoriamente.

A Tabela 4.1 mostra os valores da média da AUC e desvio padrão obtidos pelos métodos MOBJ-clas, MOBJ-VAL, SVM-RBF, SVM-Poly e LSSVM-RBF sobre as 17 bases de dados. Os melhores valores encontram-se em negrito. As quatro últimas colunas dessa tabela mostram as características de cada base, em que N_d é o número de dimensões, N é o número total de amostras. A quantidade de elementos de cada classe da base de dados são denotados por N^+ e N^- , respectivamente.

4.1 Um decisor de margem larga para o método MOBJ

Tabela 4.1: Resultados do decisor para o método MOBJ: média da AUC e características das bases de dados. Os melhores valores encontram-se em negrito.

Base de Dados	MOBJ-clas	MOBJ-VAL	SVM-RBF	SVM-Poly	LSSVM-RBF	N_d	N	N^+	N^-
<i>appendicitis</i>	0.788±0.165	0.678±0.199	0.707±0.224	0.718±0.222	0.687±0.234	7	106	21	85
<i>australian</i>	0.864±0.044	0.852±0.05	0.863±0.059	0.862±0.056	0.862±0.042	14	690	307	383
<i>banknote</i>	0.981±0.008	0.983±0.009	1±0	1±0	0.999±0.004	4	1372	610	762
<i>blood</i>	0.565±0.038	0.552±0.031	0.63±0.058	0.624±0.043	0.616±0.045	4	748	178	570
<i>breastcancer</i>	0.96±0.036	0.963±0.036	0.969±0.02	0.967±0.02	0.955±0.027	9	683	444	239
<i>breastHess</i>	0.766±0.084	0.707±0.114	0.791±0.115	0.79±0.093	0.735±0.098	30	133	99	34
<i>bupa</i>	0.674±0.06	0.675±0.074	0.663±0.053	0.665±0.052	0.674±0.076	6	345	145	200
<i>climate</i>	0.803±0.104	0.777±0.137	0.525±0.056	0.55±0.09	0.645±0.169	18	540	494	46
<i>diabetes</i>	0.721±0.045	0.723±0.047	0.711±0.048	0.716±0.05	0.714±0.058	8	768	500	268
<i>fertility</i>	0.597±0.197	0.489±0.023	0.5±0	0.5±0	0.578±0.184	9	100	12	88
<i>german</i>	0.668±0.04	0.673±0.05	0.655±0.038	0.658±0.038	0.657±0.048	24	1000	700	300
<i>glass</i>	0.883±0.158	0.833±0.192	0.846±0.202	0.846±0.202	0.822±0.203	9	214	29	185
<i>haberman</i>	0.575±0.076	0.555±0.072	0.535±0.056	0.508±0.039	0.552±0.092	3	306	225	81
<i>heart</i>	0.835±0.066	0.802±0.11	0.832±0.085	0.832±0.082	0.828±0.071	13	270	150	120
<i>ILPD</i>	0.532±0.037	0.525±0.024	0.502±0.006	0.5±0	0.572±0.07	10	579	414	165
<i>parkinsons</i>	0.765±0.063	0.743±0.068	0.753±0.062	0.769±0.059	0.813±0.116	22	195	147	48
<i>wdbc</i>	0.65±0.114	0.584±0.122	0.496±0.011	0.493±0.014	0.576±0.112	33	194	46	148
Média do Rank $R(\mathcal{L})$	2.088	3.411	3.176	3.088	3.235				

4.1.5 Teste de significância

Para comparar e analisar os resultados, foi utilizado o teste não-paramétrico de Friedman, que de acordo com Demšar (2006) é o teste mais indicado para comparar múltiplos classificadores. No teste de Friedman é criado um *rank* dos classificadores para cada base de dados, onde o valor da média do *rank* $R(\mathcal{L}_i)$ para cada classificador $\mathcal{L}_i \in \mathcal{L}$ é utilizada para formular a seguinte hipótese H_0 : Todos os classificadores são equivalentes, então a média dos *rank*s são iguais $R(\mathcal{L}) = \sum_i^N R(\mathcal{L}_i)$. A estatística de Friedman

$$F_F = \frac{(M-1)\chi_F^2}{M(L-1) - \chi_F^2} \quad (4.15)$$

é distribuída segundo a distribuição F de *Fisher-Snedecor* com $(L-1)$ e $(L-1)(M-1)$ graus de liberdade, onde

$$X_F = \frac{12 \cdot M}{K(K+1)} \left[\sum_i^N R(\mathcal{L}_i)^2 - \frac{K(K+1)^2}{4} \right]. \quad (4.16)$$

Em nosso experimento, a estatística F_F é distribuída de acordo com a distribuição F com 4 graus de liberdade para o numerador e 64 para o denominador. Considerando o valor crítico $F(4,64)$ é 2.515 para $\alpha = 0.05$, a linha final da

4.1 Um decisor de margem larga para o método MOBJ

Tabela 4.4 representa a média do *rank* ($R(\mathcal{L})$) alcançada por todos os métodos. O valor correspondente a F_F é 1.967 e assim, desde que $F_F < F(4, 64)$, H_0 não pode ser rejeitada, a um nível de significância (α) de 5%. Com base nos valores das estatísticas F_F para a AUC, pode-se afirmar que os métodos testados são estatisticamente equivalentes.

Afim de explorar a principal vantagem do decisor MOBJ-clas, que é a utilização de todo o conjunto de treinamento, foi realizado um novo teste estatístico, onde foram retiradas as bases com maior número de elementos. As bases selecionadas foram *banknote* e *german*, contendo 1372 e 1000 elementos, respectivamente.

Neste segundo experimento, a estatística F_F é distribuída de acordo com a distribuição F com 4 graus de liberdade para o numerador e 56 para o denominador. Considerando que o valor crítico $F(4, 64)$ é 2.536 para $\alpha = 0.05$, o valor correspondente a F_F é 2.671 e assim, desde que $F_F > F(4, 64)$, H_0 é rejeitada com um nível de significância (α) de 5%. Segundo Demšar (2006), caso a hipótese H_0 for rejeitada, segue-se então com o teste *post-hoc*. Ainda de acordo com Demšar (2006), dois classificadores são estatisticamente diferentes, se os *ranks* médios de dois algoritmos diferem de pelo menos,

$$CD = q_\alpha \sqrt{\frac{K(K+1)}{6N}}, \quad (4.17)$$

a diferença é considerada significativa com nível de confiança de $1-\alpha$. Para encontrar o valor crítico CD da Equação (4.17) foi utilizado o teste de *Bonferroni-Dunn* (Demšar, 2006) para 5 classificadores. A Figura 4.7 mostra um diagrama com o valor da média de cada método. Dois métodos são estatisticamente diferentes se o intervalo entre eles é disjunto, o que acontece com o método MOBJ-VAL em relação ao método RBF-clas. A partir deste teste, pode-se concluir que o método MOBJ-clas possui a melhor média entre todos os métodos comparados.

Os resultados mostram que nossa abordagem, para este experimento, foi estatisticamente equivalente aos seguintes métodos: SVM-RBF, SVM-Poly, LSSVM-RBF. O Decisor proposto provou ser melhor estatisticamente que o decisor de validação ao utilizar bases com menor número de amostras.

4.2 Encontrando parâmetros do *Kernel* Gaussiano de uma rede RBF

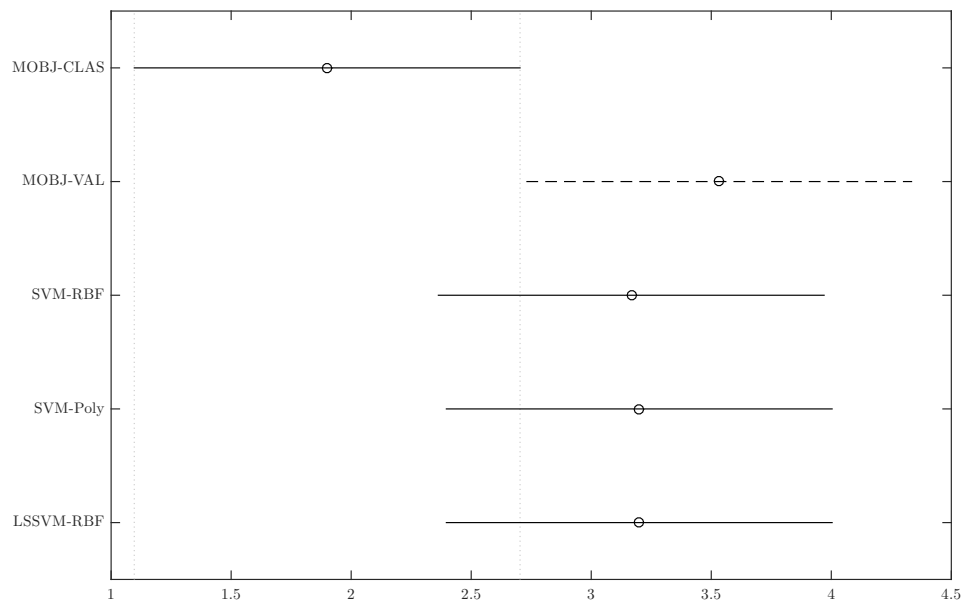


Figura 4.7: Diagrama de diferença crítica do teste *post-hoc*. Os métodos representados pelas linhas horizontais pontilhadas são estatisticamente diferentes do método MOBJ-clas.

4.2 Encontrando parâmetros do *Kernel* Gaussiano de uma rede RBF

Através de uma função não linear $\varphi(\cdot)$, a rede neural *Radial Basis Function* (RBF) realiza o mapeamento dos dados do espaço de entrada \mathbb{R}^d para um espaço de alta dimensionalidade \mathbb{R}^{d^+} , tal que $d^+ \gg d$. De acordo com o teorema de Cover (Cover, 1965), neste novo espaço, o problema pode ser separado linearmente de forma mais simples. Este princípio é compartilhado pelas *Support Vector Machines* (SVM) (Cortes & Vapnik, 1995), que utilizam *kernels* previamente ajustados para a indução de um mapeamento não linear, seguido da estimação de um hiperplano de margem máxima.

Em contraste com a rede *Multilayer Perceptron* (MLP), que pode ser projetada com mais de uma camada escondida, a rede RBF, geralmente, é formulada com apenas uma camada escondida, como mostra a Figura 4.8. O primeiro passo para a construção da topologia da rede RBF é encontrar o número de neurônios da camada escondida que corresponde ao número de funções base $\varphi(\cdot)$. Nor-

4.2 Encontrando parâmetros do *Kernel* Gaussiano de uma rede RBF

malmente, $\varphi(\cdot)$ é descrita como uma função gaussiana $\varphi(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{c}_i\|^2}{2\sigma_i^2}\right)$, sendo, portanto, necessário definir os valores do centro \mathbf{c}_i e do raio σ_i para cada i -ésimo neurônio oculto. Estudos propostos na literatura para determinação dos centros e, conseqüentemente, do número de neurônios ocultos, são baseados nos seguintes métodos: K -médias e suas variações (Sing *et al.*, 2003), *Fuzzy C-means*(FCM) (Chiu, 1994), redes *Self-Organizing Maps*(SOM) (Kohonen, 1990) e *Winner-takes-all*(WTA) (Duda *et al.*, 2012). É importante ressaltar que, no uso dos métodos k -médias e FCM, é necessário definir pelo menos um parâmetro *a priori*: o número k de centros. Os raios σ_i das funções $\varphi(\cdot)$ são normalmente definidos de forma empírica ou com base nas distâncias euclidianas entre os centros. Outra forma de se encontrar os parâmetros \mathbf{c}_i e σ_i é através do uso de validação cruzada (Kohavi *et al.*, 1995).

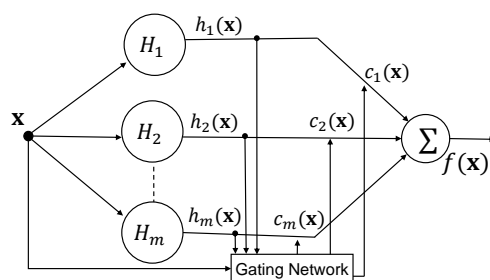


Figura 4.8: Topologia da Rede RBF.

As próximas Seções apresentam uma nova estratégia direcionada a problemas de classificação, onde valores ótimos de \mathbf{c}_i e σ_i são encontrados sem a necessidade de qualquer informação *a priori*. Usando ferramentas da Geometria Computacional (Berg *et al.*, 2008), o método proposto busca pelos padrões pertencentes à margem de separação das classes, de forma que estes exemplos são assinalados como centros das funções base $\varphi(\cdot)$ de uma rede RBF. Além de não necessitar de parâmetros adicionais para a obtenção dos centros, esta abordagem conta com as coordenadas geométricas de cada exemplo da margem para encontrar o respectivo σ_i de cada neurônio.

4.2 Encontrando parâmetros do *Kernel* Gaussiano de uma rede RBF

4.2.1 Rede Neural RBF

Uma rede neural RBF (*Radial Basis Function*) é considerada uma rede neural artificial que utiliza funções de bases radiais como funções de ativação dos neurônios da camada escondida. Formalmente, a rede RBF é construída utilizando somente três camadas, como mostra a Figura 4.8. A primeira camada é responsável pela entrada dos padrões na rede. A segunda, também conhecida como camada escondida, realiza o mapeamento dos padrões de entrada para um espaço de alta dimensionalidade, utilizando funções de ativação de bases radiais não lineares. E a terceira, a camada de saída, é responsável pela resposta da rede após os estímulos apresentados pelos padrões de entrada.

