

TESE DE DOUTORADO Nº 246

**TREINAMENTO MULTIOBJETIVO DE PERCEPTRON DE MÚLTIPLAS CAMADAS
COM REPRESENTAÇÃO ESFÉRICA DE PESOS**

Honovan Paz Rocha

DATA DA DEFESA: 17/02/2017

Universidade Federal de Minas Gerais

Escola de Engenharia

Programa de Pós-Graduação em Engenharia Elétrica

**TREINAMENTO MULTIOBJETIVO DE PERCEPTRON DE
MÚLTIPLAS CAMADAS COM REPRESENTAÇÃO ESFÉRICA DE
PESOS**

Honovan Paz Rocha

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. Marcelo Azevedo Costa

Belo Horizonte - MG

Fevereiro de 2017

R672t

Rocha, Honovan Paz.

Treinamento multiobjetivo de perceptron de múltiplas camadas com representação esférica de pesos [manuscrito] / Honovan Paz Rocha. - 2017.

xvii, 100 f., enc.: il.

Orientador: Antônio de Pádua Braga.

Coorientador: Marcelo Azevedo Costa.

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

Anexos: f.82-92.

Bibliografia: f. 93-100.

1. Engenharia elétrica - Teses. 2. Redes neurais (Computação) - Teses. 3. Perceptrons - Teses. I. Braga, Antônio de Pádua. II. Costa, Marcelo Azevedo. III. Universidade Federal de Minas Gerais. Escola de Engenharia. IV. Título.

CDU: 621.3(043)

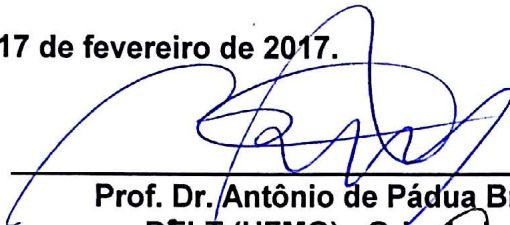
"Treinamento Multiobjetivo de Perceptron de Múltiplas Camadas com Representação Esférica de Pesos"

Honovan Paz Rocha

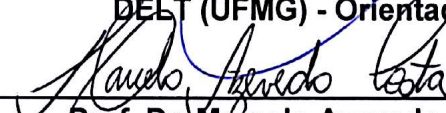
Tese de Doutorado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do grau de Doutor em Engenharia Elétrica.

Aprovada em 17 de fevereiro de 2017.

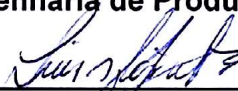
Por:



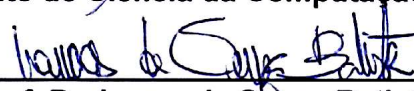
Prof. Dr. Antônio de Pádua Braga
DELT (UFMG) - Orientador



Prof. Dr. Marcelo Azevedo Costa
Engenharia de Produção (UFMG) - Coorientador



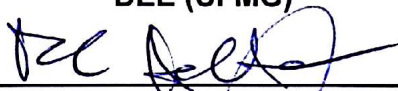
Prof. Dr. Luiz Enrique Zárate
Departamento de Ciência da Computação (PUC-MG)



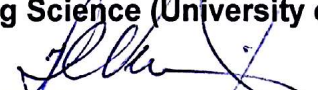
Prof. Dr. Lucas de Souza Batista
DEE (UFMG)



Prof. Dr. Rodney Rezende Saldanha
DEE (UFMG)



Prof. Dr. Robert Holte
Dep. Computing Science (University of Alberta, Canada)



Prof. Dr. Haroldo Fraga de Campos Velho
Lab Computação e Matemática Aplicada (INPE)

Dedico esta tese aos meus pais e minha irmã.

Agradecimentos

À Deus que sempre providenciou tudo em minha vida. Em quem creio e confio plenamente, que me conduziu até aqui e me levará muito mais além.

Aos meus pais, Honorato e Vanusa, que sempre se preocuparam e se esforçaram para me dar uma educação de qualidade além de apoio e incentivo em todos os momentos. À minha irmã, Adriane, por todo o carinho e compreensão. Sem vocês seria impossível chegar até aqui!

À Natália pelo carinho e compreensão em todos os momentos.

Ao Professor Braga meu orientador, pela oportunidade, ajuda, paciência e entusiasmo no dia-a-dia, além de todo o conhecimento e experiências transmitidas que contribuíram fundamentalmente para o meu amadurecimento não restrito apenas ao meio acadêmico. Obrigado pelo privilégio de trabalhar com o senhor por todos esses anos.

Ao Professor Marcelo Costa meu coorientador, pela paciência, entusiasmo e grande auxílio na construção deste trabalho.

Ao Professor Rodney por todo o auxílio e pelas boas discussões.

Aos amigos do LITC e CPDEE, por toda ajuda que me deram.

Às entidades de fomento CNPq e CAPES pelo auxílio financeiro.

”A simplicidade é a extrema
sophistication.”

Leonardo da Vinci

Resumo

Este trabalho apresenta uma nova representação para o aprendizado multiobjetivo de redes neurais artificiais (RNA), onde os pesos são descritos com base na norma Euclidiana, que é tomada como uma medida da complexidade do modelo. Os pesos são projetados em um novo espaço definido por um raio r e um vetor de ângulos Θ . Esta representação esférica simplifica ainda mais o problema de aprendizado multiobjetivo, cujo conjunto Pareto-ótimo estimado é obtido utilizando-se o método ϵ -restrito nas abordagens clássicas, minimizando o erro com múltiplos valores restritos de norma. O correspondente problema de otimização com restrições é transformado em vários problemas irrestritos, o que simplifica a formulação e esforço computacional, além de permitir que qualquer método de otimização não-linear possa ser utilizado para treinar a RNA. Os resultados indicam que a representação esférica de pesos proposta gera estimativas do conjunto Pareto mais precisas quando comparada à abordagem multiobjetivo clássica. Com relação à solução final obtida a partir do conjunto Pareto, a abordagem esférica mostrou eficácia com base no teste de Friedman, superando alguns métodos do estado da arte em vários conjuntos de dados.

Palavras-chave: Aprendizado multiobjetivo, Perceptron de múltiplas camadas, Pesos esféricos, Algoritmos de treinamento esféricos.

Abstract

This work presents a novel representation of artificial neural networks (ANN) multiobjective learning in which the weights are described based on the Euclidean norm, which is taken as a measure of model complexity. Weights are projected into a new space defined by a radius r and a vector of angles Θ . This spherical representation further simplifies the multi-objective learning problem whose estimated Pareto-optimal set is obtained in classical approaches by the ϵ -constrained method, minimizing error with multiple constrained values of norm. The corresponding constrained optimization problem is transformed into unconstrained ones, what simplifies formulation and computational efforts, besides allowing that any nonlinear optimization method could be used to train the ANN. Results indicate that the proposed spherical weights representation yields more accurate estimates of the Pareto set when compared to the classical multi-objective approach. Regarding the final solution taken from the Pareto set, our approach showed effectiveness, outperforming some state-of-the-art methods in several datasets based on Friedman's test.

Keywords: Multiobjective learning, Multi-layer Perceptron, Spherical weights, Spherical training algorithms.

Publicações

Durante o desenvolvimento desta tese alguns trabalhos foram desenvolvidos e publicados, e um artigo está em processo de revisão. Os principais trabalhos estão na lista a seguir:

- Rocha, H.P., Costa, M.A., Braga, A.P. *Bias and variance trade-off with spherical representation of parameters*. Artigo submetido para a *Neurocomputing* em 12 de Janeiro de 2017. Está em processo de revisão;
- Medeiros, T.H., Rocha, H.P., Torres, F.S., Takahashi, R.H.C., Braga, A.P. (2016). *Multi-objective decision in machine learning*. *Journal of Control, Automation and Electrical Systems*, p. 1–11;
- Rocha, H.P., Costa, M.A., Braga, A.P. (2015). *Training multi-layer perceptron with multi-objective optimization and spherical weights representation*. In *Proceedings of the European Symposium on Neural Networks*, p. 131–136;
- Rocha, H.P., Costa, M.A., Braga, A.P. (2013). Treinamento multiobjetivo de redes neurais artificiais baseado em coordenadas esféricas. In *Anais do XI Congresso Brasileiro de Inteligência Computacional - CBIC 2013*.

Sumário

Lista de Símbolos	x
Lista de Abreviaturas	xii
Lista de Figuras	xv
Lista de Tabelas	xvi
Lista de Algoritmos	xvii
1 Introdução	1
1.1 Objetivo e Contribuições	8
1.2 Organização do Trabalho	11
2 Referencial Teórico	12
2.1 Treinamento de Redes Neurais Artificiais	12
2.1.1 <i>Métodos Clássicos para Treinamento de RNA</i>	12
2.1.2 Capacidade Generalização em RNAs	13
2.2 Teoria do Aprendizado Estatístico	15
2.2.1 Princípio de Minimização do Risco Estrutural	15
2.3 Dilema entre o Viés e a Variância	16
2.4 Aprendizado Multi-objetivo	19
2.4.1 Conceitos Básicos em Otimização Multiobjetivo	19
2.4.1.1 O Conjunto Pareto-Ótimo	20
2.4.2 Formulação Multi-objetivo (MOBJ) para o Treinamento de Redes MLPs	21
2.4.3 Decisão Multi-objetivo em Aprendizado de Máquina	23
2.4.3.1 Métodos para Escolha da Solução	24
2.5 Conclusão do Capítulo	26

3	Formulação Esférica e Métodos Propostos	27
3.1	Representação Esférica dos Pesos	27
3.1.1	Pesos Esféricos Aplicados a um Perceptron Não-Linear . .	28
3.1.2	Representação Esférica de Pesos para a Rede MLP	32
3.2	Aprendizado Multi-objetivo de Redes com Pesos Esféricos	35
3.2.1	O Algoritmo <i>Gradiente Descendente</i> Utilizando Pesos Esféricos (SGD)	36
3.2.2	O Algoritmo <i>Levenberg-Marquardt</i> Esférico (SLM)	38
3.2.3	O Algoritmo <i>BFGS</i> Esférico (SBFGS)	40
3.3	Conclusão do Capítulo	45
4	Experimentos em Problemas de Regressão e Classificação	46
4.1	Descrição dos Experimentos	46
4.1.1	Descrição das Bases de Dados	46
4.1.2	Parametrização da Rede	49
4.1.3	Formato dos Experimentos	49
4.1.4	Métricas para Avaliação das Estimativas do Conjunto Pa- reto	51
4.2	Experimento 1: Avaliação dos Conjuntos Pareto Estimados	52
4.2.1	Problemas de Regressão	52
4.2.2	Problemas de Classificação	60
4.2.3	Análise dos Resultados com Relação a HV e CS	68
4.3	Experimento 2: Avaliação das Soluções Finais	71
4.3.1	Descrição do Experimento	71
4.3.2	Teste de Significância para Comparação entre Várias Máquinas de Aprendizado	72
4.3.3	Resultados para as Bases de Regressão e Classificação . . .	73
4.4	Conclusão do Capítulo	76
5	Conclusões e Trabalhos Futuros	77
5.1	Propostas de Trabalhos Futuros	80

SUMÁRIO

A Conjuntos Pareto Estimados - Bases de dados Reais	82
A.1 Problemas de Regressão	82
A.2 Problemas de Classificação	88
Referências	100

Lista de Símbolos

$R(\mathbf{w})$	Risco Esperado
$R_{emp}(\mathbf{w})$	Risco Empírico
Ω	Capacidade do Modelo
h	Complexidade da classe de funções
$f_g(\mathbf{x})$	Função geradora dos dados
$E[\cdot]$	Valor esperado
γ	Incerteza presente na amostragem
$\mu_{\{f(\mathbf{x}, \mathbf{w})\}}$	Viés para uma família de modelos
$\sigma_{\{f(\mathbf{x}, \mathbf{w})\}}^2$	Variância para uma família de modelos
$\ \cdot\ $	Norma Euclidiana
$\mathbf{x}$	Vetor de variáveis de otimização
S	Conjunto viável no espaço de busca
$f(\cdot)$	Vetor de funções objetivo
$g(\cdot)$	Vetor de restrições de desigualdade
$h(\cdot)$	Vetor de restrições de igualdade
m	Quantidade de funções objetivo
n	Quantidade de variáveis de otimização
p	Quantidade de restrições de desigualdade
q	Quantidade de restrições de igualdade
$\mathbf{X}$	Espaço de parâmetros definido no espaço de variáveis de otimização
$\mathbf{Y}$	Espaço de objetivo, gerado através do vetor de funções objetivo
$\mathbf{x}^*$	Solução Pareto-Ótima
$\mathbf{P}$	Conjunto Pareto-Ótimo
$\mathbf{w}^*$	Vetor de pesos ótimo
J_1	Representação do erro médio quadrático no método MOBJ
J_2	Representação da norma dos pesos no método MOBJ
y	Saída desejada da rede
$\hat{y}$	Saída estimada pela rede

f_U	Função de utilidade para escolha de solução em um conjunto
$\bar{y}$	Média da saída desejada
$\mathcal{PO}$	Estimativa do conjunto Pareto-ótimo
n_{sol}	Número máximo de soluções da estimativa do conjunto Pareto
δr	Diferença de norma de soluções adjacentes no conjunto de soluções
$\mathbf{w}$	Vetor contendo todos os pesos da rede
Θ^T	Vetor contendo os ângulos da rede na formulação esférica
r	Raio que representa a norma na formulação esférica
α	Taxa de aprendizado do algoritmo <i>Back-Propagation</i>
$\mathbf{N}$	Quantidade de entradas da rede
n_w	Quantidade de pesos da rede
H	Quantidade de neurônios na camada escondida
$\mathbf{T}$	Matriz de derivadas parciais dos pesos em relação aos ângulos
$\nabla f(\mathbf{w})$	Vetor gradiente da função de erro
$\nabla^2 f(\mathbf{w})$	Matriz Hessiana da função de erro
$\mathbf{B}$	Aproximação da matriz Hessiana pelo algoritmo BFGS
$\Delta_{\mathbf{w}}$	Termo de atualização dos pesos da rede
$\Delta\Theta$	Termo de atualização dos ângulos da rede
I	Matriz identidade
$J(\mathbf{w})$	Jacobiana do método <i>Levenberg-Marquardt</i> em relação aos pesos
$J(\Theta)$	Jacobiana do método <i>Levenberg-Marquardt</i> em relação aos ângulos

Lista de Abreviaturas

ACC	Acurácia
BFGS	Algoritmo Broyden-Fletcher-Goldfarb-Shanno
BP	<i>Back-Propagation</i>
CD	<i>Critical Difference</i>
CDD	<i>Critical Difference Diagram</i>
CS	Cobertura de dois conjuntos
GBM	<i>Gradient Boosting Machine</i>
GD	Gradiente Descendente
HV	Hipervolume Dominado
LM	<i>Levenberg-Marquardt</i>
ML	Aprendizado de Máquina
MLP	<i>Multilayer Perceptron</i>
MLP-WD	<i>Multilayer Perceptron Weight Decay</i>
MOBJ	Algoritmo Multiobjetivo para treinamento de Redes Neurais
MSE	Erro Médio Quadrático
RBF	<i>Radial Basis Function</i>
R_{emp}	Risco Empírico
RF	<i>Random Forest</i>
RNAs	Redes Neurais Artificiais
RPROP	<i>Resilient Back-Propagation</i>
SBFGS	Algoritmo Broyden-Fletcher-Goldfarb-Shanno Esférico
SGD	Gradiente Descendente Esférico
SLM	<i>Levenberg-Marquardt</i> Esférico
SLT	<i>Statistical Learning Theory</i>
SRM	<i>Structural Risk Minimization</i>
SVM	<i>Support Vector Machines</i>
SVM-P	<i>Support Vector Machine</i> com Kernel Polinomial
SVM-R	<i>Support Vector Machine</i> com Kernel Radial
TME	Tempo Médio de Execução

Lista de Figuras

1.1	Exemplo de duas funções objetivo convexas a serem minimizadas. A Região B é o espaço alvo, onde a solução de compromisso entre as duas funções objetivo deve ser encontrada.	3
1.2	Maximização da margem como uma função da norma dos vetores de pesos para o aprendizado com SVMs. O problema é tratado com uma formulação de otimização multiobjetivo.	4
1.3	Mapeamento do espaço de parâmetros para o espaço de objetivos com soluções restritas às regiões $\ \mathbf{w}\ \leq r_j$ e $\ \mathbf{w}\ = r_j$	6
1.4	Solução no espaço limitado por um determinado valor de norma.	9
1.5	Curvas de erro para valores constantes de norma.	10
1.6	Estimativa do conjunto Pareto e respectivos modelos fornecidos pelo algoritmo <i>Levenberg-Marquardt</i> Esférico. A solução selecionada é aquela com menor erro de validação.	11
2.1	Representação de uma estimativa do Conjunto Pareto-Ótimo	23
2.2	Pareto Discreto e Curva de Validação	25
3.1	Disposição dos pesos na elipse definida pela função norma no espaço 2D.	28
3.2	Neurônio com uma entrada.	29
3.3	Contorno da superfície de erro	30
3.4	Curvas de erro em função do ângulo θ em radianos	31
3.5	Aproximação da fronteira Pareto através da minimização de erro com norma fixa.	35
4.1	(a) Paretos estimados - f_1 . (b) Aproximação para f_1	55

LISTA DE FIGURAS

4.2	(a) Paretos estimados - Função f_2 . (b) Aproximação para f_2	56
4.3	(a) Paretos estimados - Função f_3 . (b) Aproximação para f_3	57
4.4	(a) Paretos estimados - Função f_4 . (b) Aproximação para f_4	58
4.5	(a) Paretos estimados - Função f_5 . (b) Aproximação para f_5	59
4.6	(a) Paretos estimados - Two Circles. (b) Superfície de separação - Two Circles.	63
4.7	(a) Paretos estimados - Two Moon. (b) Superfície de separação - Two Moon.	64
4.8	Paretos estimados - Base de dados Breast Cancer Wisconsin. . . .	65
4.9	Paretos estimados - Base de dados Diabetes.	65
4.10	Paretos estimados - Base de dados Sonar.	66
4.11	Paretos estimados - Base de dados Australian.	66
4.12	(a) Paretos estimados - Base de dados Banana. (b) Superfície de separação - Banana.	67
4.13	Diagrama CDD. O valor CD é a linha sobre o gráfico, o eixo mos- tra os <i>ranks</i> médios com o menor (melhor) valor à esquerda. Os grupos ligados não são significativamente diferentes ($\alpha = 0.05$). (a) CDD para problemas de regressão. (b) CDD para problemas de classificação.	75
A.1	Paretos estimados - Base de dados Servo.	83
A.2	Paretos estimados - Base de dados Auto MPG.	83
A.3	Paretos estimados - Base de dados Boston Housing.	84
A.4	Paretos estimados - Base de dados Airfoils Selfnoise.	84
A.5	Paretos estimados - Base de dados Forest Fires.	85
A.6	Paretos estimados - Base de dados Concrete Slump.	85
A.7	Paretos estimados - Base de dados Yacht Hydrodynamics.	86
A.8	Paretos estimados - Base de dados Concrete Comp. Strength. . .	86
A.9	Paretos estimados - Base de dados Mackey-glass.	87
A.10	Paretos estimados - Base de dados Four-Gaussian-class.	88
A.11	Paretos estimados - Base de dados Haberman.	89
A.12	Paretos estimados - Base de dados Bupa.	89
A.13	Paretos estimados - Base de dados Vertebral column.	90

LISTA DE FIGURAS

A.14 Paretos estimados - Base de datos Heart diseases.	90
A.15 Paretos estimados - Base de datos Blood transfusion.	91
A.16 Paretos estimados - Base de datos Banknote.	91
A.17 Paretos estimados - Base de datos Fertility.	92
A.18 Paretos estimados - Base de datos ILPD.	92

Lista de Tabelas

4.1	Bases de dados para Regressão.	48
4.2	Bases de dados para Classificação.	48
4.3	Valores médios e desvio padrão de HV para os problemas de Regressão	53
4.4	Valores médios e desvio padrão de CS para os problemas de Regressão. As abreviações $\mathcal{PO}_1$, $\mathcal{PO}_2$ e $\mathcal{PO}_3$ representam as estimativas do conjunto Pareto obtidas pelos métodos MOBJ, SLM e SBFGS, respectivamente.	54
4.5	Valores médios e desvio padrão de HV para os problemas de Classificação	61
4.6	Valores médios e desvio padrão de CS para os problemas de Classificação. As abreviações $\mathcal{PO}_1$, $\mathcal{PO}_2$ e $\mathcal{PO}_3$ representam as estimativas do conjunto Pareto obtidas pelos métodos MOBJ, SLM e SBFGS, respectivamente.	62
4.7	Tempo médio de execução em segundos para os problemas de Regressão	70
4.8	Tempo médio de execução em segundos para os problemas de Classificação	71
4.9	Média e desvio padrão do erro médio quadrático (MSE) para os problemas sintéticos e reais de regressão	74
4.10	Média e desvio padrão da acurácia (ACC) para os problemas sintéticos e reais de classificação	74

Lista de Algoritmos

1	TREINAMENTO MULTIOBJETIVO COM PESOS ESFÉRICOS	36
2	ALGORITMO LEVENBERG-MARQUARDT ESFÉRICO	43
3	ALGORITMO BFGS ESFÉRICO	44

Capítulo 1

Introdução

No desenvolvimento de modelos neurais a partir de uma amostragem de dados, é desejável que os modelos obtidos reconheçam de forma coerente padrões desconhecidos ao processo de treinamento da rede. Esta habilidade é conhecida como capacidade de generalização e depende de diversos fatores para ser alcançada de forma efetiva. Para que o processo de aprendizagem de uma RNA forneça um modelo compatível que represente bem o problema, é necessário um grande esforço por parte do projetista, tanto na definição de parâmetros adequados ao treinamento quanto na definição de uma arquitetura ideal para a construção da rede (Braga *et al.*, 2007).

A aprendizagem de RNAs está dentro da grande área do aprendizado de máquina, que tem fundamentação baseada na teoria do aprendizado estatístico. Esta teoria formaliza as bases do princípio da Minimização do Risco Estrutural como sendo a busca pela menor estrutura capaz de minimizar o Risco Empírico ($\mathbf{R}_{emp}$) (Vapnik, 1995, 1998). O $\mathbf{R}_{emp}$ representa o erro de treinamento da rede a partir de uma amostragem de dados. De maneira geral pode-se definir a capacidade de generalização como a habilidade de uma RNA, através do processo de treinamento, fornecer um modelo que se aproxime da função geradora dos dados, gerando uma boa representação do problema. Devido ao fato de não se conhecer a função de densidade de probabilidade geradora da amostragem dos dados, a tarefa de aprendizagem se torna difícil pois existe apenas uma amostra do problema, fazendo com que o objetivo passe a ser a aproximação de um modelo regressivo (Haykin, 2001). O modelo de regressão que se quer aproximar

é definido a partir de uma função determinística dos dados amostrados com o acréscimo de um erro que representa o desconhecimento da função geradora dos dados.

Para que o ajuste do risco empírico conduza à modelos com boa capacidade de generalização, é necessário que se tenha uma rede com arquitetura e parametrização adequada. Uma rede subdimensionada pode ocasionar na obtenção de modelos incapazes de aproximar a função geradora devido ao alto nível de viés. Redes superdimensionadas podem gerar modelos com grande variabilidade, com complexidade superior à função geradora dos dados, o que conduz à aproximação aos dados de treinamento mas também ao ruído presente na amostragem.

A obtenção de uma rede, que minimize o erro com complexidade compatível ao problema, nos remete a uma definição bem consolidada na literatura, de que o problema geral de aprendizado a partir de amostras de dados tem natureza multiobjetivo (Teixeira *et al.*, 2000; Vapnik, 1995). Portanto, a tarefa de aprendizagem de RNAs tem como alvo a minimização tanto do risco empírico $R_{emp}(\mathbf{w})$ quanto da capacidade do modelo h (Teixeira *et al.*, 2000; Vapnik, 1995), o que não é possível ser realizado de forma conjunta devido à natureza conflituosa no comportamento destes dois objetivos (Braga *et al.*, 2006), gerando assim a necessidade de uma solução de compromisso (*trade-off*) que forneça equilíbrio entre os dois objetivos.

Do ponto de vista da otimização a geração de uma estimativa do conjunto Pareto poderia fornecer soluções não dominadas no espaço de objetivos, $R_{emp}(\mathbf{w})$ e h , de forma que seja possível a seleção de um modelo adequado ao problema de aprendizagem a partir destas soluções. Uma solução é chamada não dominada quando não existe nenhuma outra no espaço factível que seja melhor que ela com relação a um objetivo, sem piorar pelo menos um dos outros objetivos.

O exemplo mostrado pela Figura 1.1 auxilia o entendimento do princípio, onde duas funções convexas hipotéticas, $\phi_e(\cdot)$ e $\phi_c(\cdot)$, representam os riscos empírico e estrutural, respectivamente, em função dos parâmetros livres w . As funções $\phi_e(\cdot)$ e $\phi_c(\cdot)$ possuem pontos de mínimo em w_0 e w_1 . Como $w_0 \neq w_1$, o comportamento simultâneo das duas funções pode ser observado nas regiões A, B e C definidas na figura. Na região A e C, as duas funções podem ser minimizadas de forma simultânea, enquanto na região B elas mostram um comportamento conflitante.

As soluções na região B são não-dominadas, elas representam a fronteira Pareto, e são ótimas do ponto de vista de otimização (Chankong & Haimes, 1983). Qualquer valor de w escolhido nesta região resulta de fato em uma solução de equilíbrio entre $\phi_e(\cdot)$ e $\phi_c(\cdot)$. Através da abordagem multiobjetivo, as soluções não-dominadas que representam a fronteira Pareto devem ser estimadas e uma delas deve ser selecionada de acordo com algum critério de decisão, para representar a solução final do problema.

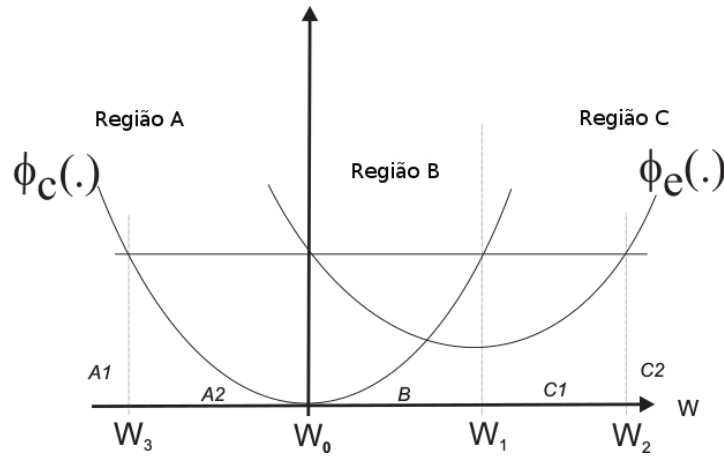


Figura 1.1: Exemplo de duas funções objetivo convexas a serem minimizadas. A Região B é o espaço alvo, onde a solução de compromisso entre as duas funções objetivo deve ser encontrada.

Diversas abordagens para resolver este problema sem o formalismo de otimização multiobjetivo tem aparecido na literatura nas últimas décadas, incluindo-se métodos construtivos e de poda (Costa *et al.*, 2002a; Reed, 1993), validação (Stone, 1974), regularização (Girosi *et al.*, 1995; Hinton, 1989), boosting (Schapire, 1990) e bagging (Breiman, 1996). Com uma abordagem diferente, o aprendizado com Máquinas de Vetores de Suporte (*Support Vector Machines - SVMs*, em inglês) (Boser *et al.*, 1992; Cortes & Vapnik, 1995) é formalizado como um problema bi-objetivo, definido pela maximização da margem dentre as soluções que produzem mínimo erro. A maximização da margem é realizada através da minimização da norma do vetor de pesos (Vapnik, 1995), como mostrado esquematicamente pela Figura 1.2. Entretanto, o problema de *trade-off* no aprendizado com SVMs é resolvido em um espaço de alta dimensionalidade (espaço de

características), após linearização do espaço de entrada dos dados, assumindo uma função de Kernel pré-estabelecida e um parâmetro de regularização C , que é normalmente fornecido pelo usuário ou estimado por meio de validação cruzada. As SVMs tornaram-se atrativas uma vez que a função de aprendizado possui um máximo devido à convexificação, no entanto, ela ainda depende de um dado Kernel e parâmetros de regularização.

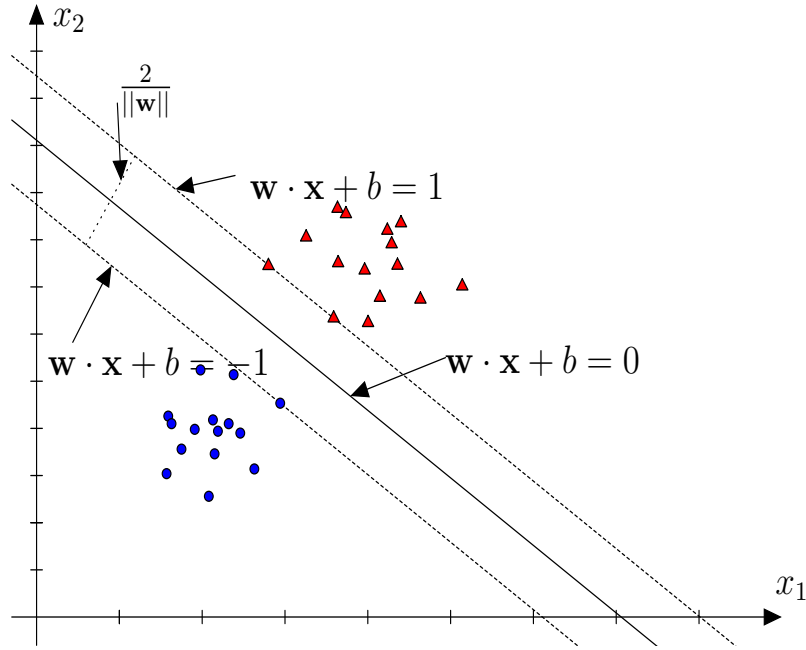


Figura 1.2: Maximização da margem como uma função da norma dos vetores de pesos para o aprendizado com SVMs. O problema é tratado com uma formulação de otimização multiobjetivo.

Em relação ao aprendizado em RNAs, o mapeamento de características geralmente não é considerado separadamente e o problema de *trade-off* é resolvido no espaço de entradas, necessitando de um tratamento diferente das SVMs. Considerando-se uma amostragem de dados representada pelas relações entrada-saída $(\mathbf{x}, y)$ de tamanho N , uma representação geralmente utilizada para $\phi_e(\cdot)$ em RNAs é a função de erro quadrático:

$$\sum e^2 = \sum_{i=1}^N (y_i - f(\mathbf{x}_i, \mathbf{w}))^2 \quad (1.1)$$

enquanto a capacidade do modelo $\phi_c(\cdot)$ é geralmente representada através da norma L_2 dos pesos da rede, definida como:

$$\|\mathbf{w}\| = \sqrt{\sum w_i^2} \quad (1.2)$$

A escolha da norma L_2 como medida de complexidade da rede está relacionada a trabalhos que mostram que a magnitude dos pesos é mais importante que a quantidade destes na rede, uma vez que o controle da magnitude conduz a uma maior capacidade de generalização (Bartlett, 1997) além de maximização da margem de separação dos dados (Boser *et al.*, 1992; Vapnik, 1995) (Vide Figura 1.2).

Para resolver o problema bi-objetivo, adota-se geralmente uma abordagem de otimização restrita onde minimiza-se a norma com restrição do erro (Boser *et al.*, 1992) ou através da minimização do erro com norma restrita (Teixeira *et al.*, 2000).

Diferentes formulações para o aprendizado multiobjetivo, com base em aproximações da fronteira Pareto, têm sido apresentadas na literatura, algumas delas no domínio das redes Perceptron de Múltiplas Camadas (*Multi-Layer Perceptrons* - *MLP*, em inglês) (Costa *et al.*, 2002b, 2007, 2012; Jin & Sendhoff, 2009; Teixeira *et al.*, 2000) e outras baseadas em diferentes estruturas como as redes de base radial (*Radial Basis Function* - *RBF*) apresentadas em (Kokshenev & Braga, 2008, 2010). As dificuldades que a maioria desses métodos têm enfrentado estão relacionadas à não-convexidade e não-linearidade inerentes ao problema de otimização destes dois objetivos. Alguns desses métodos são capazes de gerar aproximações da fronteira Pareto, porém, sem provas de convergência global (Costa *et al.*, 2012). Embora a qualidade do modelo final dependa da estratégia de seleção, a aproximação da fronteira Pareto determina a qualidade dos modelos disponíveis para seleção. Dentre as dificuldades para obtenção de boas aproximações por esses métodos, está o fato de necessitarem lidar com o problema de otimização conjunta das funções objetivo não-lineares, geralmente tratado como um problema de otimização restrita.

Neste contexto, a motivação para o desenvolvimento deste trabalho concentra-se na simplificação do problema de aprendizado multiobjetivo através de uma representação conveniente para resolução do problema formulado pela minimização

de uma função de erro $\sum e^2$ e um valor fixo da norma do vetor de pesos $|\mathbf{w}|^2$. O princípio do método proposto fundamenta-se na representação dos pesos da rede MLP através de coordenadas esféricas, de forma que a minimização do erro pode ser realizada para diferentes valores de raio r_j de uma hipersfera $w_1^2 + w_2^2 + \dots + w_n^2 = r_j^2$. O conceito geométrico é intuitivo e pode ser observado no exemplo esquemático da Figura 1.3. Abordagens baseadas em restrição de desigualdade como aquela que utiliza o método ϵ -restrito (Teixeira *et al.*, 2000) minimiza o erro dentro da esfera $\|\mathbf{w}\|^2 \leq r_j^2$. Abordagens baseadas em restrição de igualdade (Costa *et al.*, 2007) minimizam o erro sobre a superfície da esfera $\|\mathbf{w}\|^2 = r_j^2$. Nas duas abordagens, o problema de otimização é resolvido para valores pré-estabelecidos de r_j . Diferentemente destas abordagens que utilizam algoritmos que tratam as restrições no processo de otimização, a formulação proposta torna o processo de otimização irrestrito, devido ao fato de as variáveis de otimização serem os ângulos da representação esférica para um dado valor de raio r_j .

