

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO CEARÁ
PRÓ-REITORIA DE PESQUISA, PÓS-GRADUAÇÃO E INOVAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

José Alberth Vasconcelos Florêncio

Uma nova perspectiva para Máquinas de Aprendizagem Mínima

Fortaleza – CE

02 de março de 2020

José Alberth Vasconcelos Florêncio

Uma nova perspectiva para Máquinas de Aprendizagem Mínima

Dissertação submetida ao Programa de Pós-Graduação em Ciência da Computação do Instituto Federal de Educação, Ciência e Tecnologia do Ceará, como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Linha de Pesquisa: Inteligência Artificial

Orientador: Prof. Dr. Ajalmar Rêgo da Rocha Neto

Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE)

Pró-Reitoria de Pesquisa, Pós-Graduação e Inovação (PRPI)

Programa de Pós-Graduação em Ciência da Computação (PPGCC)

Fortaleza – CE

02 de março de 2020

Dados Internacionais de Catalogação na Publicação
Instituto Federal do Ceará - IFCE
Sistema de Bibliotecas - SIBI
Ficha catalográfica elaborada pelo SIBI/IFCE, com os dados fornecidos pelo(a) autor(a)

- F632n Florêncio, José Alberth Vasconcelos.
 Uma nova perspectiva para Máquinas de Aprendizagem Mínima / José Alberth Vasconcelos
 Florêncio. - 2020.
 92 f. : il. color.
- Dissertação (Mestrado) - Instituto Federal do Ceará, Mestrado em Ciência da Computação, Campus
 Fortaleza, 2020.
 Orientação: Prof. Dr. Ajalmar Rêgo da Rocha Neto.
1. Aprendizagem de máquina. 2. Aprendizagem supervisionada. 3. Máquina de aprendizagem
 mínima. I. Título.

CDD 004



Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE)
Pró-Reitoria de Pesquisa, Pós-Graduação e Inovação (PRPI)
Programa de Pós-Graduação em Ciência da Computação (PPGCC)

José Alberth Vasconcelos Florêncio

Esta dissertação foi julgada adequada para a obtenção do título de Mestre em Ciência da Computação, sendo aprovada pela Coordenação do Programa de Pós-Graduação em Ciência da Computação do Instituto Federal de Educação, Ciência e Tecnologia do Ceará e pela banca examinadora:

**Orientador: Prof. Dr. Ajalmar Rêgo da Rocha
Neto**

Instituto Federal de Educação, Ciência e Tecnologia do
Ceará (IFCE)

Prof. Dr. Guilherme de Alencar Barreto
Universidade Federal do Ceará (UFC)

Prof. Dr. Pedro Pedrosa Rebouças Filho
Instituto Federal do Ceará (IFCE)

Fortaleza – CE
02 de março de 2020

Este trabalho é dedicado aos meus pais, Lúcia de Fátima e Nilton Florêncio, que desde sempre me dão o suporte necessário para que eu possa alcançar todos os meus objetivos.

Agradecimentos

Quando optei por iniciar o mestrado, já imaginava que seria difícil, mas não imaginava que seria tão difícil assim. O que eu também não imaginava seria o suporte que eu receberia para poder concluir esta etapa tão importante da minha vida com sucesso.

Obviamente, começo os meus agradecimentos mais do que especiais para a minha família, minha mãe Lúcia de Fátima e meu pai José Nilton, por todo amor que recebi e pelo apoio em minha carreira acadêmica desde sempre. Sem vocês, eu tenho certeza que não teria chegado tão longe. E claro, também agradeço as ajudas oferecidas por meus irmãos, que seja com conversa jogada fora ou uma pequena ajuda, foram capazes de me dar o apoio necessário para a conclusão do meu mestrado.