De acordo com o teorema de Cover (Cover, 1965), o problema de classificação inicialmente formulado no espaço de entrada, que é posteriormente transformado em um espaço de alta dimensão, torna-se mais fácil de ser separado linearmente. Na formulação da rede RBF, o problema de classificação é transformado em um problema de aproximação de função em um espaço multi-dimensional. Neste novo espaço, também conhecido por espaço de características, o objetivo é encontrar o modelo que gera a melhor aproximação dos dados de treinamento (Haykin, 1999).

Para realizar o mapeamento dos dados para o espaço de características, a RBF faz uso de funções bases diferente das sigmóides utilizadas na rede MLP. Estas funções bases são utilizadas como funções de ativação dos neurônios da camada intermediária da rede (segunda camada), vide Figura 4.8. Dado o conjunto de dados $\mathcal{S} = \{\mathbf{x}_i, \mathbf{y}_i | i = 1 \dots n\}$ onde $\mathbf{x}_i = \{\mathbf{x}_i \in \mathbb{R}^d | i = 1, 2 \dots n\}$ e $\mathbf{y}_i \in \{-1, 1\}$, sendo n o tamanho de \mathcal{S} . O problema se traduz em minimizar a função

$$\arg \min_{\mathbf{w}} \sum_{i=1}^n \|\hat{F}(\mathbf{x}_i) - \mathbf{y}_i\|, \quad (4.18)$$

em que

$$\hat{F}(\mathbf{x}) = \sum_{i=1}^m \mathbf{w}_i \varphi(\mathbf{x}, c_i), \quad (4.19)$$

$$\varphi(\mathbf{x}, \mathbf{c}_i) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{c}_i\|^2}{2\sigma_i^2}\right), \quad (4.20)$$

4.2 Encontrando parâmetros do *Kernel* Gaussiano de uma rede RBF

\mathbf{c}_i é o centro da função de ativação do neurônio i , e σ_i é o raio de abrangência da função.

Para encontrar os parâmetros σ_i e \mathbf{c}_i , utilizados na função de ativação gaussiana mostrada na Equação (4.20), são utilizadas heurísticas. Os métodos mais utilizados são: K-médias, FCM, rede SOM e WTA. Na maioria das vezes, esses métodos necessitam de mais parâmetros além do número de centros já definidos. Após encontrados estes parâmetros, o vetor de pesos \mathbf{w} que minimiza a Equação (4.18) é calculado através da Equação (4.21).

$$\mathbf{w} = \Psi^\dagger \mathbf{y}, \quad (4.21)$$

onde

$$\Psi = \begin{pmatrix} \beta_{11} & \beta_{12} & \cdots & \beta_{1N} \\ \beta_{21} & \beta_{2,2} & \cdots & \beta_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{m1} & \beta_{m2} & \cdots & \beta_{mN} \end{pmatrix}, \quad (4.22)$$

$$\beta_{ij} = \mathbf{w}_i \varphi(\mathbf{x}_j, \mathbf{c}_i).$$

e Ψ^\dagger é a pseudo-inversa linear *Moore-Penrose* (Albert, 1972) de Ψ .

4.2.2 Metodologia

Dado o conjunto de arestas de suporte \mathcal{AS} representado o conjunto de dados $\mathcal{S} = \{(\mathbf{x}, \mathbf{y}) \mid i = 1 \dots n\}$, sendo $\mathbf{x}_i \in \mathbb{R}^d$, $\mathbf{y}_i = \{-1, 1\}$ e n representando o tamanho de \mathcal{S} . Cada par de vértices que forma uma aresta de \mathcal{AS} é utilizado como centro da função de ativação dos neurônios da rede RBF, ou seja, cada exemplo que compõe \mathcal{AS} é um centro $\mathbf{c}_i \in \mathcal{AS}$ para função de ativação φ_i de um neurônio da rede. O número de exemplos no conjunto \mathcal{AS} é igual ao número m de neurônios da camada escondida da rede RBF. O raio σ_i do i -ésimo neurônio da rede é dado como $\Sigma_i = I_d \cdot \sigma^2$, ou seja, o centro de cada uma dessas hipersferas é associado à coordenada de um vértice de \mathcal{AS} . Baseando-se na metodologia encontrada em Bishop (1995), onde o σ é igual a duas vezes o espaçamento médio

4.2 Encontrando parâmetros do *Kernel* Gaussiano de uma rede RBF

entre os centros. Nesta abordagem, o valor do σ_i é dado pelo valor do diâmetro da hiperesfera calculado como

$$\sigma_i = 2\mathcal{R}_i \quad (4.23)$$

onde o raio

$$\mathcal{R}_i = \frac{1}{2}(\mathbf{c}_i + \mathbf{d}_i), \quad \forall i (\mathbf{c}_i, \mathbf{d}_i) \in \mathcal{AS}_i, \quad (4.24)$$

sendo $(\mathbf{c}_i, \mathbf{d}_i) \in \mathcal{AS}_i$, um par de vértices de uma aresta de suporte. A Figura 4.9 mostra os respectivos centros \mathbf{c}_i e raios σ_i de cada neurônio, onde $(\mathbf{d}_i = \sigma_i)$, e \mathbf{j} é o ponto médio mais próximo de \mathbf{c}_i . A Figura 4.10 apresenta o resultado do método aplicado ao *benchmark Two Moons*.

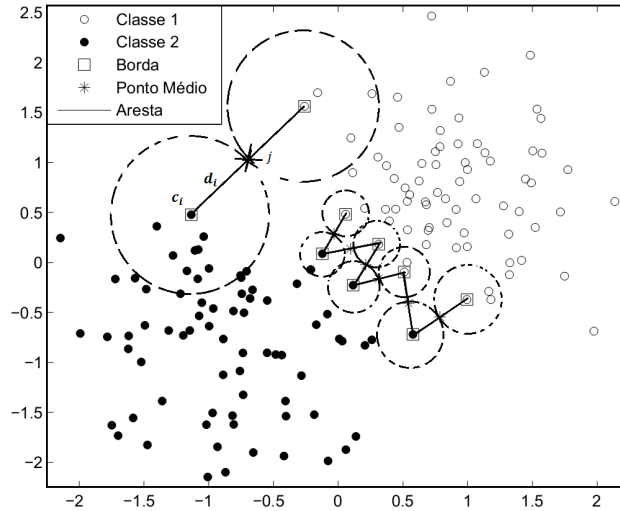


Figura 4.9: Centros e raios das funções de ativação dos neurônios da camada intermediária.

4.2 Encontrando parâmetros do *Kernel* Gaussiano de uma rede RBF

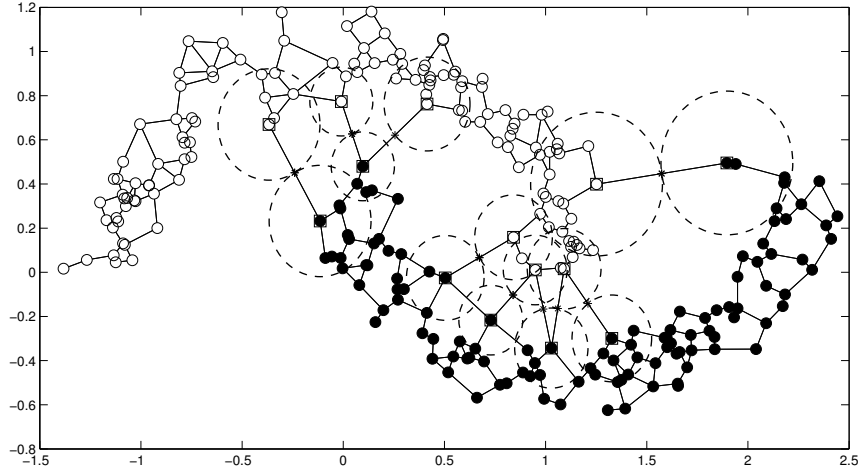


Figura 4.10: Metodologia proposta aplicada ao *benchmark Two Moons*.

4.2.3 Resultados

As bases de dados utilizadas neste método estão descritas na Seção 4.1.3, juntamente com a metodologia de processamento das mesmas. Afim de testar o desempenho do método RBF-clas, foram realizados 2 experimentos.

4.2.3.1 Experimento I

No primeiro experimento, para selecionar o número de centros da rede RBF foram utilizados os seguintes métodos: *Silhouette* (Rousseeuw, 1987), *Affinity propagation* (Frey & Dueck, 2007) e um método aleatório. Em seguida, foram utilizados os métodos K-médias e FCM para seleção dos centros. O nome utilizado para cada metodologia está descrito na Tabela 4.2.

Tabela 4.2: Representação dos nomes utilizados em cada metodologia para seleção e número K de centros para serem utilizados no projeto da rede RBF.

Nome	Método para seleção dos centros.	Método para seleção do número dos centros.
K- <i>Silhouette</i>	K-médias	Silhouette
K- <i>Affinity propagation</i>	K-médias	Affinity
K-Aleatório	K-médias	Aleatório
F- <i>Silhouette</i>	FCM	Silhouette
F- <i>Affinity propagation</i>	FCM	Affinity
F-Aleatório	FCM	Aleatório

4.2 Encontrando parâmetros do *Kernel* Gaussiano de uma rede RBF

Para encontrar o valor do σ foram utilizadas duas metodologias descritas em Bishop (1995). Na primeira metodologia, o valor do sigma σ é igual a duas vezes o espaçamento médio entre os centros, ou seja,

$$\frac{2}{K} \sum_{i=1}^K \sum_{j=i+1}^K \|\mathbf{c}_i - \mathbf{c}_j\|, \quad (4.25)$$

onde K é o número de centros e $\|\cdot\|$ o operador que realiza a distância Euclidiana. Na segunda, o valor de σ é encontrado através da média dos L vizinhos mais próximos de cada centro. Neste experimento, foi utilizado $L = 1$, onde cada σ_i pode ser encontrado como

$$\sigma_i = \min(\mathbf{c}_i, \mathcal{S}), \quad \forall i. \quad (4.26)$$

As Tabelas 4.3 e 4.4 mostram os valores da média da AUC e desvio padrão obtidos pelas metodologias RBF-clas, K-*Silhouette*, K-*Affinity propagation*, K-Aleatório, F-*Silhouette*, F-*Affinity propagation* e F-Aleatório, sobre as 17 bases de dados apresentadas na Seção 4.1.3. Para os resultados da Tabela 4.3, foi utilizada a metodologia para encontrar o σ de acordo com a Equação (4.25). Já para Tabela 4.4, foi utilizada a Equação (4.26) para encontrar o valor do σ . Os melhores valores encontram-se em negrito. As quatro últimas colunas das Tabelas 4.3 e 4.4 mostram as características de cada base, em que N_d é o número de dimensões, N é o número total de amostras. A quantidade de elementos de cada classe da base de dados são denotados por N^+ e N^- respectivamente.

4.2 Encontrando parâmetros do *Kernel* Gaussiano de uma rede RBF

Tabela 4.3: Resultados para o método RBF-clas: média da AUC e características das bases de dados. Foi utilizada a Equação (4.25) para encontrar o valor do σ .

Base de Dados	RBF-clas	K-Silhouette	K-Affinity	K-Aleatório	F-Silhouette	F-Affinity	F-Aleatório	N_d	N	N^+	N^-
<i>appendicitis</i>	0.818±0.17	0.742±0.199	0.793±0.164	0.569±0.21	0.717±0.227	0.793±0.169	0.668±0.236	7	106	21	85
<i>australian</i>	0.854±0.05	0.868±0.057	0.871±0.047	0.725±0.097	0.868±0.053	0.861±0.042	0.844±0.048	14	690	307	383
<i>banknote</i>	0.999±0.003	0.926±0.042	1±0	1±0	0.94±0.026	1±0	1±0	4	1372	610	762
<i>blood</i>	0.682±0.06	0.5±0	0.628±0.058	0.632±0.082	0.5±0	0.636±0.06	0.633±0.079	4	748	178	570
<i>breastcancer</i>	0.968±0.029	0.968±0.017	0.965±0.022	0.922±0.06	0.968±0.017	0.966±0.021	0.969±0.025	9	683	444	239
<i>breastHess</i>	0.816±0.113	0.756±0.128	0.79±0.122	0.762±0.17	0.812±0.114	0.807±0.094	0.714±0.135	30	133	99	34
<i>bupa</i>	0.567±0.095	0.491±0.018	0.698±0.056	0.614±0.099	0.483±0.026	0.733±0.045	0.645±0.064	6	345	145	200
<i>climate</i>	0.818±0.132	0.5±0	0.584±0.104	0.739±0.13	0.5±0	0.5±0	0.5±0	18	540	494	46
<i>diabetes</i>	0.733±0.051	0.657±0.056	0.712±0.07	0.624±0.113	0.689±0.056	0.7±0.063	0.708±0.053	8	768	500	268
<i>fertility</i>	0.522±0.174	0.5±0	0.5±0	0.504±0.189	0.5±0	0.5±0	0.483±0.118	9	100	12	88
<i>german</i>	0.704±0.075	0.574±0.041	0.667±0.058	0.651±0.064	0.5±0	0.5±0	0.5±0	24	1000	700	300
<i>glass</i>	0.851±0.207	0.819±0.206	0.883±0.159	0.872±0.157	0.829±0.205	0.886±0.161	0.867±0.156	9	214	29	185
<i>haberman</i>	0.56±0.091	0.5±0	0.6±0.071	0.55±0.069	0.5±0	0.605±0.085	0.545±0.096	3	306	225	81
<i>heart</i>	0.815±0.082	0.81±0.082	0.82±0.077	0.71±0.083	0.818±0.092	0.813±0.087	0.792±0.102	13	270	150	120
<i>ILPD</i>	0.594±0.059	0.5±0	0.54±0.04	0.53±0.064	0.5±0	0.558±0.055	0.567±0.048	10	579	414	165
<i>parkinsons</i>	0.784±0.117	0.694±0.067	0.744±0.076	0.836±0.086	0.69±0.07	0.739±0.06	0.832±0.117	22	195	147	48
<i>upbc</i>	0.579±0.159	0.5±0	0.582±0.12	0.637±0.134	0.5±0	0.568±0.086	0.551±0.11	33	194	46	148
Média do Rank $R(\mathcal{L})$	2.529	5.441	2.911	4.205	5.147	3.411	4.352				

Tabela 4.4: Resultados para o método RBF-clas: média da AUC e características das bases de dados. Foi utilizada a metodologia para encontrar o σ de acordo com a Equação (4.26).