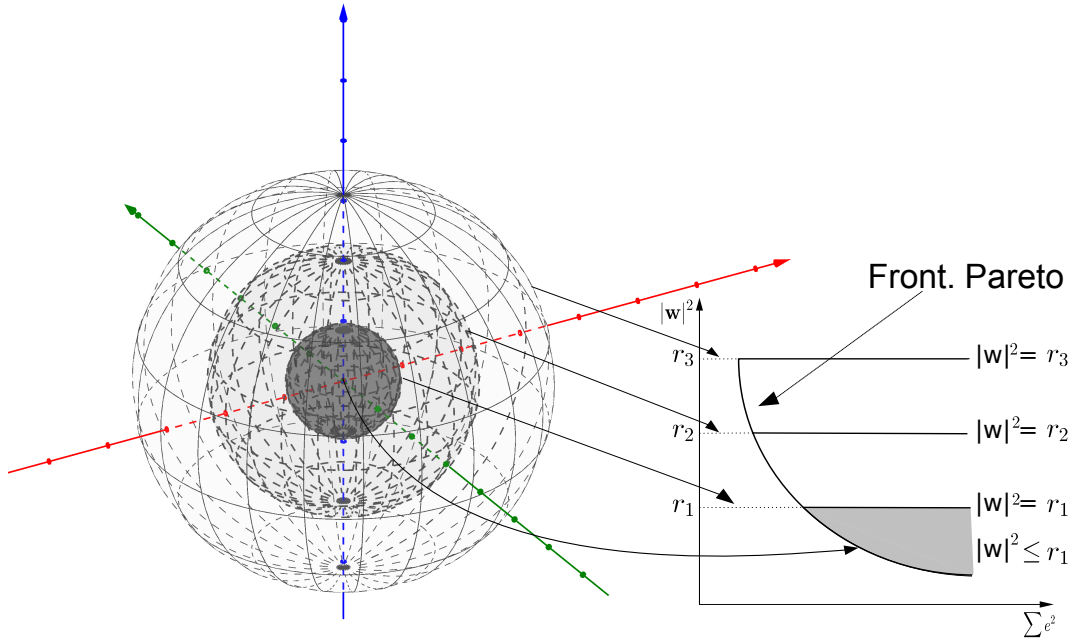


Figura 1.3: Mapeamento do espaço de parâmetros para o espaço de objetivos com soluções restritas às regiões $\|\mathbf{w}\| \leq r_j$ e $\|\mathbf{w}\| = r_j$.

Ao se tratar a minimização da norma como uma restrição de igualdade de forma intrínseca, a representação proposta permite a resolução do problema de otimização através de algoritmos para otimização mono-objetivo sem tratamento de restrições, dado que visa-se encontrar o mínimo valor de erro em um espaço limitado pela norma. Variando-se os valores alvos de norma pode-se efetuar a minimização do erro para os subproblemas gerados de maneira que se obtenha uma estimativa eficiente do conjunto Pareto. A transformação do problema original em subproblemas mono-objetivo irrestritos junto à facilidade de se obter derivadas num problema baseado em coordenadas esféricas motiva a utilização de algoritmos determinísticos para o treinamento da rede nesta formulação.

Este trabalho apresenta ainda versões dos algoritmos *Back-Propagation* baseado em gradiente descendente (GD), *Levenberg-Marquardt* (LM) e *BFGS* de maneira que a direção de busca é obtida em termos da formulação dos pesos no sistema de coordenadas esféricas, gerando soluções que minimizam o erro de treinamento sem escapar da superfície de norma fixa definida, fazendo com que o erro mínimo encontrado para um determinado valor de norma gere uma solução específica no conjunto Pareto estimado. Aplicando-se os algoritmos propostos a problemas de regressão e classificação, foram efetuadas análises comparativas quanto à estimativa do conjunto Pareto e com relação ao erro/acurácia de teste mensurado através solução escolhida pelo decisor de mínimo erro de validação. As estimativas de soluções Pareto obtidas pelos métodos propostos foram comparadas às estimativas geradas pelo clássico método MOBJ (Teixeira *et al.*, 2000), pioneiro no treinamento multiobjetivo no domínio da complexidade aparente. Foi possível verificar que os métodos propostos geram soluções que cobrem, com significância estatística, aquelas geradas pelo MOBJ nos problemas avaliados. O algoritmo *Levenberg-Marquardt* com pesos esféricos (SLM) obteve as melhores estimativas do conjunto Pareto para a maioria dos problemas em todos os experimentos realizados. Com relação às soluções finais obtidas pelo decisor de validação a partir do conjunto estimado, o SLM obteve os melhores resultados quando comparado a métodos do estado da arte presentes na literatura, com base na estatística de *rank* médio. Entretanto não se comprovou significância estatística com relação à maioria dos métodos utilizados na comparação.

1.1 Objetivo e Contribuições

O objetivo deste trabalho é o desenvolvimento de uma nova representação para os pesos de redes MLP que simplifica o processo de otimização para o aprendizado multiobjetivo, além do desenvolvimento de algoritmos para o treinamento da rede com a representação proposta, que sejam capazes de gerar estimativas eficientes do conjunto Pareto. Este objetivo é expresso pelas contribuições do trabalho, que são apresentadas a seguir de maneira resumida:

- Formulação de uma nova representação para o vetor de pesos em redes MLP, de forma que o procedimento de otimização restrita geralmente utilizado para o aprendizado multiobjetivo da rede seja simplificado, sem a necessidade do tratamento de restrições. A formulação proposta define a representação dos pesos através de coordenadas esféricas:

$$\begin{aligned}
 w_1 &= r_j \sin \theta_1 \\
 w_2 &= r_j \cos \theta_1 \sin \theta_2 \\
 w_3 &= r_j \cos \theta_1 \cos \theta_2 \sin \theta_3 \\
 &\vdots \\
 w_{n_w-1} &= r_j \cos \theta_1 \cdots \cos \theta_{n-2} \sin \theta_{n-1} \\
 w_{n_w} &= r_j \cos \theta_1 \cdots \cos \theta_{n-2} \cos \theta_{n-1}
 \end{aligned}$$

A modelagem do problema de otimização utilizando esta abordagem torna possível o tratamento de um dos objetivos ($\min ||\mathbf{w}||$) de forma intrínseca, fazendo com que o objetivo do treinamento seja buscar pelo menor erro no espaço limitado pelo valor fixo de norma. Na Figura 1.4 é apresentada a superfície de um valor específico de norma onde podem ser visualizados vários exemplos de soluções para o erro, sendo possível visualizar uma solução que representa o erro mínimo neste espaço restrito. A geração de uma estimativa do conjunto Pareto, consiste em minimizar o erro a partir de diferentes raios r_j de uma hipersfera (Vide, Figura 1.3). A Figura 1.5 mostra a limitação imposta à superfície de erro considerando-se três valores de r_j , de

maneira que resolvendo-se os subproblemas de otimização para cada um destes valores serão obtidas três soluções de mínimo erro que comporão a estimativa final do conjunto Pareto.

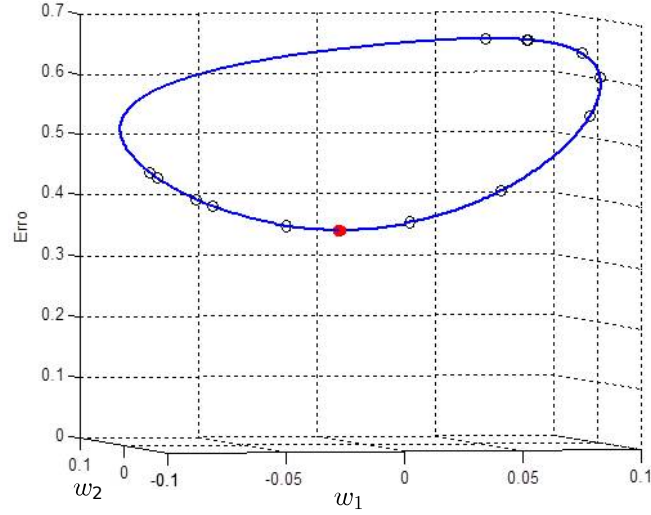


Figura 1.4: Solução no espaço limitado por um determinado valor de norma.

Adicionalmente a nova representação fornece um recurso útil ao processo de otimização, onde ao minimizar o erro considerando-se um valor particular de r_j pode-se obter a solução inicial para o procedimento de otimização em r_{j+1} apenas projetando a solução obtida para o espaço com norma maior, mantendo os valores de ângulos obtidos e consequentemente a direção de busca neste espaço. Através da Figura 1.5 pode-se visualizar também a diferença dos espaços de busca considerando-se a formulação tradicional com restrições de desigualdade do método MOBJ e os métodos que trabalham com representação esférica de pesos.

- Desenvolvimento do algoritmo *Gradiente descendente* esférico. Este método é utilizado com um modelo teórico para a implementação de algoritmos esféricos de otimização baseados em direção de busca. A descrição deste algoritmo está na Seção 3.2.1, onde os principais recursos para obtenção de direção de busca são apresentados. Embora este algoritmo não tenha sido avaliado neste trabalho, dado que existem algoritmos mais eficientes para o

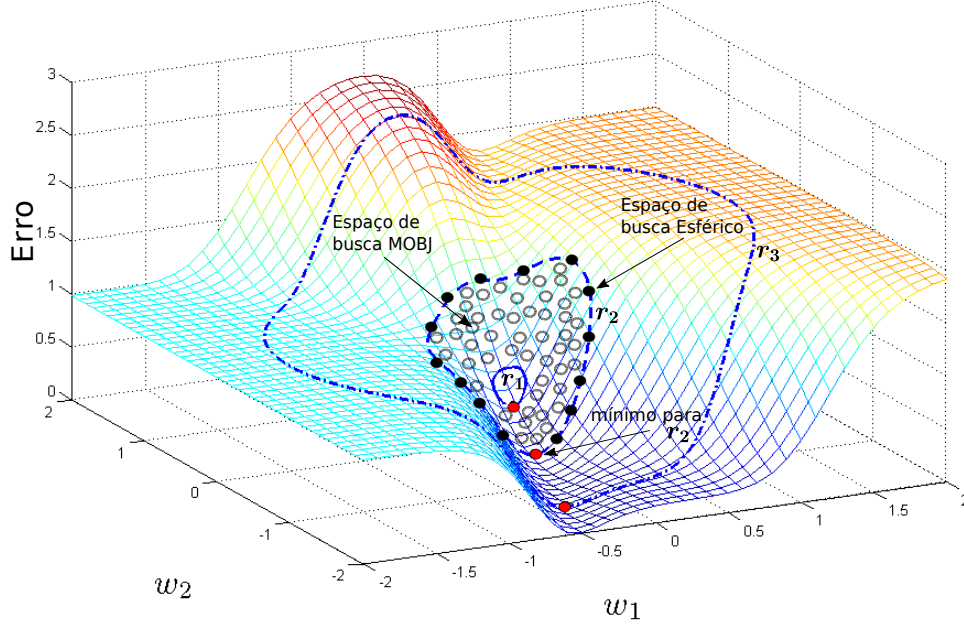


Figura 1.5: Curvas de erro para valores constantes de norma.

problema, ele demonstrou eficiência para estimativa de soluções, como pode ser visto no trabalho apresentado em (Rocha *et al.*, 2015).

- Desenvolvimento dos algoritmos *Levenberg-Marquardt* Esférico (SLM) e *BFGS* esférico (SBFGS). Estes algoritmos têm habilidades que eliminam a maioria dos problemas encontrados em relação à convergência e velocidade presentes em outros métodos de treinamento de MLPs como o *Back-Propagation* clássico que é baseado no gradiente descendente. Ao serem adaptados ao espaço esférico, demonstraram eficiência para geração de estimativas do conjunto Pareto além de rápida convergência com relação ao gradiente descendente. A Figura 1.6 exemplifica um problema de aproximação de função utilizando-se o SLM. A aproximação da função Seno é utilizada no exemplo, onde uma amostragem com ruído Gaussiano na saída representa o conjunto de dados para treinamento. Cada solução da estimativa obtida para o conjunto Pareto pode ser visualizada junto ao modelos que cada uma representa, variando de soluções com *underfitting* a soluções com *overfitting*. A solução de *trade-off* selecionada pelo decisor de validação está destacada.

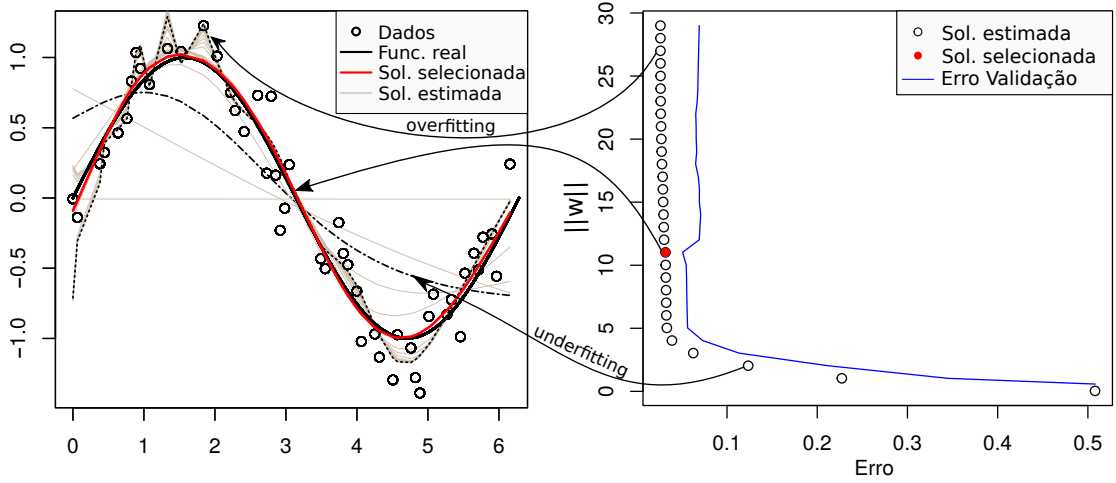


Figura 1.6: Estimativa do conjunto Pareto e respectivos modelos fornecidos pelo algoritmo *Levenberg-Marquardt* Esférico. A solução selecionada é aquela com menor erro de validação.

1.2 Organização do Trabalho

O capítulo 2 apresenta brevemente alguns dos principais métodos de treinamento de Redes Neurais Artificiais (RNA) do tipo *Multi-layer Perceptron*, define conceitos fundamentais acerca da Teoria do aprendizado estatístico e do dilema entre o viés e a variância. Ao final são apresentados conceitos básicos de otimização multiobjetivo e definições fundamentais sobre o aprendizado multiobjetivo de RNAs.

O capítulo 3 explica detalhadamente a formulação para representação esférica de redes MLP. Neste capítulo ainda é detalhado o desenvolvimento de três métodos para treinamento da rede a partir da nova formulação, o *Back-Propagation* esférico, o *Levenberg-Marquardt* esférico e o *BFGS* esférico.

O capítulo 4 apresenta os resultados obtidos após os experimentos realizados utilizando os métodos para treinamento multiobjetivo propostos e o MOBJ, aplicados a problemas de regressão e classificação. Os experimentos realizados consistem na avaliação das estimativas do conjunto Pareto obtidas pelos métodos bem como a solução final selecionada.

O capítulo 5 apresenta as conclusões desta tese de doutorado além da discussão acerca de pontos a serem investigados em trabalhos futuros.

Capítulo 2

Referencial Teórico

Neste capítulo, são apresentados inicialmente alguns dos principais métodos para o treinamento de Redes Neurais Artificiais (RNA) do tipo *Multi-layer Perceptron*. Subsequentemente o problema geral de aprendizado de RNAs é apresentado sob a perspectiva da Teoria do Aprendizado Estatístico (*Statistical Learning Theory* - *SLT*, em inglês), onde são descritos alguns conceitos fundamentais acerca desta teoria. Descreve-se ainda o Dilema entre o Viés e a Variância, que junto à *SLT*, enfatizam o aspecto multiobjetivo do aprendizado a partir de dados. Por fim, definições acerca do aprendizado multiobjetivo de redes neurais são detalhadas e a formulação clássica para o problema de treinamento multiobjetivo é apresentada.

2.1 Treinamento de Redes Neurais Artificiais

2.1.1 *Métodos Clássicos para Treinamento de RNA*

A maneira clássica de se ajustar uma rede neural do tipo *Multi-layer Perceptron* (MLP) (Haykin, 2009) é através do processo de treinamento que visa apenas a minimização do erro médio quadrático, sendo que o mais conhecido método para realização deste processo é o *Error Back-Propagation* (BP) (Rumelhart & McClelland, 1986), que utiliza o gradiente descendente como direção de busca no processo numérico de otimização, através da geração de aproximações lineares da função de erro em relação aos pesos. Este método tem problemas conhecidos

quanto à dificuldade de convergência, utilizando em geral, uma forma de treinamento onde a atualização dos pesos ocorre a cada apresentação de um padrão à rede, de forma que a cada ciclo N correções de pesos são efetuadas.

Algumas variações deste método foram propostas com o acréscimo de alterações que conduzem a um melhor desempenho tanto em relação à eficiência do ajuste quanto ao custo computacional dispendido. Dentre estes métodos podem ser destacados o de gradientes conjugados (Charalambous, 1992), Gauss-Newton, *Resilient Back-Propagation* (Riedmiller & Braun, 1992) e *Levenberg-Marquardt* (Hagan & Menhaj, 1994). Com este último, foram obtidos bons resultados quanto à eficiência em se obter a solução em diversos problemas.

O algoritmo *Levenberg-Marquardt* (LM) é um eficiente método de treinamento de RNAs. Pode ser definido como uma aproximação do método de Newton, sendo caracterizado como uma versão híbrida dos métodos Gauss-Newton e Gradiente Descendente (GD). O LM explora a velocidade de convergência do método Gauss-Newton em regiões próximas a um ótimo local, e a robustez do Gradiente Descendente em regiões mais distantes, dado que diferentemente do método Gauss-Newton, o GD possui menor sensibilidade quanto ao ponto de partida do processo de busca. O LM em sua implementação clássica utiliza um treinamento em lote, onde todos os pesos da rede só são atualizados após um ciclo completo sob os dados de treinamento. O principal recurso fornecido pelo LM é a matriz Jacobiana, cujo cálculo se torna o gargalo do método em problemas com muitos parâmetros devido à grande demanda de memória para armazenamento, além da necessidade da operação de inversão de matriz que tem alto custo computacional.

2.1.2 Capacidade Generalização em RNAs

Como mostrado em (Costa *et al.*, 2002b), o desempenho de generalização é a principal preocupação no treinamento de redes neurais. O *Back-Propagation* e suas variações, em suas definições originais, não levam em consideração esta dificuldade, dado que por levar em consideração somente o erro do modelo gerado, tem-se garantias de aproximação apenas para os dados usados no treinamento da rede, que em geral são amostrados de uma distribuição desconhecida.

2.1 Treinamento de Redes Neurais Artificiais

Para que a utilização de $\mathbf{R}_{emp}$ seja consistente e se aproxime do risco esperado, algumas premissas devem ser verdadeiras, como por exemplo ter uma amostragem de dados grande o suficiente e bem distribuída, além de uma rede com topologia e parametrização adequada à complexidade do problema. Essas características normalmente não estão disponíveis, o que pode ocasionar em dois problemas no treinamento. O primeiro problema é o sub-ajuste da rede devido à utilização de uma rede subdimensionada, onde a arquitetura é insuficiente para aproximação da função geradora expressa pelos dados amostrados, ocasionando na chamada polarização. Redes com esta característica geram modelos similares quanto ao erro de treinamento e validação, com pouca variabilidade, impossibilitando o ajuste à complexidade do problema. Por outro lado, a construção de uma rede com topologia mais complexa resolveria o problema da polarização mas poderia provocar um outro efeito indesejado, o sobre-ajuste. Soluções super-dimensionadas podem minimizar o erro de treinamento mas poderão gerar soluções que modelam também o ruído presente nos dados, não representando bem a função geradora. Estas soluções apresentam alta variância e também provocam perda da capacidade de generalização. Desta maneira, a obtenção de soluções eficientes, *i.e.*, com boa capacidade de generalização, depende do ajuste da complexidade do modelo junto à minimização do erro. O controle destes dois fatores direciona a busca por um modelo que não seja demasiadamente rígido a ponto de não modelar os dados e nem flexível demais a ponto de modelar o ruído inerente ao processo de amostragem.

De maneira geral a capacidade de generalização de Redes Neurais Artificiais (RNAs) pode ser entendida através da Teoria do Aprendizado Estatístico, que formaliza as bases do princípio da Minimização do Risco Estrutural. Outras definições importantes para o entendimento da capacidade de generalização concentram-se no dilema entre o viés e a variância, que é consistente com os conceitos apresentados na SLT. A seção 2.2 detalha aspectos fundamentais deste princípio enquanto na seção 2.3 são apresentados conceitos relativos ao dilema entre o viés e a variância.

2.2 Teoria do Aprendizado Estatístico

A definição das condições de convergência de uma aproximação empírica para o risco esperado é uma das principais contribuições da Teoria do Aprendizado Estatístico Vapnik (1995, 1998) para o aprendizado de máquina (*Machine Learning* - *ML*, em inglês). Considerando-se uma função de aproximação $f(\mathbf{x}, \mathbf{w})$ obtida através da seleção de um vetor de parâmetros $\mathbf{w}$ e, sob condições ideais, considerando-se ainda que a função de densidade de probabilidade conjunta $p(\mathbf{x}, y)$ é conhecida e todas as relações entrada-saída $(\mathbf{x}, y)$ estão disponíveis, a estimativa do Risco Esperado é representada através da Equação (2.1),

$$R(\mathbf{w}) = \int Q(y, f(\mathbf{x}, \mathbf{w}))p(\mathbf{x}, y)dxdy \quad (2.1)$$

onde $Q(y, f(\mathbf{x}, \mathbf{w}))$ é uma função de perda. Com base na Equação (2.1) o aprendizado pode ser definido como sendo o processo de busca pela função de aproximação $f(\mathbf{x}, \mathbf{w}^*)$, onde $\mathbf{w}^*$ é o vetor de parâmetros que minimiza $R(\mathbf{w})$. Entretanto, $p(\mathbf{x}, y)$ é desconhecida e tem-se apenas uma amostragem das relações entrada-saída $(\mathbf{x}, y)$, o que impossibilita o cálculo da integral na Equação (2.1). Devido à impossibilidade de se calcular $R(\mathbf{w})$ o aprendizado baseado em dados visa uma aproximação de $R(\mathbf{w})$ através da Minimização do Risco Empírico ($\mathbf{R}_{\text{emp}}$) calculado a partir da amostragem de dados disponíveis, o chamado conjunto de treinamento. A indução de função a partir de um conjunto limitado de dados é um dos princípios básicos dos modelos indutivos em aprendizado de máquina, entretanto, a convergência de $\mathbf{R}_{\text{emp}}$ para $R(\mathbf{w})$ é válida apenas quando se tem um conjunto de treinamento suficientemente grande e representativo, o que não ocorre para a maioria dos problemas reais. Quando se tem apenas uma amostragem do problema a convergência de $\mathbf{R}_{\text{emp}}$ para $R(\mathbf{w})$ depende também da capacidade do modelo.

2.2.1 Princípio de Minimização do Risco Estrutural

Definido em Vapnik (1995, 1998) o Princípio de Minimização do Risco Estrutural (*Structural Risk Minimization* - *SRM*, em inglês) introduz o conceito de Capacidade de Modelo. A partir deste princípio, é estabelecido um limite superior para

2.3 Dilema entre o Viés e a Variância

o Risco Esperado de uma máquina de aprendizado. A Expressão (2.2) define este limite, onde é mostrado que este depende de dois termos. O primeiro termo é o $\mathbf{R}_{\text{emp}}$ enquanto o segundo representa a Capacidade do Modelo Ω , que depende do tamanho da amostragem de dados (N) e de uma medida da complexidade h da classe de funções implementadas pela máquina de aprendizagem, onde h denota a dimensão Vapnik-Chervonenkis (VC) (Vapnik, 1995) e η é um intervalo de confiança.

$$R(\mathbf{w}) \leq R_{\text{emp}}(\mathbf{w}) + \sqrt{\left(\frac{h(\log(\frac{2N}{h}) + 1) - \log(\frac{\eta}{4})}{N} \right)} \quad (2.2)$$

Através da expressão (2.2) fica claro que quando N tende a infinito, $R(\mathbf{w})$ é limitado apenas por $\mathbf{R}_{\text{emp}}$. Entretanto, como geralmente não se tem uma grande amostragem de dados nos casos reais, um pequeno $\mathbf{R}_{\text{emp}}$ não implica em um baixo valor de $R(\mathbf{w})$. De maneira geral, para que se tenha um baixo erro de aproximação é necessário minimizar tanto $\mathbf{R}_{\text{emp}}$ quanto h .

2.3 Dilema entre o Viés e a Variância

Outra maneira de se entender melhor os fatores que afetam o erro de aproximação é através do Dilema entre o Viés e a Variância (Geman *et al.*, 1992), que considera a utilização do erro quadrático para representação de $\mathbf{R}_{\text{emp}}$ conforme a Equação (2.3). O valor esperado do erro quadrático $(y - f(\mathbf{x}, \mathbf{w}))^2$ é decomposto em dois termos, como mostrado na Equação (2.4). O primeiro termo é o valor esperado do erro de amostragem, que representa a incerteza inerente ao processo de obtenção da amostra de dados. O segundo termo corresponde ao valor esperado do erro de aproximação. O valor esperado de y dado que $\mathbf{x}$ foi produzido pela função geradora $f_g(\mathbf{x})$ é descrito por $E[y|\mathbf{x}]$.

$$R_{\text{emp}} = \frac{1}{N} \sum_{i=1}^N (y - f(\mathbf{x}, \mathbf{w}))^2 \quad (2.3)$$

$$E[(y - f(\mathbf{x}, \mathbf{w}))^2] = E[(y - E[y|\mathbf{x}])^2] + E[(f(\mathbf{x}, \mathbf{w}) - E[y|\mathbf{x}])^2] \quad (2.4)$$

2.3 Dilema entre o Viés e a Variância

Considera-se que os dados de saída (y) amostrados a partir do sistema a ser modelado correspondem ao resultado da função geradora $f_g(\mathbf{x})$ com um ruído adicional γ representando a incerteza dos dados. Desta forma o aprendizado a partir de uma amostragem de dados visa a aproximação de $f_g(\mathbf{x})$ e não de $f_g(\mathbf{x}) + \gamma$. Desta forma, um modelo estimado apropriado considerando-se a incerteza γ é aquele onde o resíduo obtido se aproxima da incerteza dos dados e, portanto, $f(\mathbf{x}, \mathbf{w}) \approx f_g(\mathbf{x})$.

Uma conclusão importante obtida a partir da Equação (2.4), é que o controle do erro de aproximação através da minimização do erro somente ocorrerá em um cenário sem a presença de incerteza, o que não ocorre na maioria dos problemas reais. Desta forma, para um melhor entendimento dos aspectos que interferem no erro de aproximação, decompõe-se o segundo termo da Equação (2.4) obtendo-se a Equação (2.5), onde o primeiro termo obtido representa o viés $\mu_{\{f(\mathbf{x}, \mathbf{w})\}}$ e o segundo é a variância $\sigma_{\{f(\mathbf{x}, \mathbf{w})\}}^2$, para uma família de modelos.

$$E [(f(\mathbf{x}, \mathbf{w}) - E[y|\mathbf{x}])^2] = (E[f(\mathbf{x}, \mathbf{w})] - E[y|\mathbf{x}])^2 + E [(f(\mathbf{x}, \mathbf{w}) - E[f(\mathbf{x}, \mathbf{w})])^2] \quad (2.5)$$

A partir das Equações (2.4) e (2.5) pode-se concluir que o valor esperado do Risco Empírico $\mathbf{R}_{\text{emp}}$, definido através da Equação (2.3), depende dos seguintes fatores: incerteza da amostragem (γ), viés do modelo ($\mu_{\{f(\mathbf{x}, \mathbf{w})\}}$) e variância do modelo ($\sigma_{\{f(\mathbf{x}, \mathbf{w})\}}^2$).

O termo viés representa o valor esperado da distância entre o valor esperado de uma família de modelos e o valor esperado da função geradora $f_g(\mathbf{x})$. De forma similar, a variância é uma medida da distância entre um modelo selecionado e o valor esperado da família de modelos. A variância é controlada através da geração de modelos com menor complexidade, entretanto quando limita-se a complexidade um maior nível de viés é acrescentado aos modelos. Desta maneira, sabendo-se que o ajuste de parâmetros afeta somente o viés e a variância, pode-se afirmar também que o objetivo do aprendizado é balancear estes dois fatores conflitantes.

De acordo com a Teoria do Aprendizado Estatístico apresentada na Seção 2.2, o erro de aproximação estimado pelo Risco Esperado $R(\mathbf{w})$ representado pela Expressão (2.2), depende apenas do Risco Empírico $\mathbf{R}_{\text{emp}}$ e da capacidade do

2.3 Dilema entre o Viés e a Variância

modelo h para um valor fixo de N . Esta propriedade de máquinas de aprendizado também pode ser observada através da decomposição do erro quadrático mostrada na Equação (2.4). Esta relação mostra que quanto maior a capacidade do modelo maior é a habilidade do modelo se ajustar aos dados mas também ao ruído inerente ao processo de amostragem ($f(\mathbf{x}, \mathbf{w}) \approx f_g(\mathbf{x}) + \gamma$), fazendo com que o modelo esteja distante de $f_g(\mathbf{x})$ por uma distância γ . Uma capacidade maior que a necessária para o problema pode conduzir a modelos com baixo viés e grande variância, gerando o efeito de sobreajuste (*overfitting*, em inglês). Por outro lado, se uma limitação muito grande for imposta à capacidade do modelo, de forma que esta seja menor que a necessária para o problema de aprendizado em questão, pode-se gerar modelos com grande viés e pequena variância, conduzindo ao efeito de subajuste (*underfitting*, em inglês), onde modelo é demasiadamente limitado para aproximar o problema. Desta forma, para se obter um baixo erro de aproximação, os modelos devem tender à função geradora (baixo viés) com baixa variabilidade (baixa variância).

Embora o viés não possa ser estimado de forma direta devido ao desconhecimento do termo $E[y|\mathbf{x}]$, ele pode ser indiretamente avaliado verificando-se o quanto o modelo se adapta à incerteza. Um modelo com erro muito baixo, provavelmente está distante de $f_g(\mathbf{x})$, dado que deve-se considerar a premissa de existência de incerteza ($\gamma \neq 0$) e, sabe-se que um baixo valor de erro obtido pelo modelo, não implica em um baixo erro de aproximação. Portanto, deve-se controlar a capacidade para que se limite a adaptação ao ruído γ , o que é também consistente com os princípios da *SLT*. Ademais, o controle da capacidade no espaço de parâmetros afeta diretamente a variância do modelo, dado que o segundo termo da Equação (2.5) depende somente da imposição de restrições aos parâmetros das funções candidatas. Portanto, os termos Viés e Variância, que compõem o erro de aproximação $E[(f(\mathbf{x}, \mathbf{w}) - E[y|\mathbf{x}])^2]$ podem ser aferidos por meio do erro $\mathbf{R}_{\text{emp}}$ e da capacidade h das funções, que possuem comportamento conflitante. Esta relação sugere que o problema geral de aprendizado indutivo deveria ser tratado como um problema de aprendizado bi-objetivo. A seção 2.4.2 discute a formulação do problema de aprendizado considerando-se os dois objetivos apresentados, após alguns conceitos básicos de suporte.

2.4 Aprendizado Multi-objetivo

Nesta seção, são apresentados inicialmente conceitos básicos de otimização multiobjetivo e posteriormente a formulação do treinamento de redes MLP visando a minimização do risco estrutural como um problema multiobjetivo. Ao fim da seção são mostrados alguns métodos para o problema de decisão multiobjetivo em aprendizado de máquina.

2.4.1 Conceitos Básicos em Otimização Multiobjetivo

Um problema de otimização corresponde à busca das melhores variáveis presentes em um determinado problema com o propósito de maximizar ou minimizar um ou mais objetivos definidos através de funções ou funcionais, que fornecem uma forma de mensurar tais objetivos. Este problema envolve ainda a adequação do procedimento de busca à um espaço definido por restrições que podem existir. Um problema de otimização com apenas um único objetivo será representado por apenas uma função objetivo a partir da qual será possível identificar a solução mais apropriada ao projeto. Por outro lado, um problema de otimização multiobjetivo, além de satisfazer as possíveis restrições, deverá levar em consideração a existência de dois ou mais objetivos conflitantes que serão representados por um vetor de funções objetivo a serem minimizadas e (ou) maximizadas, gerando ao final do processo um conjunto de soluções com diferentes *trade-offs* (Branke *et al.*, 2008). Desta maneira, objetiva-se encontrar uma solução do conjunto obtido em que os valores de todas as funções são considerados aceitáveis pelo projetista (Osyczka, 1985). A formulação matemática para problemas desta natureza é detalhada nos trabalhos de (Sawaragi *et al.*, 1985), (Ferreira., 1999) e (Korhonen, 1998), sendo que a forma geral do problema pode ser definido como:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \{f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})\} \\ \text{s.a :} \quad & \mathbf{x} \in S \end{aligned} \tag{2.6}$$

onde $m \geq 2$ define a quantidade de funções objetivo, $f_i(\mathbf{x}) : \mathbf{R}^n \rightarrow \mathbf{R}$, e o vetor de variáveis de decisão $\mathbf{x} = (x_1, x_2, \dots, x_n)$ pertence à região factível definida pelo vetor S . O conjunto S pode ser definido como:

$$\mathcal{S} = \begin{cases} g_i(\mathbf{x}) \leq 0 & \forall i = 1, 2, \dots, p \\ h_j(\mathbf{x}) = 0 & \forall j = 1, 2, \dots, q \\ x \in \mathbf{X} \end{cases} \quad (2.7)$$

onde $g_i(\mathbf{x})$ para $i = 1, 2, \dots, p$ e $h_j(\mathbf{x})$ para $j = 1, 2, \dots, q$ são, respectivamente, as restrições de desigualdade e igualdade, e $\mathbf{X} \subset \mathbf{R}^n$. O espaço de variáveis $\mathbf{X} \subset \mathbf{R}^n$ é mapeado para o espaço de objetivos $\mathbf{Y} \subset \mathbf{R}^m$ através do vetor de funções objetivo $f(\cdot) : \mathbf{X} \mapsto \mathbf{Y}$.

2.4.1.1 O Conjunto Pareto-Ótimo

Considerando-se todas as soluções encontradas por um algoritmo de otimização multiobjetivo e observando-se que estes objetivos são conflitantes e nem sempre será possível melhorar um objetivo sem piorar o outro, existe a necessidade de se definir o que seria uma solução ótima ou um conjunto ótimo de soluções. Neste contexto, o conceito de eficiência ([Pareto, 1897](#)) faz-se necessário, consistindo em dizer que uma determinada solução é considerada ótima se não existe nenhuma outra solução factível que seja melhor em relação a um objetivo sem piorar pelo menos um dos outros presentes no vetor de objetivos. Desta forma esta solução é dita pertencente ao conjunto Pareto-Ótimo, modificando-se a definição de ótimo em problemas com um único objetivo. O conjunto Pareto-Ótimo é definido por todas as soluções que satisfaçam as condições para serem consideradas eficientes. Algumas definições que são necessárias para um melhor entendimento deste conjunto de soluções são expressas a seguir ([Takahashi, 2007](#)):

Definição 1 (*Dominância*) *Seja uma solução $\mathbf{x}_1 \in S$, esta solução domina uma solução $\mathbf{x}_2 \in S$ se $f_i(\mathbf{x}_1) \leq f_i(\mathbf{x}_2) \forall i = 1, \dots, m$ e $\exists i \mid f_i(\mathbf{x}_1) \neq f_i(\mathbf{x}_2)$. De maneira equivalente, diz-se que $f(\mathbf{x}_1) \in \mathbf{Y}$ domina $f(\mathbf{x}_2) \in \mathbf{Y}$ no espaço de objetivos. Geralmente, esta relação é denominada Pareto dominância sendo normalmente escrita como $f(\mathbf{x}_1) \prec f(\mathbf{x}_2)$.*

Definição 2 (*Solução Pareto-Ótima*) *Seja uma solução $\mathbf{x}^* \in S$, diz-se que esta é uma solução Pareto-Ótima se não existe nenhuma outra solução $\mathbf{x} \in S$ tal que $f(\mathbf{x}) \prec f(\mathbf{x}^*)$, ou seja, não existe solução factível que domina $\mathbf{x}^*$.*

A partir das definições pode-se expressar o Conjunto Pareto-Ótimo como:

$$P = \{\mathbf{x}^* \in S | \nexists \mathbf{x} \in S : f(\mathbf{x}) \prec f(\mathbf{x}^*)\} \quad (2.8)$$

e a partir de P deseja-se escolher uma solução $\mathbf{x}^* \in S$ que represente uma solução final para o problema de otimização. Entretanto não há como se definir a partir das funções objetivo, que uma solução é melhor que outra, fazendo com que as soluções presentes no conjunto Pareto-Ótimo sejam apenas candidatas a solução final. Com o propósito de se escolher uma solução final pertencente ao conjunto P existe a necessidade de um processo de decisão, que poderia ser feita, por exemplo, com o auxílio de uma entidade externa especialista ou a partir da definição de critérios dependentes de cada problema (Chankong & Haimes, 1983).

2.4.2 Formulação Multi-objetivo (MOBJ) para o Treinamento de Redes MLPs

Considerando-se uma rede MLP e, conhecendo-se a relação entre a capacidade de aprendizagem e a magnitude dos pesos da rede, pode-se definir a norma dos pesos da rede $\|\mathbf{w}\|$ como uma medida de controle da variância (Bartlett, 1997). Desta maneira pode-se definir um algoritmo que controle erro e complexidade em termos de um modelo de otimização definido pela minimização conjunta das funções objetivo apresentadas nas Equações (2.9) e (2.10), sendo J_1 o erro médio quadrático (*Mean Square Error - MSE*, em inglês) e J_2 a norma dos pesos da rede.

$$J_1 = \frac{1}{2} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (2.9)$$

$$J_2 = \|\mathbf{w}\| = \sqrt{\sum_{i=1}^{n_w} w_i^2} \quad (2.10)$$

Na Equação (2.9), y_i e $\hat{y}_i$ são os valores para a saída desejada e a saída da rede para i -ésima amostra, respectivamente, sendo que N é o número total de amostras do conjunto de treinamento. Em (2.10), w_i é o i -ésimo peso da rede e n_w é o número total de pesos.

Uma das possíveis soluções do problema de otimização caracterizado pela minimização conjunta das funções J_1 e J_2 é apresentada em (Teixeira *et al.*, 2000) através do algoritmo MOBJ. O MOBJ foi pioneiro no treinamento multiobjetivo de RNAs tratando os objetivos de maneira explícita e desacoplada. A abordagem utilizada até então para resolução do problema de aprendizado multiobjetivo adicionando-se a função J_2 ao treinamento era o conhecido método de regularização Weight Decay (MLP-WD) (Hinton, 1989). O MOBJ utiliza o método ϵ -restrito (Chankong & Haimes, 1983), de forma que o problema multiobjetivo é descrito por vários problemas de otimização mono-objetivo em que a função J_1 é otimizada tendo $J_2 \leq \epsilon$ como restrição, conforme mostrado pela Equação (2.11).

$$\begin{aligned} \min_{\mathbf{w}} \quad & J_1 \\ \text{s.a :} \quad & J_2 \leq \epsilon \end{aligned} \tag{2.11}$$

Para resolução de cada problema de otimização mono-objetivo restrito, o MOBJ utiliza o algoritmo Elipsoidal (Shor, 1977). Este algoritmo é um método de exclusão de semi-espacos que possui a capacidade de tratar problemas de otimização não-linear restritos. O método tem como finalidade circundar a solução ótima do problema com um elipsoide. A cada iteração do algoritmo são feitos cortes que excluem as regiões onde não está contida a solução ótima e são gerados novos elipsoides cada vez menores até que se consiga um elipsoide degenerado sobre o ponto ótimo. A cada execução do algoritmo obtem-se uma solução para determinado valor de norma, o que consistirá ao final em um conjunto de soluções onde em cada solução não há mais como melhorar um dos objetivos sem degradar o outro. Este conjunto de soluções consiste numa estimativa $\mathcal{PO}$ do conjunto Pareto-Ótimo, cuja ilustração pode ser visualizada pela Figura 2.1.

Após a geração de $\mathcal{PO}$ é necessário que se selecione uma solução final, sendo que na ausência de qualquer informação *a priori* ou entidade especialista no problema, qualquer solução é uma potencial candidata. Em (Teixeira, 2001), é utilizado o decisor baseado no menor erro para o conjunto de validação. A etapa de decisão utilizada em Costa *et al.* (2007) e também neste trabalho tem como base este mesmo decisor, de maneira a facilitar a comparação com o método clássico (Teixeira, 2001). A seção 2.4.3 descreve o processo de escolha de uma

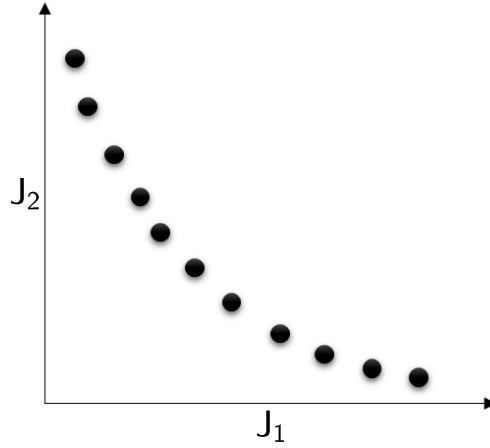


Figura 2.1: Representação de uma estimativa do Conjunto Pareto-Ótimo

solução final para o problema de aprendizado multiobjetivo, além de mostrar o decisor utilizado neste trabalho e alguns outros que estão entre os principais decisores utilizados na literatura neste tipo de problema.

2.4.3 Decisão Multi-objetivo em Aprendizado de Máquina

Considerando-se problemas de otimização multiobjetivo, existe a necessidade de um processo adicional após a geração de um conjunto de soluções eficientes, onde uma solução final deve ser selecionada com base em algum critério, para ser implementada. A solução final deve ter como base as preferências de um Decisor, que pode variar de acordo com o problema. Existem 3 três formas principais para identificação de preferências referentes à escolha da solução final: *a priori*, interativa ou progressiva e *a posteriori* (Parreiras & Vasconcelos, 2005). A abordagem *a posteriori* fornece facilidades quanto à inclusão ou modificação dos critérios de preferência, sem a necessidade de alterações no algoritmo de otimização utilizado para estimação do conjunto de soluções. Entretanto, a abordagem possui custo elevado pois o decisor analisa todas as alternativas (soluções) disponíveis com a finalidade de definir a solução preferencial, diferentemente das abordagens *a priori* e interativa, que avaliam uma menor quantidade de soluções dado que são acopladas ao método de otimização multiobjetivo (Parreiras, 2006). Como este trabalho foca no problema de treinamento multiobjetivo e estimação de um

conjunto de soluções Pareto, apenas um decisor *a posteriori* será abordado.

Considerando-se todas as soluções que compõem o conjunto Pareto estimado $\mathcal{PO}$ pelo treinamento multiobjetivo, o problema de tomada de decisão para seleção de uma destas soluções é definido na Equação (2.12) (Medeiros *et al.*, 2016).

$$\mathbf{w}^* = \arg \min_{\mathbf{w} \in \mathcal{W}^*} f_U \quad (2.12)$$

onde f_U é uma função de utilidade que classifica as soluções do conjunto Pareto estimado com base em critérios de preferência previamente definidos. Considerando-se o problema de aprendizado indutivo, o objetivo de f_U é encontrar a função $f(\mathbf{x}, \mathbf{w})$, dentre aquelas geradas pelo método multiobjetivo, que melhor aproxima a função geradora $f_g(\mathbf{x})$.

2.4.3.1 Métodos para Escolha da Solução

O trabalho desenvolvido em (Teixeira *et al.*, 2000) utilizou um processo decisório cuja solução selecionada é aquela com o menor valor de erro em um conjunto de dados de validação. Esta abordagem para o treinamento multiobjetivo de RNAs é considerada clássica, devido ao pioneirismo do trabalho em representar de maneira explícita o problema multiobjetivo em RNAs. O decisor baseado em erro de validação foi utilizado em vários trabalhos subsequentes (Costa *et al.*, 2002b, 2007; Rocha *et al.*, 2011, 2013, 2015), influenciados pela abordagem clássica.

Este decisor caracteriza-se pela utilização de uma re-amostragem de dados, retirada dos dados disponíveis para o aprendizado, compondo o chamado conjunto de validação. O processo decisório fundamenta-se na geração de uma curva discreta do erro de validação (vide Figura 2.2) de maneira que a solução em $\mathcal{PO}$ com o menor erro apresentado nesta curva é aquela que deve ser implementada.

Conforme destacado em (Teixeira *et al.*, 2002) este decisor é consistente com princípio de minimização do risco estrutural *SRM*, desde que o conjunto Pareto estimado mapeie de maneira representativa a região contendo soluções sub e superdimensionadas aproximando o problema em questão. Porém, a eficiência deste decisor depende da quantidade de dados disponíveis para o problema, obtendo piores resultados em conjuntos reduzidos, devido à complexidade da tarefa de

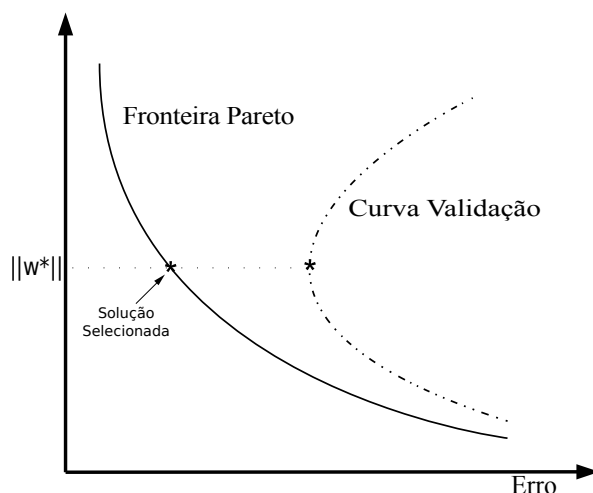


Figura 2.2: Pareto Discreto e Curva de Validação

aprendizado em cenários como este. Além disso, devido à necessidade de se retirar parte dos dados disponíveis, a utilização do decisor limita o poder do algoritmo de aprendizado, que terá menos dados disponíveis para o treinamento.

Alguns decisores alternativos têm sido utilizados com o objetivo de eliminar a necessidade de se utilizar um subconjunto de dados no processo de decisão. Uma destas alternativas é o decisor Curva L, cuja utilização prática é sugerida inicialmente em (Lawson & Hanson, 1974). Uma versão atualizada deste decisor é utilizada no problema de aprendizado desenvolvido em (Tavares, 2015), gerando resultados compatíveis com outros da literatura. Entretanto, este decisor é geralmente utilizado em formulações ponderadas do problema de aprendizado multiobjetivo, tornando-se melhor aplicável a problemas convexos, o que não é garantido nos casos gerais do problema de aprendizado onde os objetivos são tratados de maneira explícita e desacoplada (Medeiros *et al.*, 2009, 2016). Outra alternativa trata-se do método desenvolvido em (Medeiros *et al.*, 2007) que tem como base a informação relativa à autocorrelação dos resíduos das soluções. A funcionalidade do método é demonstrada em (Barroso *et al.*, 2007) que aplica o método a problemas na área de identificação de sistemas. Contudo, a formulação deste decisor concentra-se apenas em problemas com saída contínua, *i.e.* problemas de regressão. O decisor descrito em (Torres *et al.*, 2012) apresenta uma

abordagem baseada na geometria da margem de separação entre classes através da construção do grafo de Gabriel (Zhang & King, 2002). A solução selecionada é aquela situada à menor distância de um separador alvo obtido através do grafo. Este método apresentou robustez em bases de dados de *benchmark*, entretanto limita-se à problemas binários de classificação. No trabalho apresentado em (Medeiros *et al.*, 2009), o processo de tomada de decisão tem como base a informação *a priori* acerca de imprecisões nos dados coletados, considerando-se uma distribuição igualitária binomial para as classes. Em (Medeiros *et al.*, 2016), a abordagem é estendida, descrevendo uma forma genérica de se utilizar a informação *a priori* considerando-se qualquer distribuição. Embora demonstre eficiência, esta abordagem depende da disponibilidade de informação *a priori* sobre os dados e aplica-se apenas a problemas de classificação.

Considerando-se a formulação de uso geral do decisor baseado no menor erro de validação, este será utilizado para seleção da solução final em todos os métodos de treinamento multiobjetivo utilizados neste trabalho. Um dos fatores que contribuiu para a escolha deste decisor se deve às limitações de outros métodos que são aplicáveis a tipos específicos de tarefas de aprendizado, enquanto o decisor por erro de validação é robusto neste quesito, mesmo perdendo eficiência em bases de dados com tamanho reduzido. Outro aspecto considerado foi a ampla utilização deste decisor em trabalhos anteriores, o que facilita comparações entre soluções finais obtidas em outras abordagens.

2.5 Conclusão do Capítulo

Este capítulo apresentou conceitos fundamentais que suportam o desenvolvimento deste trabalho. Métodos clássicos para o treinamento de redes MLP foram brevemente descritos e o problema acerca da capacidade de generalização foi elencado. A Teoria do aprendizado estatístico foi apresentada junto ao dilema entre o viés e a variância, visando o entendimento dos fundamentos que cercam o aprendizado multiobjetivo. Por fim, a formulação clássica para o treinamento multiobjetivo de redes foi apresentada, passando por conceitos básicos de otimização multiobjetivo até o processo de tomada de decisão para seleção da solução final.

Capítulo 3

Formulação Esférica e Métodos Propostos

3.1 Representação Esférica dos Pesos

Esta seção apresenta a principal contribuição deste trabalho, um novo algoritmo para o treinamento multiobjetivo de redes MLP. A abordagem proposta consiste no desenvolvimento de um esquema de treinamento de redes neurais em que os pesos da rede são representados em forma de ângulos, utilizando as funções trigonométricas seno e cosseno. É importante ressaltar que a estrutura da rede não está sendo alterada, *i.e.*, o que está sendo apresentado é uma nova representação dos pesos para fins de treinamento de uma rede com estrutura previamente definida. A abordagem reduz o espaço de soluções a um sub-espaço particular em que a norma dos pesos tem valor garantidamente fixado durante o treinamento. Cada peso da rede define um eixo no espaço esférico n -dimensional de forma que, para um determinado valor de norma, pode-se gerar os valores dos pesos em função dos ângulos existentes entre o vetor de pesos com tamanho r e os eixos de um elipse (vide exemplo 2D, Figura 3.1), podendo-se assim garantir que variando-se os valores dos ângulos obtêm-se novos valores para os pesos sem que a solução escape da superfície da norma. Esta formulação elimina a necessidade de se utilizar funções de penalidade ou qualquer algoritmo de otimização multiobjetivo. A solução do problema de aprendizado multiobjetivo é alcançada através da resolução de vários subproblemas mono-objetivo irrestritos, permitindo assim

a utilização de algoritmos clássicos para otimização não linear, como um simples *Gradiente Descendente* ou o *Levenberg-Marquardt*, para estimação do conjunto Pareto do problema.

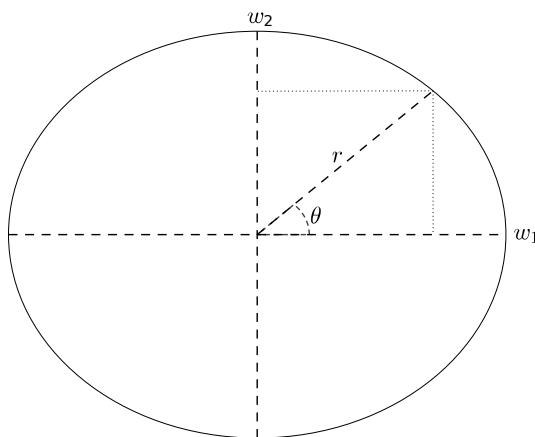


Figura 3.1: Disposição dos pesos na elipse definida pela função norma no espaço 2D.

Ao longo deste trabalho o termo "Representação Esférica" denotará a representação proposta, embora haja grande variação quanto ao termo ideal para nomear a extensão desta forma de coordenadas considerando-se um espaço n-dimensional (Nguyen, 2014).

A representação esférica é ilustrada através de um exemplo simples e, logo em seguida, estendida para RNAs do tipo Multi-layer Perceptron. Pode-se ainda estender esta representação a outros modelos não lineares, mas isso não faz parte do escopo deste trabalho.

3.1.1 Pesos Esféricos Aplicados a um Perceptron Não-Linear

Como exemplo, será utilizado um único neurônio com função de ativação do tipo tangente hiperbólica, considerando-se uma única entrada e consequentemente 2 pesos $\mathbf{w} = [w_1, w_2]$, onde w_1 é o peso aplicado à entrada enquanto w_2 é o termo bias (vide, Figura 3.2). Os dados de treinamento são compostos por dois padrões entrada-saída $\{(-3, -0.4), (2, -0.9)\}$ e a saída do neurônio é dada por

$y(x_i) = \tanh(w_1 \cdot x_i + w_2)$. Para este problema, a Figura 3.3 apresenta o gráfico do contorno da superfície de erro. Esferas, ou hiperesferas, neste problema cor-

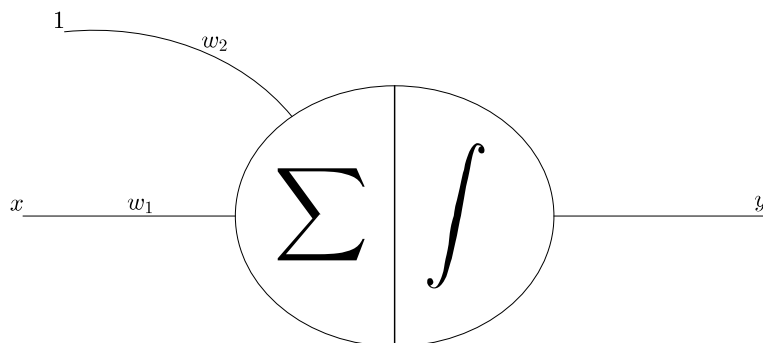


Figura 3.2: Neurônio com uma entrada.

respondem a círculos no plano $w_1 \times w_2$ projetadas à superfície apresentada na Figura 3.3. Cada isolinha de contorno corresponde às soluções com mesmo valor de norma, $w_1^2 + w_2^2 = r_j^2$, destacando o espaço de busca limitado por valores fixos de norma. Os pontos na figura representam as soluções de mínimo erro e, portanto, correspondem às soluções alvo do conjunto Pareto estimado.

A equação da restrição de norma pode também ser representada utilizando-se o raio r_j e um novo parâmetro, o ângulo θ , de maneira que $w_1 = r_j \cdot \sin \theta$ e $w_2 = r_j \cdot \cos \theta$. Portanto, o espaço original de parâmetros formado por pesos, é agora representado por um novo vetor de parâmetros, $[r_j, \theta]$. Porém, nesta nova representação, a norma, ou o raio r_j é fixo e, portanto, o procedimento de treinamento visa encontrar o valor de θ que minimiza a função de erro, onde $0 \leq \theta < 2\pi$.

A Figura 3.4 mostra as curvas de erro do problema apresentado na Figura 3.3 como uma função do ângulo θ , para diferentes valores de r_j . Os pontos na figura representam soluções de mínimo erro ou, em outras palavras, as soluções do conjunto Pareto estimado para cada valor de r_j . Pode ser observado que, para pequenos valores de r_j , as curvas são aproximadamente quadráticas com um único mínimo, contudo a região contendo este mínimo apresenta-se maior quando comparada a regiões definidas por maiores valores de r_j . Quanto maior o valor de r_j maior é o nível de não-linearidade das curvas, e menores são os valores de erro que podem ser atingidos. Essas observações são consistentes com

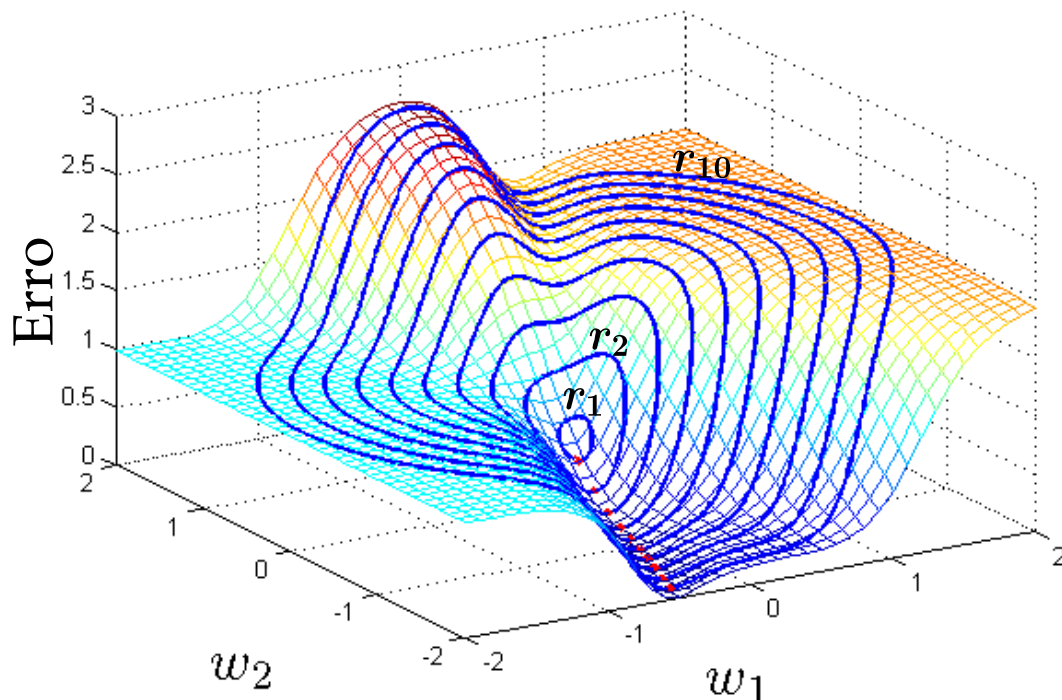


Figura 3.3: Contorno da superfície de erro

o princípio de que a norma dos pesos está associada à complexidade da função de erro (Bartlet, 1997). De forma resumida, para $j = 1, 2, \dots, n_{sol}$ são obtidas n_{sol} soluções que representam um conjunto Pareto estimado, através da resolução de n_{sol} problemas de otimização com r_j fixo em cada um deles.

No trabalho apresentado em (Matthysen & Engelbrecht, 2011) os autores apresentam conclusões similares às deste trabalho com relação à alteração sofrida por uma superfície no espaço cartesiano quando esta é transformada num espaço esférico. O trabalho descreve a utilização de um algoritmo evolutivo de otimização para resolução de problemas clássicos de *benchmark* utilizando coordenadas esféricas n-dimensionais para representar os parâmetros. Através de exemplos gráficos os autores ilustram o efeito de compressão sofrido pelo espaço original de parâmetros ao passar para o espaço esférico, mostrando que grandes regiões contendo um ótimo local são geradas para valores pequenos de raio enquanto para grandes valores de raio as regiões de ótimo decrescem em tamanho. Entretanto, este trabalho aplica o uso de coordenadas esféricas no âmbito do treinamento

de redes neurais e, diferentemente do problema apresentado em (Matthysen & Engelbrecht, 2011), onde o raio também é uma variável de otimização, neste trabalho o problema é discretizado sobre a superfície de erro quadrático, gerando-se um problema de otimização para cada valor de raio.

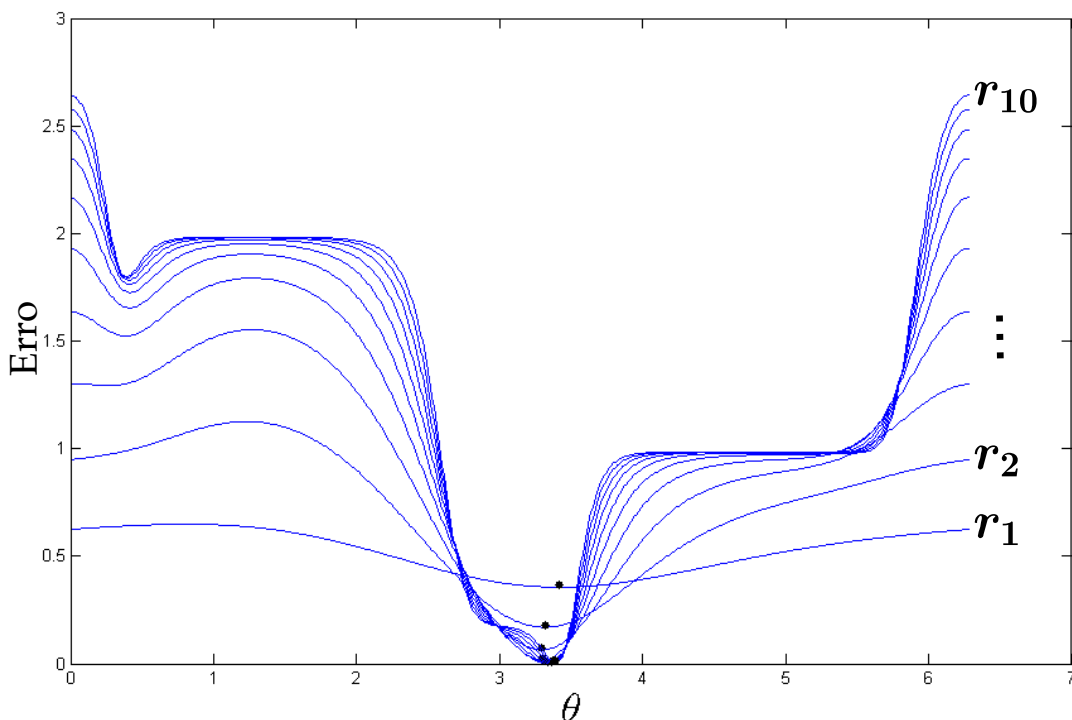


Figura 3.4: Curvas de erro em função do ângulo θ em radianos

Vale ressaltar que com esta representação a dimensão do novo espaço de parâmetros é reduzida em uma unidade. Ademais, para pequenos valores de raio r_j , as observações realizadas indicam que a superfície de erro possui um único mínimo situado numa região com maior nível de suavidade, de forma que este mínimo pode ser alcançado utilizando-se algoritmos básicos de otimização. Esta característica é particularmente útil para geração de soluções com maiores raios. Tomando estas considerações como base, a estratégia usada no processo de minimização é iniciada com um pequeno valor de raio, por exemplo, o valor médio de y . Em seguida o raio é aumentado progressivamente, utilizando pequenos incrementos, visando encontrar uma nova estimativa de solução para o conjunto Pareto. Para cada valor de r_j um simples procedimento de minimização

é aplicado. O processo global pode então ser simplificado, de maneira que uma solução de mínimo erro obtida para um valor particular de r_j , pode ser utilizada como solução inicial para o problema de otimização com raio r_{j+1} , consistindo no procedimento conhecido como *warm start* (Friedman *et al.*, 2010). A utilização deste procedimento visa aproveitar informações acerca da direção obtida por uma solução anterior de maneira que as soluções subsequentes estejam situadas nas proximidades da bacia de atração do subproblema de minimização em questão, visando aumentar a velocidade de convergência.

Para resumir a utilização de coordenadas esféricas aplicadas a um único neurônio, consideremos agora um modelo neural com 4 pesos, $\mathbf{w} = [w_1, w_2, w_3, w_4]$. A representação esférica dos pesos é dada pela Equação (3.1).

$$\begin{aligned} w_1 &= r \sin \theta_1 \\ w_2 &= r \cos \theta_1 \sin \theta_2 \\ w_3 &= r \cos \theta_1 \cos \theta_2 \sin \theta_3 \\ w_4 &= r \cos \theta_1 \cos \theta_2 \cos \theta_3 \end{aligned} \tag{3.1}$$

onde $0 \leq \theta_3 < 2\pi$ e $-\frac{\pi}{2} \leq (\theta_1, \theta_2) < +\frac{\pi}{2}$. Como o raio r é fixo, o novo espaço de parâmetros é representado pelo vetor $\Theta = [\theta_1, \theta_2, \theta_3]$. Vale ressaltar que os ângulos podem ser definidos recursivamente em função dos pesos, como mostrado pela Equação (3.2).

$$\begin{aligned} \theta_3 &= \arctan \left(\frac{w_3}{w_4} \right) \\ \theta_2 &= \arctan \left(\frac{w_2}{w_3} \cdot \sin \theta_3 \right) \\ \theta_1 &= \arctan \left(\frac{w_1}{w_2} \cdot \sin \theta_2 \right) \end{aligned} \tag{3.2}$$

3.1.2 Representação Esférica de Pesos para a Rede MLP

A representação esférica de pesos pode ser facilmente estendida para redes *Multi-Layer Perceptron*. Para demonstrar a representação da rede MLP, consideraremos a estrutura com uma camada escondida e um neurônio linear de saída. A Equação (3.3) representa a saída da rede, dado um vetor de entrada arbitrário $\mathbf{x}$ de dimensão N , $\mathbf{x}^T = [x_1, \dots, x_N]$.

3.1 Representação Esférica dos Pesos

$$y_{(\mathbf{x})} = \sum_{i=1}^{H+1} \varphi_i \cdot f_i^h \left[\sum_{j=1}^{N+1} (\psi_{ji} \cdot x_j) \right] \quad (3.3)$$

onde H é a quantidade de neurônios da camada escondida, ψ_{ji} representa um peso da camada escondida, φ_i é um peso da camada de saída e f^h é a função de ativação dos neurônios da camada escondida. Os componentes *bias* foram acrescentados aos vetores de entrada e de pesos de forma que: $x_{N+1} = 1$, $f_{H+1}^h = 1$.

Podemos definir o vetor $\mathbf{w}$ como o vetor que contém todos os pesos das camadas escondida e de saída, $\mathbf{w}^T = [\varphi_1, \dots, \varphi_{H+1}, \psi_{1,1}, \dots, \psi_{N+1,H+1}]$, de forma que cada elemento do vetor $\mathbf{w}$ é representado por w_i . Portanto, a norma Euclidiana do vetor de pesos é definida como $|\mathbf{w}|^2 = \sum_i w_i^2$. Utilizando pesos esféricos podemos nos referir à norma do vetor de pesos através do raio r , onde $r^2 = |\mathbf{w}|^2$.