Base de Dados	RBF-clas	K-Silhouette	K-Affinity	K-Aleatório	F-Silhouette	F-Affinity	F-Aleatório	N_d	N	N^+	N^-
<i>appendicitis</i>	0.818±0.17	0.711±0.214	0.787±0.159	0.638±0.163	0.711±0.214	0.788±0.166	0.705±0.195	7	106	21	85
<i>australian</i>	0.854±0.05	0.858±0.053	0.859±0.044	0.761±0.066	0.873±0.044	0.867±0.045	0.837±0.058	14	690	307	383
<i>banknote</i>	0.999±0.003	0.883±0.039	1±0	0.994±0.009	0.885±0.036	1±0	0.996±0.007	4	1372	610	762
<i>blood</i>	0.682±0.06	0.5±0	0.624±0.066	0.584±0.071	0.5±0	0.632±0.053	0.577±0.063	4	748	178	570
<i>breastcancer</i>	0.968±0.029	0.97±0.022	0.97±0.022	0.94±0.044	0.971±0.021	0.973±0.015	0.975±0.024	9	683	444	239
<i>breastHess</i>	0.816±0.113	0.807±0.12	0.769±0.086	0.751±0.145	0.801±0.126	0.802±0.087	0.71±0.143	30	133	99	34
<i>bupa</i>	0.567±0.095	0.477±0.044	0.674±0.053	0.606±0.121	0.484±0.033	0.614±0.088	0.615±0.085	6	345	145	200
<i>climate</i>	0.818±0.132	0.5±0	0.59±0.102	0.664±0.148	0.5±0	0.562±0.08	0.743±0.125	18	540	494	46
<i>diabetes</i>	0.733±0.051	0.639±0.054	0.727±0.058	0.694±0.039	0.667±0.041	0.745±0.035	0.71±0.063	8	768	500	268
<i>fertility</i>	0.522±0.174	0.5±0	0.5±0	0.51±0.176	0.5±0	0.5±0	0.56±0.178	9	100	12	88
<i>german</i>	0.704±0.075	0.57±0.066	0.668±0.052	0.646±0.055	0.579±0.071	0.653±0.033	0.655±0.043	24	1000	700	300
<i>glass</i>	0.851±0.207	0.786±0.226	0.911±0.157	0.874±0.154	0.819±0.201	0.886±0.161	0.859±0.174	9	214	29	185
<i>haberman</i>	0.56±0.091	0.5±0	0.567±0.075	0.541±0.104	0.5±0	0.533±0.046	0.607±0.07	3	306	225	81
<i>heart</i>	0.815±0.082	0.819±0.097	0.81±0.084	0.741±0.063	0.81±0.097	0.803±0.094	0.792±0.084	13	270	150	120
<i>ILPD</i>	0.594±0.059	0.5±0	0.513±0.063	0.553±0.084	0.5±0	0.524±0.074	0.574±0.066	10	579	414	165
<i>parkinsons</i>	0.784±0.117	0.688±0.069	0.749±0.094	0.805±0.114	0.663±0.094	0.793±0.105	0.852±0.109	22	195	147	48
<i>upbc</i>	0.579±0.159	0.5±0	0.616±0.083	0.625±0.115	0.5±0	0.57±0.109	0.581±0.096	33	194	46	148
Média do Rank $R(\mathcal{L})$	2.764	5.588	3.176	4.470	5.352	3.235	3.411				

4.2.3.2 Experimento II

No segundo experimento, os métodos SVM e LSSVM-RBF (Suykens & Vandewalle, 1999) foram utilizados para comparar o nosso decisor (RBF-clas). Duas configurações de *Kernels* foram utilizadas. A primeira, SVM-RBF e LSSVM-RBF com função de bases radiais, e SVM-Poly com função polinomial. Os parâmetros de *Kernel* e regularização para SVM-RBF e SVM-Poly foram encontrados através de validação cruzada com 10 partições e busca em *grid*. A implementação destes métodos se deu através dos pacotes *Kernlab* (Karatzoglou *et al.*, 2004) e *Caret* (Kuhn, 2008) disponíveis para linguagem *R* (R Core Team, 2015). A

4.2 Encontrando parâmetros do *Kernel* Gaussiano de uma rede RBF

Tabela 4.5 mostra os valores da média da AUC e desvio padrão obtidos pelos métodos RBF-clas, SVM-RBF, SVM-Poly e LSSVM-RBF sobre as 17 bases de dados. Os melhores valores encontram-se em negrito. As quatro últimas colunas desta tabela mostram as características de cada base, em que N_d é o número de dimensões, N é o número total de amostras. A quantidade de elementos de cada classe da base de dados são denotados por N^+ e N^- respectivamente.

Tabela 4.5: Resultados do método RBF-clas: média da AUC e características das bases de dados.

Base de Dados	RBF-clas	SVM-RBF	SVM-Poly	LSSVM-RBF	N_d	N	N^+	N^-
<i>appendicitis</i>	0.818±0.17	0.707±0.224	0.718±0.222	0.687±0.234	7	106	21	85
<i>australian</i>	0.854±0.05	0.863±0.059	0.862±0.056	0.862±0.042	14	690	307	383
<i>banknote</i>	0.999±0.003	1±0	1±0	0.999±0.004	4	1372	610	762
<i>blood</i>	0.682±0.06	0.63±0.058	0.624±0.043	0.616±0.045	4	748	178	570
<i>breastcancer</i>	0.968±0.029	0.969±0.02	0.967±0.02	0.955±0.027	9	683	444	239
<i>breastHess</i>	0.816±0.113	0.791±0.115	0.79±0.093	0.735±0.098	30	133	99	34
<i>bupa</i>	0.567±0.095	0.663±0.053	0.665±0.052	0.674±0.076	6	345	145	200
<i>climate</i>	0.818±0.132	0.525±0.056	0.55±0.09	0.645±0.169	18	540	494	46
<i>diabetes</i>	0.733±0.051	0.711±0.048	0.716±0.05	0.714±0.058	8	768	500	268
<i>fertility</i>	0.522±0.174	0.5±0	0.5±0	0.578±0.184	9	100	12	88
<i>german</i>	0.704±0.075	0.655±0.038	0.658±0.038	0.657±0.048	24	1000	700	300
<i>glass</i>	0.851±0.207	0.846±0.202	0.846±0.202	0.822±0.203	9	214	29	185
<i>haberman</i>	0.56±0.091	0.535±0.056	0.508±0.039	0.552±0.092	3	306	225	81
<i>heart</i>	0.815±0.082	0.832±0.085	0.832±0.082	0.828±0.071	13	270	150	120
<i>ILPD</i>	0.594±0.059	0.502±0.006	0.5±0	0.572±0.07	10	579	414	165
<i>parkinsons</i>	0.784±0.117	0.753±0.062	0.769±0.059	0.813±0.116	22	195	147	48
<i>wdbc</i>	0.579±0.159	0.496±0.011	0.493±0.014	0.576±0.112	33	194	46	148
Média do Rank $R(\mathcal{L})$	1.852	2.705	2.735	2.705				

4.2.4 Teste de significância para o experimento I

O teste de Friedman (Demšar, 2006) foi aplicado nos resultados da Tabela 4.3 e Tabela 4.4, assumindo a hipótese nula H_0 que todos os métodos são equivalentes. Em nosso experimento, a estatística F_F (Equação 4.15) é distribuída de acordo com a distribuição F de Fisher-Snedecor com 6 graus de liberdade para o numerador e 96 para o denominador. Considerando que o valor crítico $F(6, 96)$ é 2.194 para $\alpha = 0.05$, as linhas finais da Tabela 4.3 e da Tabela 4.4 representam a média do *rank* ($R(\mathcal{L})$) alcançada por todos os métodos.

4.2 Encontrando parâmetros do *Kernel* Gaussiano de uma rede RBF

Metodologia I para escolha do σ conforme a Equação (4.25).

O valor correspondente a F_F é 5.593 e assim, desde que $F_F > F(6, 96)$, H_0 é rejeitada, a um nível de significância (α) de 5%. Seguindo com o teste *post-hoc*, foi utilizado o teste de *Bonferroni-Dunn* para 7 classificadores. A Figura 4.11 mostra um diagrama com o valor da média de cada método. Dois métodos são estatisticamente diferentes se o intervalo entre eles é disjunto. O que acontece com os métodos *K-Silhouette* e *F-Silhouette* em relação ao método RBF-clas. A partir deste teste podemos concluir que o método RBF-clas possui a melhor média entre todos os métodos comparados.

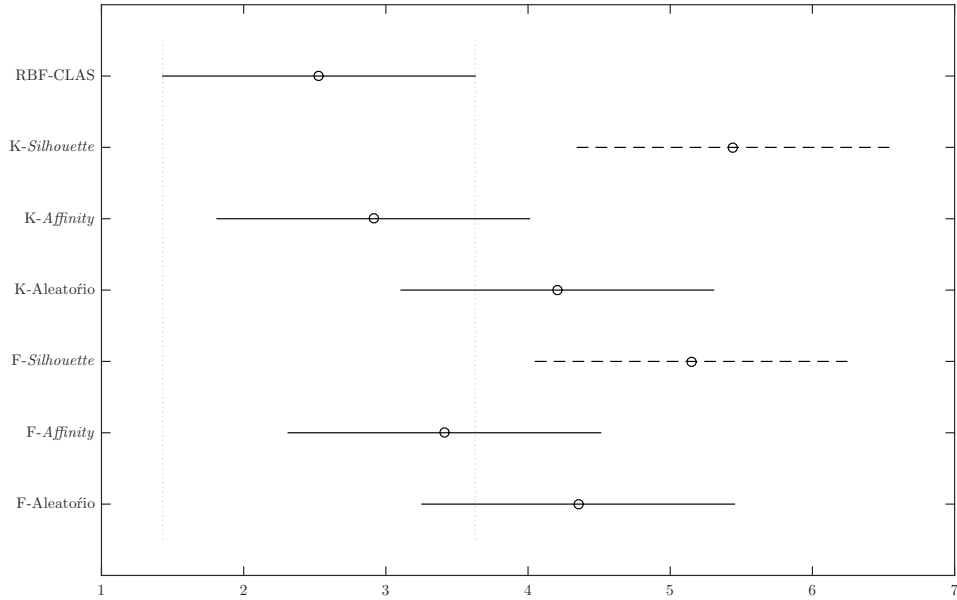


Figura 4.11: Diagrama de diferença crítica do teste *post-hoc*. Os métodos representados pelas linhas horizontais pontilhadas são estatisticamente diferentes do método RBF-clas.

Metodologia II para escolha do σ conforme a Equação (4.26).

O valor correspondente a F_F é 6.079 e assim, desde que $F_F > F(6, 96)$, H_0 é rejeitada com um nível de significância (α) de 5%. Seguindo com o teste *post-hoc*, foi utilizado o teste de *Bonferroni-Dunn* para 7 classificadores. A Figura 4.12 mostra um diagrama com o valor da média de cada método. Dois métodos são

4.2 Encontrando parâmetros do *Kernel* Gaussiano de uma rede RBF

estatisticamente diferentes se o intervalo entre eles é disjunto. O que acontece com os métodos *K-Silhouette* e *F-Silhouette* em relação ao método RBF-clas. A partir deste teste podemos concluir que o método RBF-clas possui a melhor média entre todos os métodos comparados.

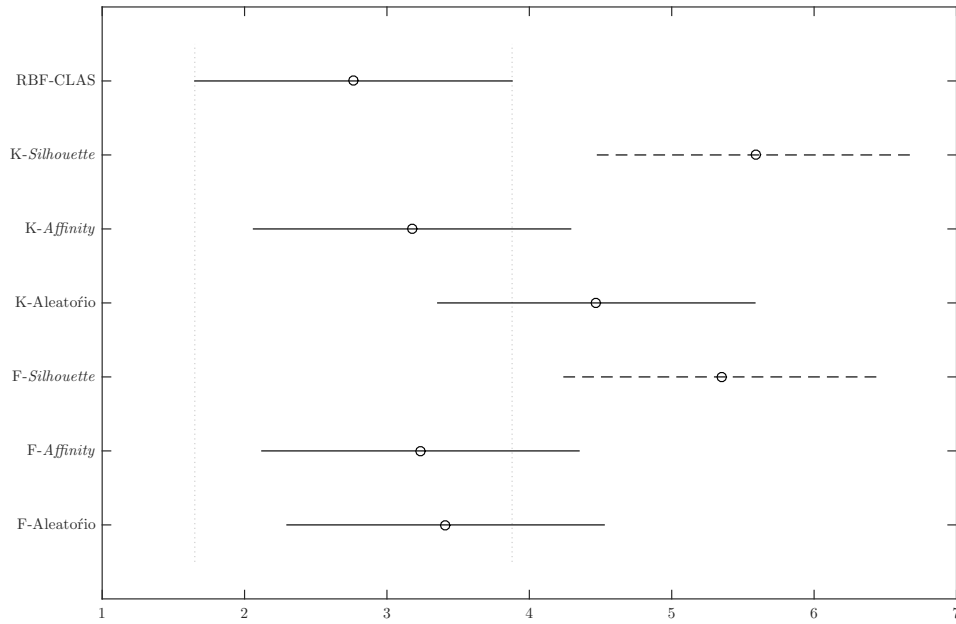


Figura 4.12: Diagrama de diferença crítica do teste *post-hoc*. Os métodos representados pelas linhas horizontais pontilhadas são estatisticamente diferentes do método RBF-clas

4.2.5 Teste de significância para o experimento II

O teste de Friedman (Demšar, 2006) foi aplicado nos resultados da Tabela 4.5, assumindo a hipótese nula H_0 que todos os métodos são equivalentes. Em nosso experimento, a estatística F_F (Equação 4.15) é distribuída de acordo com a distribuição F de Fisher-Snedecor com 3 graus de liberdade para o numerador e 48 para o denominador. Considerando que o valor crítico $F(3, 48)$ é 2.798 para $\alpha = 0.05$, a linha final da Tabela 4.5 representa a média do *rank* ($R(\mathcal{L})$) alcançada por todos os métodos. O valor correspondente a F_F é 2.013 e assim, desde que $F_F < F(3, 48)$, a hipótese H_0 não pode ser rejeitada.