Expandindo-se esta formulação a um problema n -dimensional tem-se a estrutura de uma n -sphere (esfera n -dimensional), onde o vetor de pesos $\mathbf{w}$, de tamanho n_w , pode ser escrito por coordenadas hiperesféricas, como formulado em (Kendall, 1961). Esta formulação é consistente e se reduz às coordenadas polares para $n_w = 2$ e coordenadas esféricas para $n_w = 3$ (Nguyen, 2014). Desta forma o vetor $\mathbf{w}$ pode ser escrito em função de $n_w - 1$ ângulos, definindo o vetor $\Theta^T = [\theta_1, \dots, \theta_{n_w-1}]$ e raio r , como mostrado pela Equação (3.4).

$$w_i = \begin{cases} r \sin(\theta_1) & i = 1. \\ r \prod_{k=1}^{i-1} \cos(\theta_k) \sin(\theta_i) & 2 \leq i \leq n_w - 1. \\ r \prod_{k=1}^{i-1} \cos(\theta_k) & i = n_w. \end{cases} \quad (3.4)$$

A partir da Equação (3.4) a estimativa dos ângulos $\theta_i \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ para $i = 1, 2, \dots, n_w - 2$ e $0 \leq \theta_{n_w-1} < 2\pi$, pode ser obtida através dos valores dos pesos, como mostrado pela Equação (3.5).

$$\theta_i = \begin{cases} \tan^{-1} \left(\frac{w_i}{w_{i+1}} \sin(\theta_{i+1}) \right) & 1 \leq i < n_w - 1. \\ \tan^{-1} \left(\frac{w_{n_w-1}}{w_{n_w}} \right) & i = n_w - 1. \end{cases} \quad (3.5)$$

A utilização de pesos esféricos cria um novo conjunto de parâmetros Θ de dimensão $n_w - 1$ que pode ser otimizado com o objetivo de se encontrar os pesos ótimos da rede MLP, com mínimo erro. Neste novo espaço de parâmetros, a norma dos pesos, ou raio, tem valor fixo. Portanto, a minimização da função de erro, com

3.1 Representação Esférica dos Pesos

relação ao vetor de ângulos Θ , pode ser descrito com um problema de otimização mono-objetivo sem restrições. Como resultado, os vetor final de pesos tem norma fixa que, de fato, representa uma solução do problema de otimização restrito, no espaço original ($\mathbf{w}$). Segundo esta representação, o novo problema de treinamento da rede passa a ser o de encontrar os valores dos ângulos que minimizam o erro caracterizado pela função J_1 definida na Equação (2.9), definindo o novo problema de otimização mostrado na Equação (3.6).

$$\arg \min_{\theta_1 \dots \theta_{n_w-1}} J_1 \quad (3.6)$$

A Figura 3.5 ilustra os múltiplos problemas mono-objetivo a serem resolvidos com o objetivo de se estimar o conjunto Pareto utilizando-se a formulação esférica. Uma outra abordagem para a aproximação do conjunto Pareto através do controle da trajetória no espaço de objetivos, utilizando Controle por Modos Deslizantes e otimização com o algoritmo *Levenberg-Marquadt*, foi descrita em trabalho anterior (Costa *et al.*, 2007). Entretanto os resultados obtidos neste trabalho não demonstram que ele supera a abordagem clássica (Teixeira, 2001) quanto às estimativas do conjunto Pareto.

Conforme mencionado no início desta seção a utilização da formulação baseada em coordenadas esféricas torna o problema de otimização multiobjetivo da rede irrestrito, dado que o controle de complexidade está acoplado à função objetivo. A partir da transformação gerada, podem ser utilizados diversos algoritmos para otimização não linear sem a necessidade do tratamento de restrições, onde cada subproblema de otimização é definido para um valor particular de norma. Com o objetivo de gerar uma estimativa do conjunto Pareto, foram desenvolvidos neste trabalho alguns algoritmos para o treinamento multiobjetivo de redes MLP utilizando a representação proposta, conforme será detalhado na seção a seguir.

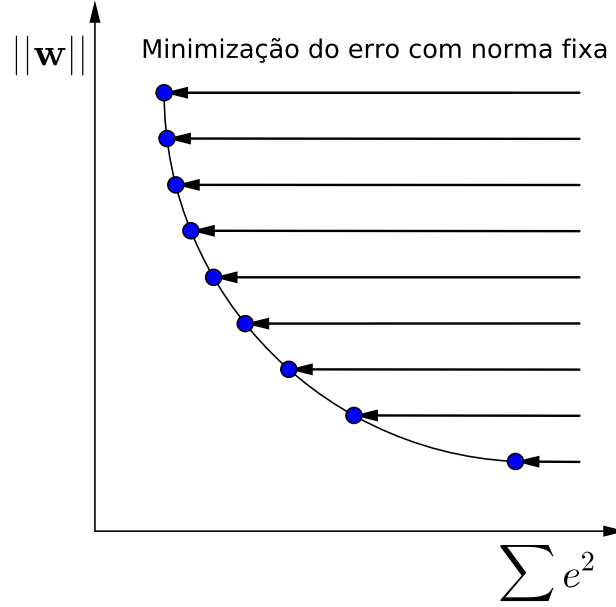


Figura 3.5: Aproximação da fronteira Pareto através da minimização de erro com norma fixa.

3.2 Aprendizado Multi-objetivo de Redes com Pesos Esféricos

Nesta seção será abordado o processo de treinamento multiobjetivo de uma rede MLP com pesos representados pela formulação esférica. O *framework* geral que representa o aprendizado multiobjetivo será apresentado enquanto os métodos mono-objetivo para o treinamento de cada subproblema serão apresentados nas subseções seguintes.

O **Algoritmo 1** descreve o processo de treinamento multiobjetivo da rede, considerando-se a necessidade de se variar o valor de r para geração de cada solução que compõe a estimativa do conjunto Pareto. A linha **7** no algoritmo representa o método de otimização utilizado para minimização do problema definido por um valor particular de norma r .

As subseções seguintes nesta seção abordarão alguns algoritmos possíveis para implementação com pesos esféricos, sendo que o método *Gradiente Descendente* será utilizado como base para as principais definições enquanto algoritmos mais

3.2 Aprendizado Multi-objetivo de Redes com Pesos Esféricos

eficientes serão implementados e avaliados. Maiores detalhes acerca da implementação e avaliação do algoritmo *Gradiente Descendente* com pesos esféricos podem ser encontrados em (Rocha *et al.*, 2015).

Algoritmo 1: TREINAMENTO MULTIOBJETIVO COM PESOS ESFÉRICOS

Entrada: $n_{sol}, \delta r$

Saída: Estimativa do conjunto Pareto

```
1 início
2   Inicializar Variáveis
   /* (Atribuir valor aleatório e pequeno a cada peso  $w_i$ .) */
3    $w_i \leftarrow Normal(\mu = 0, \sigma^2 = 1)/1000$ 
   /* Atribuir à norma alvo inicial um valor próximo a  $\bar{y}$  */
4    $r_1 \leftarrow \bar{y} + \delta r$ 
5   Converter o vetor inicial de pesos  $\mathbf{w}$  em  $\Theta$  utilizando a Equação (3.5)
6   para  $j = 1, 2, \dots, n_{sol}$  faça
7     Aplicar algoritmo de otimização até a convergência, dado  $[\Theta, r_j]$  .
8      $\mathcal{PO}_j \leftarrow$  solução final do subproblema, dado  $[\Theta, r_j]$  // armazena
       solução Pareto
9      $r_{j+1} \leftarrow r_j + \delta r$ 
10  fim
11  Selecionar solução final por erro de validação
12 fim
13 retorna  $[\mathcal{PO}, \text{Solução selecionada}]$ 
```

3.2.1 O Algoritmo *Gradiente Descendente* Utilizando Pesos Esféricos (SGD)

O algoritmo *Back-Propagation* clássico (Rumelhart & McClelland, 1986) visa a minimização de uma função de erro, geralmente o erro médio quadrático, utilizando o método *Gradiente Descendente*, onde a atualização do vetor de pesos ocorre na direção oposta do gradiente no ponto $\mathbf{w}_0$, onde $\mathbf{w}_0$ é o vetor de pesos inicial. A Equação para atualização dos pesos é mostrada na Equação (3.7).

3.2 Aprendizado Multi-objetivo de Redes com Pesos Esféricos

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \alpha \cdot \left. \frac{\partial \sum e_i^2}{\partial \mathbf{w}} \right|_{\mathbf{w}_k} \quad (3.7)$$

onde $\sum e_i^2$ é a função de erro e α é a taxa de aprendizado. Ao transformar o espaço de pesos no espaço esférico o objetivo do algoritmo passa a ser o de otimizar o vetor de ângulos da rede, o que pode ser obtido acrescentando-se a derivada dos pesos em relação aos ângulos através do emprego da regra da cadeia em relação ao gradiente, como definido na Equação (3.8).

$$\frac{\partial \sum e_i^2}{\partial \theta} = \frac{\partial \sum e_i^2}{\partial \mathbf{w}} \times \frac{\partial \mathbf{w}}{\partial \theta} \quad (3.8)$$

Para definir a regra de atualização do *Back-Propagation* no espaço esférico será utilizada uma notação matricial, definindo-se a matriz $\mathbf{T} = \frac{\partial \mathbf{w}}{\partial \Theta}$, definida na Equação (3.9). A expansão genérica da matriz $\mathbf{T}$ é obtida pela Equação (3.10), onde cada derivada parcial que define os elementos de $\mathbf{T}$ são mostrados.

$$\mathbf{T} = \frac{\partial \mathbf{w}}{\partial \theta} = \begin{bmatrix} \frac{\partial w_1}{\partial \theta_1} & \cdots & \frac{\partial w_{n_w}}{\partial \theta_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial w_1}{\partial \theta_{n_w-1}} & \cdots & \frac{\partial w_{n_w}}{\partial \theta_{n_w-1}} \end{bmatrix} \quad (3.9)$$

$$\mathbf{T} = r \cdot \begin{bmatrix} \cos \theta_1 & \cdots & -\sin \theta_1 \cdots \cos \theta_{n_w-2} \sin \theta_{n_w-1} & -\sin \theta_1 \cos \theta_2 \cdots \cos \theta_{n_w-1} \\ 0 & \cdots & -\cos \theta_1 \sin \theta_2 \cdots \cos \theta_{n_w-2} \sin \theta_{n_w-1} & -\cos \theta_1 \sin \theta_2 \cdots \cos \theta_{n_w-1} \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & \cos \theta_1 \cos \theta_2 \cdots \cos \theta_{n_w-1} & -\cos \theta_1 \cos \theta_2 \cdots \sin \theta_{n_w-1} \end{bmatrix} \quad (3.10)$$

Uma maneira simples e eficiente de se calcular os elementos da matriz $\mathbf{T}$ através do vetor de pesos $\mathbf{w}$ é mostrada na Equação (3.11), que é obtida utilizando-se a Equação (3.4) em (3.10).

$$[\mathbf{T}]_{ij} = r \cdot \begin{cases} 0, & i = 2, \dots, n_w - 1; j = 1, \dots, i - 1 \\ \prod_{k=1}^i \cos \theta_k & i = j \\ -\hat{w}_j \tan \theta_i & i = 1, \dots, n_w - 1; j = i + 1, \dots, n_w \\ -\hat{w}_{n_w-1} & i = n_w - 1; j = n_w \end{cases} \quad (3.11)$$

onde $\hat{\mathbf{w}} = \mathbf{w}/r$. Portanto, podemos definir a equação de ajuste do *Back-Propagation* adaptado à formulação com pesos esféricos, como:

3.2 Aprendizado Multi-objetivo de Redes com Pesos Esféricos

$$\Theta_{k+1} = \Theta_k - \alpha \cdot \mathbf{T} \times \nabla f(\mathbf{w}) \quad (3.12)$$

onde $\nabla f(\mathbf{w})$ é o gradiente da função de erro. Comparando as Equações (3.7) e (3.12) pode-se observar que a diferença entre o termo de ajuste utilizado no *Gradiente Descendente* convencional e aquele utilizado no *Gradiente Descendente* com pesos esféricos resume-se na inclusão da matriz $\mathbf{T}$. Esta característica torna possível a utilização de outros algoritmos de otimização para treinamento da rede com simples modificações nas regras de ajuste de parâmetros, incluindo-se a matriz $\mathbf{T}$. Alguns destes algoritmos passíveis de utilização serão vistos ao longo desta seção.

Além dos benefícios obtidos com relação à transformação do problema de maneira que não existam restrições no procedimento de otimização, devido à otimização dos parâmetros com valor fixo de norma, um dos principais atrativos do método é a simplicidade de implementação dado que as derivadas acrescentadas são facilmente calculadas para a formulação esférica do problema, consistindo basicamente na utilização das funções trigonométricas seno e cosseno (Vide, Equação (3.10)), que têm derivadas simples e conhecidas.

3.2.2 O Algoritmo *Levenberg-Marquardt* Esférico (SLM)

Diferentemente do *Back-Propagation* que é um algoritmo baseado no gradiente descendente o algoritmo *Levenberg-Marquardt* (Marquardt, 1963) pode ser considerado uma aproximação do método de Newton (Fun & Hagan, 1996). Este algoritmo foi adaptado ao treinamento de redes neurais (Hagan & Menhaj, 1994), tornando-se um dos mais eficientes dentre os algoritmos para o treinamento de rede, sendo que a velocidade de convergência é um dos seus principais atrativos. A dedução do algoritmo *Levenberg-Marquardt* pode ser realizada tomando-se como base a utilização do algoritmo de Newton com o objetivo de minimizar uma função $f(\mathbf{w})$. O ajuste necessário poderia ser definido como:

$$\Delta_{\mathbf{w}} = - [\nabla^2 f(\mathbf{w})]^{-1} \nabla f(\mathbf{w}) \quad (3.13)$$

onde $\nabla^2 f(\mathbf{w})$ é a matriz Hessiana de $f(\mathbf{w})$ e $\nabla f(\mathbf{w})$ é o gradiente. Definindo-se $f(\mathbf{w})$ como a soma do erro quadrático ($e(\mathbf{w})$), função de desempenho utilizada

3.2 Aprendizado Multi-objetivo de Redes com Pesos Esféricos

no treinamento de RNAs, pode-se obter uma aproximação para a Hessiana e o gradiente como pode ser verificado em (Hagan & Menhaj, 1994), onde tem-se:

$$\nabla f(\mathbf{w}) = J^T(\mathbf{w})e(\mathbf{w}) \quad (3.14)$$

$$\nabla^2 f(\mathbf{w}) = J^T(\mathbf{w})J(\mathbf{w}) + S(\mathbf{w}) \quad (3.15)$$

onde $S(\mathbf{w}) = \sum_{i=1}^N e_i(\mathbf{w})\nabla^2 e_i(\mathbf{w})$ e $J(\mathbf{w})$ é a matriz Jacobiana definida em termos de n pesos e N padrões, como pode ser visualizada:

$$J(\mathbf{w}) = \begin{bmatrix} \frac{\partial e_1(\mathbf{w})}{\partial w_1} & \frac{\partial e_1(\mathbf{w})}{\partial w_2} & \cdots & \frac{\partial e_1(\mathbf{w})}{\partial w_n} \\ \frac{\partial e_2(\mathbf{w})}{\partial w_1} & \frac{\partial e_2(\mathbf{w})}{\partial w_2} & \cdots & \frac{\partial e_2(\mathbf{w})}{\partial w_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_N(\mathbf{w})}{\partial w_1} & \frac{\partial e_N(\mathbf{w})}{\partial w_2} & \cdots & \frac{\partial e_N(\mathbf{w})}{\partial w_n} \end{bmatrix} \quad (3.16)$$

De acordo com o método Gauss-Newton (Björck, 1996) pode-se assumir $S(\mathbf{w}) \approx 0$ podendo-se retirar este termo, gerando a Equação (3.17).

$$\Delta_{\mathbf{w}} = [J^T(\mathbf{w})J(\mathbf{w})]^{-1} J^T(\mathbf{w})e(\mathbf{w}) \quad (3.17)$$

A contribuição do algoritmo *Levenberg-Marquardt* ao método Gauss-Newton consiste na adição de um parâmetro μ multiplicado por uma matriz identidade. A Equação (3.17) adaptada pelo *Levenberg-Marquardt* é definida na Equação (3.18).

$$\Delta_{\mathbf{w}} = [J^T(\mathbf{w})J(\mathbf{w}) + \mu I]^{-1} J^T(\mathbf{w})e(\mathbf{w}) \quad (3.18)$$

onde I é a matriz identidade e μ é um parâmetro positivo que quando tem valores muito altos se aproxima do gradiente descendente e com valores baixos se aproxima do método Gauss-Newton. O algoritmo deve começar com valor inicial para μ que será multiplicado por um fator β quando se obter um aumento do erro numa determinada iteração e, será dividido por β quando o erro diminuir.

A matriz Jacobiana definida na Equação (3.16) representa o principal recurso do algoritmo com relação à obtenção de uma direção de busca eficiente. Considerando-se que esta matriz consiste nas derivadas primeiras do erro em

3.2 Aprendizado Multi-objetivo de Redes com Pesos Esféricos

relação aos pesos para cada padrão de treinamento, pode-se reescrevê-la de maneira que se obtenha uma direção de busca em relação ao vetor de ângulos de maneira semelhante ao processo utilizado para modificação do *Back-Propagation* clássico utilizando o gradiente descendente. Utilizando-se a regra da cadeia pode-se reescrever a matriz Jacobiana de maneira que possa ser definida como:

$$J(\Theta) = \begin{bmatrix} \frac{\partial e_1(\Theta)}{\partial \theta_1} & \frac{\partial e_1(\Theta)}{\partial \theta_2} & \cdots & \frac{\partial e_1(\Theta)}{\partial \theta_{n-1}} \\ \frac{\partial e_2(\Theta)}{\partial \theta_1} & \frac{\partial e_2(\Theta)}{\partial \theta_2} & \cdots & \frac{\partial e_2(\Theta)}{\partial \theta_{n-1}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_N(\Theta)}{\partial \theta_1} & \frac{\partial e_N(\Theta)}{\partial \theta_2} & \cdots & \frac{\partial e_N(\Theta)}{\partial \theta_{n-1}} \end{bmatrix} \quad (3.19)$$

onde os elementos desta matriz podem ser obtidos de maneira similar à forma vista na Equação (3.8), com o auxílio da Matriz $\mathbf{T}$ definida na Equação (3.9). Desta maneira o ajuste dos ângulos é definido pelas Equações (3.20), (3.21) e (3.22).

$$J(\Theta) = J(\mathbf{w}) \times \mathbf{T}^T \quad (3.20)$$

$$\Delta_{\Theta} = [J^T(\Theta)J(\Theta) + \mu I]^{-1} J^T(\Theta)e(\mathbf{w}) \quad (3.21)$$

A partir da definição da matriz mostrada na Equação (3.19) torna-se possível realizar o treinamento de uma rede MLP com o algoritmo *Levenberg-Marquardt* adaptado à formulação dos pesos no espaço esférico, cujo processo completo resume-se no **Algoritmo 2**.

$$\Theta_{k+1} = \Theta_k + [J^T(\Theta_k)J(\Theta_k) + \mu I]^{-1} J_k^T(\Theta_k)e(\mathbf{w}_k) \quad (3.22)$$

3.2.3 O Algoritmo *BFGS* Esférico (SBFGS)

O algoritmo *Broyden-Fletcher-Goldfarb-Shanno* - *BFGS* (Broyden, 1970), cujo nome se refere às iniciais dos nomes dos autores que propuseram o método de maneira independente, é classificado como um método Quasi-Newton, onde a

3.2 Aprendizado Multi-objetivo de Redes com Pesos Esféricos

informação de segunda ordem acerca da direção de busca é obtida através de uma aproximação da inversa da matriz Hessiana $[\nabla^2 f(\mathbf{w})]^{-1}$ definida pela matriz $\mathbf{B}$. Este método, bem como variações dele, são utilizados na forma original para otimização de problemas não-lineares irrestritos e tem como um dos principais atrativos o baixo custo computacional para obtenção da direção de busca no processo de otimização. O processo de otimização consiste na geração iterativa da aproximação $\mathbf{B}_k$ de forma que utilizando-se informação de primeira ordem a cada iteração k , obtém-se a matriz com informação de segunda ordem atualizada e melhorada. A regra de atualização de $\mathbf{B}_k$ no algoritmo *BFGS* é definida pela Equação (3.23) (Nocedal & Wright, 1999).

$$\mathbf{B}_{k+1} = (I - \rho_k \mathbf{s}_k \mathbf{y}_k^T) \mathbf{B}_k (I - \rho_k \mathbf{y}_k \mathbf{s}_k^T) + \rho_k \mathbf{s}_k \mathbf{s}_k^T \quad (3.23)$$

onde $\mathbf{s}_k$ e $\mathbf{y}_k$ são definidos como:

$$\mathbf{s}_k = \mathbf{w}_{k+1} - \mathbf{w}_k \quad (3.24)$$

$$\mathbf{y}_k = \nabla f(\mathbf{w}_{k+1}) - \nabla f(\mathbf{w}_k) \quad (3.25)$$

e ρ_k é definido da seguinte maneira:

$$\rho_k = \frac{1}{\mathbf{y}_k^T \mathbf{s}_k} \quad (3.26)$$

A regra de ajuste dos pesos utilizando o *BFGS* segue o mesmo formato do ajuste definido para o método de Newton, de forma similar ao algoritmo *Levenberg-Marquardt* mostrado pela Equação (3.13), porém, o termo $[\nabla^2 f(\mathbf{w})]^{-1}$ é substituído por $\mathbf{B}$, gerando o termo de ajuste:

$$\Delta_{\mathbf{w}} = -\mathbf{B} \nabla f(\mathbf{w}) \quad (3.27)$$

Considerando-se que as variáveis no espaço definido pela formulação esférica são os ângulos, o ajuste do algoritmo *BFGS* na versão esférica passa a ser definido como:

$$\Delta_{\Theta} = -\mathbf{B} \nabla f(\Theta) \quad (3.28)$$

3.2 Aprendizado Multi-objetivo de Redes com Pesos Esféricos

onde $\nabla f(\Theta)$ é obtido a partir da regra definida na Equação (3.8), com o auxílio da Matriz $\mathbf{T}$ definida na Equação (3.9), consistindo portanto em:

$$\nabla f(\Theta) = \mathbf{T} \times \nabla f(\mathbf{w}) \quad (3.29)$$

e a matriz $\mathbf{B}$ é definida de forma similar ao *BFGS* no espaço cartesiano, mas com valores de $\mathbf{s}_k$ e $\mathbf{y}_k$ definidos no espaço esférico:

$$\mathbf{s}_k = \Theta_{k+1} - \Theta_k \quad (3.30)$$

$$\mathbf{y}_k = \nabla f(\Theta_{k+1}) - \nabla f(\Theta_k) \quad (3.31)$$

A partir da definição de $\nabla f(\Theta)$ pela Equação (3.29) torna-se possível realizar o treinamento da rede MLP no espaço esférico, onde a forma final para ajuste dos parâmetros pode ser vista na Equação (3.32), considerando-se que a matriz $\mathbf{B}$ é definida pelos novos valores de $\mathbf{s}_k$ e $\mathbf{y}_k$ definidos nas Equações (3.30) e (3.31), respectivamente.

$$\Theta_{k+1} = \Theta_k - \alpha \cdot \mathbf{B}_k \nabla f(\Theta) \quad (3.32)$$

onde o parâmetro α é obtido através do procedimento chamado *backtracking line search* (Nash & Varadhan, 2011) e a matriz I é utilizada como valor inicial para $\mathbf{B}_1$, de forma que o primeiro passo do algoritmo é equivalente ao método *Gradiente Descendente*. A implementação do algoritmo *BFGS* na versão esférica (*SBFGS*) bem como o procedimento *backtracking line search* foram realizados com o auxílio do pacote Optimx (Nash & Varadhan, 2011) do software R (R Core Team, 2016).

O processo completo de treinamento da rede utilizando o *SBFGS* pode ser visto no Algoritmo 3.

3.2 Aprendizado Multi-objetivo de Redes com Pesos Esféricos

Algoritmo 2: ALGORITMO LEVENBERG-MARQUARDT ESFÉRICO

Entrada: $r, max_it, \Theta, \mu, \beta$

Saída: Solução com menor erro ($e(\mathbf{w})$) no espaço restrito por r

```
1 início
2    $k \leftarrow 1$ 
3   enquanto  $((k \leq max\_it) \text{ ou } (e(\mathbf{w}) \text{ estagnado por } 30 \text{ iterações}))$  faça
4       /* Calcular função de erro ( $e(\mathbf{w})$ ) */
5        $MSE \leftarrow e(\mathbf{w})$ 
6       Calcular matriz Jacobiana. Usar (3.16), (3.9) e  $J(\Theta) = J(\mathbf{w}) \times \mathbf{T}^T$ 
7       repita
8           Obter  $\Delta\Theta$  resolvendo-se a Equação (3.21)
9           /* Re-calcular erro ( $e(\mathbf{w})_{novo}$ ), utilizando as Equações
10              (3.22) e (3.4) */
11            $MSE_{novo} \leftarrow e(\mathbf{w})$ 
12           se  $(MSE \leq MSE_{novo})$  então
13                $\mu \leftarrow \mu \times \beta$ 
14           fim
15       até  $(MSE_{novo} \leq MSE)$ ;
16        $\Theta_{k+1} = \Theta_k + \Delta\Theta$ 
17        $MSE \leftarrow MSE_{novo}$ 
18        $\mu \leftarrow \mu \div \beta$ 
19        $k \leftarrow k + 1$ 
20   fim
21 fim
22 retorna  $[\Theta, \mathbf{w}, MSE]$ 
```

3.2 Aprendizado Multi-objetivo de Redes com Pesos Esféricos

Algoritmo 3: ALGORITMO BFGS ESFÉRICO

Entrada: r, max_it, Θ

Saída: Solução com menor erro ($e(\mathbf{w})$) no espaço restrito por r

```
1 início
2    $k \leftarrow 1$ 
3    $\mathbf{B}_1 = I$ 
4   enquanto  $((k \leq max\_it) \text{ ou } (e(\mathbf{w}) \text{ estagnado por } 30 \text{ iterações}))$  faça
5       /* Calcular função de erro ( $e(\mathbf{w})$ ) */
6        $MSE \leftarrow e(\mathbf{w})$ 
7       Obter  $\Delta\Theta$  resolvendo-se a Equação (3.28)
8       /* Atualiza ângulos através da Equação (3.32), com  $\alpha$ 
9       obtido por backtracking line search */
10       $\Theta_{k+1} = \Theta_k + \alpha\Delta\Theta$ 
11      Obter  $\mathbf{s}_k$  e  $\mathbf{y}_k$  resolvendo-se as Equações (3.30) e (3.31)
12      Calcular  $\mathbf{B}_k$  utilizando as Equações (3.23) e (3.26)
13      /* Re-calcular erro ( $e(\mathbf{w})_{novo}$ ), utilizando a Equação
14      (3.4) */
15       $MSE \leftarrow MSE_{novo}$ 
16       $k \leftarrow k + 1$ 
17 fim
18 fim
19 retorna  $[\Theta, \mathbf{w}, MSE]$ 
```

3.3 Conclusão do Capítulo

Este capítulo apresentou as contribuições desta tese. A formulação que define a representação dos pesos por coordenadas esféricas foi apresentada bem como os benefícios gerados por esta forma de representação. De maneira didática, demonstrou-se a aplicação desta representação a um modelo simples com base em um único neurônio com função de ativação não-linear e, posteriormente, foram apresentadas definições acerca desta representação em redes MLP. O algoritmo para resolução do problema de aprendizado multiobjetivo foi apresentado junto à estratégia de busca utilizada para o novo espaço gerado. Por fim, foram descritos três algoritmos para o treinamento de redes MLP com a representação esférica, sendo que o primeiro consiste em um modelo teórico baseado no método *Gradiente Descendente*, onde as principais definições acerca da direção de busca obtida no espaço esférico foram detalhadas. Os dois outros algoritmos *Levenberg-Marquardt* esférico e *BFGS* esférico, considerados eficientes para o treinamento clássico de redes neurais, foram então definidos com a formulação esférica.

Capítulo 4

Experimentos em Problemas de Regressão e Classificação

O capítulo anterior apresentou os métodos propostos para o treinamento de redes MLP utilizando a formulação de pesos no espaço esférico. Este capítulo detalha os experimentos realizados para avaliação dos métodos propostos (**SLM** e **SBFGS**) aplicados a problemas de regressão e classificação. Foram realizados dois tipos de experimentos distintos para comparação dos métodos, como o objetivo de se avaliar tanto as estimativas do Conjunto Pareto quanto a solução final selecionada a partir do conjunto.

4.1 Descrição dos Experimentos

Para realização dos experimentos o método MOBJ (Teixeira *et al.*, 2000), pioneiro no treinamento multiobjetivo de redes MLP, foi tomado como referência. Este algoritmo mostrou resultados eficientes para obtenção de modelos neurais com boa capacidade de generalização, além de ser baseado no controle da complexidade aparente da rede, de maneira análoga aos métodos propostos. Uma versão deste algoritmo desenvolvido em Linguagem Matlab está descrita em (Teixeira, 2001).

4.1.1 Descrição das Bases de Dados

Os algoritmos desenvolvidos foram avaliados em 14 problemas de regressão e 16 de classificação, considerando-se problemas sintéticos e reais. Todas as bases reais

4.1 Descrição dos Experimentos

foram normalizadas para o intervalo $[-1, +1]$, da mesma maneira que em outros trabalhos com abordagens semelhantes para regressão e classificação (Tavares, 2015; Torres, 2016; Torres *et al.*, 2015).

Os problemas sintéticos de regressão utilizados foram gerados a partir de funções clássicas referenciadas em outros trabalhos como (Costa *et al.*, 2007; Teixeira *et al.*, 2000, 2007) e são definidos da seguinte forma:

$$\begin{aligned} f_1(x_i) &= \sin(x_i) + \gamma_i \\ f_2(x_i) &= (x_i - 2)(2x_i + 1)/(1 + x_i^2) + \gamma_i \\ f_3(x_i) &= 4.26(e^{-x_i} - 4e^{-2x_i} + 3e^{-3x_i}) + \gamma_i \\ f_4(x_i) &= e^{-0.2x_i} + (2e^{-0.2x_i} \times \sin(2\pi \times 0.2x_i - \pi/4) - 0.27) + \gamma_i \\ f_5(x_i) &= \sin(\pi x_i)/(\pi x_i) + \gamma_i \end{aligned} \tag{4.1}$$

onde $\gamma_i \sim Normal(\mu = 0; \sigma^2 = 0.2^2)$. Para cada base de dados foram geradas 100 amostras, $i = 1, \dots, 100$, geradas de forma uniforme, com valores limitados aos intervalos $[0; 2\pi]$, $[-8; 12]$, $[0; 3.25]$, $[0, 10]$ e $[0; 4]$ para as funções f_1 , f_2 , f_3 , f_4 e f_5 respectivamente.

Os outros 9 problemas de regressão foram obtidos do repositório UCI (Blake & Merz, 1998), exceto a base **mackey-glass** (Flake & Lawrence, 2002) que foi obtida através do repositório disponível em (Chang & Lin, 2011).

Os 16 problemas de classificação consistem em 3 bases de dados sintéticas: o primeiro é o problema four-Gaussian-class, que consiste em 4 classes Gaussianas geradas da mesma forma que em (Teixeira *et al.*, 2000). Os outros 2 problemas são o two.moon e o circles2 que são gerados utilizando-se rotinas disponíveis no pacote clusterSim (Walesiak & Dudek, 2014) no software R (R Core Team, 2016). As outras 13 bases de dados consistem em problemas reais obtidas a partir dos repositórios UCI (Blake & Merz, 1998) e KEEL (Alcalá *et al.*, 2011).

Dados adicionais acerca das bases de dados podem ser visualizados nas Tabelas 4.1 e 4.2, onde **Base de dados** descreve o nome da base, **Alias** mostra uma versão abreviada do nome da base, **N** é o tamanho da amostra de dados e **D** representa a quantidade de dimensões da base.

4.1 Descrição dos Experimentos

Tabela 4.1: Bases de dados para Regressão.

Base de dados	Alias	N	D
$f_1(x)$	f_1	100	1
$f_2(x)$	f_2	100	1
$f_3(x)$	f_3	100	1
$f_4(x)$	f_4	100	1
$f_5(x)$	f_5	100	1
servo	srv	167	4
auto mpg	mpg	392	8
boston housing	bos	506	13
airfoils selfnoise	air	1503	5
forest fires	frt	517	12
concrete slump	con	103	7
yacht hydrodynamics	yac	308	6
concrete comp. strength	ccs	1030	8
mackey-glass	mck	1385	6

Tabela 4.2: Bases de dados para Classificação.

Base de dados	Alias	N	D
four-Gaussian-class	gau	200	2
circles2	cir	200	2
two.moon	moo	200	2
haberman	hab	306	3
diabetes	dia	768	8
sonar	son	208	60
bupa	bup	345	6
vertebral column	vtc	310	6
heart diseases	hrt	270	13
blood transfusion	btf	748	4
banana	ban	5300	2
australian credit approval	aus	690	14
wisconsin breast cancer	wbc	569	31
banknote	bak	1372	4
fertility	fer	100	9
ILPD	ilp	579	10

4.1.2 Parametrização da Rede

A estrutura da rede MLP para os 3 métodos de treinamento é composta por 20 neurônios na camada escondida, com função de ativação tangente hiperbólica, enquanto a camada de saída tem 1 neurônio com saída linear. Os conjuntos Pareto estimados pelos métodos consistem em um máximo de soluções $n_{sol} = 30$, igualmente espaçadas com relação à norma, com variação $\delta r = 1$ para os 5 problemas sintéticos de regressão e $\delta r = 2$ para os demais problemas, de forma que n_{sol} soluções são geradas e um filtro é utilizado para selecionar apenas as soluções não-dominadas. A estrutura da rede e os parâmetros n_{sol} e δr para todos os métodos considerados seguem o padrão utilizado em (Teixeira, 2001) para experimentos com bases de dados sintéticas e reais de regressão e classificação. Outros trabalhos, visando redes com estruturas previamente sobre-parametrizadas, utilizam estruturas e parâmetros semelhantes com pequenas variações. Em (Costa & Braga, 2006) foram utilizados 15 neurônios na camada escondida para problemas de regressão e classificação, considerando-se problemas sintéticos e reais obtidos no repositório UCI (Blake & Merz, 1998). Nos trabalhos desenvolvidos em (Horta *et al.*, 2008) e (Medeiros *et al.*, 2009), foram utilizados 10 neurônios para problemas reais e sintéticos de classificação, respectivamente. Em (Costa & Braga, 2011) 10 neurônios foram utilizados na camada escondidos, apenas para problemas de regressão. Em (Rocha *et al.*, 2011), foram utilizados 15 neurônios para regressão e classificação. Para todos os algoritmos foi definido um máximo de 3000 iterações no processo de otimização para obtenção de cada solução do conjunto e o critério de parada foi definido de maneira que os algoritmos são finalizados quando não ocorrerem alterações significativas (da ordem de 10^{-6}) no valor do erro durante 30 iterações consecutivas.