4.3 Classificador geométrico de margem larga

De acordo com a Figura 4.13, nenhum método possui a média estatisticamente diferente do método RBF-clas. Embora a média do RBF-clas seja a menor dentre os métodos testados.

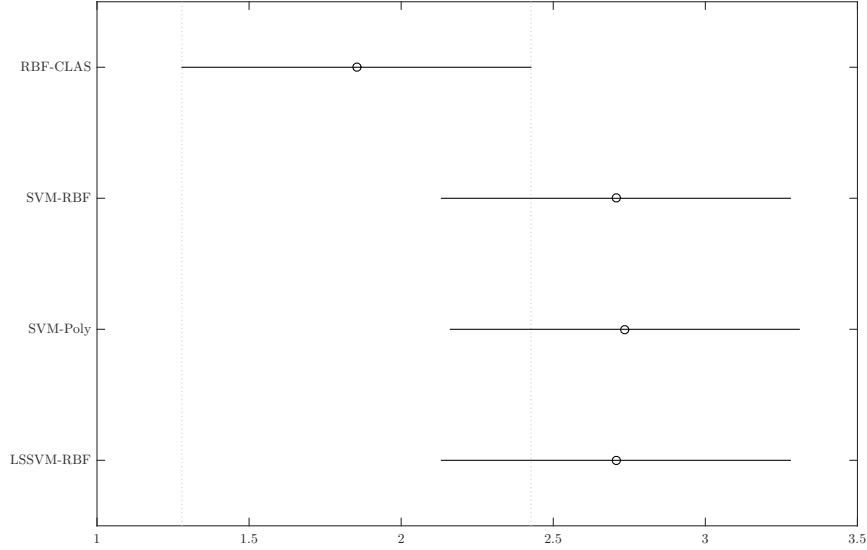


Figura 4.13: Diagrama de diferença crítica do teste *post-hoc*. Os métodos representados pelas linhas horizontais pontilhadas são estatisticamente diferentes do método RBF-clas. Neste caso, todos os métodos são equivalente.

4.3 Classificador geométrico de margem larga

Nesta abordagem, foi projetado um classificador de margem larga para problemas binários usando um modelo de mistura gaussiana (MMG). Uma vez encontrada a função de probabilidade de densidade para cada classe, uma classificação Bayesiana é aplicada. Baseado nos conceitos de margem larga das SVMs, o critério adotado para esta abordagem é que a superfície de separação passa o mais próximo possível dos pontos médios, assim maximizando a margem separação entre as classes. Isto pode ser alcançado modelando o problema através de duas componentes de MMG, onde cada componente representa uma função de distribuição normal multivariada (FDM) mostrada na Figura 4.14. Neste modelo, os pares de \mathcal{AS} representam o centro (média) de cada FDM, e sua distribuição tem desvio padrão de 3σ , assim representando 99.7% dos dados. Esta definição é dada

observando que valor de $p(\mathbf{x}|C)$ é arbitrariamente pequeno, onde \mathbf{x} é um ponto médio, ou seja, esta é a região onde deseja-se que o hiperplano intersecte.

Seguindo este conceito pode-se extrair as informações para configurar cada MMG atendendo aos critérios estabelecidos acima. Uma vez que todas as informações necessárias para projetar o classificador são extraídas a partir da estrutura geométrica dos próprios dados.

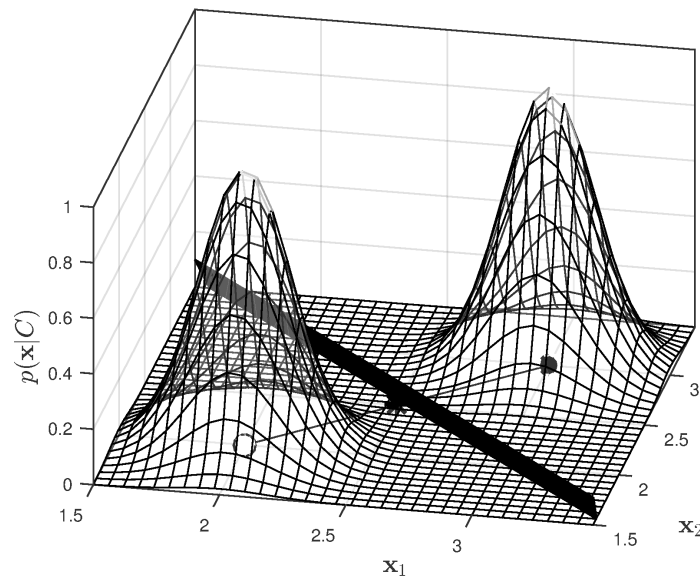


Figura 4.14: Superfície de separação de duas gaussianas, onde o hiperplano intersecta o ponto médio entre um par de vértices de uma aresta de suporte.

4.3.1 Extração de parâmetros

A primeira informação que obtém-se são os padrões localizados na região de separação entre as classe, que são chamados neste trabalho de vetores de arestas de suporte (\mathcal{AS}). Estes padrões correspondem as arestas do grafo de Gabriel GG que possuem vértices de classes distintas, como descrito na Seção 3.1. Como mencionado anteriormente, a importância dos \mathcal{AS} para nosso método é similar as vetores de suporte da SVM. A próxima informação retirada do GG é necessária para modelar a função de distribuição normal multivariada (FDM), onde são necessários os parâmetros Σ e μ , que são a matriz de covariância e a média

4.3 Classificador geométrico de margem larga

respectivamente. A seguir será mostrado como estes padrões são selecionados utilizando o critério de margem larga. Inicia-se escrevendo a FDM em d dimensões como

$$p(\mathbf{x}|\mu, \Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp \left[-\frac{1}{2}(\mathbf{x} - \mu)^T \Sigma^{-1}(\mathbf{x} - \mu) \right], \quad (4.27)$$

nesta abordagem, a distribuição pode ser representada geometricamente por uma hipersfera, onde cada dimensão tem a mesma variância σ^2 . Assim, a matriz de covariância pode ser dada como $\Sigma = I_d \sigma^2$, onde I_d é uma matriz identidade $d \times d$ contendo o valor 1 na diagonal e o valor zero nas demais posições. Nesse caso, o determinante $|\Sigma|$ e a inversa Σ^{-1} podem ser calculados como $|\Sigma| = \sigma^2$ e $\Sigma^{-1} = 1/\sigma^2$, respectivamente (Duda *et al.*, 2012). Então, a Equação (4.27) pode ser reescrita como

$$p(\mathbf{x}|\mu, \Sigma) = \frac{1}{(2\pi)^{d/2} \sqrt{\sigma^{2d}}} \exp \left[-\frac{1}{2} \left(\frac{\|\mathbf{x} - \mu\|^2}{\sigma^2} \right) \right], \quad (4.28)$$

onde $\|\cdot\|$ a distância euclidiana, ou seja, $\|\mathbf{x} - \mu\|^2 = (\mathbf{x} - \mu)(\mathbf{x} - \mu)$. Como já foi mencionado, o valor de $p(\mathbf{x}|\mu, \Sigma)$ para o ponto médio \mathbf{x} possui um valor arbitrariamente pequeno, desde que esteja próximo da faixa $\mu + 3\sigma$ de distribuição, ou seja, $p(\mathbf{x}|\mu, \Sigma) = z$, onde $z \approx 0$. Removendo as constantes independentes de \mathbf{x} da Equação (4.28), o valor de σ pode ser encontrado por

$$p(\mathbf{x}|\mu, \Sigma) = \exp \left[-\frac{1}{2} \left(\frac{\|\mathbf{x} - \mu\|^2}{\sigma^2} \right) \right],$$

substituindo $p(\mathbf{x}|\mu, \Sigma)$ por z , tem-se

$$z = \exp \left[-\frac{1}{2} \left(\frac{\|\mathbf{x} - \mu\|^2}{\sigma^2} \right) \right],$$

isolando o termo σ , encontra-se

$$\sigma = \sqrt{-\frac{1}{2} \left(\frac{\|\mathbf{x} - \mu\|^2}{\ln(z)} \right)}, \quad (4.29)$$

onde z representa o valor de saída de $p(\mathbf{x}|\mu, \Sigma)$ para $\mathbf{x} = (3\sigma + \mu)$, $\mu = 0$ e $\sigma = 1$, ou seja, o valor da função de probabilidade de densidade de \mathbf{x} . Substituindo-se z

4.3 Classificador geométrico de margem larga

por $p(\mathbf{x}|\mu, \Sigma)$ na Equação (4.28), pode-se encontrar o valor de z no espaço \mathbb{R}^d da seguinte maneira:

$$z = \frac{1}{(2\pi)^{d/2}\sigma^{2d}} \exp \left[-\frac{1}{2} \left(\frac{\|\mathbf{x} - \mu\|^2}{\sigma^2} \right) \right],$$

fazendo $\sigma^{2d} = 1$ e $\mu = 0$, encontra-se

$$= \frac{1}{(2\pi)^{d/2}} \exp \left[-\frac{1}{2} \left(\frac{\|\mathbf{x}\|^2}{\sigma^2} \right) \right],$$

generalizando $\mathbf{x} = \|3\sigma + \mu\|^2$ em \mathbb{R}^d , obtém-se

$$\begin{aligned} &= \frac{1}{(2\pi)^{d/2}} \exp \left(-\frac{3\sqrt{d}}{2} \right), \\ &= \frac{\exp \left(-\frac{3}{2}\sqrt{d} \right)}{(2\pi)^{1/d}}, \end{aligned} \tag{4.30}$$

quando a dimensão d cresce, o valor de z decresce. Seja $f_z(d)$ representando z no espaço \mathbb{R}^d , então o limite de $f_z(d)$ em d aproximando-se do infinito é 0. Em notação matemática,

$$\lim_{d \rightarrow \infty} \frac{\exp \left(-\frac{3}{2}\sqrt{d} \right)}{(2\pi)^{1/d}} = 0.$$

4.3.2 Metodologia para construção do classificador

A partir das informações obtidas na Seção anterior pode-se formular o método em 3 passos que se seguem:

1. *Gaussiana Multivariada.* Para cada par $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{AS}$, tal que $y_i \neq y_j$, obter a densidade Gaussiana $G_\tau(\mathbf{x}, \mu_\tau, \Sigma_\tau)$, sendo a média $\mu_\tau = \mathbf{x}_\tau$ e a matriz de covariância diagonal $\Sigma_\tau = I_d \sigma^2$, para $\tau = i, j$. O parâmetro σ pode ser encontrado através da Equação (4.29).
2. *Mistura de Densidade por classe.* Calcule o modelo de densidade de mistura (McLachlan & Peel, 2004) para cada classe C_k a partir da soma ponderada

4.3 Classificador geométrico de margem larga

das funções Gaussianas a quais os centros pertencem a mesma classe, ou seja,

$$p(\mathbf{x}, \theta_k | C_k) = \sum_{j=1}^{N_k} w_j G_j(\mathbf{x}, \mu_j, \Sigma_j), \text{ for } k = 1, 2 \quad (4.31)$$

onde $\theta_k = [\{\mu_1, \Sigma_1\}, \dots, \{\mu_{N_k}, \Sigma_{N_k}\}]$ é o vetor de parâmetros de N_k vetores geométricos da classe C_k , w_j corresponde ao peso para o j -ésima densidade $G_j(\cdot)$, sujeito a $\sum_{j=1}^{N_k} w_j = 1$. As Figuras 4.15(a) e 4.15(b) ilustram os modelos de mistura $p(\mathbf{x}, \theta_1 | C_1)$ e $p(\mathbf{x}, \theta_2 | C_2)$ para o *benchmark Spiral*.

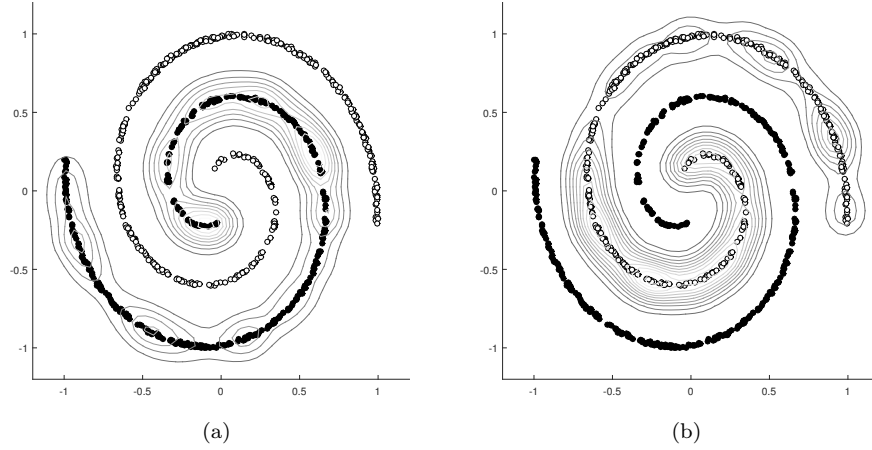


Figura 4.15: (a) Mistura de densidades de $f_{\mathcal{P}}(\mathbf{x}|\theta_A)$. (b) Mistura de densidades de $f_{\mathcal{P}}(\mathbf{x}|\theta_B)$.