4.1.3 Formato dos Experimentos

Foram realizados dois tipos de experimentos para avaliação dos métodos propostos. O primeiro experimento tem o objetivo de comparar as estimativas de conjuntos Pareto geradas pelos métodos enquanto o segundo tem o propósito de comparar o erro médio quadrático (*mean squared error* - MSE, em inglês) e a acurácia (ACC) obtidos pela solução final selecionada, para um conjunto de dados de teste, em relação aos problemas de regressão e classificação respectivamente.

4.1 Descrição dos Experimentos

O Experimento 1 consiste na avaliação dos conjuntos Pareto estimados por cada método desenvolvido em contraposição a um algoritmo de referência. As métricas Hipervolume Dominado (HV) e Cobertura de dois conjuntos - (CS) são utilizadas para comparação dos conjuntos Pareto estimados e foram escolhidas por não necessitarem de uma amostra da fronteira Pareto-ótima global do problema, que é difícil de ser determinada em problemas de aprendizado a partir de amostragens de dados. A métrica HV estima tanto convergência quanto diversidade do conjunto gerado, o que de certa maneira indica uma tendência geral para o desempenho dos algoritmos. Por outro lado, a métrica CS é mais apropriada para se avaliar o percentual de soluções que um conjunto não-dominado obtido por um determinado algoritmo domina em relação ao conjunto obtido por outro algoritmo.

O Experimento 2 consiste na avaliação das soluções finais obtidas a partir dos conjuntos Pareto estimados. Dentre os algoritmos desenvolvidos, aquele com maiores valores médios de HV e CS no Experimento 1 será utilizado junto ao algoritmo referência como representantes do aprendizado multiobjetivo no Experimento 2. Adicionalmente, alguns métodos do estado da arte serão considerados com o objetivo de se avaliar de maneira geral os algoritmos desenvolvidos quanto à solução final.

Em todos os experimentos utilizou-se um procedimento de validação cruzada com 10-*folds* (em inglês, *10-fold cross-validation*), *i.e.*, o conjunto de dados foi dividido em blocos mutuamente exclusivos, onde 9 destes foram utilizados para formar um conjunto de dados para treinamento e o décimo foi usado como conjunto de teste. Durante cada iteração dentre as 10 do procedimento de validação cruzada, 1 bloco foi utilizado como conjunto de teste e os dados restantes (9 blocos), foram divididos aleatoriamente em duas porções, sendo 80% para treinamento e 20% para validação, onde esta última porção é utilizada com o objetivo de selecionar um modelo do conjunto Pareto estimado para o bloco em questão.

Os experimentos foram realizadas em um computador com processador Intel Core i7 3630QM 2.4GHz, 64bits com 16GB de memória. Para codificação utilizou-se o software R (R Core Team, 2016) na versão 3.3.1 num ambiente com sistema operacional Ubuntu 16.04 LTS. Adicionalmente, é mostrado também o tempo médio de execução (TME) em segundos e o respectivo desvio padrão considerando-se os 10 *folds*, para todos os métodos.

4.1.4 Métricas para Avaliação das Estimativas do Conjunto Pareto

Como definido na Seção 2.4.1 um conjunto Pareto é formado por várias soluções onde cada uma destas pode ser composta por dois ou mais valores, que consistem em valores relativos a cada critério avaliado no problema em questão. Devido ao conflito das funções objetivo em um conjunto Pareto, existe a necessidade de se definir um indicador que faça um mapeamento deste conjunto para um valor particular, com o propósito de facilitar a comparação do desempenho de algoritmos para otimização multiobjetivo. Neste trabalho duas métricas foram utilizadas:

- Métrica Hipervolume Dominado (em inglês, *Hypervolume* - *HV*)
Ao se considerar indicadores para mensurar a qualidade de um conjunto de soluções, existe a necessidade de se analisar tanto o aspecto convergência, que mede o quanto este conjunto está próximo do conjunto Pareto-Ótimo global do problema, quanto o aspecto diversidade, que define o quanto as soluções estão bem distribuídas ao longo da fronteira Pareto. A métrica que calcula o hipervolume dominado (*HV*), também conhecida como *S-Metric* (Zitzler, 1999), gera um indicador através do cálculo do hipervolume entre uma região multidimensional, definida por soluções Pareto estimadas, e uma solução dominada utilizada como ponto de referência, onde conjuntos com maiores valores de hipervolume dominam um espaço maior. A motivação na escolha desta métrica se encontra no fato dela levar em consideração tanto o aspecto convergência quanto diversidade de um conjunto de soluções. Nos experimentos deste trabalho, a solução utilizada como ponto de referência foi definida através do máximo valor de cada dimensão do conjunto Pareto com adição de 10%, considerando-se todos os métodos.
- Métrica Cobertura de Dois Conjuntos (em inglês, *Coverage of Two Sets* - *CS*)
Diferente de *HV*, que mensura um conjunto de soluções não-dominadas com relação a um ponto de referência, a métrica **CS**, ou *C-Metric*, (Zitzler, 1999) é classificada como binária, dado que ela consiste em uma medida para comparação entre dois conjuntos de soluções não-dominadas. A métrica

4.2 Experimento 1: Avaliação dos Conjuntos Pareto Estimados

CS mapeia um par ordenado de conjuntos de soluções $(\mathcal{PO}_1, \mathcal{PO}_2)$ para um intervalo $[0, 1]$, sendo definida da seguinte maneira (Batista, 2011):

$$CS(\mathcal{PO}_1, \mathcal{PO}_2) = \frac{|\{s_2 \in \mathcal{PO}_2 \mid \exists s_1 \in \mathcal{PO}_1 : s_1 \preceq s_2\}|}{|\mathcal{PO}_2|} \quad (4.2)$$

onde $\mathcal{PO}_1$ e $\mathcal{PO}_2$ representam as estimativas do conjunto Pareto obtidas por dois algoritmos diferentes, e $s_1 \preceq s_2$ denota que s_1 cobre s_2 , *i.e.*, $s_1 \prec s_2$ ou $s_1 = s_2$. Observe que $CS(\mathcal{PO}_1, \mathcal{PO}_2) = 1$ indica que todas as soluções em $\mathcal{PO}_2$ são dominadas por ou iguais às soluções contidas em $\mathcal{PO}_1$ enquanto $CS(\mathcal{PO}_1, \mathcal{PO}_2) = 0$ determina que nenhum elemento em $\mathcal{PO}_2$ é coberto pelos elementos de $\mathcal{PO}_1$. É importante que se obtenha tanto $CS(\mathcal{PO}_1, \mathcal{PO}_2)$ quanto $CS(\mathcal{PO}_2, \mathcal{PO}_1)$ dado que estes valores não são necessariamente complementares.

4.2 Experimento 1: Avaliação dos Conjuntos Pareto Estimados

Através do procedimento de validação cruzada adotado, foram obtidas 10 estimativas do conjunto Pareto aplicando-se cada método aos problemas teste. Para cada estimativa gerada, calcula-se o valor de HV e CS, sendo que para este último, considera-se o pareamento entre todos os métodos. Dentre as 10 estimativas obtidas para cada base de dados aquela com o maior valor de HV é utilizada nas figuras das Subseções 4.2.1 e 4.2.2 com o objetivo de se poder visualizar e comparar as estimativas. Nas Subseções 4.2.1 e 4.2.2, as Figuras geradas mostram as estimativas do conjunto Pareto e soluções finais selecionadas, onde uma análise a partir destas é realizada. Uma análise mais apropriada com base numa estatística de teste é realizada na Subseção 4.2.3.

4.2.1 Problemas de Regressão

Considerando-se os problemas sintéticos de regressão, as Figuras 4.1 a 4.5 apresentam as estimativas do conjunto Pareto e a solução selecionada considerando-se o *fold* onde se obteve o maior valor de HV em cada algoritmo. Estas figuras apenas ilustram as estimativas obtidas pelos métodos além das curvas representando

4.2 Experimento 1: Avaliação dos Conjuntos Pareto Estimados

a aproximação obtida em relação à função geradora dos dados. Por este motivo, apenas os problemas sintéticos são representados, enquanto as figuras referentes aos conjuntos estimados para os problemas reais de regressão podem ser visualizadas no Apêndice A.1. As Tabelas 4.3 e 4.4 resumem de maneira mais adequada o desempenho dos algoritmos para todas as bases de regressão. A Tabela 4.3 apresenta os valores médios de HV obtidos pelos algoritmos para cada base de dados enquanto a Tabela 4.4 mostra os valores médios de CS considerando-se cada par de algoritmos nos dois sentidos de avaliação. Os melhores resultados obtidos para cada métrica estão em negrito.

Tabela 4.3: Valores médios e desvio padrão de **HV** para os problemas de Regressão

	MOBJ	SLM	SBFGS
f_1	16.668±0.578	16.692±0.566	16.601±0.641
f_2	39.901±6.48	39.832±6.561	32.376±16.162
f_3	7.367±0.403	7.415±0.424	7.401±0.394
f_4	15.779±0.662	15.845±0.673	15.791±0.689
f_5	2.519±0.146	2.518±0.148	2.523±0.15
srv	5.084±1.054	5.083±1.032	1.512±2.164
mpg	8.201±0.235	8.387±0.223	8.151±0.374
bos	5.538±0.119	5.57±0.121	5.57±0.118
air	3.98±0.047	4.01±0.037	3.648±0.366
frt	0.923±0.197	0.96±0.208	0.413±0.464
con	17.317±4.388	17.297±4.386	17.205±4.428
yac	14.526±0.379	14.525±0.379	14.518±0.376
ccs	11.234±0.181	11.18±0.232	11.244±0.189
mck	16.372±0.241	16.412±0.256	16.411±0.247

Embora os conjuntos de soluções ilustrados pelas figuras sejam muito próximos de maneira visual, nota-se que os conjuntos obtidos pelos métodos SLM e SBFGS são melhores que aqueles gerados pelo MOBJ, exceto para a função f_2 , onde o método SBFGS se mostra menos preciso que o MOBJ para valores elevados de norma. A Figura 4.3(a) demonstra mais claramente a dominância dos métodos propostos em relação ao MOBJ para a função f_3 , e através da Figura 4.5(a) pode-se notar dificuldades enfrentadas pelo MOBJ em gerar soluções não dominadas mesmo para uma função simples de *benchmark*. Pode ser também verificado pelas figuras, que na maioria dos problemas simulados, as soluções obtidas pelo decisor

4.2 Experimento 1: Avaliação dos Conjuntos Pareto Estimados

Tabela 4.4: Valores médios e desvio padrão de **CS** para os problemas de Regressão. As abreviações $\mathcal{PO}_1$, $\mathcal{PO}_2$ e $\mathcal{PO}_3$ representam as estimativas do conjunto Pareto obtidas pelos métodos MOBJ, SLM e SBFGS, respectivamente.

	$(\mathcal{PO}_1, \mathcal{PO}_2)$	$(\mathcal{PO}_1, \mathcal{PO}_3)$	$(\mathcal{PO}_2, \mathcal{PO}_1)$	$(\mathcal{PO}_2, \mathcal{PO}_3)$	$(\mathcal{PO}_3, \mathcal{PO}_1)$	$(\mathcal{PO}_3, \mathcal{PO}_2)$
f_1	0.061±0.078	0.214±0.178	0.767±0.168	0.469±0.161	0.622±0.182	0.321±0.159
f_2	0.317±0.379	0.188±0.275	0.535±0.346	0.186±0.276	0.51±0.356	0.483±0.387
f_3	0.013±0.023	0.035±0.05	0.669±0.093	0.507±0.208	0.513±0.263	0.203±0.196
f_4	0.087±0.072	0.117±0.14	0.682±0.072	0.407±0.149	0.637±0.136	0.287±0.146
f_5	0.071±0.054	0.115±0.131	0.484±0.162	0.474±0.203	0.425±0.216	0.208±0.138
srv	0.087±0.058	0.033±0.094	0.73±0.053	0.057±0.148	0.154±0.292	0.09±0.245
mpg	0.194±0.088	0.346±0.183	0.5±0.156	0.574±0.17	0.35±0.158	0.236±0.171
bos	0.081±0.057	0.108±0.068	0.728±0.099	0.445±0.24	0.676±0.192	0.39±0.263
air	0.047±0.043	0.553±0.348	0.653±0.083	0.71±0.229	0.296±0.281	0.175±0.185
frt	0.177±0.271	0.185±0.381	0.768±0.099	0.336±0.437	0.271±0.428	0.16±0.304
con	0.236±0.141	0.262±0.154	0.702±0.19	0.87±0.084	0.598±0.169	0.217±0.125
yac	0.008±0.017	0.107±0.232	0.897±0.051	0.369±0.271	0.668±0.349	0.623±0.286
ccs	0.043±0.074	0.16±0.098	0.752±0.121	0.47±0.251	0.603±0.204	0.522±0.257
mck	0.237±0.309	0.16±0.194	0.487±0.324	0.53±0.318	0.517±0.254	0.47±0.318

de validação indicam valores de erro semelhantes, mas na maioria dos problemas as soluções finais para os métodos propostos possuem norma menor ou igual às soluções finais do MOBJ. Embora hajam diferenças em relação aos conjuntos obtidos para os problemas sintéticos de regressão verifica-se que a solução final obtida para todos os métodos são semelhantes, um exemplo deste fato pode ser observado na Figura 4.5(b), onde todos os métodos obtiveram curvas semelhantes, mesmo após o MOBJ enfrentar dificuldades para gerar soluções com valores mais elevados de norma. Ao considerarmos os problemas reais de regressão, os métodos propostos obtiveram maiores valores para a média de HV em 6 dos 9 problemas, como pode ser visto na Tabela 4.3. Com base na métrica CS , os valores médios mostrados na Tabela 4.4 indicam que o método SLM foi o que obteve os melhores resultados em todas as bases de dados, demonstrando que os conjuntos de soluções obtidas por ele, na média, cobrem todos os outros conjuntos.

4.2 Experimento 1: Avaliação dos Conjuntos Pareto Estimados

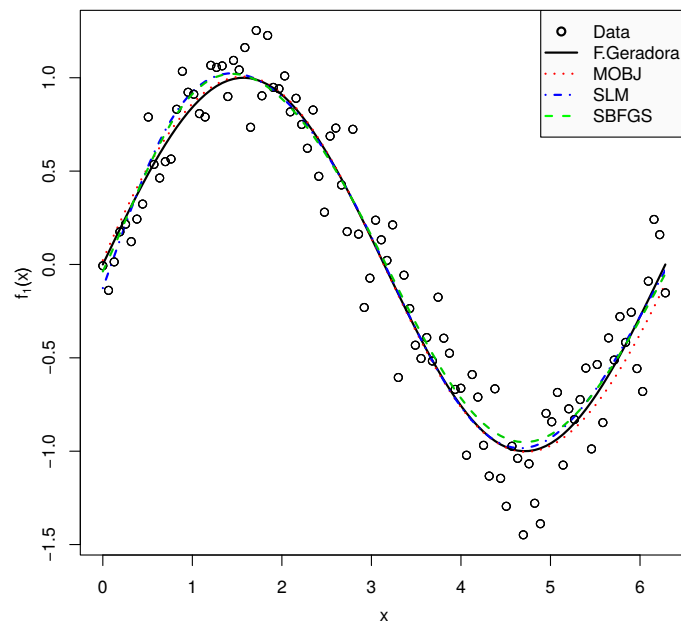
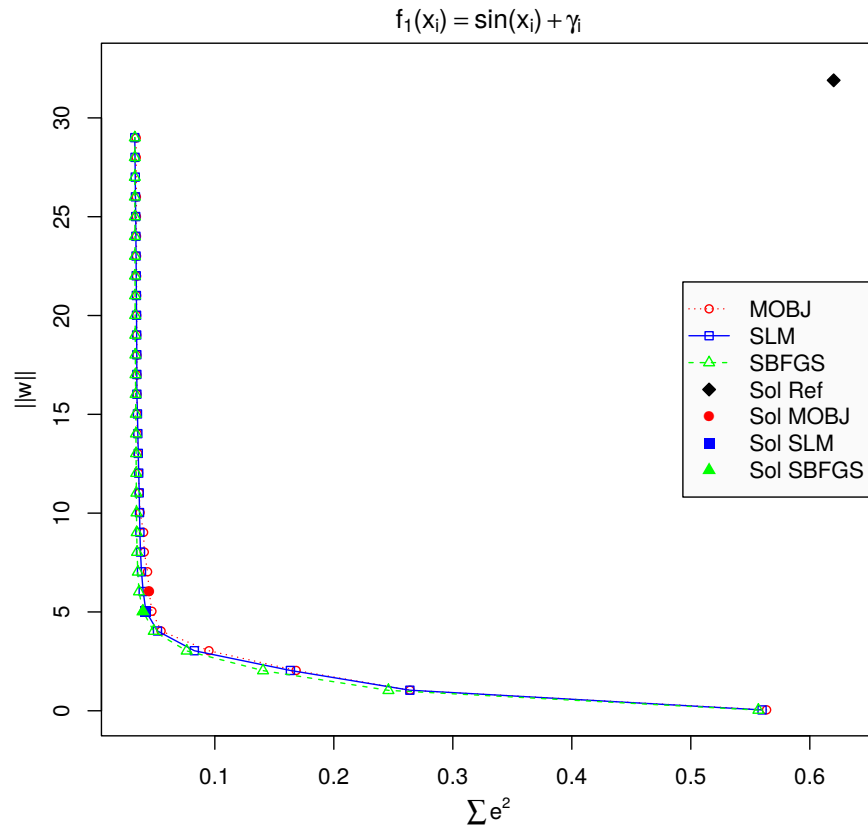


Figura 4.1: (a) Paretos estimados - f_1 . (b) Aproximação para f_1 .

4.2 Experimento 1: Avaliação dos Conjuntos Pareto Estimados

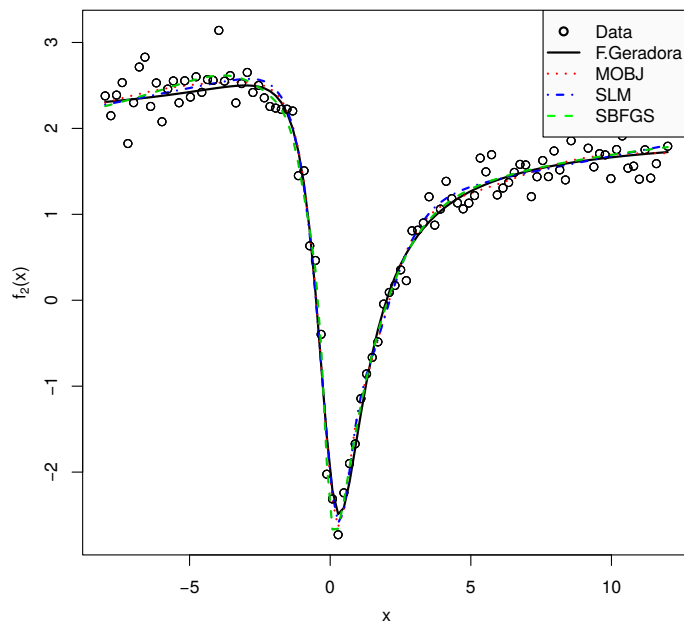
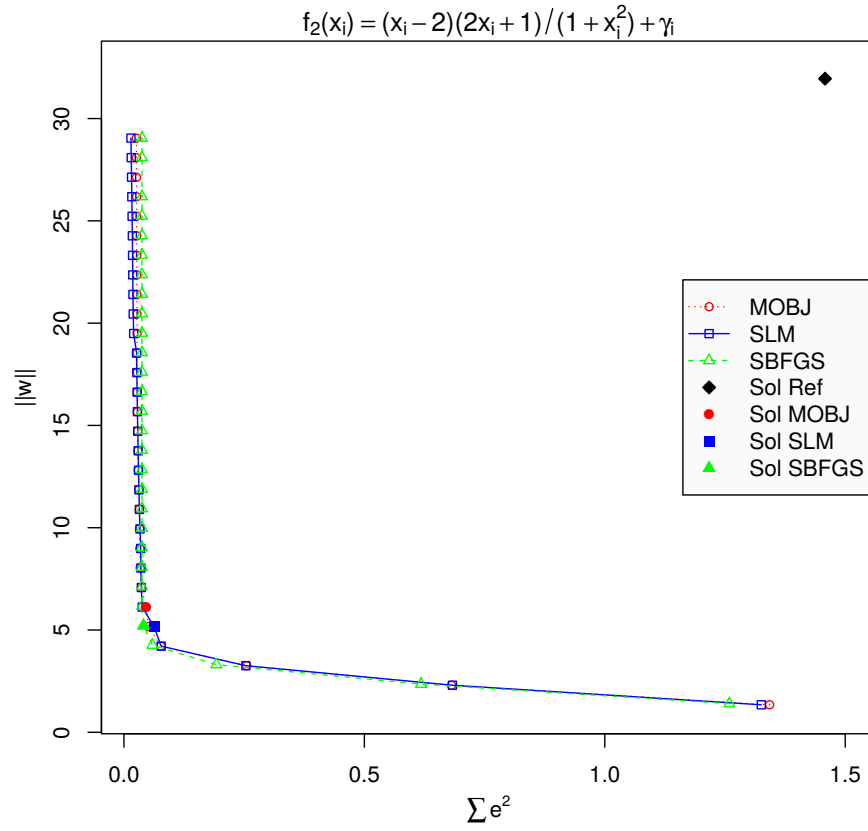


Figura 4.2: (a) Paretos estimados - Função f_2 . (b) Aproximação para f_2 .

4.2 Experimento 1: Avaliação dos Conjuntos Pareto Estimados

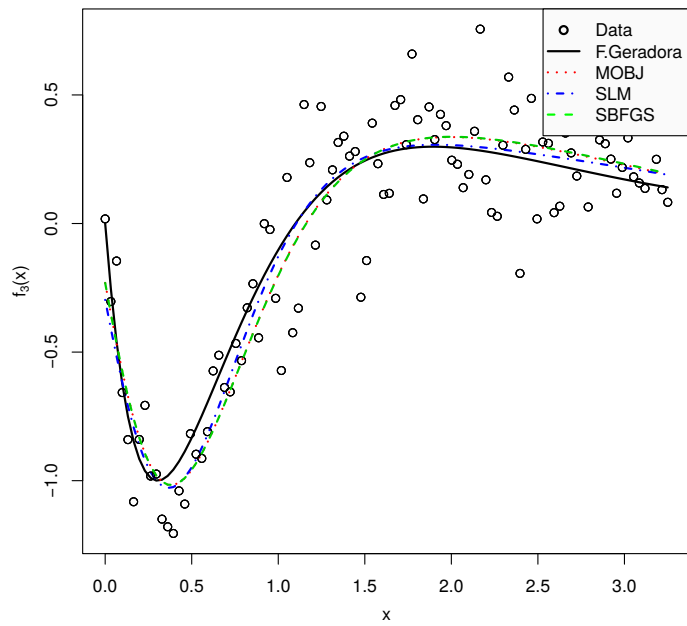
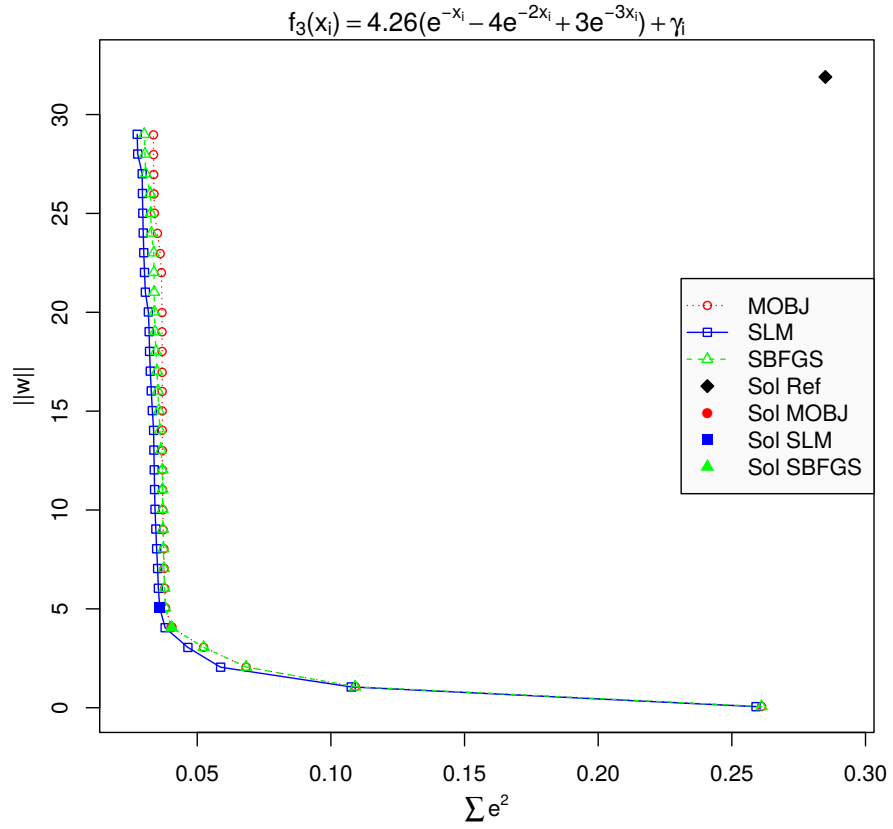
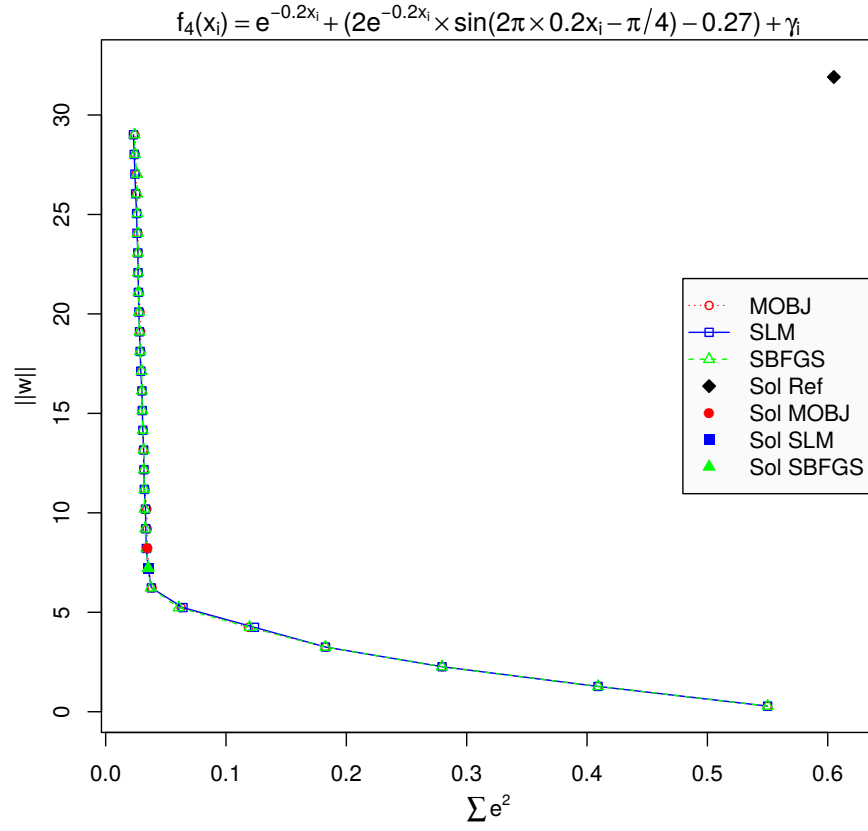
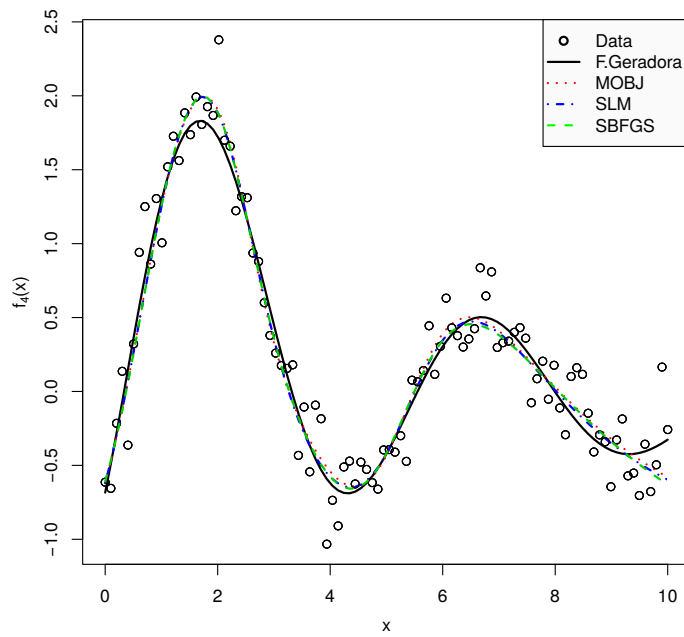


Figura 4.3: (a) Paretos estimados - Função f_3 . (b) Aproximação para f_3 .

4.2 Experimento 1: Avaliação dos Conjuntos Pareto Estimados



(a)



(b)

Figura 4.4: (a) Paretos estimados - Função f_4 . (b) Aproximação para f_4 .

4.2 Experimento 1: Avaliação dos Conjuntos Pareto Estimados

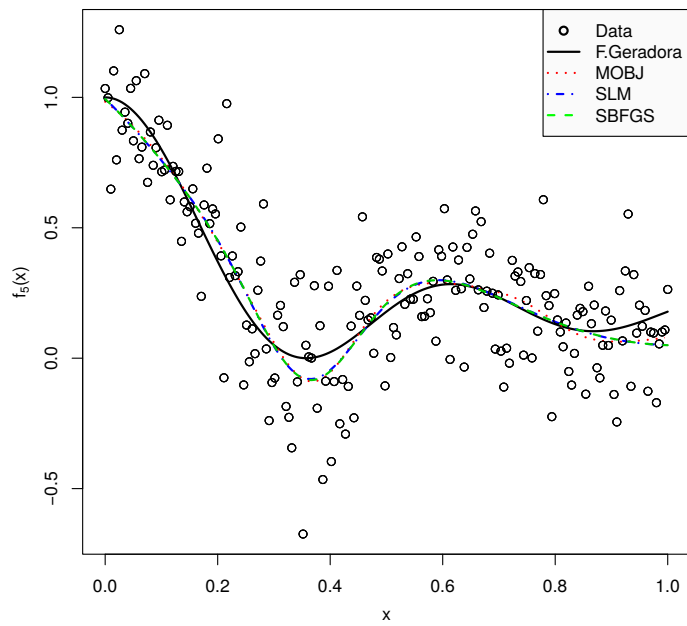
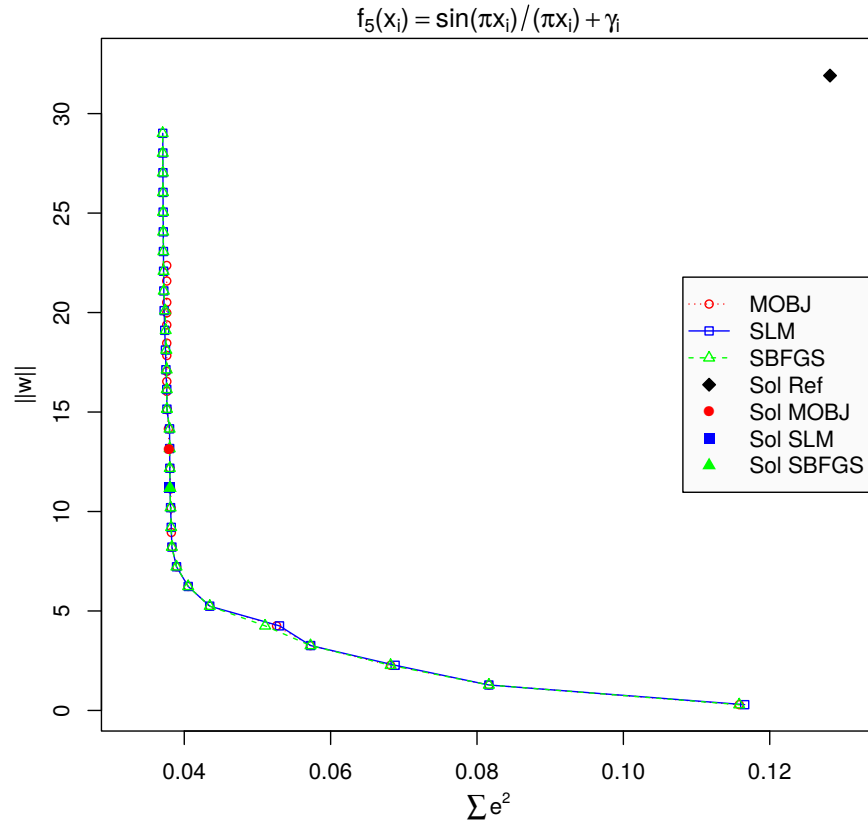


Figura 4.5: (a) Paretos estimados - Função f_5 . (b) Aproximação para f_5 .

4.2.2 Problemas de Classificação

Consideremos as Figuras de 4.6 a 4.12 para realização de uma análise visual acerca das estimativas do conjunto Pareto obtidas pelos métodos para bases de dados de classificação. A escolha destas bases se deve a algumas características particulares: as Figuras 4.6 e 4.7 referem-se aos problemas sintéticos binários, o que permite a visualização bidimensional da superfície de separação que representa a solução selecionada; as Figuras de 4.8 a 4.11 ilustram as maiores bases de dados quanto à quantidade de dimensões, e a Figura 4.12 mostra a única base de dados real com 2 dimensões, o que permite a visualização da solução selecionada e, além disso, esta é a maior base quanto à quantidade de amostras. Além dos conjuntos de soluções pode-se visualizar nas Figuras 4.6(b), 4.7(b) e 4.12(b) a solução final selecionada, que mostra as curvas de separação entre as classes representando a aproximação obtida em relação à função geradora dos dados. Os conjuntos de soluções representados nas figuras são referentes ao *fold* em que os métodos obtiveram o maior valor de *HV*. As figuras desta subseção apenas ilustram algumas das estimativas obtidas, enquanto uma análise mais apropriada pode ser realizada por meio dos resultados contidos nas Tabelas 4.5 e 4.6, que descrevem os valores médios de *HV* e *CS*, respectivamente. Os conjuntos de soluções estimados para os outros problemas de classificação podem ser visualizados no Apêndice A.2.