3. *Regra de Decisão.* Uma regra de decisão Bayesiana é então formulada. Através do vetor de parâmetros θ_1 e θ_2 , conhecidos *a priori* e provenientes de um conjunto de dados \mathcal{S} , a classificação binária é feita minimizando a probabilidade do erro por intermédio da regra de decisão de Bayes (Duda *et al.*, 2012)

$$f(\mathbf{x}) = \begin{cases} C_1 & \text{se } \frac{p(\mathbf{x}, \theta_1 | C_1)}{p(\mathbf{x}, \theta_2 | C_2)} \geq \frac{P(C_2)}{P(C_1)} \\ C_2 & \text{Caso contrário.} \end{cases} \quad (4.32)$$

A Figura 4.16 mostra a solução final da nossa metodologia aplicado ao *benchmark Spiral*.

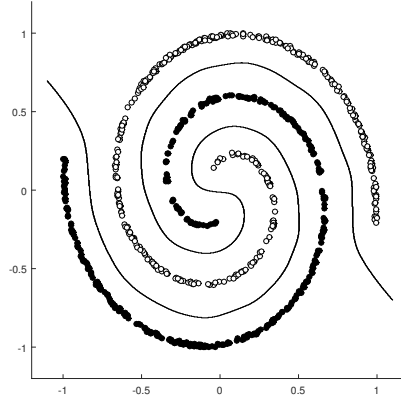


Figura 4.16: Hiperplano separador obtido através do nosso método.

4.3.3 Espaço de Verossimilhanças

No espaço de verossimilhanças pode-se ver a similaridade do nosso método com um simples problema de separação de duas classes Gaussianas mostrado na Figura 4.17(b). Projetando os dados do *benchmark Spiral* neste espaço com valor aleatório para o parâmetro σ da Equação (4.29), obtém-se a representação mostrada na Figura 4.18(a), onde os círculos representam os vértices das arestas de suporte. Já os pontos médios são representados através do símbolo asterisco. Na Figura 4.18(a), os pontos médios e os vértices do conjunto AS estão desordenados, diferentemente de quando projeta-se a mesma informação usando nossa abordagem para encontrar o valor do σ , que pode ser visto na Figura 4.19(a). A projeção é refletida na complexidade do classificador, como pode ser visto na Figura 4.19(b) através da nossa abordagem e na Figura 4.18(b) com σ aleatório. Por fim, através da Figura 4.17(a) é fácil observar a similaridade da nossa abordagem com o espaço projetado das duas Gaussianas.

4.3 Classificador geométrico de margem larga

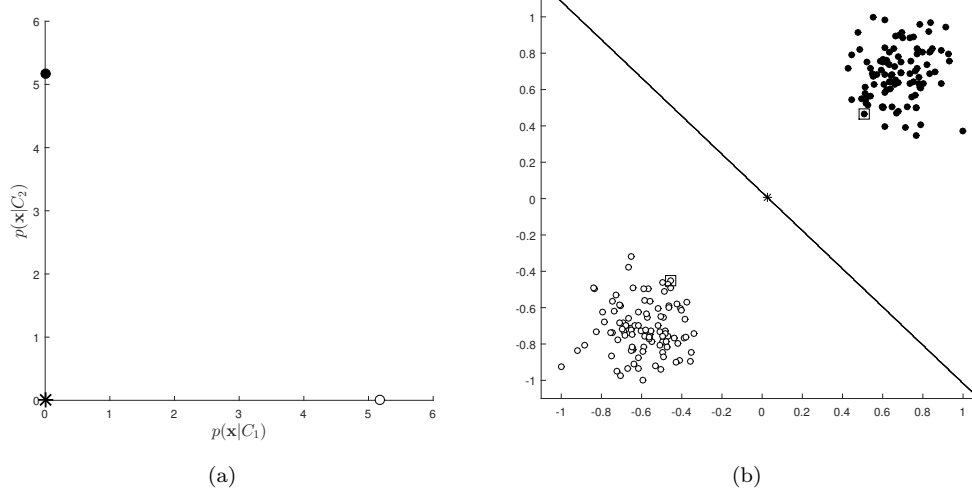
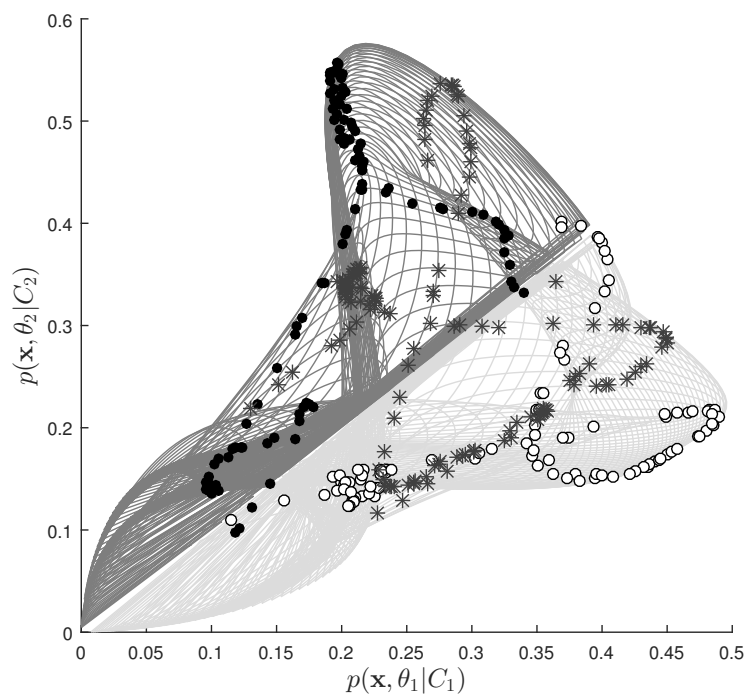
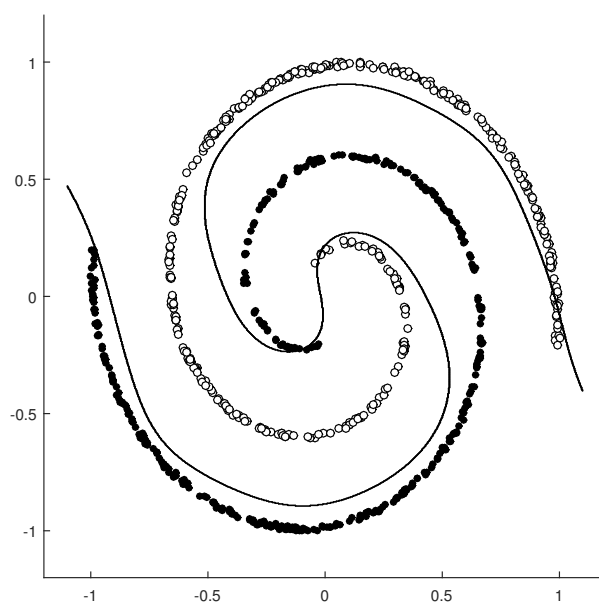


Figura 4.17: (a) Projeção no espaço de verossimilhanças de duas distribuições Gaussianas. (b) Hiperplano separador no espaço de entrada maximizando a margem entre as duas classes.

4.3 Classificador geométrico de margem larga

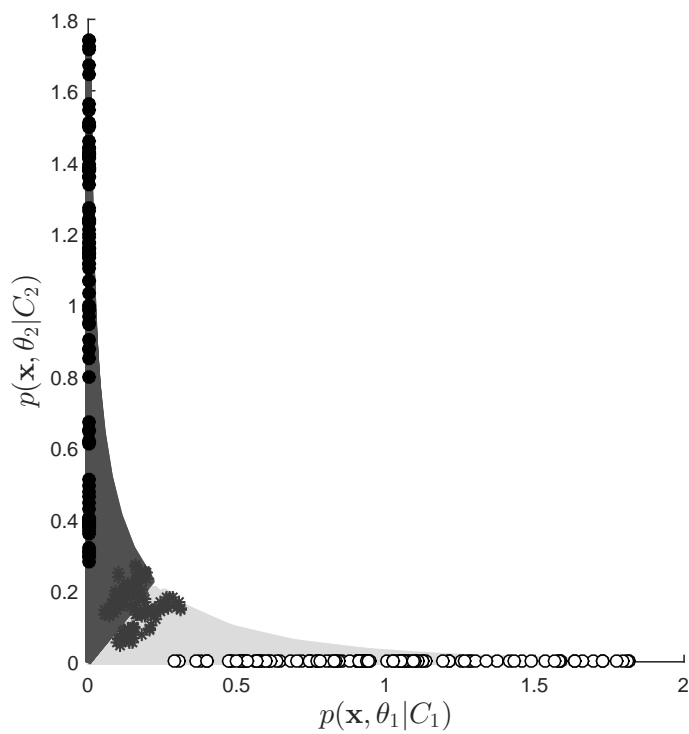


(a)

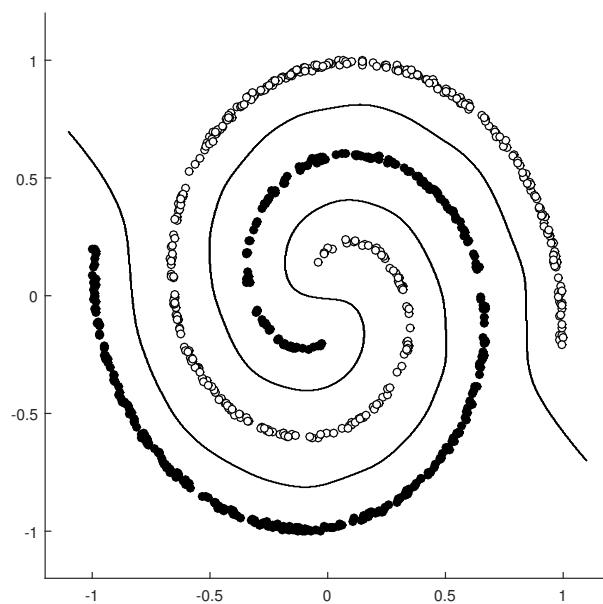


(b)

Figura 4.18: (a) Projeção no espaço de verossimilhanças utilizando σ aleatório
(b) Hiperplano separador no espaço de entrada fazendo σ aleatório.



(a)



(b)

Figura 4.19: (a) Projeção no espaço de verossimilhanças utilizando nossa abordagem (b) Hiperplano separador resultante da nossa abordagem.

4.3.4 Analogia com as SVMs

O resultado final da classificação da SVM para um padrão de entrada \mathbf{x}_i é obtido como

$$f(\mathbf{x}_i) = \text{sign} \left(\sum_j^N y_j \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \right), \quad (4.33)$$

que é, de fato, o sinal de uma soma ponderada dos rótulos y_j . Embora a soma é realizada sobre todos N padrões de treinamento, somente os termos associados aos vértices SVs, os quais tem valor não nulo α_j (Multiplicador de Lagrange), são na verdade computados. Se a magnitude dos termos positivos ($y_j = +1$) dominam a soma, então o resultado é positivo ($y_i = +1$); caso contrário, se os termos negativos dominam a soma ($y_j = -1$) então a saída é negativa ($y_i = -1$). A Equação (4.34) mostra a regra de classificação da SVM com os termos de soma positiva e negativa separadas, e os valores dos rótulos atribuídos.

$$f(\mathbf{x}_i) = \text{sign} \left(\overbrace{\sum_{j=1}^{N_1} \alpha_j K(\mathbf{x}_i, \mathbf{x}_j)}^{\text{Classe Positiva}} - \overbrace{\sum_{l=1}^{N_2} \alpha_l K(\mathbf{x}_i, \mathbf{x}_l)}^{\text{Classe Negativa}} \right) \quad (4.34)$$

O método apresentado tem uma regra de classificação análoga, uma vez que a regra geral de classificação de Bayes da Equação (4.32) possa ser reescrita como

$$f(\mathbf{x}_i) = \text{sign} \left(p(\mathbf{x}_i, \theta_1 | C_1) - \frac{P(C_2)}{P(C_1)} p(\mathbf{x}_i, \theta_2 | C_2) \right). \quad (4.35)$$

Uma vez que as verossimilhanças $p(\mathbf{x}, \theta_1 | C_1)$ e $p(\mathbf{x}, \theta_2 | C_2)$ são descritas aqui como misturas de densidades gaussianas, ou funções de *Kernel* centrados nos vértices do conjunto \mathcal{AS} , a regra geral de classificação pode ser reescrita da seguinte forma

$$f(\mathbf{x}_i) = \text{sign} \left(\overbrace{\sum_{j=1}^{N_1} w_j K(\mathbf{x}_i, \mathbf{x}_j)}^{\text{Classe Positiva}} - \overbrace{\frac{N_1}{N_2} \sum_{l=1}^{N_2} w_l K(\mathbf{x}_i, \mathbf{x}_l)}^{\text{Classe Negativa}} \right). \quad (4.36)$$

Desde que para cada vértice de uma aresta de suporte de uma classe existe um vértice correspondente para a outra classe, $N_1 = N_2$, e as duas regras de classificação diferem apenas sobre a forma como os parâmetros de mistura α_j e w_j são calculados. Nos experimentos apresentados nesta tese, $w_j = 1$ em todos os experimentos.

4.3.5 Resultados

As bases de dados utilizadas neste método estão descritas na Seção 4.1.3, juntamente com a metodologia de processamento das mesmas. Os métodos SVM e LSSVM-RBF (Suykens & Vandewalle, 1999) foram utilizados para comparar o nosso classificador (MIS-clas). Duas configurações de *Kernels* foram utilizadas. A primeira, SVM-RBF e LSSVM-RBF com função de bases radiais, e SVM-Poly com função polinomial. Os parâmetros de *Kernel* e regularização para SVM-RBF e SVM-Poly foram encontrados através de validação cruzada com 10 partições e busca em *grid*. A implementação destes métodos se deu através dos pacotes *Kernlab* (Karatzoglou *et al.*, 2004) e *Caret* (Kuhn, 2008), disponíveis para linguagem *R* (R Core Team, 2015).

A Tabela 4.6 mostra os valores da média da AUC e desvio padrão obtidos pelos métodos MIS-clas, SVM-RBF, SVM-Poly e LSSVM-RBF sobre as 17 bases de dados. Os melhores valores encontram-se em negrito. As quatro últimas colunas desta tabela mostram as características de cada base, em que N_d é o número de dimensões, N é o número total de amostras. A quantidade de elementos de cada classe da base de dados são denotados por N^+ e N^- respectivamente.

4.3 Classificador geométrico de margem larga

Tabela 4.6: Resultados do classificador Gaussiano: média da AUC e características das bases de dados.

Base de Dados	MIS-clas	SVM-RBF	SVM-Poly	LSSVM-RBF	N_d	N	N^+	N^-
<i>appendicitis</i>	0.802±0.163	0.707±0.224	0.718±0.222	0.687±0.234	7	106	21	85
<i>australian</i>	0.856±0.048	0.863±0.059	0.862±0.056	0.862±0.042	14	690	307	383
<i>banknote</i>	0.998±0.005	1±0	1±0	0.999±0.004	4	1372	610	762
<i>blood</i>	0.619±0.063	0.63±0.058	0.624±0.043	0.616±0.045	4	748	178	570
<i>breastcancer</i>	0.96±0.033	0.969±0.02	0.967±0.02	0.955±0.027	9	683	444	239
<i>breastHess</i>	0.828±0.08	0.791±0.115	0.79±0.093	0.735±0.098	30	133	99	34
<i>bupa</i>	0.596±0.095	0.663±0.053	0.665±0.052	0.674±0.076	6	345	145	200
<i>climate</i>	0.757±0.11	0.525±0.056	0.55±0.09	0.645±0.169	18	540	494	46
<i>diabetes</i>	0.725±0.065	0.711±0.048	0.716±0.05	0.714±0.058	8	768	500	268
<i>fertility</i>	0.585±0.226	0.5±0	0.5±0	0.578±0.184	9	100	12	88
<i>german</i>	0.7±0.074	0.655±0.038	0.658±0.038	0.657±0.048	24	1000	700	300
<i>glass</i>	0.842±0.206	0.846±0.202	0.846±0.202	0.822±0.203	9	214	29	185
<i>haberman</i>	0.578±0.073	0.535±0.056	0.508±0.039	0.552±0.092	3	306	225	81
<i>heart</i>	0.816±0.065	0.832±0.085	0.832±0.082	0.828±0.071	13	270	150	120
<i>ILPD</i>	0.598±0.064	0.502±0.006	0.5±0	0.572±0.07	10	579	414	165
<i>parkinsons</i>	0.768±0.121	0.753±0.062	0.769±0.059	0.813±0.116	22	195	147	48
<i>wdbc</i>	0.586±0.164	0.496±0.011	0.493±0.014	0.576±0.112	33	194	46	148
Média do rank $R(\mathcal{L})$	2.176	2.588	2.5	2.735				

4.3.6 Teste de significância

O teste de Friedman ([Demšar, 2006](#)) foi aplicado assumindo a hipótese nula H_0 que todos os métodos são equivalentes. Em nosso experimento, a estatística F_F é distribuída de acordo com a distribuição F de *Fisher-Snedecor* com 3 graus de liberdade para o numerador e 48 para o denominador. Considerando o valor crítico $F(3, 48)$ é 2.827 para $\alpha = 0.05$, a linha final da Tabela 4.6 apresenta a média do rank ($R(\mathcal{L})$) alcançada por todos os métodos. O valor correspondente a F_F é 0.370 e assim, desde que $F_F < F(3, 48)$, a hipótese H_0 não pode ser rejeitada. De acordo com a Figura 4.20, nenhum método possui a média significamente diferente do método MIS-clas. Embora a média do MIS-clas seja a menor dentre os métodos testados.

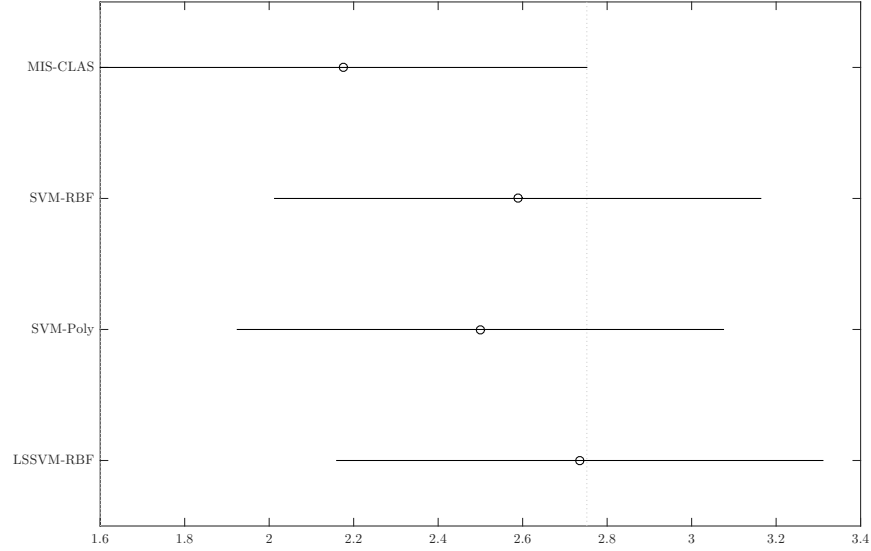


Figura 4.20: Diagrama de diferença crítica do teste *post-hoc*.

4.4 Classificador para Sistemas Embarcados

O projeto de aprendizagem de máquina geralmente requer a interação do usuário para definir a estrutura de aprendizagem do modelo e de parâmetros, e pode também envolver a implementação de algoritmos de otimização complexos. Além disso, o aprendizado envolve o *trade-off* entre o *bias* e a variância (Geman *et al.*, 1992), que também pode exigir o procedimento de regularização de parâmetros e validação cruzada. Estas podem ser consideradas como as principais limitações para a implementação de métodos de aprendizagem de máquinas, tais como Redes Neurais Artificiais (RNA) em nível de circuito integrado (CI) (He *et al.*, 2008). Durante os anos de 1990 e início dos anos 2000, placas de circuitos integrados de RNA apareceram no mercado, de forma motivada principalmente pela convergência lenta da taxa de algoritmos de aprendizagem. Embora o chip neural da Intel ETANN (Holler *et al.*, 1989) lançado em 1989 tivesse *synapses* adaptativas, um computador externo era necessário para processar os algoritmos de otimização e fornecer uma interface para o usuário. Recentemente, um novo interesse acerca dos sistemas de aprendizado embarcados tem sido motivado pelo surgimento da Internet das Coisas (*Internet of Things*). Tal que a IBM lançou recentemente um chip neural (Hsu, 2014) de alto desempenho, que é mais um

passo para próxima geração de sistemas adaptativos on-line.

As antigas limitações, no entanto, ainda existem, já que métodos de alto desempenho necessitam de interação do usuário e ainda de cálculos complexos, que não são viáveis para serem diretamente implementados em circuitos integrados. Neste trabalho, é apresentado um novo algoritmo de aprendizagem que não depende da interação do usuário e de parâmetros pré-definidos para lidar com o dilema do *bias* e variância. O método leva em consideração somente a estrutura dos dados para minimizar o erro do conjunto de treinamento e maximizar a margem de separação entre as classes. Uma vez que somente cálculos de distância são requeridos, o novo método é particularmente adequado para implementação em ASICs (*Application Specific Integrated Circuits*) e FPGAs (*Field-Programmable Gate Arrays*).

4.4.1 Combinação de Classificadores de Margem Larga

Não é esperado que um único hiperplano H_i separe todos os padrões em um conjunto $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i) \mid i = 1 \dots N\}$, uma vez que ele é baseado em um único par de vértices de uma aresta de suporte. Entretanto, a combinação de todos os hiperplanos produz um classificador com a informação espacial de todos os padrões em \mathcal{D} . A Figura 4.21(a) mostra um conjunto de dados com duas classes concêntricas, e sua representação através do grafo de Gabriel pode ser vista na Figura 4.21(b), onde as arestas de suporte e pontos médios são dados nas Figuras 4.21(c) e 4.21(d), respectivamente. Na Figura 4.21(e) são mostrados todos os classificadores correspondentes a todos pontos médios e a Figura 4.21(f) mostra a superfície de separação resultante da combinação de todos classificadores. Nota-se, que além de separar as duas classes, o classificador também maximiza a margem de separação entre elas.

4.4.2 Mistura Hierárquica de Especialistas

A classificação final é o resultado de uma Mistura Hierárquica de Especialistas (MHE), onde cada hiperplano (classificador) terá um peso diferente dado um padrão de entrada \mathbf{x} . A arquitetura da MHE é representada através de uma rede mostrada na Figura 4.22, onde a primeira camada corresponde aos especialistas

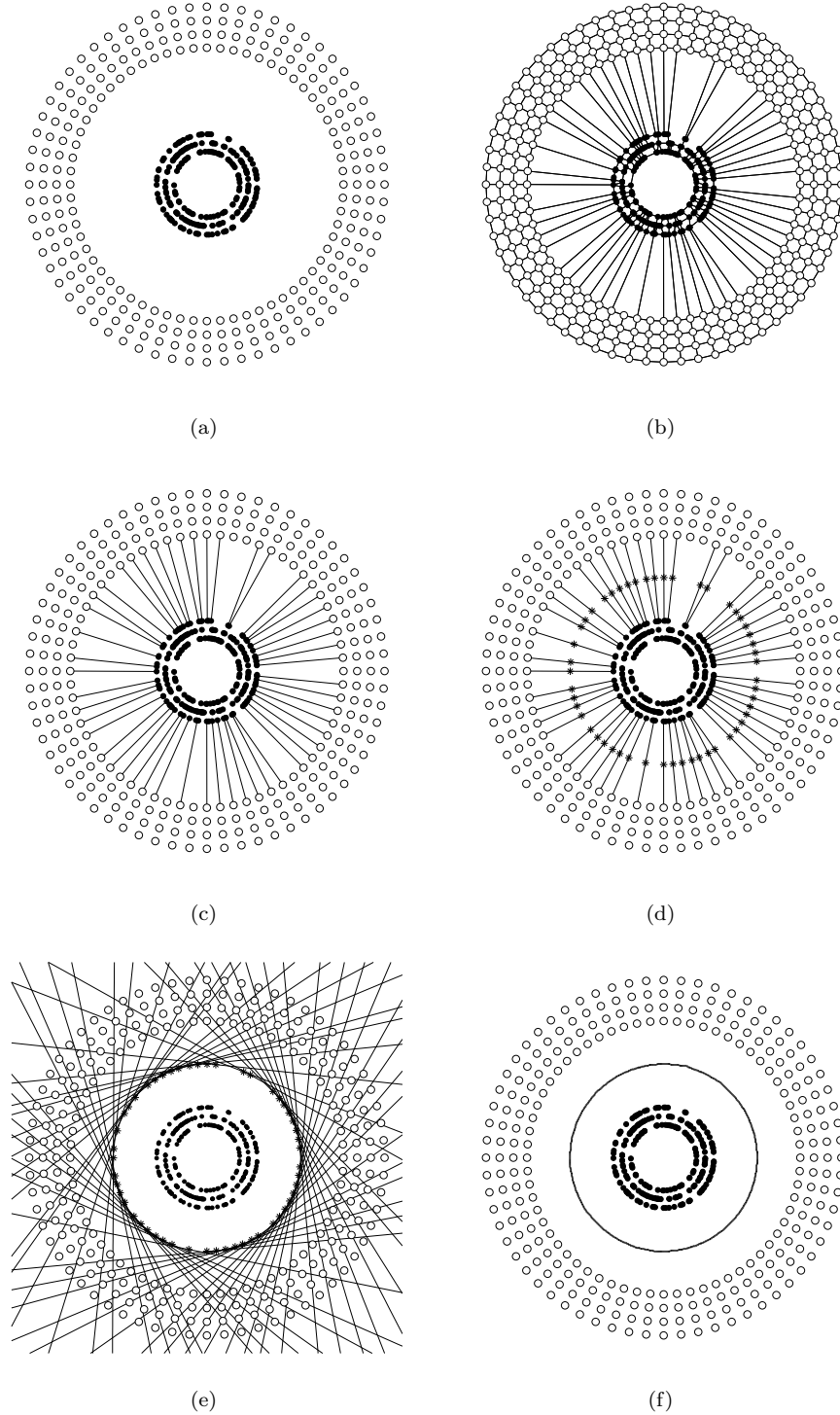


Figura 4.21: (a) Conjunto de padrões no espaço de entrada. (b) Conjunto de padrões no espaço de entrada modelado com o Grafo de Gabriel. (c) Conjunto de arestas de suporte. (d) Conjunto de pontos médios representado por asteriscos. (e) Classificadores gerados através de cada aresta de suporte. (f) Superfície de separação gerada pelo classificador.

locais $\{H_1, \dots, H_m\}$, ou seja, os hiperplanos. Já a saída de m especialistas para um padrão de entrada \mathbf{x} é representada pelas funções $h_1(\mathbf{x}), \dots, h_m(\mathbf{x})$, onde cada especialista é ponderado por um módulo *Gating Network*, onde o peso $c_l(\mathbf{x})$ para o l -ésimo especialista é obtido de acordo com uma função de distância mostrada a seguir.