As soluções estimadas em relação aos problemas sintéticos representados nas Figuras 4.6(a) e 4.7(a) mostram que os métodos geram conjuntos semelhantes na maior parte da fronteira estimada, com leve perda para o SBFGS na região de norma entre 2 e 12 na base Two-moon. Entretanto, pode-se observar que os métodos SLM e SBFGS conseguem gerar soluções não-dominadas durante toda a faixa pré-definida de valores de norma enquanto o MOBJ não conseguiu gerar soluções para valores mais elevados, sendo que para a base Two-moon as soluções geradas por ele não chegam à norma de valor 30. Com relação às soluções finais, verifica-se que o decisor por erro de validação escolhe soluções com valores semelhantes de erro para todos os métodos, mas para o SLM e SBFGS as soluções escolhidas possuem menor complexidade, dado que possuem menor valor de norma. O menor nível de complexidade é perceptível com relação ao método SLM, como pode ser visualizado nas Figuras 4.6(b) e 4.7(b) que mostram maior suavidade nas curvas geradas pelas soluções oriundas do SLM.

4.2 Experimento 1: Avaliação dos Conjuntos Pareto Estimados

Tabela 4.5: Valores médios e desvio padrão de **HV** para os problemas de Classificação

	MOBJ	SLM	SBFGS
gau	45.825±19.297	45.078±18.547	40.516±26.279
cir	65.069±0.134	65.044±0.146	30.194±30.096
moo	67.195±0.238	67.282±0.156	62.621±4.912
hab	29.023±1.318	30.401±1.267	29.178±3.186
dia	43.134±0.557	43.697±0.83	28.097±19.35
son	13.88±17.865	13.983±17.855	13.93±17.863
bup	46.166±1.071	48.797±0.886	46.696±2.123
vtc	46.887±1.07	49.184±0.988	49.083±0.846
hrt	47.898±16.501	47.665±16.679	47.136±17.029
btf	16.306±0.716	16.899±0.784	16.517±2.03
ban	46.264±0.353	46.718±0.229	46.734±0.327
aus	60.072±0.595	61.397±0.428	61.244±0.37
wbc	55.829±10.853	55.561±11.321	26.34±8.558
bak	66.135±0.268	66.317±0.253	66.339±0.24
fer	9.824±8.657	9.819±8.636	8.443±9.532
ilp	31.476±0.777	34.612±1.384	34.853±0.773

Com relação às bases de dados com maior quantidade de dimensões, pode-se perceber que o método SLM foi mais estável de maneira geral. Os conjuntos de soluções apresentados na Figura 4.8 mostram que para a base Breast cancer wisconsin o SLM obteve as melhores estimativas, gerando soluções em toda a faixa pré-definida de valores de norma enquanto o MOBJ e o SBFGS sofrem estagnação antes da norma 30 sendo que este último obteve o pior conjunto de soluções. Como mostra a Figura 4.9, em relação à base Diabetes os métodos SLM e MOBJ geram soluções muito semelhantes com leve superioridade do SLM a partir da norma 15, enquanto o SBFGS mostra comportamento irregular, mas com as melhores soluções após a norma 28. A base de dados Sonar, por possuir a maior quantidade de dimensões, gerou dificuldades a todos os métodos para obtenção de soluções não-dominadas. Através da Figura 4.10 pode-se verificar que todos os métodos obtiveram poucas soluções, sendo que o MOBJ gerou mais soluções não-dominadas a partir da norma 6. Contudo, o SLM gerou as melhores soluções enquanto o SBFGS foi o pior para esta base. Quanto às bases de dados Australian credit approval e Banana, os métodos SLM e MOBJ foram novamente

4.2 Experimento 1: Avaliação dos Conjuntos Pareto Estimados

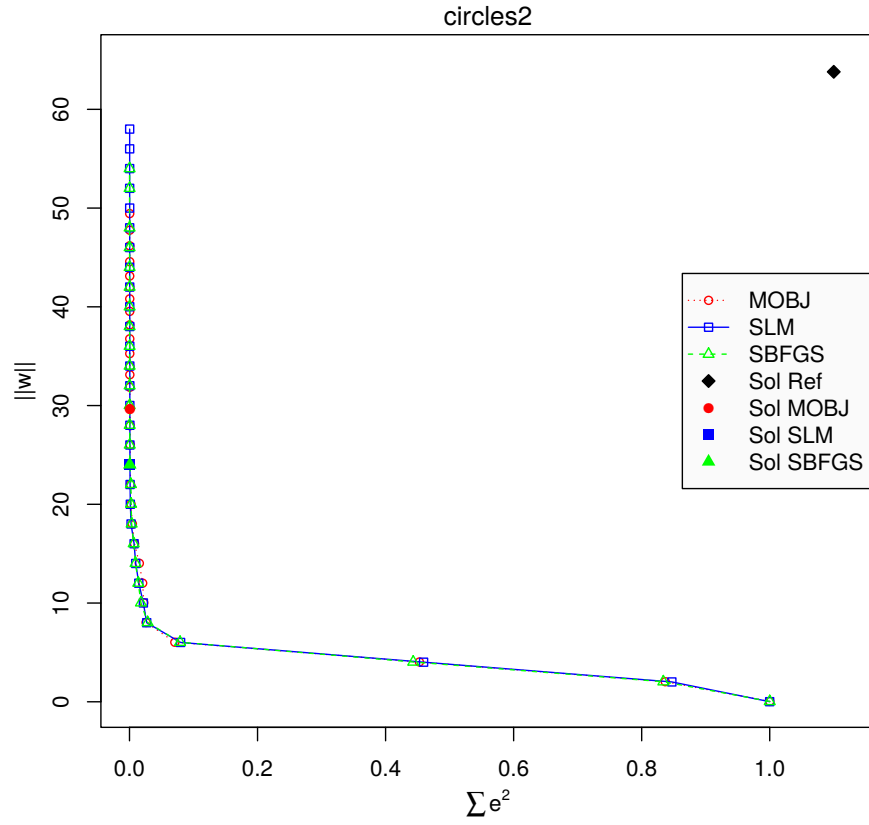
Tabela 4.6: Valores médios e desvio padrão de **CS** para os problemas de Classificação. As abreviações $\mathcal{PO}_1$, $\mathcal{PO}_2$ e $\mathcal{PO}_3$ representam as estimativas do conjunto Pareto obtidas pelos métodos MOBJ, SLM e SBFGS, respectivamente.

	$(\mathcal{PO}_1, \mathcal{PO}_2)$	$(\mathcal{PO}_1, \mathcal{PO}_3)$	$(\mathcal{PO}_2, \mathcal{PO}_1)$	$(\mathcal{PO}_2, \mathcal{PO}_3)$	$(\mathcal{PO}_3, \mathcal{PO}_1)$	$(\mathcal{PO}_3, \mathcal{PO}_2)$
gau	0.42±0.194	0.273±0.225	0.257±0.172	0.349±0.243	0.273±0.246	0.435±0.285
cir	0.247±0.209	0.502±0.351	0.6±0.204	0.659±0.214	0.261±0.312	0.19±0.217
moo	0.14±0.198	0.195±0.196	0.691±0.217	0.485±0.281	0.516±0.319	0.393±0.331
hab	0.395±0.205	0.453±0.282	0.217±0.174	0.52±0.264	0.287±0.228	0.477±0.268
dia	0.266±0.237	0.219±0.271	0.53±0.242	0.419±0.399	0.26±0.278	0.177±0.23
son	0.0±0.0	0.092±0.143	0.726±0.095	0.413±0.346	0.565±0.265	0.098±0.132
bup	0.34±0.247	0.458±0.262	0.357±0.218	0.566±0.223	0.33±0.242	0.413±0.218
vtc	0.293±0.214	0.337±0.244	0.402±0.211	0.52±0.239	0.348±0.204	0.48±0.239
hrt	0.464±0.198	0.337±0.263	0.414±0.28	0.663±0.292	0.412±0.315	0.468±0.256
btf	0.353±0.281	0.526±0.263	0.4±0.258	0.616±0.303	0.25±0.253	0.37±0.312
ban	0.207±0.153	0.29±0.195	0.535±0.131	0.52±0.261	0.448±0.188	0.48±0.261
aus	0.11±0.069	0.247±0.267	0.671±0.061	0.493±0.308	0.565±0.243	0.507±0.308
wbc	0.062±0.058	0.32±0.473	0.651±0.135	0.04±0.126	0.0±0.0	0.003±0.011
bak	0.217±0.189	0.103±0.096	0.533±0.222	0.387±0.148	0.607±0.201	0.613±0.148
fer	0.286±0.232	0.166±0.198	0.377±0.293	0.273±0.359	0.244±0.31	0.34±0.337
ilp	0.193±0.091	0.227±0.154	0.463±0.179	0.46±0.271	0.457±0.196	0.428±0.272

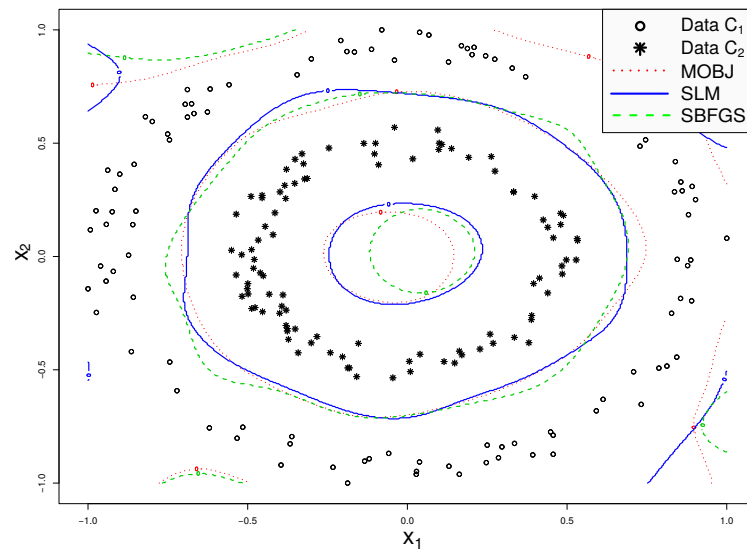
semelhantes, com leve superioridade para o SLM como pode ser verificado nas Figuras 4.11 e 4.12. O SBFGS gerou estimativas semelhantes aos outros métodos para a base Banana, mas demonstrou instabilidade para a base Australian credit approval. Em relação às soluções finais selecionadas, as figuras mostram que os métodos foram novamente semelhantes de maneira geral, porém as soluções selecionadas para os métodos SLM e SBFGS foram as de menor norma para a maioria dos problemas. A Figura 4.12(b) mostra as curvas de separação obtidas pelos métodos, onde pode-se notar que são similares, mas aquela referente ao SLM apresenta maior nível de suavidade.

Considerando agora todos os problemas de classificação, a Tabela 4.5 mostra que os métodos propostos nesta tese obtiveram os melhores valores de HV em 11 das 16 bases, com superioridade para o SLM que obteve o melhor resultado em 8. Quanto à métrica CS os resultados foram ainda mais favoráveis aos métodos propostos como mostra a Tabela 4.6, onde o método SLM obteve os melhores resultados em 14 das 16 bases, superando amplamente o MOBJ e o SBFGS.

4.2 Experimento 1: Avaliação dos Conjuntos Pareto Estimados



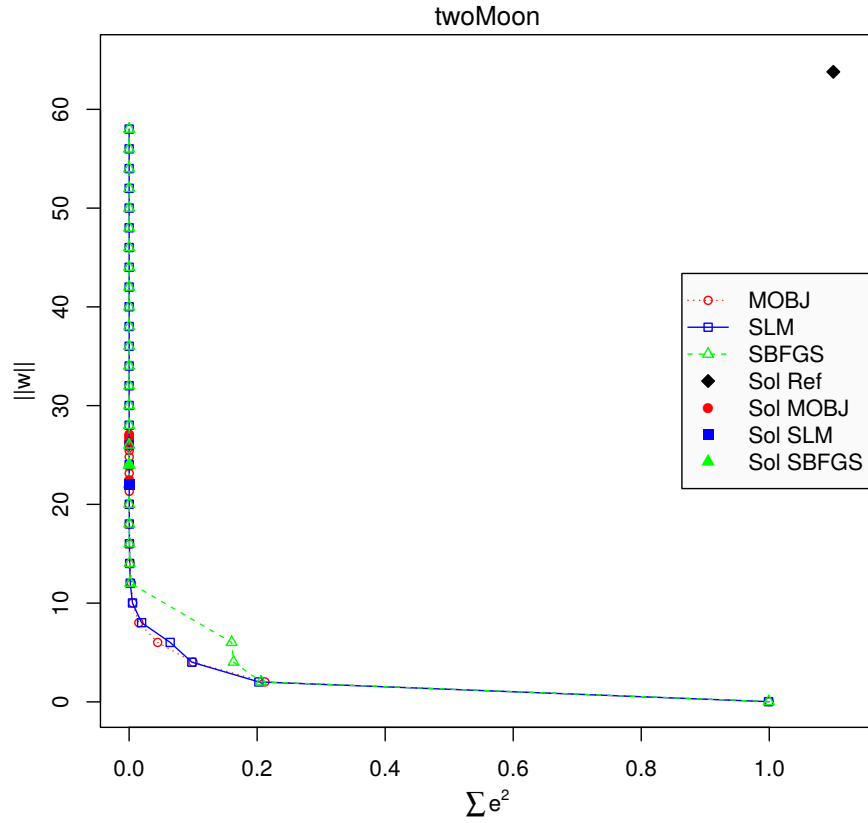
(a)



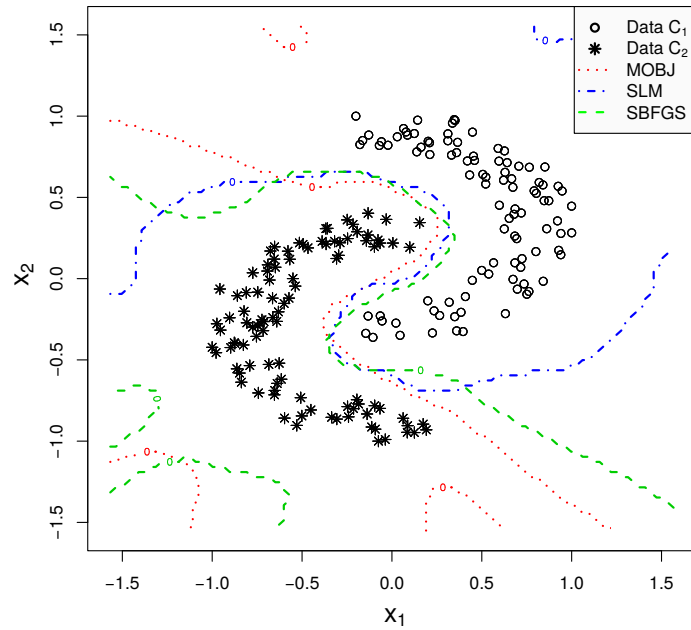
(b)

Figura 4.6: (a) Paretos estimados - Two Circles. (b) Superfície de separação - Two Circles.

4.2 Experimento 1: Avaliação dos Conjuntos Pareto Estimados



(a)



(b)

Figura 4.7: (a) Paretos estimados - Two Moon. (b) Superfície de separação - Two Moon.

4.2 Experimento 1: Avaliação dos Conjuntos Pareto Estimados

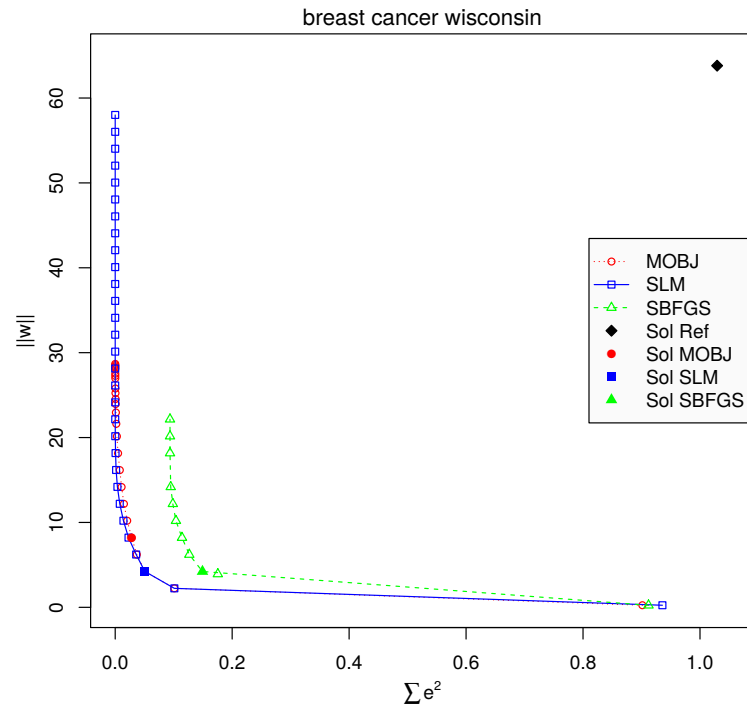


Figura 4.8: Paretos estimados - Base de dados Breast Cancer Wisconsin.

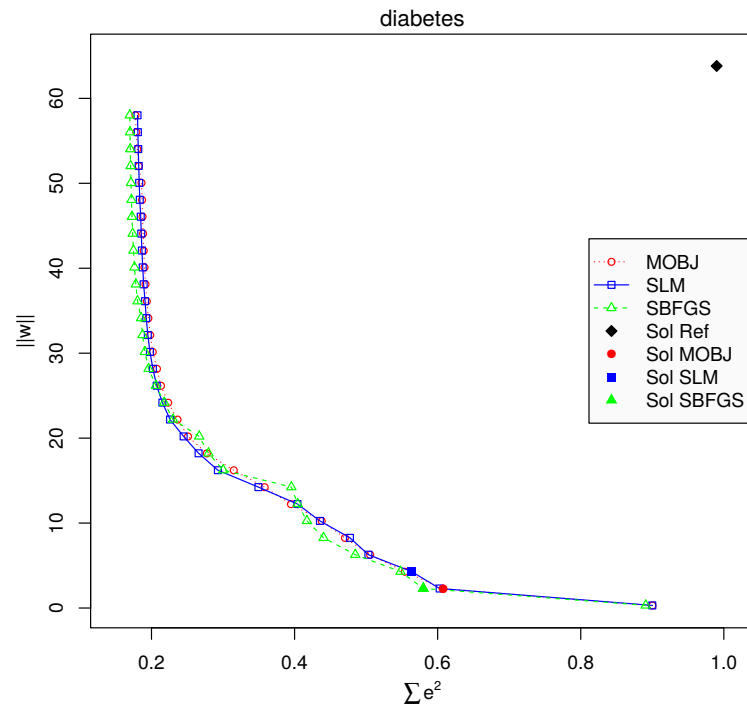


Figura 4.9: Paretos estimados - Base de dados Diabetes.

4.2 Experimento 1: Avaliação dos Conjuntos Pareto Estimados

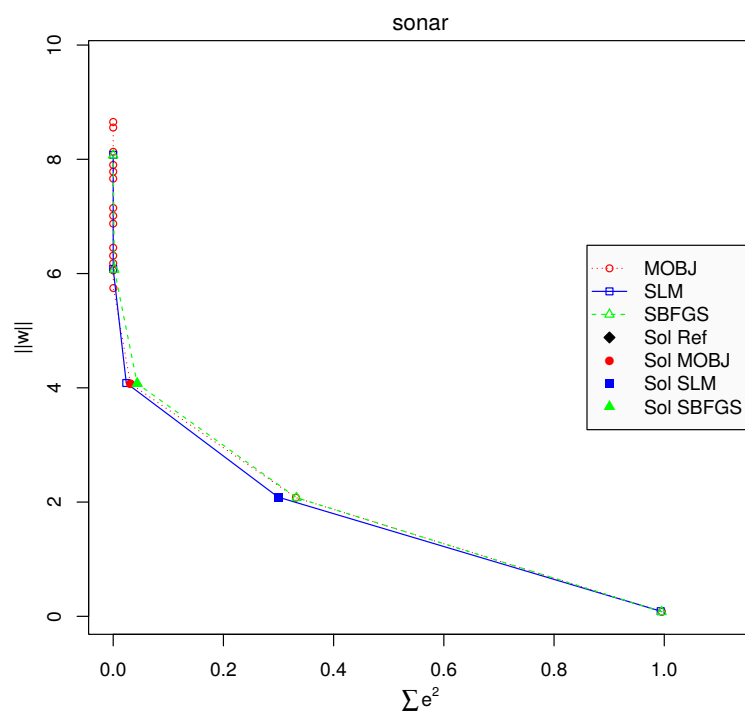


Figura 4.10: Paretos estimados - Base de dados Sonar.

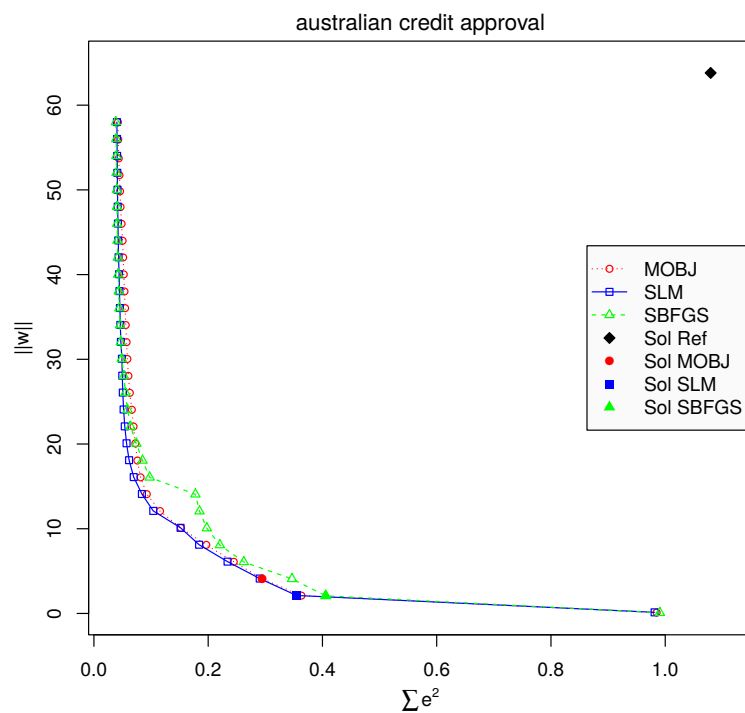


Figura 4.11: Paretos estimados - Base de dados Australian.

4.2 Experimento 1: Avaliação dos Conjuntos Pareto Estimados

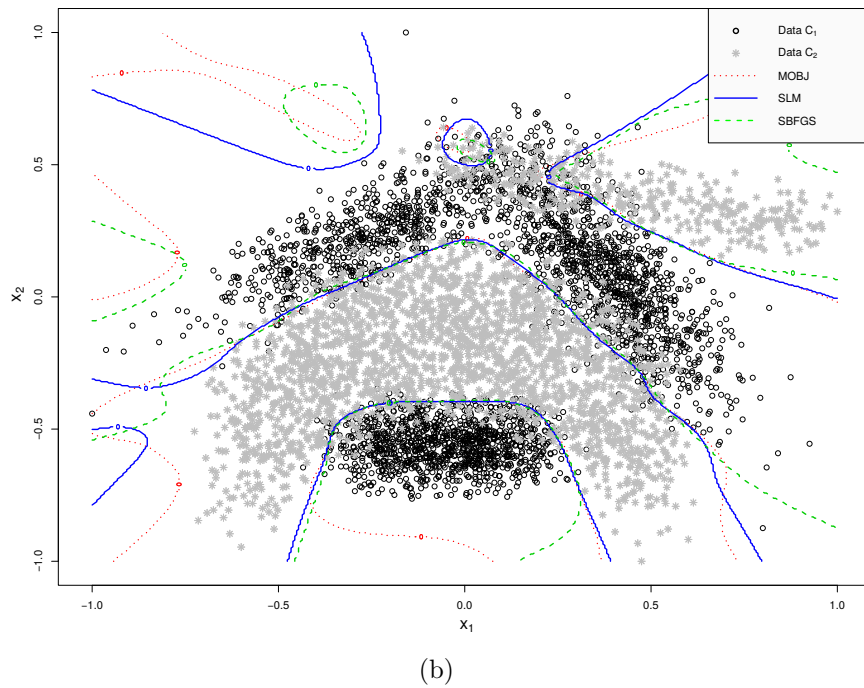
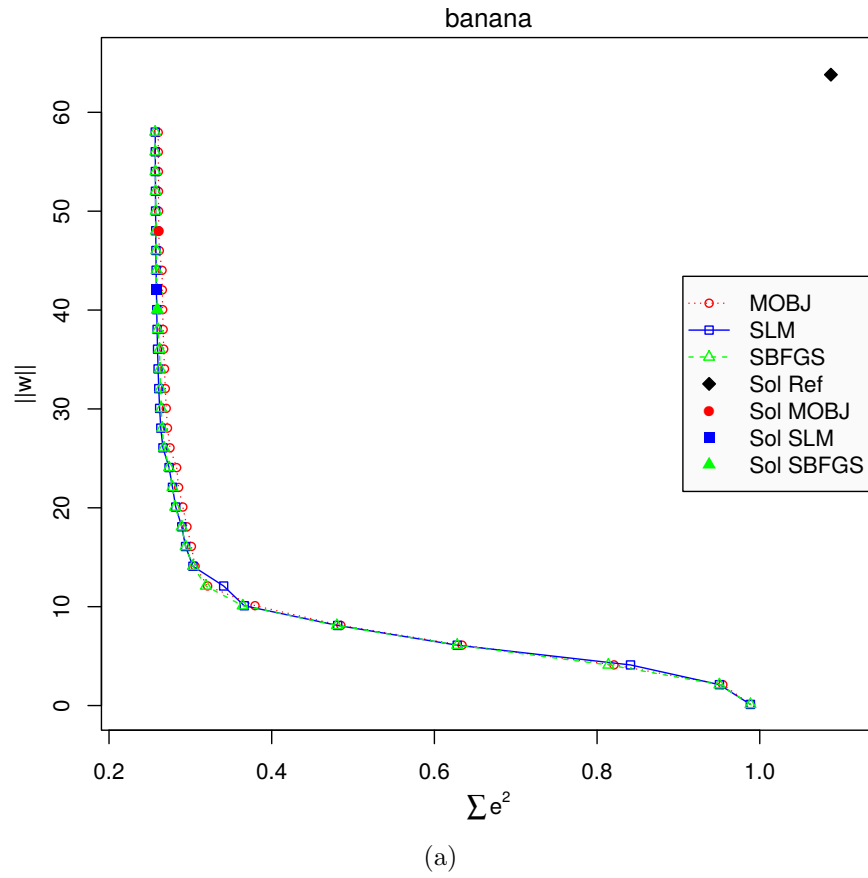


Figura 4.12: (a) Paretos estimados - Base de dados Banana. (b) Superfície de separação - Banana.

4.2.3 Análise dos Resultados com Relação a HV e CS

Conforme visto nas subseções anteriores os métodos propostos demonstram que podem obter melhores estimativas do conjunto Pareto do que o clássico algoritmo MOBJ, com ênfase para os resultados obtidos pelo método SLM. Contudo, para a maioria das bases de dados tanto de regressão quanto de classificação estas diferenças não podem ser determinadas através das figuras apresentadas, o que gera a necessidade de se utilizar um teste estatístico para verificar de forma mais precisa os resultados obtidos. Para se analisar os resultados gerais com relação aos conjuntos de soluções geradas por cada método os problemas de regressão e classificação, serão considerados em conjunto, dado que os parâmetros utilizados e a estrutura da rede foram os mesmos para ambos os tipos de problema. Desta maneira, as informações contidas nas Tabelas 4.3 e 4.5 foram consideradas de forma conjunta para representar todos os valores de HV e, de forma análoga, as Tabelas 4.4 e 4.6 foram consideradas para uma análise global de CS para os três algoritmos. Com base nos valores unificados de HV e CS , o teste estatístico *Wilcoxon signed rank test* (Wilcoxon, 1945) foi utilizado, como recomendado em (Demšar, 2006) para comparação de dois diferentes métodos de aprendizado aplicados a múltiplas bases de dados. Este teste pertence à categoria de métodos não-paramétricos para comparação de duas amostras pareadas. Os testes foram realizados comparando-se os métodos dois a dois, totalizando duas seções com três testes (S_iT_1 : MOBJ vs SLM, S_iT_2 : MOBJ vs SBFGS, S_iT_3 : SLM vs SBFGS), onde a seção S_i com $i = 1$ representa testes em relação à HV enquanto para $i = 2$ a seção considera a métrica CS . Como no caso da métrica CS existem duas vias de avaliação, considerou-se a diferença entre as coberturas mútuas dos pares de métodos. As hipóteses para realização dos testes foram formuladas da seguinte maneira:

- H_0 : Não há diferença em relação ao desempenho dos métodos. (Hipótese Nula).
- H_1 : Existe diferença em relação ao desempenho dos métodos. (Hipótese Alternativa).

Os testes foram realizados com um nível de significância igual a 5% levando-se em consideração os valores médios obtidos em todas as bases de dados.

4.2 Experimento 1: Avaliação dos Conjuntos Pareto Estimados

Aplicando-se o teste de Wilcoxon em relação aos hipervolumes médios, cujos maiores valores estão destacados em negrito nas Tabelas 4.3 e 4.5, em S_1T_1 obteve-se um p-valor igual a 0.0164, para S_1T_2 o p-valor foi igual a 0.21498 e para S_1T_3 um p-valor de 0.0001. Verificando-se o p-valor ≤ 0.05 em S_1T_1 e S_1T_3 , pode-se concluir com um nível de significância de 5% que a hipótese nula foi rejeitada em todos os casos envolvendo o método SLM, enquanto a comparação efetuada em S_1T_2 demonstra que os métodos MOBJ e SBFGS não são estatisticamente diferentes. Portanto, podemos afirmar que dentre os métodos propostos, o SLM gera estimativas do conjunto Pareto mais eficientes que o método MOBJ, considerando-se as bases de dados utilizadas e a métrica HV . Entretanto o método SBFGS não gera estimativas melhores que o MOBJ, sendo também inferior ao SLM quanto aos conjuntos obtidos.

É importante ressaltar que com relação à HV os resultados obtidos pelo teste de Wilcoxon conferem com as expectativas geradas de maneira visual através das figuras e das Tabelas 4.3 e 4.5, onde foi possível notar superioridade do método SLM, enquanto entre os métodos MOBJ e o SBFGS as diferenças foram imperceptíveis. Considerando-se o total de 30 bases de dados, na comparação realizada em S_1T_1 o método SLM foi melhor em 19 bases, em S_1T_2 tanto o MOBJ quanto o SBFGS obtiveram os melhores resultados em 15 bases, e para S_1T_3 o SLM foi melhor em 24 bases.

Em relação à métrica CS , os maiores valores também estão destacados em negrito nas Tabelas 4.4 e 4.6, neste caso utilizou-se o teste de Wilcoxon nos mesmos moldes em que o teste foi conduzido para HV . Nos três testes realizados, obteve-se um p-valor igual a $3.148 \cdot 10^{-7}$ em S_2T_1 , 0.00168 em S_2T_2 e 0.00634 em S_2T_3 , concluindo-se que para a métrica CS a hipótese nula foi rejeitada em todos os casos. Portanto, podemos afirmar que os métodos propostos (SLM e SBFGS) obtiveram os melhores resultados quanto à métrica CS , demonstrando que os algoritmos propostos cobrem os conjuntos de soluções gerados pelo MOBJ na maioria das bases de dados, com diferença estatisticamente significativa. A evidência numérica apresentada é coerente com a expectativa de que os métodos propostos seriam mais eficientes quanto à estimativa do conjunto Pareto, dado que percorrem um espaço de busca menor quando comparados ao MOBJ, devido à inicialização da busca nas proximidades de um ótimo local em cada subproblema de otimização. Ressalta-se que o método SLM obteve os melhores resultados em

4.2 Experimento 1: Avaliação dos Conjuntos Pareto Estimados

todos os testes. Esta informação pode ser observada através dos valores destacados nas Tabelas 4.4 e 4.6 para as 30 bases de dados, onde para o teste S_2T_1 o SLM foi melhor que o MOBJ em 27 bases, no teste S_2T_2 verifica-se que o SBFGS obteve melhores resultados em 24 bases e para S_2T_3 o SLM foi melhor em 22 bases.

Como informação adicional as Tabelas 4.7 e 4.8 mostram o tempo médio de execução, em segundos, e o respectivo desvio padrão, para todos os métodos aplicados aos problemas de teste. Os valores em negrito representam os menores tempos. É possível observar que em 18 das 30 bases de dados os método SBFGS obteve os menores tempos computacionais, enquanto o MOBJ obteve os menores tempos em 10 bases. Este resultado era esperado devido à maneira eficiente do método SBFGS para o cálculo de informação de segunda ordem utilizado para obtenção da direção de busca. Embora tenha sido o mais eficiente em relação às estimativas do conjunto Pareto o método SLM obteve pior desempenho quanto ao tempo de execução, devido ao custo de se computar as inversões de matrizes durante o processo de otimização. Embora o SLM tenha obtido o menor tempo em apenas 2 bases de dados, este resultado sobe para 7 quando compara-se o SLM apenas com o MOBJ, além de se aproximar em algumas outras bases.