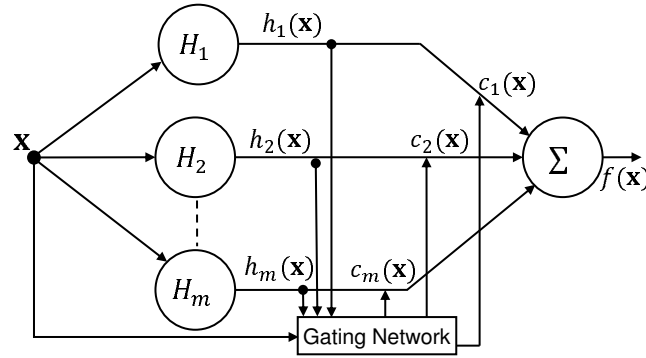


Figura 4.22: Estrutura do modelo de Mistura Hierárquica de Especialistas.

Considere-se H_l um hiperplano local gerado por uma aresta de suporte \mathcal{AS} de vértices $(\mathbf{x}_i, \mathbf{x}_j)$, tal que $y_i = -1$ e $y_j = +1$. O resultado da classificação devido ao hiperplano H_l para um padrão arbitrário \mathbf{x} é dado por $h_l(\mathbf{x}) = \text{sign}(\mathbf{x}^T \mathbf{w}_l - b_l)$, onde $\text{sign}(\cdot)$ é a função sinal e $\mathbf{w}_l = (\mathbf{x}_i - \mathbf{x}_j)$, uma vez que corresponde ao hiperplano que é perpendicular a linha (aresta) que conecta \mathbf{x}_i e \mathbf{x}_j . Da mesma forma, o termo de polarização b_l é obtido pela expressão $b_l = [(1/2)(\mathbf{x}_i + \mathbf{x}_j)] \mathbf{w}_l^T$. O ponto médio de cada \mathcal{AS} é também obtido diretamente como $p_l = (\mathbf{x}_i + \mathbf{x}_j)/2$. Portanto, os parâmetros dos especialistas locais dependem apenas dos vértices \mathbf{x}_i e \mathbf{x}_j e são obtidos por cálculo direto. O parâmetro de ponderação $c_l(\mathbf{x})$ de H_l para um padrão arbitrário \mathbf{x} é calculado de acordo com a Equação (4.37).

$$c_l(\mathbf{x}) = \exp - \left(\frac{(\max(\delta(\mathbf{x}, p_k)))^2}{\delta(\mathbf{x}, p_l)} \right), \quad \forall k = 1, \dots, m \quad (4.37)$$

Uma normalização também é imposta de forma que, $\sum_{l=1}^m c_l(\mathbf{x}) = 1$, e o resultado final da classificação é obtido por

$$f(\mathbf{x}) = \text{sign} \left(\sum_{l=1}^m h_l(\mathbf{x}) c_l(\mathbf{x}) \right), \quad (4.38)$$

onde os coeficientes $c_l(\mathbf{x})$ da Equação (4.37), que são combinados com $h_l(\mathbf{x})$, apresentam valores maiores para os hiperplanos que estão mais próximos de \mathbf{x} . Na prática, para aplicação no nível do circuito, o resultado final pode ser estimado apenas pelo hiperplano mais próximo.

4.4.3 Resultados

Os experimentos foram realizados com 13 bases de dados reais retiradas do repositório UCI (Bache & Lichman, 2013) e 2 problemas de expressão gênica: “Golub” (Golub *et al.*, 1999) e “BcrHess” (Hess *et al.*, 2006). Todos estes conjuntos de dados passaram pelas seguintes etapas de pré-processamento: filtragem de ruído, remoção de amostras contendo atributos faltantes e normalização dos dados entre $\{-1, 1\}$. Para garantir a relevância estatística, o experimento foi repetido usando validação cruzada com 10 partições. O desempenho médio (AUC) e o desvio padrão são listados na Tabela 4.7, juntamente com algumas características dos conjuntos de dados na última coluna, N e N_a , que correspondem ao total de padrões e atributos, respectivamente. Os valores em negrito representam os métodos com melhor desempenho para cada base.

Os métodos SVM-RBF e SVM-Poly foram utilizados para comparar o nosso método (CHIP-clas), de forma que duas configurações de *Kernel* foram utilizadas. A primeira, SVM-RBF com função de bases radiais, e a segunda SVM-Poly com função polinomial. Os parâmetros de *Kernel* e regularização para os métodos SVM-RBF e SVM-Poly foram encontrados através de validação cruzada com 10 partições e busca em *grid*. A implementação destes métodos se deu através dos pacotes *Kernlab* e *Caret* disponíveis para linguagem *R* (R Core Team, 2015).

O teste de Friedman (Demšar, 2006) foi aplicado assumindo-se a hipótese nula H_0 que todos os métodos são equivalentes. Em nosso experimento, a estatística F_F é distribuída de acordo com a distribuição F de Fisher-Snedecor com 2 graus de liberdade para o numerador e 28 para o denominador. Considerando que o valor crítico $F(2, 28)$ é 3.34 para $\alpha = 0.05$, a linha final da Tabela 4.7 representa a média do rank ($R(\mathcal{L})$) alcançada por todos os métodos. O valor correspondente a F_F é 0.60 e assim, desde que $F_F < F(2, 28)$, a hipótese H_0 não pode ser rejeitada.

4.4 Classificador para Sistemas Embarcados

Tabela 4.7: Resultados: AUC média e desvio padrão

Base de dados	CHIP-clas	SVM-RBF	SVM-Poly	N/N_d
<i>Australian Cr.</i>	0.85±0.04	0.86±0.04	0.87±0.04	690/14
<i>Banknote Auth.</i>	0.98±0.03	1±0	1±0	1372/4
<i>BcrHess</i>	0.81±0.12	0.76±0.11	0.77±0.15	133/30
<i>B. Cancer W.P</i>	0.96±0.03	0.97±0.01	0.96±0.03	683/9
<i>Climate M.S.C.</i>	0.84±0.07	0.53±0.06	0.72±0.11	540/18
<i>Fertility</i>	0.59±0.26	0.5±0	0.5±0	100/9
<i>German Cr</i>	0.67±0.04	0.66±0.07	0.68±0.05	1000/24
<i>Golub</i>	0.77±0.17	0.8±0.16	0.78±0.17	72/50
<i>Habermans S.</i>	0.57±0.09	0.52±0.06	0.5±0.02	306/3
<i>ILPD</i>	0.56±0.09	0.49±0.02	0.5±0	579/10
<i>Liver Disorders</i>	0.61±0.1	0.67±0.05	0.72±0.07	345/6
<i>P. ind. Diabetes</i>	0.72±0.04	0.71±0.05	0.71±0.07	768/8
<i>Parkinsons</i>	0.9±0.15	0.77±0.11	0.81±0.12	195/22
<i>Sonar. M vs. R.</i>	0.88±0.08	0.84±0.09	0.87±0.08	208/60
<i>Stalog Heart</i>	0.8±0.08	0.83±0.07	0.83±0.07	270/13
Média do rank $R(\mathcal{L})$	1.87	2.23	1.90	

Diante destes resultados, foi possível concluir que nossa abordagem é equivalente aos métodos SVM-RBF e SVM-Poly para os conjuntos de dados testados.

Capítulo 5

Conclusões e Propostas de Continuidade

Essa tese apresentou uma metodologia baseada no grafo de Gabriel para construção de modelos de inteligência computacional, assim como sua aplicação no desenvolvimento de quatro abordagens direcionadas a problemas de classificação binária. A partir da estrutura do grafo de Gabriel, foi possível extrair informações para o projeto de cada abordagem. Isso resultou em uma característica intrínseca pertencente a todos os métodos apresentados: a ausência do especialista para selecionar parâmetros e de outros métodos para encontrá-los. A metodologia geométrica também possibilitou uma interpretação mais simples e intuitiva dos métodos, o que leva a um melhor entendimento didático e contribui para a extensão e melhoria das abordagens.

O processo de obtenção do conjunto de arestas de suporte \mathcal{AS} possibilitou o desenvolvimento de um método para retirar a sobreposição entre as classes e possíveis elementos ruidosos. A metodologia também se mostrou invariante ao problema de desbalanceamento entre as classes, uma vez que o processamento é realizado de modo independente nos dados de cada classe.

Na primeira abordagem, foi desenvolvido um decisor denominado MOBJ-clas para o método de treinamento multiobjetivo de redes neurais (MOBJ), que em um primeiro experimento se mostrou estatisticamente equivalente aos métodos MOBJ-VAL, SVM-RBF, SVM-Poly e LSSVM-RBF, sobre o domínio de 17 bases de dados. Entretanto, ao retirar bases de dados cujo tamanho era maior ou

igual a 1000 amostras, o método MOBJ-clas foi estatisticamente melhor do que o MOBJ-VAL. Nota-se que, embora apresente bons resultados, esse último método necessita de um subconjunto do conjunto de treinamento, e que o mesmo seja representativo. Uma vez utilizado em problemas com poucas amostras disponíveis, como no caso de algumas bases de dados médicas, o MOBJ-VAL pode ter uma perda de performance acentuada. Em contrapartida, o MOBJ-clas pode utilizar todo o conjunto de treinamento para seu projeto. Além disso, ele pode ser aplicado em bases de dados desbalanceadas, já que a tomada de decisão acontece na margem entre as classes.

Na segunda abordagem, foi desenvolvida uma metodologia chamada RBF-clas para encontrar os parâmetros σ e c da função de *Kernel* de uma rede neural RBF. Nossa abordagem foi comparada com 12 arranjos de estratégias para selecionar parâmetros existentes na literatura. Em todas elas, nosso método obteve um desempenho superior. O método RBF-clas também foi comparado com métodos de classificação de margem larga tidos como estado da arte, a saber: SVM-RBF, SVM-Poly e LSLSVM-RBF. De acordo com o teste de Friedman, nosso método foi estatisticamente igual a todos os citados, entretanto, foi o que apresentou o melhor *rank*.

Na terceira abordagem, foi concebido um classificador de mistura Gaussiana de margem larga chamado MIS-clas. A partir do conjunto de arestas de suporte \mathcal{AS} , foram extraídas informações para a construção de funções de densidade multivariada. Sabendo *a priori* que a complexidade do classificador é dada pelo parâmetro de variância σ^2 da função de densidade, o valor de σ^2 foi encontrado de modo determinístico, seguindo-se a premissa de que cada componente da mistura gaussiana está a ± 3 desvios padrões do ponto médio mais próximo. O classificador MIS-clas foi comparado com os métodos SVM-RBF, SVM-Poly e LSLSVM-RBF e foi considerado estatisticamente equivalente a eles pelo teste de Friedman. Como aconteceu com os métodos MOBJ-clas e RBF-clas, o classificador MIS-clas foi o que apresentou o melhor *rank*.

Na quarta e última abordagem, foi projetado um classificador direcionado a implementação em sistemas embarcados, denominado CHIP-clas. O classificador é projetado com base na métrica de distância Euclidiana. Isso permite que ele seja implementado facilmente por meio de operações matemáticas básicas e otimizado

com o paralelismo proporcionado por FPGAs e GPUs. Esse ponto é uma grande vantagem em relação aos métodos que dependem de operações complexas. Os resultados obtidos de testes realizados com bases reais e comparados com os métodos SVM-RBF e SVM-poly mostraram que o método CHIP-clas, apesar de sua simplicidade, é equivalente aos métodos testados.

Por fim, essa tese possibilitou a construção de uma nova família de classificadores de margem máxima. Apesar do desempenho dos métodos ser avaliado a partir de bases de dados reais, com diferentes características, cada método pode ser melhor aproveitado se for direcionado a certos tipos de problemas e situações. O método MOBJ-clas, por exemplo, pode ser utilizado em bases de dados pequenas em conjunto com outros métodos de aprendizado multiobjetivo. O método RBF-clas, por sua vez, pode ser uma alternativa para utilizadores da rede RBF, pois não necessita de um arranjo de metodologias para encontrar parâmetros. Já o método MIS-clas, ainda que pouco explorado, possui no espaço de verossimilhanças a possibilidade de bases de dados no espaço \mathbb{R}^n serem visualizadas no espaço \mathbb{R}^2 . Dessa forma, novas características e informações sobre os dados podem ser geometricamente explorados. Por último, o método CHIP-clas pode ser aplicado a problemas que exigem tomadas de decisão rápidas, como é comum no setor industrial. Entretanto, alguns pontos do método CHIP-clas ainda precisam ser estudados e fazem parte da proposta de continuidade desta tese.

5.1 Propostas de Continuidade

Seguem algumas propostas de continuidade desse trabalho:

- Aplicação dos métodos direcionado a problemas específicos. Por exemplo, aplicar o método MOBJ-clas em base de dados médicas, onde é possível encontrar base de dados com um número pequeno de amostras.
- Desenvolvimento e implementação de um algoritmo incremental para construção do grafo de Gabriel.
- Extensão do método CHIP-clas para problemas que exigem trabalhar em tempo real. Como o estudo de estratégias para aprendizado incremental a

5.1 Propostas de Continuidade

partir de *data-streams* e dos problemas industriais que podem ser solucionados através desta abordagem.

Referências

- ALBERT, A. (1972). *Regression and the Moore-Penrose pseudoinverse*, vol. 3. Academic Press New York. [30](#), [51](#)
- ALCALÁ, J., FERNÁNDEZ, A., LUENGO, J., DERRAC, J., GARCÍA, S., SÁNCHEZ, L. & HERRERA, F. (2010). Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic and Soft Computing*, **17**, 11. [44](#)
- AUPETIT, M. & CATZ, T. (2005). High-dimensional labeled data analysis with topology representing graphs. *Neurocomputing*, **63**, 139 – 169. [22](#), [24](#)
- AURENHAMMER, F. & KLEIN, R. (1990). Voronoi diagrams. In *Handbook of Computational Geometry*, 152–159, Elsevier. [17](#)
- BACHE, K. & LICHMAN, M. (2013). UCI machine learning repository. [44](#), [75](#)
- BARROSO, M.M.A. (2007). Operações elementares em grafos. *EMARC*, **7**. [16](#)
- BENNETT, K.P. & BREDENSTEINER, E.J. (2000). Duality and geometry in svm classifiers. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, 57–64, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. [2](#), [28](#), [31](#)
- BERG, M.D., CHEONG, O., KREVELD, M.V. & OVERMARS, M. (2008). *Computational Geometry: Algorithms and Applications*. Springer-Verlag TELOS, Santa Clara, CA, USA, 3rd edn. [17](#), [18](#), [19](#), [28](#), [33](#), [49](#)
- BISHOP, C.M. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press, Inc., New York, NY, USA. [51](#), [54](#)

- BOSER, B.E., GUYON, I.M. & VAPNIK, V.N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, 144–152, ACM, New York, NY, USA. [1](#), [27](#)
- CASTRO, C. & BRAGA, A. (2013). Novel cost-sensitive approach to improve the multilayer perceptron performance on imbalanced data. *Neural Networks and Learning Systems, IEEE Transactions on*, **24**, 888–899. [44](#)
- CEVIKALP, H. & TRIGGS, B. (2013). Hyperdisk based large margin classifier. *Pattern Recognition*, **46**, 1523–1531. [1](#)
- CEVIKALP, H., TRIGGS, B., YAVUZ, H.S., KÜÇÜK, Y., KÜÇÜK, M. & BARKANA, A. (2010). Large margin classifiers based on affine hulls. *Neurocomputing*, **73**, 3160–3168. [1](#)
- CHIU, S.L. (1994). Fuzzy model identification based on cluster estimation. *Journal of Intelligent & Fuzzy Systems*, **2**, 267–278. [49](#)
- CHRISTOFIDES, N. (1975). *Graph Theory: An Algorithmic Approach (Computer Science and Applied Mathematics)*. Academic Press, Inc., Orlando, FL, USA. [16](#), [22](#)
- CORTES, C. & VAPNIK, V. (1995). Support-vector networks. *Machine learning*, **20**, 273–297. [27](#), [48](#)
- COSTA, M.A., BRAGA, A.P. & MENEZES, B.R. (2007). Improving generalization of mlps with sliding mode control and the levenberg-marquardt algorithm. *Neurocomputing*, **70**, 1342–1347. [14](#), [33](#)
- COVER, T.M. (1965). Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers*, **14**, 326–334. [48](#), [50](#)
- DEMŠAR, J. (2006). Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, **7**, 1–30. [46](#), [47](#), [56](#), [58](#), [70](#), [75](#)

- DUDA, R.O., HART, P.E. & STORK, D.G. (2012). *Pattern classification*. John Wiley & Sons. [5](#), [49](#), [61](#), [63](#)
- FAWCETT, T. (2006). An introduction to roc analysis. *Pattern recognition letters*, **27**, 861–874. [44](#)
- FERNANDEZ-DELGADO, M., RIBEIRO, J., CERNADAS, E. & AMENEIRO, S.B. (2011). Direct parallel perceptrons (dpps): Fast analytical calculation of the parallel perceptrons weights with margin control for classification tasks. *Neural Networks, IEEE Transactions on*, **22**, 1837–1848. [1](#)
- FIGUEIREDO, L.H. (1991). *Introdução a Geometria Computacional*. Impa, Rio de Janeiro. [17](#), [18](#)
- FREUND, Y. & SCHAPIRE, R.E. (1996). Game theory, on-line prediction and boosting. In *Proceedings of the ninth annual conference on Computational learning theory*, 325–332, ACM. [1](#)
- FREUND, Y., SCHAPIRE, R.E. *et al.* (1996). Experiments with a new boosting algorithm. In *ICML*, vol. 96, 148–156. [1](#)
- FREY, B.J. & DUECK, D. (2007). Clustering by passing messages between data points. *Science*, **315**, 972–976. [53](#)
- GABRIEL, K.R. & SOKAL, R.R. (1969). A new statistical approach to geographic variation analysis. *Systematic Biology*, **18**, 259–278. [2](#)
- GARCIA, L.P., DE CARVALHO, A.C. & LORENA, A.C. (2015). Effect of label noise in the complexity of classification problems. *Neurocomputing*, **160**, 108 – 119. [22](#)
- GEMAN, S., BIENENSTOCK, E. & DOURSAT, R. (1992). Neural networks and the bias / variance dilemma. *Neural Computation*, **4**, 1–58. [32](#), [71](#)
- GIROSI, F., JONES, M. & POGGIO, T. (1995). Regularization theory and neural network architectures. *Neural computation*, **7**, 219–269. [32](#)

REFERÊNCIAS

- GOLUB, T.R., SLONIM, D.K., TAMAYO, P., HUARD, C., GAASENBEEK, M., MESIROV, J.P., COLLIER, H., LOH, M.L., DOWNING, J.R., CALIGIURI, M.A. *et al.* (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *science*, **286**, 531–537. [75](#)
- GUYON, I. (2006). *Feature extraction: foundations and applications*, vol. 207. Springer. [30](#)
- HAYKIN, S. (1999). *Neural networks: A Comprehensive Foundation*. Prentice Hall, New Jersey, 2nd edn. [10](#), [50](#)
- HAYKIN, S. (2009). *Neural networks and learning machines*, vol. 3. Prentice Hall. [9](#), [11](#), [13](#)
- HE, M., KLEIN, J.O. & BELHAIRE, E. (2008). Design and electrical simulation of on-chip neural learning based on nanocomponents. *Electronics Letters*, **44**, 575–576. [71](#)
- HESS, K.R., ANDERSON, K., SYMMANS, W.F., VALERO, V., IBRAHIM, N., MEJIA, J.A., BOOSER, D., THERIAULT, R.L., BUZDAR, A.U., DEMPSEY, P.J. *et al.* (2006). Pharmacogenomic predictor of sensitivity to preoperative chemotherapy with paclitaxel and fluorouracil, doxorubicin, and cyclophosphamide in breast cancer. *Journal of clinical oncology*, **24**, 4236–4244. [44](#), [75](#)
- HINTON, G. (1989). Connectionist learning procedures. *Artificial intelligence*, **40**, 185–234. [14](#), [32](#)
- HOGG, R.V. & LEDOLTER, J. (1987). *Engineering statistics*, vol. 358. MacMillan New York. [36](#)
- HOLLER, M., TAM, S., CASTRO, H. & BENSON, R. (1989). An electrically trainable artificial neural network (etann) with 10240 'floating gate' synapses. In *Neural Networks, 1989. IJCNN., International Joint Conference on*, 191–196 2. [71](#)
- HSU, J. (2014). IBM's new brain [News]. *Spectrum, IEEE*, **51**, 17–19. [71](#)

- JIN, Y. & SENDHOFF, B. (2009). Pareto-based multiobjective machine learning: An overview and case studies. *IEEE Transactions on Systems Science and Cybernetics*, **39**, 373. [33](#)
- KARATZOGLU, A., SMOLA, A., HORNIK, K. & ZEILEIS, A. (2004). kernlab – an S4 package for kernel methods in R. *Journal of Statistical Software*, **11**, 1–20. [55](#), [69](#)
- KOHAVER, R. *et al.* (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *International joint Conference on artificial intelligence*, vol. 14, 1137–1145, Lawrence Erlbaum Associates Ltd. [49](#)
- KOHONEN, T. (1990). The self-organizing map. *Proceedings of the IEEE*, **78**, 1464–1480. [49](#)
- KOKSHENEV, I. & BRAGA, A.P. (2010). An efficient multi-objective learning algorithm for rbf neural network. *Neurocomputing*, **37**, 2799–2808. [33](#)
- KUHN, M. (2008). Building predictive models in r using the caret package. *Journal of Statistical Software*, **28**, 1–26. [55](#), [69](#)
- LI, J. & KUO, C.C. (1998). A dual graph approach to 3d triangular mesh compression. In *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, vol. 2, 891–894 2. [18](#)
- MCLACHLAN, G. & PEEL, D. (2004). *Finite mixture models*. John Wiley & Sons. [62](#)
- MEDEIROS, T.H., TAKAHASHI, H.C.R. & BRAGA, A. (2009). A incorporaco do conhecimento prvio na tomada de deciso do aprendizado multiobjetivo. *Congresso Brasileiro de Redes Neurais - Inteligncia Computacional*, **9**, 25–28. [15](#), [16](#), [33](#)
- PENG, X. & WANG, Y. (2012). Geometric algorithms to large margin classifier based on affine hulls. *Neural Networks and Learning Systems, IEEE Transactions on*, **23**, 236–246. [1](#)

- PENG, X. & XU, D. (2013). Geometric algorithms for parametric-margin ν -support vector machine. *Neurocomputing*, **99**, 197–205. [1](#)
- R CORE TEAM (2015). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. [37](#), [45](#), [55](#), [69](#), [75](#)
- REED, R. (1993). Pruning algorithms-a survey. *IEEE Transactions on Neural Networks*, **4**, 740–747. [32](#)
- ROUSSEEUW, P.J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, **20**, 53–65. [53](#)
- SCHAPIRE, R.E. (1990). The strength of weak learnability. *Machine learning*, **5**, 197–227. [1](#)
- SHA, F. & SAUL, L.K. (2006). Large margin gaussian mixture modeling for phonetic classification and recognition. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 1, I–I, IEEE. [1](#)
- SING, J., BASU, D., NASIPURI, M. & KUNDU, M. (2003). Improved k-means algorithm in the design of rbf neural networks. In *TENCON 2003. Conference on Convergent Technologies for Asia-Pacific Region*, vol. 2, 841–845, IEEE. [49](#)
- SUYKENS, J. & VANDEWALLE, J. (1999). Least squares support vector machine classifiers. *Neural Processing Letters*, **9**, 293–300. [44](#), [55](#), [69](#)
- TEIXEIRA, R.A. (2001). *Treinamento de Redes Neurais Artificiais Através De Otimização Multi-objetivo: Uma Nova Abordagem Para O Equilíbrio Entre A Polarização e A Variância*. Tese de doutorado, UFMG. [37](#), [45](#)
- TEIXEIRA, R.A., BRAGA, A.P., TAKAHASHI, R.H.C. & SALDANHA, R.R. (2000). Improving generalization of mlps with multi-objective optimization. *Neurocomputing*, **35**, 189–194. [4](#), [14](#), [16](#), [33](#), [37](#), [45](#)

REFERÊNCIAS

- TORRES, L., CASTRO, C., COELHO, F., SILL TORRES, F. & BRAGA, A. (2015a). Distance-based large margin classifier suitable for integrated circuit implementation. *Electronics Letters*, **51**, 1967–1969. [2](#), [3](#)
- TORRES, L.C., CASTRO, C.L. & BRAGA, A.P. (2012). A computational geometry approach for pareto-optimal selection of neural networks. In *Artificial Neural Networks and Machine Learning–ICANN 2012*, 100–107, Springer. [2](#)
- TORRES, L.C., COELHO, C., FREDERICO, CASTRO, C.L. & BRAGA, A.P. (2014a). A graph of gabriel approach for large margin classifiers. In *The Latin American Congress on Computational Intelligence Co-located with ARGENCON–LA-CCI*, 25–29, San Carlos de Bariloche. [3](#), [29](#)
- TORRES, L.C., CASTRO, C.L. & BRAGA, A.P. (2015b). Gabriel graph for dataset structure and large margin classification: A bayesian approach. In *Proceedings of the European Symposium on Neural Networks–ESANN*, 237–242, Bruges. [3](#)
- TORRES, L.C.B., LEMOS, A.P., CASTRO, C.L. & BRAGA, A.P. (2014b). *Artificial Neural Networks and Machine Learning – ICANN 2014: 24th International Conference on Artificial Neural Networks, Hamburg, Germany, September 15-19, 2014. Proceedings*, chap. A Geometrical Approach for Parameter Selection of Radial Basis Functions Networks, 531–538. Springer International Publishing, Cham. [2](#)
- TOUSSAINT, G.T. (1980). The relative neighbourhood graph of a finite planar set. *Pattern recognition*, **12**, 261–268. [19](#)
- VAPNIK, V.N. (1995). *The nature of statistical learning theory*. Springer-Verlag New York, Inc. [8](#), [13](#), [27](#), [32](#)
- VAPNIK, V.N. (1998). *Statistical Learning Theory*. [8](#), [13](#), [32](#)
- ZHANG, H. & HE, X. (2006). On simultaneous straight-line grid embedding of a planar graph and its dual. *Information Processing Letters*, **99**, 1 – 6. [18](#)

REFERÊNCIAS

ZHANG, W. & KING, I. (2002). A study of the relationship between support vector machine and gabriel graph. In *Neural Networks, 2002. IJCNN '02. Proceedings of the 2002 International Joint Conference on*, vol. 1, 239–244. [19](#)