Tabela 4.7: Tempo médio de execução em segundos para os problemas de Regressão

	MOBJ	SLM	SBFGS
f_1	74.32±4.05	66.49±17.9	37.26±5.41
f_2	79.49±4.73	80.01±22.9	32.27±20.62
f_3	43.06±5.17	35.37±7.09	30.83±11.17
f_4	78.89±1.46	88.14±18.28	41.73±5.94
f_5	44.48±2.93	32.08±7.25	19.87±5.45
srv	165.53±12.57	197.55±43.4	138.44±282.74
mpg	553.56±64.95	1533.33±260.83	469.77±109.69
bos	871.29±48.51	2981.9±631.7	4050.99±588.93
air	1209.72±67.81	1720.09±307.43	289.34±224.01
frt	685±144.89	1852.86±423.65	1047.69±1537.82
con	135.65±1.08	273.81±89.76	297.36±34.06
yac	236.2±16.55	132.58±11.89	1122.61±413.6
ccs	906.91±196.1	1834.53±408.48	1052.67±111.45
mck	636.28±19.65	927.91±287.81	503.25±58.81

4.3 Experimento 2: Avaliação das Soluções Finais

Tabela 4.8: Tempo médio de execução em segundos para os problemas de Classificação

	MOBJ	SLM	SBFGS
gau	106.47±5.18	227.49±59.67	69.8±34.67
cir	106.16±1.46	79.57±13.7	69.9±83.36
moo	104.83±14.8	86±23.64	145.01±72.22
hab	172.39±2.01	340.59±101.01	102.14±27.16
dia	712.6±15.72	1714.12±273.7	311.72±267.48
son	8298.96±164.13	22123.12±979.28	27830.03±2130.33
bup	295.69±3.64	900.08±131.48	307.06±53.59
vtc	274.4±9.49	1238.57±163.37	346.83±57.12
hrt	527.09±63.73	1379.5±599.28	1238.11±179.14
btf	437.27±8.4	787.91±184.53	176.93±61.05
ban	2771.47±168.2	2013.39±546.13	143.94±17.53
aus	1148.22±33.15	5226.01±864.21	1995.04±123.93
wbc	5383.22±64.02	8769.11±4274.84	659.96±1252.81
bak	664.23±14.7	1186.26±174.86	536.29±108.86
fer	204.55±3.68	303.84±156.83	200.34±180.8
ilp	752.26±31.47	4308.08±1628.64	1075.83±132.88

4.3 Experimento 2: Avaliação das Soluções Finais

4.3.1 Descrição do Experimento

Esta seção descreve o experimento realizado para avaliação das soluções finais obtidas pelos métodos de aprendizado multiobjetivo MOBJ e SLM. Estes dois algoritmos foram definidos como representantes do aprendizado multiobjetivo no experimento, sendo que o primeiro é o algoritmo de referência devido ao pioneirismo da abordagem enquanto o segundo foi o que obteve os melhores resultados no Experimento 1, dentre os algoritmos desenvolvidos neste trabalho. As soluções finais obtidas pelos algoritmos MOBJ e SLM são comparadas às encontradas por algoritmos do estado da arte, previamente utilizados no trabalho desenvolvido em (Fernández-Delgado *et al.*, 2014), para que se possa ter uma visão geral do desempenho dos métodos desenvolvidos. Os métodos descritos em (Fernández-

4.3 Experimento 2: Avaliação das Soluções Finais

Delgado *et al.*, 2014) utilizados neste experimento são: *Random Forest* (RF), *Support Vector Machine* com funções de *kernel* radial (SVM-R), *Support Vector Machine* com funções de *kernel* polinomiais (SVM-P), rede *Multi-layer Perceptron* com algoritmo *Weight Decay* (MLP-WD) e *Gradient Boosting Machine* (GBM). Alguns destes métodos estão entre os 10 melhores dentre os 179 algoritmos avaliados em 121 bases de dados (Fernández-Delgado *et al.*, 2014), além disso são aplicáveis tanto a problemas de classificação quanto de regressão. Para o ajuste de parâmetros destes algoritmos, foi adotado um procedimento de busca em *grid*, através de um procedimento *10-fold cross-validation* utilizando o pacote *Caret* (Kuhn, 2012) do software R (R Core Team, 2016).

4.3.2 Teste de Significância para Comparação entre Várias Máquinas de Aprendizado

Diferente do Experimento 1, onde os métodos eram comparados dois a dois, neste experimento 7 algoritmos serão comparados de maneira conjunta o que gera a necessidade de um teste estatístico mais apropriado nesta situação. Como recomendado em (Demšar, 2006), o desempenho global dos 7 métodos envolvidos neste experimento foi avaliado pela utilização do teste estatístico de Friedman, que assume a hipótese nula H_0 de que todos os métodos são equivalentes. O teste de Friedman verifica se há diferenças entre os métodos analisados através da geração de um *rank*, que neste experimento, consiste na ordenação dos métodos avaliados com relação às funções de desempenho: minimização do erro médio quadrático (**mse**) para problemas de regressão e maximização da acurácia (**acc**) em problemas de classificação. Sob H_0 , a estatística do teste de Friedman

$$F_F = \frac{(N-1)X_F^2}{N(k-1) - X_F^2} \quad (4.3)$$

segue a distribuição F de *Fisher-Snedecor* com $(k-1)$ e $(k-1)(N-1)$ graus de liberdade, onde N é a quantidade de bases de dados e k é a quantidade de algoritmos analisados. X_F^2 é definido como:

$$X_F^2 = \frac{12N}{k(k+1)} \left[\sum_{j=1}^k R_j^2 - \frac{k(k+1)^2}{4} \right] \quad (4.4)$$

4.3 Experimento 2: Avaliação das Soluções Finais

onde R_j é a média do *rank* para o algoritmo j .

Este teste tem sido utilizado em diversos trabalhos onde vários métodos são aplicados a várias bases de dados (Castro, 2011; Kokshenev, 2010; Silvestre, 2015; Torres, 2016). Entretanto o teste não aponta quais métodos são diferentes. Para realizar esta análise o teste *post-hoc* Bonferroni-Dunn é utilizado neste trabalho, indicando que dois métodos apresentam diferenças significativas se os valores de *rank* médio correspondentes aos métodos diferem de pelo menos uma diferença crítica (em inglês, *critical difference* - *CD*). O diagrama de diferenças críticas (em inglês, *critical difference diagram* - *CDD*) (Demšar, 2006) será utilizado como ferramenta gráfica para visualizar as diferenças, tomando-se o método SLM como base. Maiores detalhes acerca do teste de Friedman e do teste *post-hoc* Bonferroni-Dunn podem ser encontrados em (Demšar, 2006).

4.3.3 Resultados para as Bases de Regressão e Classificação

A Tabela 4.9 mostra os valores médios do erro médio quadrático (**mse**) para as bases de dados de regressão. Estes valores representam a média do **mse** para os 10-*folds* e o respectivo desvio padrão. Para os problemas de classificação, a Tabela 4.10 mostra os valores médios de acurácia para os conjuntos de teste além do respectivo desvio padrão. O método com o melhor desempenho para cada base de dados é mostrado em negrito. O valor do *rank* médio para cada método é mostrado na parte inferior das Tabelas 4.9 e 4.10, para problemas de regressão e classificação, respectivamente.

Considerando-se os problemas de regressão, o valor crítico do teste de Friedman (F_c), assumindo $\alpha = 0.05$, é $F_c(6; 78) = 2.217$. O p-valor obtido é 0.00063 e $F_F = 7.980$, o que indica, dado que $F_F > F(6, 78)$, que a hipótese nula H_0 foi rejeitada. Desta forma, para verificar se os algoritmos avaliados são diferentes do algoritmo SLM, deve-se proceder com o teste *post-hoc*, que indicou $CD = 2.154$, mostrando que há diferenças significativas entre o SLM e os métodos SVM-P e MLP-WD. A Figura 4.13(a) mostra o diagrama de diferenças críticas para problemas de regressão. O valor obtido de CD é mostrado como uma linha horizontal em negrito. Pode-se ver que os métodos SLM, MOBJ, RF e GBM obtiveram valores de *rank* médio estatisticamente similares, enquanto os métodos MLP-WD

4.3 Experimento 2: Avaliação das Soluções Finais

Tabela 4.9: Média e desvio padrão do erro médio quadrático (MSE) para os problemas sintéticos e reais de regressão

	RF	SVM-R	SVM-P	MLP-WD	GBM	MOBJ	SLM
f_1	0.052±0.023	0.050±0.022	0.281±0.118	0.115±0.086	0.082±0.035	0.045±0.017	0.043±0.018
f_2	0.121±0.172	0.139±0.239	1.259±1.532	0.099±0.073	0.109±0.174	0.067±0.055	0.064±0.054
f_3	0.049±0.026	0.048±0.030	0.087±0.045	0.062±0.032	0.061±0.036	0.042±0.025	0.047±0.025
f_4	0.065±0.032	0.053±0.029	0.525±0.154	0.166±0.092	0.183±0.157	0.058±0.032	0.057±0.084
f_5	0.054±0.016	0.041±0.011	0.063±0.027	0.055±0.020	0.041±0.014	0.039±0.011	0.041±0.012
srv	0.020±0.014	0.042±0.049	0.046±0.042	0.046±0.036	0.027±0.029	0.036±0.042	0.041±0.045
mpg	0.283±0.074	0.311±0.070	0.314±0.053	0.312±0.071	0.291±0.061	0.309±0.066	0.305±0.065
bos	0.021±0.009	0.030±0.017	0.021±0.013	0.033±0.020	0.022±0.012	0.037±0.025	0.028±0.024
air	0.009±0.002	0.027±0.003	0.038±0.004	0.041±0.005	0.020±0.003	0.013±0.004	0.011±0.002
frt	0.014±0.024	0.014±0.025	0.014±0.025	0.015±0.024	0.014±0.025	0.014±0.024	0.014±0.024
con	0.265±0.199	0.254±0.159	0.243±0.158	0.214±0.130	0.283±0.151	0.210±0.160	0.217±0.134
yac	0.001±0.000	0.042±0.019	0.011±0.005	0.001±0.000	0.001±0.000	0.000±0.000	0.000±0.000
ccs	0.014±0.004	0.027±0.005	0.026±0.006	0.028±0.005	0.016±0.003	0.017±0.003	0.019±0.004
mck	0.020±0.006	0.044±0.007	0.054±0.007	0.054±0.003	0.041±0.004	0.019±0.006	0.018±0.006
avg.rank	3.143	4.464	5.786	5.571	3.893	2.643	2.500

Tabela 4.10: Média e desvio padrão da acurácia (ACC) para os problemas sintéticos e reais de classificação

	RF	SVM-R	SVM-P	MLP-WD	GBM	MOBJ	SLM
gau	87.50±7.2	87.50±7.2	72.00±8.9	77.50±8.4	88.00±6.7	84.50±8.6	84.50±7.6
cir	100.00±0.0	100.00±0.0	100.00±0.0	85.50±3.9	100.00±0.0	100.00±0.0	100.00±0.0
moo	95.50±6.0	100.00±0.0	93.50±6.1	94.00±5.9	96.50±6.3	100.00±0.0	100.00±0.0
hab	69.91±5.6	73.16±7.6	73.52±7.4	72.52±7.5	74.14±7.7	71.54±8.5	75.14±8.7
dia	76.84±4.7	76.31±6.3	76.57±5.3	76.31±5.3	76.05±4.7	76.83±6.5	77.22±6.0
son	83.19±6.8	84.10±11.7	87.10±10.6	84.17±6.4	84.19±5.6	80.79±11.1	86.02±7.4
bup	75.10±8.9	67.86±6.5	67.85±7.5	70.47±7.9	73.10±11.7	69.00±6.8	69.61±8.9
vtc	84.84±8.3	85.81±7.0	77.74±7.1	83.87±7.0	84.19±4.9	83.87±7.8	85.81±6.8
hrt	83.70±7.0	83.70±8.2	84.07±8.9	82.22±8.7	83.70±10.8	82.96±7.3	83.33±8.1
btf	75.12±6.9	79.01±6.8	76.87±6.7	76.20±6.7	77.94±7.3	77.54±5.8	79.01±6.5
ban	88.94±1.4	90.45±1.6	70.08±4.1	75.79±4.2	88.53±1.3	90.21±1.6	90.47±1.7
aus	86.09±5.0	85.36±5.3	85.80±4.3	86.96±4.9	86.67±4.4	86.67±4.5	86.09±5.3
wbc	95.78±2.1	96.83±2.6	94.90±2.9	97.72±1.2	96.13±2.2	97.19±1.5	97.36±1.5
bak	99.34±0.5	99.64±0.4	98.54±1.1	99.13±0.7	99.49±0.5	99.85±0.5	100.00±0.0
fer	87.00±4.8	88.00±6.3	87.00±6.3	88.00±6.3	87.00±8.2	87.00±6.7	87.00±6.7
ilp	70.63±6.3	71.15±6.9	71.14±6.8	71.49±6.3	70.28±5.7	69.94±5.7	70.80±6.4
avg.rank	4.344	3.375	5.031	4.469	3.750	4.313	2.719

e SVM-P obtiveram valores estatisticamente diferentes do SLM.

Para os problemas de classificação, o valor crítico $F(6, 90) = 2.201$ para $\alpha = 0.05$. O p-valor obtido é 0.02671 e $F_F = 2.209$, o que mostra que a hipótese nula H_0 foi novamente rejeitada dado que $F_F > F(6, 90)$. Seguindo com o teste *post-hoc* o valor fornecido para a diferença crítica é $CD = 2.015$, o que possibi-

4.3 Experimento 2: Avaliação das Soluções Finais

lita verificar que somente SLM e SVM-P são significativamente diferentes, como mostra a Figura 4.13(b).

É importante ressaltar que, embora não haja diferença estatística significativa com relação à maioria dos métodos, o método desenvolvido neste trabalho (SLM), obteve o menor valor de *rank* médio tanto para os problemas de regressão quanto para os de classificação. Assim, em geral, o SLM obteve os melhores níveis de *rank* em comparação com os métodos avaliados. Para os problemas de regressão, o MOBJ obteve o valor de *rank* médio mais próximo do SLM. No entanto, para os problemas de classificação, o método SVM-R obteve o *rank* médio mais próximo do SLM, seguido por GBM e MOBJ.

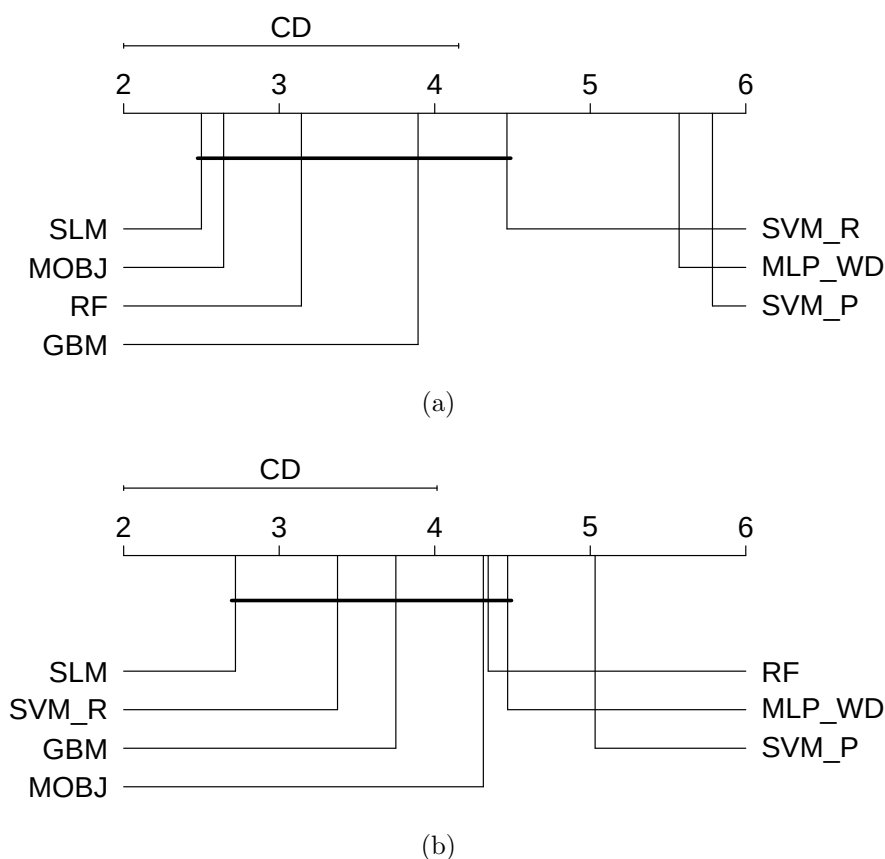


Figura 4.13: Diagrama CDD. O valor CD é a linha sobre o gráfico, o eixo mostra os *rank*s médios com o menor (melhor) valor à esquerda. Os grupos ligados não são significativamente diferentes ($\alpha = 0.05$). (a) CDD para problemas de regressão. (b) CDD para problemas de classificação.

É importante observar que os métodos de treinamento multiobjetivo utilizaram o decisor de validação para seleção da solução no Pareto estimado, o que pode ter contribuído para a escolha de soluções similares mesmo após os resultados mostrarem que o método SLM gera as melhores estimativas do conjunto Pareto, dentre os métodos analisados no Experimento 1.

4.4 Conclusão do Capítulo

Neste capítulo, foi realizado um estudo para avaliação dos algoritmos esféricos propostos na tese quanto à eficiência na geração de uma estimativa do conjunto Pareto e a consequente escolha de uma solução, tanto em problemas de regressão quanto em problemas de classificação. Foi possível verificar que dentre os métodos propostos o SLM obteve os melhores resultados em relação à eficiência para se estimar soluções não-dominadas, obtendo os melhores valores para as métricas de desempenho HV e CS nas comparações com o clássico método MOBJ e a versão esférica do método BFGS (SBFGS). Foi mostrado que o método SBFGS não gerou conjuntos de soluções tão bons quanto aqueles gerados pelo SLM e não se mostrou melhor que o método MOBJ considerando-se o indicador HV . Porém o SBFGS se mostrou superior ao MOBJ com significância estatística, quando a métrica de avaliação em questão foi CS .

Os resultados também indicaram que a solução final selecionada para o método SLM supera métodos tradicionais como SVM-R para os problemas de regressão e SVM-P para os problemas de regressão e classificação. Além disso o SLM se mostrou estatisticamente equivalente a alguns dos principais métodos da literatura, com eficiência comprovada em estudos recentes (Fernández-Delgado *et al.*, 2014). Adicionalmente, embora não se tenha comprovado diferença com significância estatística entre o SLM e o MOBJ para a solução final, o SLM obteve os melhores valores de *rank* médio tanto para regressão quanto para classificação, o que sugere que ele pode gerar modelos com desempenho competitivo com a literatura, mesmo utilizando o decisor de validação para seleção da solução final, sendo este um decisor com limitações conhecidas.

Capítulo 5

Conclusões e Trabalhos Futuros

Esta tese de doutorado apresentou uma nova representação para os parâmetros de redes MLP que simplifica o processo de otimização para o aprendizado multiobjetivo. A formulação esférica proposta para representação dos pesos da rede transformam a forma tradicional de resolução do problema de aprendizado multiobjetivo normalmente baseado em otimização restrita (Costa *et al.*, 2007; Teixeira *et al.*, 2000), de maneira que o problema de otimização com valor fixo de norma é resolvido como um problema mono-objetivo irrestrito, e consequentemente, resolvendo-se várias instâncias deste problema para diferentes valores fixos de norma, pode-se gerar uma estimativa do conjunto Pareto-ótimo para o problema.

Os conceitos necessários ao embasamento da representação proposta foram apresentados no Capítulo 2, onde o aparato que fundamenta o aprendizado multiobjetivo de redes neurais foi apresentado. Foram definidos desde conceitos essenciais referentes ao aprendizado estatístico até a formulação multiobjetivo do algoritmo de referência na literatura (Teixeira *et al.*, 2000).

As contribuições desta tese foram apresentadas no Capítulo 3, onde a generalização da formulação por coordenadas esféricas foi detalhada definindo-se as possibilidades de aplicação desde a um simples neurônio não-linear até uma rede MLP. Foi mostrado que a representação proposta restringe o espaço de busca sobre a superfície de uma hiperesfera $w_1^2 + w_2^2 + \dots + w_n^2 = r_j^2$ dada a definição dos pesos em função do valor fixo de norma representado por r_j , como mostra o exemplo a seguir para uma rede com cinco pesos:

$$\begin{aligned}
w_1 &= r_j \sin \theta_1 \\
w_2 &= r_j \cos \theta_1 \sin \theta_2 \\
w_3 &= r_j \cos \theta_1 \cos \theta_2 \sin \theta_3 \\
w_4 &= r_j \cos \theta_1 \cos \theta_2 \cos \theta_3 \sin \theta_4 \\
w_5 &= r_j \cos \theta_1 \cos \theta_2 \cos \theta_3 \cos \theta_4
\end{aligned} \tag{5.1}$$

Ao definir os pesos desta maneira o problema de aprendizado multiobjetivo tratado da maneira clássica (Teixeira *et al.*, 2000):

$$\begin{aligned}
&\arg \min_{\mathbf{w}} J_1 \\
&s.a : J_2 \leq \epsilon
\end{aligned} \tag{5.2}$$

onde as funções J_1 e J_2 são definidas como:

$$J_1 = \frac{1}{2} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \tag{5.3}$$

$$J_2 = \sqrt{\sum_{i=1}^{n_w} w_i^2} \tag{5.4}$$

passa a ser tratado de forma simplificada, onde minimiza-se o erro da rede sem a necessidade do tratamento da restrição de norma:

$$\arg \min_{\theta_1 \dots \theta_{n_w-1}} J_1 \tag{5.5}$$

Esta forma de tratamento do problema simplifica a formulação matemática além de eliminar o esforço de se lidar com o tratamento de restrições, necessário no processo de otimização definido pela formulação clássica.

O Capítulo 3 ainda definiu o algoritmo de aprendizado multiobjetivo onde é mostrado que a representação proposta fornece um mecanismo interessante como estratégia para geração de soluções. De maneira natural pode-se aproveitar a direção de busca obtida por cada passo anterior no algoritmo, onde a solução obtida após a resolução do subproblema para um valor particular de r_j é utilizada como solução inicial para resolução do problema seguinte para r_{j+1} , o que

defini o processo de *warm start* (Friedman *et al.*, 2010). Finalizando as contribuições deste trabalho foram desenvolvidos três algoritmos para o treinamento da rede com a representação esférica. O primeiro algoritmo é a versão esférica do clássico *Gradiente Descendente*, base do treinamento de redes por meio do *Back-propagation* (Rumelhart & McClelland, 1986). Este algoritmo foi utilizado como um modelo teórico para a definição dos principais recursos para geração da direção de busca no espaço esférico para os algoritmos determinísticos implementados. Os outros dois algoritmos desenvolvidos consistem em versões esféricas dos algoritmos *Levenberg-Marquardt* (LM) e *BFGS*.

Os resultados obtidos através dos experimentos realizados no Capítulo 4 demonstraram que as soluções estimadas pelos métodos desenvolvidos cobrem aquelas encontrados pelo algoritmo MOBJ, com ênfase para o método SLM que superou com significância estatística o MOBJ com relação às duas métricas utilizadas, *HV* e *CS*. Portanto, foi possível verificar que o método proposto estima soluções com maior refinamento que o método clássico. Com relação à solução final obtida, foi mostrado que o método SLM é competitivo com alguns métodos do estado da arte da literatura, sendo que ele obteve os melhores resultados com relação ao *rank* médio, no entanto, com significância estatística apenas com relação a dois dentre os seis métodos comparados.

De maneira geral os objetivos desta tese foram alcançados, dado que a formulação esférica foi desenvolvida e os algoritmos propostos se mostraram mais eficientes que o método de referência (MOBJ) com relação à capacidade de gerar estimativas melhoradas do conjunto Pareto, além de se mostrarem competitivos com alguns dos principais métodos da literatura quando a solução final foi levada em consideração. O estudo realizado neste trabalho ainda mostrou que o fato de se lidar com ângulos ao invés de pesos escalares traz novas perspectivas para representação e aprendizado em redes neurais, por apresentar uma modelagem intuitiva e simplificada de redes para resolução do problema de aprendizado multiobjetivo. As facilidades oriundas da representação esférica demonstraram que com simples modificações podem ser criados novos algoritmos para o treinamento multiobjetivo, e ainda levam a crer que esta formulação pode ser facilmente aplicada em outros métodos de aprendizado além das redes neurais.

Na seção a seguir é feita uma análise acerca das dificuldades dos algoritmos propostos, de maneira conjunta à proposição de trabalhos futuros com o objetivo

de sanar os problemas elencados.

5.1 Propostas de Trabalhos Futuros

Com foco na melhoria dos métodos propostos bem como na minimização de algumas limitações destes, os itens a seguir expressam algumas sugestões de trabalhos futuros como possível continuidade desta pesquisa.

- Os resultados encontrados a partir dos experimentos desenvolvidos nesta tese mostraram que em relação à eficiência da solução escolhida no Pareto estimado, os métodos desenvolvidos não melhoram significativamente em relação ao MOBJ. Entretanto foi utilizado apenas um decisor nos experimentos, que seleciona a solução com menor erro de validação. A utilização de outros decisores pode contribuir para uma melhor avaliação da etapa de seleção do modelo. Testes iniciais indicam que a utilização do ruído obtido por cada solução estimada do conjunto Pareto pode consistir em um recurso interessante para o desenvolvimento de um novo decisor, de maneira que uma parte do conjunto de dados não precisaria ser utilizada durante o treinamento da rede, contribuindo assim para o melhor ajuste dos modelos gerados devido ao aumento da representatividade do conjunto de dados. Entretanto inúmeros testes com diferentes tipos de bases de dados são necessários para verificação destes argumentos.
- Outros algoritmos podem ser desenvolvidos para explorar a formulação esférica, visando o alcance de melhores níveis quanto à convergência para o conjunto Pareto e menores níveis de complexidade computacional. Considerando-se o quesito convergência, melhorias poderiam ser obtidas através da utilização de algoritmos evolutivos, onde resultados iniciais mostram eficiência desta abordagem (Rocha *et al.*, 2013), embora com maior custo computacional. Com relação ao aspecto complexidade computacional, foi possível perceber através dos experimentos que o método SLM proposto gerou as melhores estimativas do conjunto Pareto, mas com custo computacional superior aos demais avaliados. Para obtenção de melhorias quanto a este aspecto poderiam ser utilizadas versões esféricas de outros algoritmos

5.1 Propostas de Trabalhos Futuros

clássicos onde este quesito é priorizado, como é o caso dos algoritmos *Resilient Back-Propagation - RPROP* (Riedmiller & Braun, 1992) e *Quikprop* (Fahlman, 1988).

- Poderia-se explorar uma implementação paralela dos algoritmos desenvolvidos, uma vez que cada solução estimada não precisa necessariamente ser obtida de forma sequencial. Estratégias poderiam ser estudadas a fim de se utilizar uma implementação paralela com benefícios semelhantes ao processo de *warm start*.

Apêndice A

Conjuntos Pareto Estimados - Bases de dados Reais

A.1 Problemas de Regressão

As Figuras de [A.1](#) a [A.9](#) ilustram as estimativas do conjunto Pareto obtidas para as bases de dados reais de regressão. Cada conjunto representado bem como a solução selecionada presente nas Figuras representam o experimento cujo fold gerou maior valor da métrica HV .

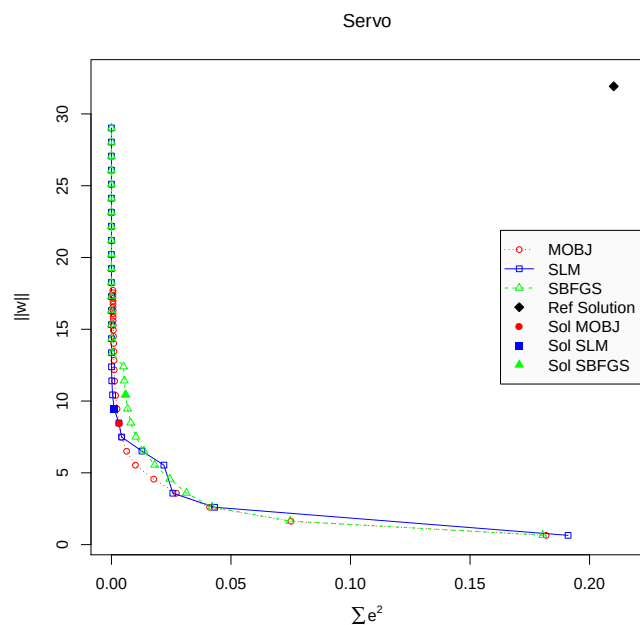


Figura A.1: Paretos estimados - Base de dados Servo.

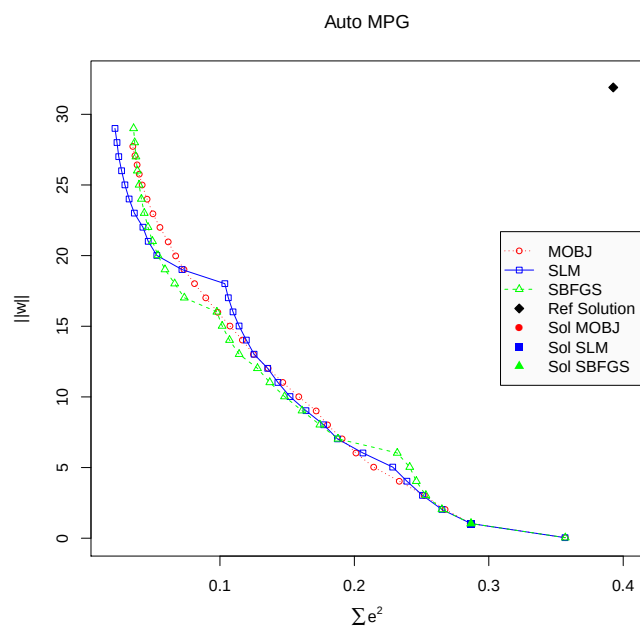


Figura A.2: Paretos estimados - Base de dados Auto MPG.

A.1 Problemas de Regressão

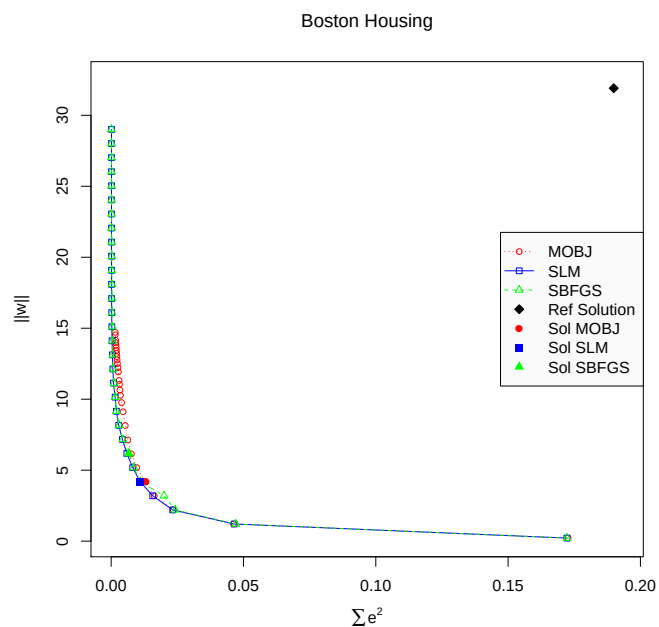


Figura A.3: Paretos estimados - Base de dados Boston Housing.

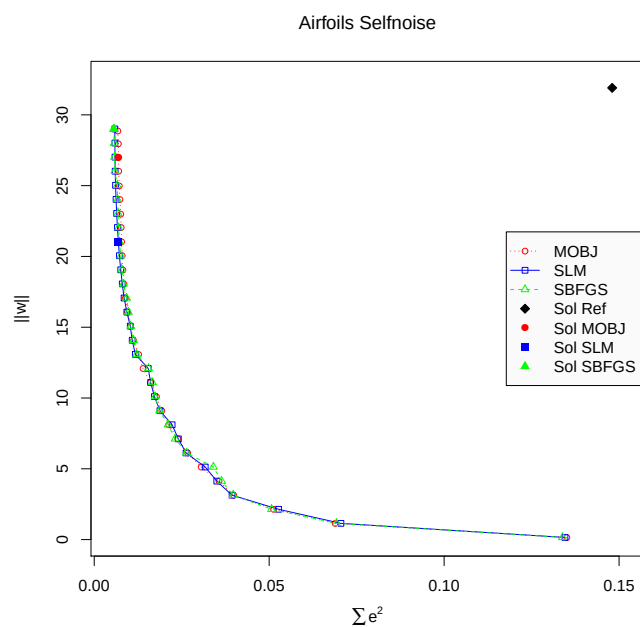


Figura A.4: Paretos estimados - Base de dados Airfoils Selfnoise.

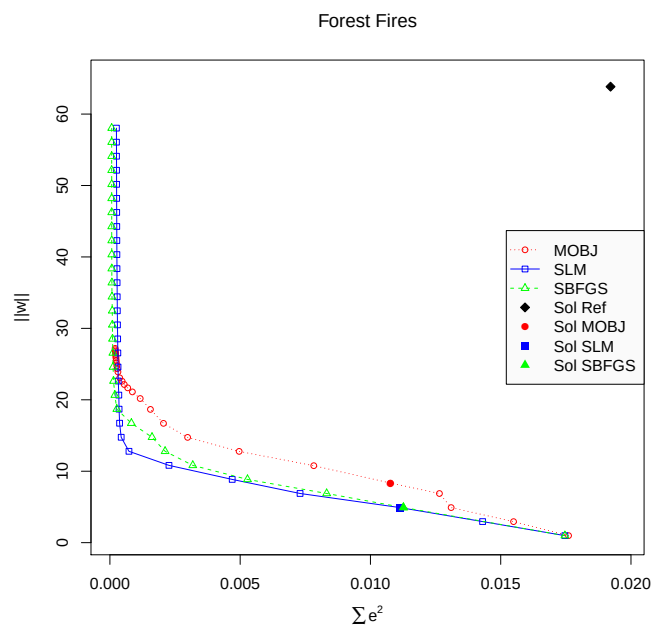


Figura A.5: Paretos estimados - Base de dados Forest Fires.

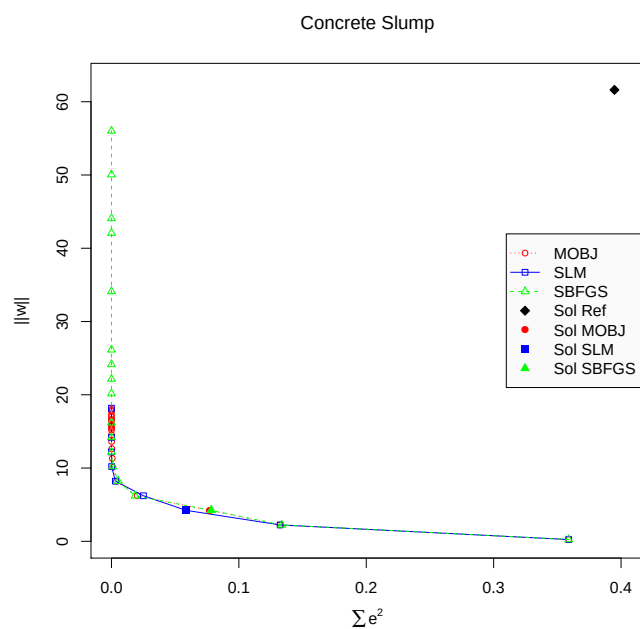


Figura A.6: Paretos estimados - Base de dados Concrete Slump.

A.1 Problemas de Regressão

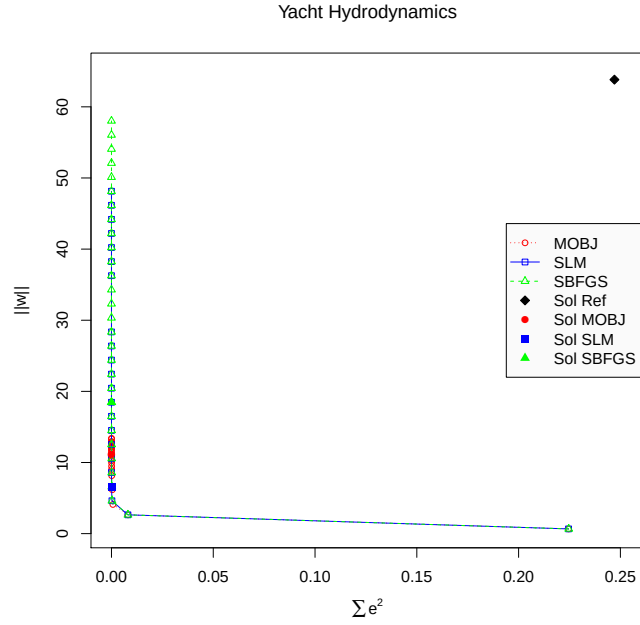


Figura A.7: Paretos estimados - Base de dados Yacht Hydrodynamics.

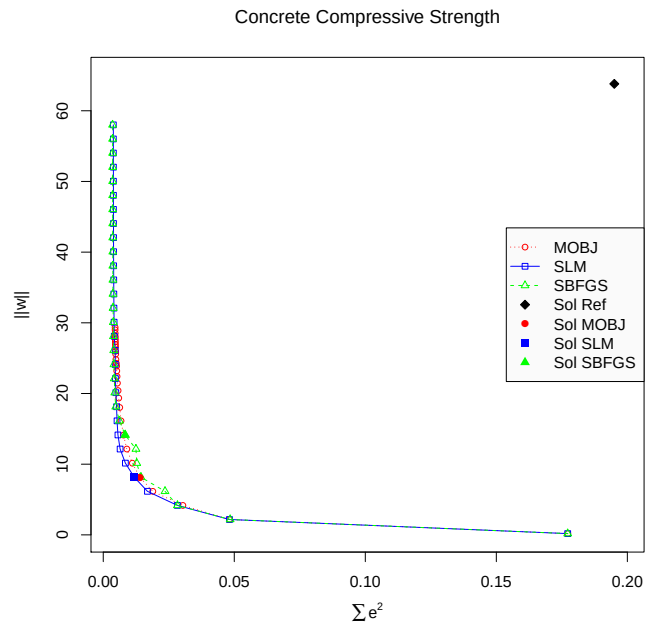


Figura A.8: Paretos estimados - Base de dados Concrete Comp. Strength.

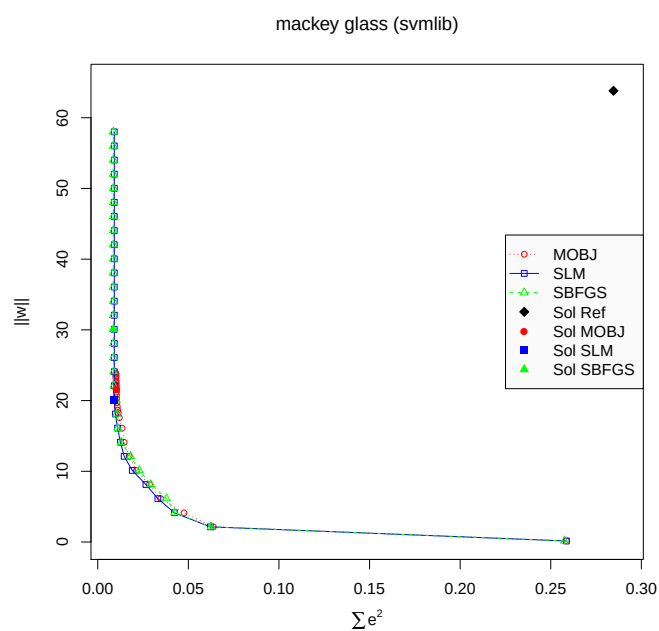


Figura A.9: Paretos estimados - Base de dados Mackey-glass.

A.2 Problemas de Classificação

As Figuras de A.10 a A.18 ilustram as estimativas do conjunto Pareto obtidas para as bases de dados de classificação que não foram apresentadas na Seção 4.2.2. Cada conjunto representado bem como a solução selecionada representam o experimento cujo fold gerou maior valor para a métrica HV .

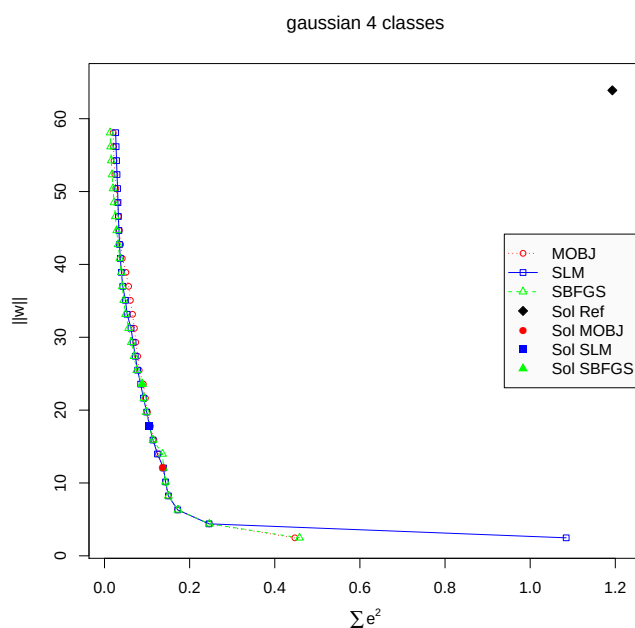


Figura A.10: Paretos estimados - Base de dados Four-Gaussian-class.

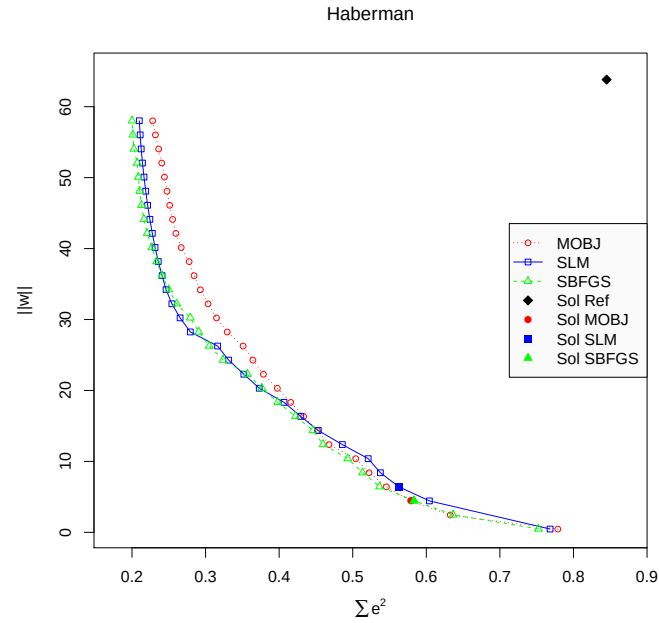


Figura A.11: Paretos estimados - Base de dados Haberman.

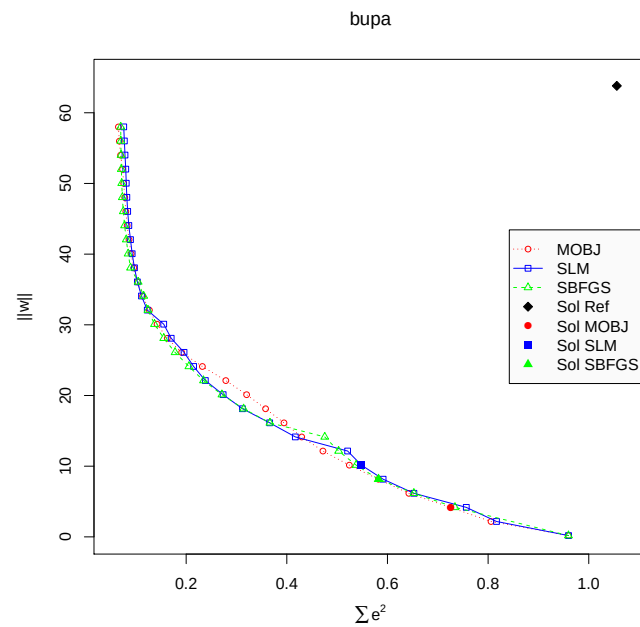


Figura A.12: Paretos estimados - Base de dados Bupa.

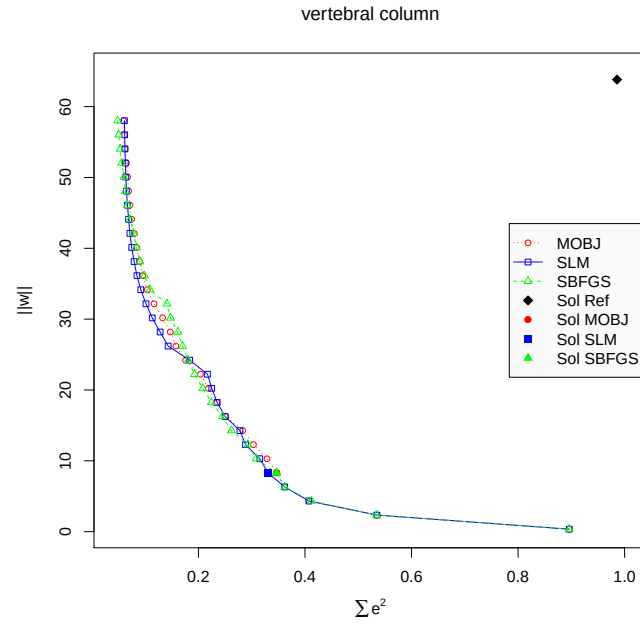


Figura A.13: Paretos estimados - Base de dados Vertebral column.

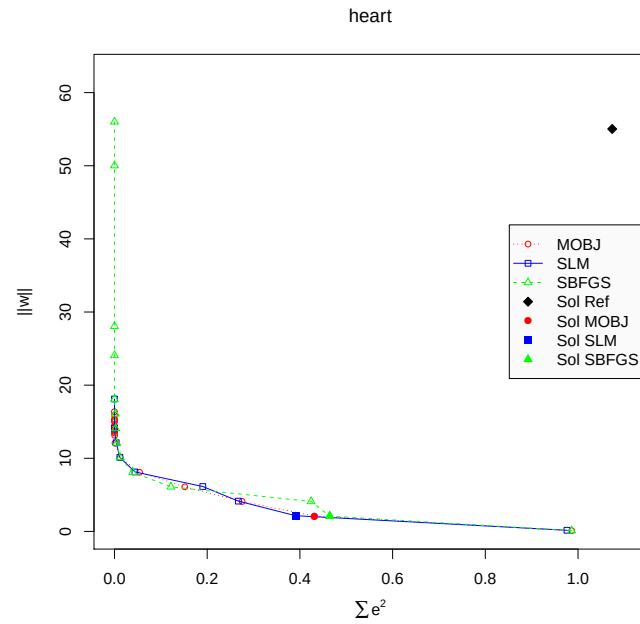


Figura A.14: Paretos estimados - Base de dados Heart diseases.

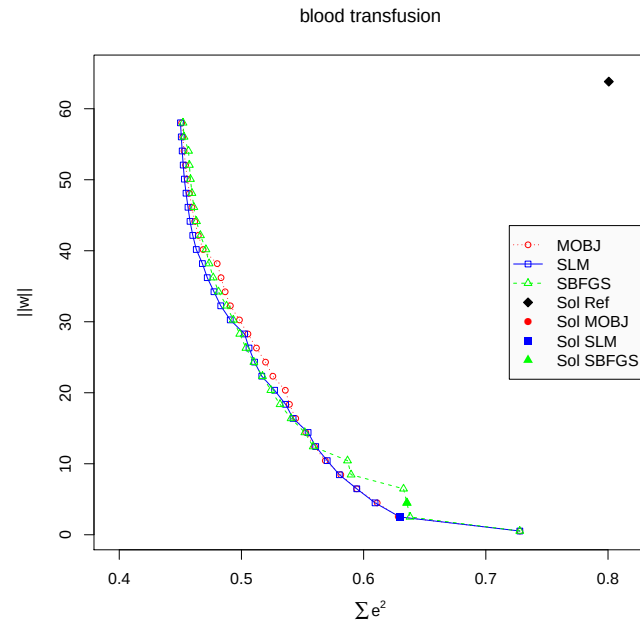


Figura A.15: Paretos estimados - Base de dados Blood transfusion.

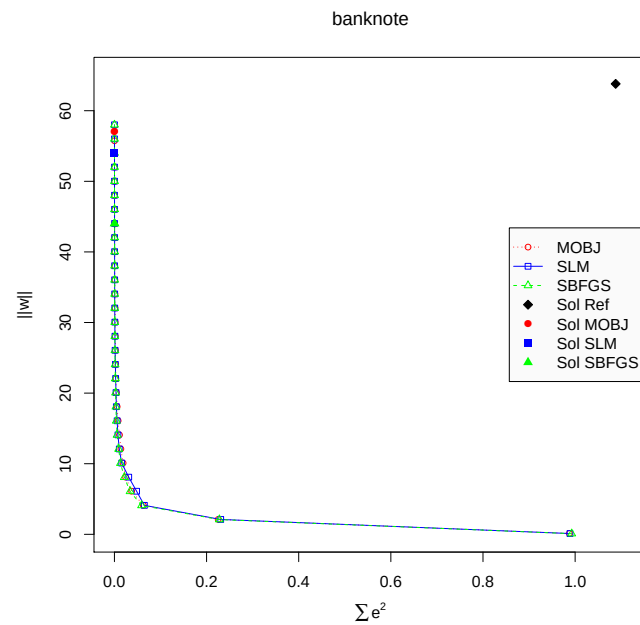


Figura A.16: Paretos estimados - Base de dados Banknote.

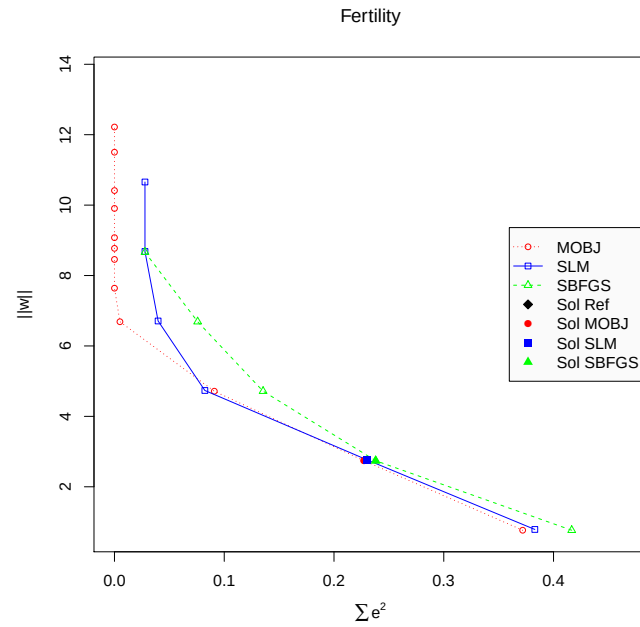


Figura A.17: Paretos estimados - Base de dados Fertility.

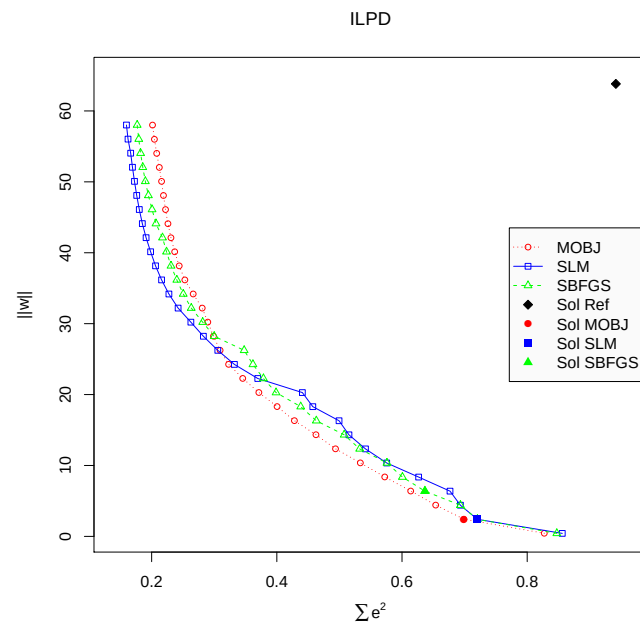


Figura A.18: Paretos estimados - Base de dados ILPD.

Referências

- ALCALÁ, J., FERNANDEZ, A., LUENGO, J., DERRAC, J., GARCÍA, S., SÁNCHEZ, L. & HERRERA, F. (2011). Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic and Soft Computing*, **17**, 255–287. [47](#)
- BARROSO, M., TAKAHASHI, R.H.C. & AGUIRRE, L. (2007). Multi-objective parameter estimation via minimal correlation criterion. *Journal of Process Control*, **17**, 321–332. [25](#)
- BARTLET, P. (1997). For valid generalization the size of the weights is more important than the size of the network. *Advances in Neural Information Processing Systems*, **9**, 134–140. [5](#), [21](#), [30](#)
- BATISTA, L.S. (2011). *Investigação de Novas Abordagens para Otimização Multiobjetivo em Algoritmos Evolutivos*. Tese de doutorado, Universidade Federal de Minas Gerais - UFMG. [52](#)
- BJÖRCK, A. (1996). *Numerical methods for least squares problems*. Siam. [39](#)
- BLAKE, C. & MERZ, C. (1998). Uci repository of machine learning databases. [47](#), [49](#)
- BOSER, B.E., GUYON, I.M. & VAPNIK, V.N. (1992). A training algorithm for optimal margin classifiers. In *COLT '92: Proceedings of the fifth annual workshop on Computational learning theory*, 144–152, ACM, New York, NY, USA. [3](#), [5](#)
- BRAGA, A.P., TAKAHASHI, R., COSTA, M.A. & TEIXEIRA, R. (2006). Multi-objective algorithms for neural networks learning. In Y. Jin, ed., *Multi-*

- Objective Machine Learning*, vol. 16 of *Studies in Computational Intelligence*, 151–171, Springer Berlin / Heidelberg. [2](#)
- BRAGA, A.P., LUDEMIR, T.B. & CARVALHO, A.C.P.F. (2007). *Redes Neurais Artificiais. Teoria e aplicações. 2.ed.* LTC Editora, Rio de Janeiro. [1](#)
- BRANKE, J., DEB, K., MIETTINEN., K. & SLOWINSKI, R. (2008). *Multiobjective Optimization - Interactives and Evolutionary Approaches..* Springer. [19](#)
- BREIMAN, L. (1996). Bagging predictors. *Machine Learning*, **24**, 123–140. [3](#)
- BROYDEN, C.G. (1970). The convergence of a class of double-rank minimization algorithms 1. general considerations. *IMA Journal of Applied Mathematics*, **6**, 76–90. [40](#)
- CASTRO, C.L. (2011). *Novos critérios para seleção de modelos neurais em problemas de classificação com dados desbalanceados.* Tese de doutorado, UFMG. [73](#)
- CHANG, C.C. & LIN, C.J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, **2**, 27:1–27:27, software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>. [47](#)
- CHANKONG, V. & HAIMES, Y. (1983). *Multiobjective Decision Making: Theory and Methodology.* Elsevier Science, New York, NY. [3](#), [21](#), [22](#)
- CHARALAMBOUS, C. (1992). Conjugate gradient algorithm for efficient training of artificial neural networks. *Proc. Inst. Elect. Eng. G*, **139**, 301 – 310. [13](#)
- CORTES, C. & VAPNIK, V. (1995). Support-vector networks. *Mach. Learn.*, **20**, 273–297. [3](#)
- COSTA, M.A. & BRAGA, A.P. (2006). Optimization of neural networks with multi-objective lasso algorithm. In *Neural Networks, 2006. IJCNN '06. International Joint Conference on*, 3312 – 3318. [49](#)
- COSTA, M.A. & BRAGA, A.P. (2011). *Gradient Descent Decomposition for Multi-objective Learning*, 377–384. Springer Berlin Heidelberg, Berlin, Heidelberg. [49](#)

- COSTA, M.A., BRAGA, A.P. & MENEZES, B.R. (2002a). Improving neural networks generalization with new constructive and pruning methods. *J. Intell. Fuzzy Syst.*, **13**, 75–83. [3](#)
- COSTA, M.A., BRAGA, A.P., MENEZES, B.R., TEIXEIRA, R. & PARMA, G. (2002b). Training neural networks with a multi-objective sliding mode control algorithm. *Neurocomputing*, **51**, 467–473. [5](#), [13](#), [24](#)
- COSTA, M.A., BRAGA, A.P. & MENEZES, B.R. (2007). Improviing generalization of mlps with multi-objective sliding mode control and the levenberg-maquardt algorithm. *Neurocomputing*, **70**, 1342–1347. [5](#), [6](#), [22](#), [24](#), [34](#), [47](#), [77](#)
- COSTA, M.A., BRAGA, A.P. & MENEZES, B.R. (2012). Convergence analysis of sliding mode trajectories in multi-objective neural networks learning. *Neural Networks*, **33**, 21–31. [5](#)
- DEMŠAR, J. (2006). Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, **7**, 1–30. [68](#), [72](#), [73](#)
- FAHLMAN, S. (1988). An empirical study of learning speed in back-propagation networks. *Technical Report, Carnegie Mellow University*. [81](#)
- FERNÁNDEZ-DELGADO, M., CERNADAS, E., BARRO, S. & AMORIM, D. (2014). Do we need hundreds of classifiers to solve real world classification problems? *J. Mach. Learn. Res.*, **15**, 3133–3181. [71](#), [72](#), [76](#)
- FERREIRA., P.A.V. (1999). Otimização multiobjetivo: Teoria e aplicações. tese de livre docência. [19](#)
- FLAKE, G.W. & LAWRENCE, S. (2002). Efficient svm regression training with smo. *Machine Learning*, **46**, 271–290. [47](#)
- FRIEDMAN, J., HASTIE, T. & TIBSHIRANI, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, **33**, 1–22. [32](#), [79](#)

- FUN, M.H. & HAGAN, M.T. (1996). Levenberg-marquardt training for modular networks. *IEEE International Conference on Neural Networks*, **1**, 468 – 473. [38](#)
- GEMAN, S., BIENENSTOCK, E. & DOURSAT, R. (1992). Neural networks and the bias/variance dilemma. *Neural Comput.*, **4**, 1–58. [16](#)
- GIROSI, F., JONES, M. & POGGIO, T. (1995). Regularization theory and neural network architectures. *Neural computation*, **vol. 7**, 219–269. [3](#)
- HAGAN, M.T. & MENHAJ, M.B. (1994). Training feedforward networks with the marquardt algorithm. *IEEE Transactions on Neural Networks*, **5**, 989 – 993. [13](#), [38](#), [39](#)
- HAYKIN, S. (2001). *Redes Neurais: Princípios e Práticas*. Bookman. [1](#)
- HAYKIN, S. (2009). *Neural networks and learning machines*, vol. 3. Prentice Hall. [12](#)
- HINTON, G.E. (1989). Connectionist learning procedures. *Artificial intelligence*, **40**, 185–234. [3](#), [22](#)
- HORTA, E.G., BRAGA, A.P. & SALDANHA, R.R. (2008). Acelerando o treinamento multiobjetivo de rnas pelo método de gradiente projetado. In *Congresso Brasileiro de Automática, Setembro*. [49](#)
- JIN, Y. & SENDHOFF, B. (2009). Pareto-based multiobjective machine learning: An overview and case studies. *IEEE Transactions on Systems Science and Cybernetics*, **Vol. 39**, 373. [5](#)
- KENDALL, M.G. (1961). *A Course in The Geometry of n Dimensions*. Griffin’s Statistical Monographs & Courses, Charles Griffin & Company Limited, London, UK. [33](#)
- KOKSHENEV, I. (2010). *Aprendizado Multiobjetivo de Redes RBF e de Máquinas de Kernel*. Tese de doutorado, UFMG. [73](#)
- KOKSHENEV, I. & BRAGA, A.P. (2008). A multi-objective approach to rbf network learning. *Neurocomputing*, **71**, 1203–1209. [5](#)

- KOKSHENEV, I. & BRAGA, A.P. (2010). An efficient multi-objective learning algorithm for rbf neural network. *Neurocomputing*, **37**, 2799–2808. [5](#)
- KORHONEN, P. (1998). Multiple objective programming support. Tech. rep., International Institute for Applied Systems Analysis, Laxenburg, Austria. [19](#)
- KUHN, M. (2012). *caret: Classification and Regression Training*. R package version 6.0-47. [72](#)
- LAWSON, C.L. & HANSON, J.R. (1974). *Solving Least Square Problems*. Prentice Hall, Englewoods Clifs, NJ. [25](#)
- MARQUARDT, D.W. (1963). An algorithm for least squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, **11**, 431–441. [38](#)
- MATTHYSEN, W. & ENGELBRECHT, A.P. (2011). A polar coordinate particle swarm optimiser. *Applied Soft Computing*, **11**, 1322–1339. [30](#), [31](#)
- MEDEIROS, T.H., TAKAHASHI, R.H.C. & BRAGA, A.P. (2007). A new decision strategy in multi-objective training of artificial neural networks. In *European Symposium on Neural Networks, Brugges*, 555–560, Proceedings of the European Symposium on Neural Networks, D-Side Publications, Bruxelas. [25](#)
- MEDEIROS, T.H., TAKAHASHI, H.C.R. & BRAGA, A.P. (2009). A incorporação do conhecimento prévio na tomada de decisão do aprendizado multi-objetivo. *Congresso Brasileiro de Redes Neurais - Inteligência Computacional*, vol. **9**, 25–28. [25](#), [26](#), [49](#)
- MEDEIROS, T.H., ROCHA, H.P., TORRES, F.S., TAKAHASHI, R.H.C. & BRAGA, A.P. (2016). Multi-objective decision in machine learning. *Journal of Control, Automation and Electrical Systems*, 1–11. [24](#), [25](#), [26](#)
- NASH, J.C. & VARADHAN, R. (2011). Unifying optimization algorithms to aid software system users: optimx for R. *Journal of Statistical Software*, **43**, 1–14. [42](#)
- NGUYEN, T.M. (2014). *N-Dimensional Quasipolar Coordinates - Theory and Application*. M.s. thesis, University of Nevada, Las Vegas, USA. [28](#), [33](#)

REFERÊNCIAS

- NOCEDAL, J. & WRIGHT, S.J. (1999). *Numerical Optimization*. Springer Series in Operations Research, Springer-Verlag, New York, NY. [41](#)
- OSY CZKA, A. (1985). Multicriteria optimization for engineering design. *Design optimization*, **1**, 193–227. [19](#)
- PARETO, V. (1897). *Cours d'Economie Politique*, vol. I e II. F. Rouge & Cie. Lausanne, Switzerland. [20](#)
- PARREIRAS, R.O. (2006). *Algoritmos Evolucionários e Técnicas de Tomada de Decisão em Análise Multicritério*. Tese de doutorado, Universidade Federal de Minas Gerais - UFMG. [23](#)
- PARREIRAS, R.O. & VASCONCELOS, J.A. (2005). Decision Making in Multiobjective Optimization Problems. In N. Nedjah & L. de Macedo Mourelle, eds., *Real-World Multi-Objective System Engineering*, 29–52, Nova Science Publishers, New York. [23](#)
- R CORE TEAM (2016). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. [42](#), [47](#), [50](#), [72](#)
- REED, R. (1993). Pruning algorithms: A survey. *IEEE Transactions on Neural Networks*, **4**, 740–746. [3](#)
- RIEDMILLER, M. & BRAUN, H. (1992). Rprop - a fast adaptive learning algorithm. Tech. rep., Proc. of ISCIS VII), Universitat Karlsruhe. [13](#), [81](#)
- ROCHA, H.P., CASTRO, C.L. & BRAGA, A.P. (2011). Seleção de modelos neurais utilizando evolução diferencial através do controle de erro e norma do vetor de pesos. In *X Congresso Brasileiro de Inteligência Computacional (CBIC-2011)*. [24](#), [49](#)
- ROCHA, H.P., COSTA, M.A. & BRAGA, A.P. (2013). Treinamento multiobjetivo de redes neurais artificiais baseado em coordenadas esféricas. In *Anais do XI Congresso Brasileiro de Inteligência Computacional - CBIC 2013*. [24](#), [80](#)

REFERÊNCIAS

- ROCHA, H.P., COSTA, M.A. & BRAGA, A.P. (2015). Training multi-layer perceptron with multi-objective optimization and spherical weights representation. In *Proc. of the European Symposium on Neural Networks*, 131–136. [10](#), [24](#), [36](#)
- RUMELHART, D. & MCCLELLAND, J. (1986). *Parallel Distributed Processing, vol1: Foundations..* The MIT Press. [12](#), [36](#), [79](#)
- SAWARAGI, Y., NAKAYAMA, H. & TANINO., T. (1985). *Theory of multiobjective optimization..* Academic Press. [19](#)
- SCHAPIRE, R.E. (1990). The strength of weak learnability. *Machine learning*, **5**, 197–227. [3](#)
- SHOR, N.Z. (1977). Cut-off method with space extension in convex programming problems. *Cybernetics*, **12**, 94–96. [22](#)
- SILVESTRE, L.J. (2015). *Regularização de Extreme Learning Machines: Uma Abordagem com Matrizes de Afinidade*. Tese de doutorado, UFMG. [73](#)
- STONE, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society. Series B (Methodological)*, 111–147. [3](#)
- TAKAHASHI, R.H.C. (2007). Otimização escalar e vetorial. Notas de aula em Otimização, universidade Federal de Minas Gerais, Dept. de Matemática. [20](#)
- TAVARES, L.D. (2015). *Treinamento Multiobjetivo de Extreme Learning Machines Utilizando Decomposição em Valores Singulares*. Tese de doutorado, Universidade Federal de Minas Gerais - UFMG. [25](#), [47](#)
- 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, Universidade Federal de Minas Gerais - UFMG. [22](#), [34](#), [46](#), [49](#)
- 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. [2](#), [5](#), [6](#), [7](#), [22](#), [24](#), [46](#), [47](#), [77](#), [78](#)

- TEIXEIRA, R.A., BRAGA, A.P., TAKAHASHI, R.H.C. & SALDANHA, R.R. (2002). Decisor implementation in neural model selection by multiobjective optimization. In *VII Brazilian Symposium on Neural Networks*. [24](#)
- TEIXEIRA, R.A., BRAGA, A.P., SALDANHA, R.R., TAKAHASHI, R.H. & MEDEIROS, T.H. (2007). The usage of golden section in calculating the efficient solution in artificial neural networks training by multi-objective optimization. In *Inter. Conf. on Artificial Neural Networks (ICANN)*, vol. 4668, 289–298. [47](#)
- TORRES, L.C.B. (2016). *Classificador por Arestas de Suporte (clas): Métodos de Aprendizado Baseados em Grafos de Gabriel*. Tese de doutorado, UFMG. [47](#), [73](#)
- TORRES, L.C.B., CASTRO, C.L. & BRAGA, A.P. (2012). A computational geometry approach for pareto-optimal selection of neural networks. In *Inter. Conf. on Artificial Neural Networks (ICANN)*, 100–107. [25](#)
- TORRES, L.C.B., CASTRO, C.L., COELHO, F., TORRES, F.S. & BRAGA, A.P. (2015). Distance-based large margin classifier suitable for integrated circuit implementation. *Electronics Letters*, **51**, 1967–1969. [47](#)
- VAPNIK, V.N. (1995). *The nature of statistical learning theory*. Springer Verlag New York. [1](#), [2](#), [3](#), [5](#), [15](#), [16](#)
- VAPNIK, V.N. (1998). *Statistical Learning Theory*. Wiley-Interscience. [1](#), [15](#)
- WALESIK, M. & DUDEK, A. (2014). Searching for optimal clustering procedure for a data set. [47](#)
- WILCOXON, F. (1945). Individual comparisons by ranking methods. *Biometrics bulletin*, **1**, 80–83. [68](#)
- ZHANG, W. & KING, I. (2002). A study of the relationship between support vector machine and gabriel graph. *Neural Networks, IJCNN*, **vol. 1**, 239–244, proceedings of the 2002 International Joint Conference. [26](#)
- ZITZLER, E. (1999). *Evolutionary Algorithms for Multiobjective Optimization: Methods and Application*. Ph.d. dissertation, Swiss Federal Int. Technology. [51](#)