Também agradeço ao IFCE, que direta e indiretamente, me ajudou na conclusão deste mestrado. Seja pela mão do meu orientador Prof. Dr. Ajalmar Rocha, que me recebeu no LIA quando solicitei fazer parte do seu time de pesquisadores, como também por todas as amizades adquiridas no laboratório, que sem elas, eu não teria conseguido realizar tantos feitos durante o curso. Em especial, um abraço aos meus queridos amigos Madson Dias e Saulo Oliveira, que me ajudaram quando eu estava mais perdido sobre o que fazer na minha pesquisa, e que além da ajuda acadêmica, se mostraram como amigos extraordinários para a vida toda. Um abraço também para meus amigos Átilla Negreiros, Felipe Moreira, Lucas Matias e Lucas Silva, que também estavam sempre presentes no laboratório, gerando discussões relevantes para nossas pesquisas, como também participando do nosso bom coffee-break recheado de salgados e refri. E por fim, também vale o agradecimentos ao PPGCC/IFCE e seus professores, por contribuírem de maneira excepcional na condução do curso.

“Faça. Ou não faça. Tentativa não há.”
(Mestre Yoda)

Resumo

Este trabalho introduz três novas propostas para a máquina de aprendizagem mínima (*Minimal Learning Machine*, MLM), chamadas de *overlap removed* MLM (OR-MLM), *fuzzy C-means* MLM (FCM-MLM) e a máquina de aprendizagem mínima de peso leve (*Lightweight Minimal Learning Machine*, LW-MLM), onde as duas primeiras versões utilizam algoritmos de agrupamento na etapa de seleção de pontos de referência (*reference points*, RPs), enquanto que no LW-MLM a etapa de seleção de RPs é ignorada. O procedimento básico de treinamento do MLM está relacionado à seleção de um subconjunto de padrões de treinamento, chamados de pontos de referência. Tal subconjunto é utilizado para a realização de um mapeamento entre as configurações geométricas das amostras (pontos de entrada) e seus rótulos (pontos de saída) correspondentes. A proposição original sugere que a seleção dos pontos de referência seja feita de maneira aleatória, deixando a cargo do utilizador apenas a definição do número de pontos a serem utilizados. O método de busca em grade, combinado à validação cruzada, pode ser utilizado para seleção automática da quantidade de pontos. Porém, esse método de seleção frequentemente acarreta uma escolha inadequada dos pontos de referência, fazendo com que amostras ruidosas ou mesmo discrepantes componham tal subconjunto. Além disso, a utilização de todo o conjunto de treinamento como RPs resulta em um modelo sobreajustado (*overfitted*). Por este motivo, as propostas OR-MLM e FCM-MLM apresentam novas abordagens que buscam escolher um melhor subconjunto como RPs, porém com o custo de algoritmos externos para isso. A fim de superar o problema na escolha de pontos de referência e evitar *overfitting*, o LW-MLM apresenta uma versão regularizada do MLM em que cada padrão é penalizado individualmente, além de modificar a etapa de predição de novos padrões, reduzindo a quantidade de cálculos inerentes ao processo de multiliteração da proposta original. As propostas foram validadas com a realização de experimentos com conjuntos de dados do mundo real e artificiais. Os experimentos computacionais atestam o desempenho satisfatórios do LW-MLM quanto a capacidade de generalização e redução de complexidade, em comparativo com outras versões do MLM e demais regressores da literatura, e os experimentos para o OR-MLM e FCM-MLM também atestam a qualidade do modelo perante outras versões de classificadores da máquina de aprendizagem mínima.

Palavras-chave: Aprendizagem de máquina. Aprendizagem supervisionada. Máquina de aprendizagem mínima.

Abstract

This work introduces three new proposals for the minimal learning machines (MLM), called overlap removed MLM (OR-MLM), fuzzy *C*-means MLM (FCM-MLM) and the lightweight minimal learning machine (LW-MLM), where the first two versions use clustering algorithms in the references points (RPs) selection step, whereas in LW-MLM the RPs selection step is ignored. The original proposition suggests that the selection of the reference points be done in a random way, leaving only the definition of the number of points to be used to the user. The grid search method, combined with cross-validation, can be used to automatically select the number of points. However, this selection method often leads to an inadequate choice of reference points, causing noisy or even discrepant samples to make up such a subset. In addition, the use of the entire training set as PRs results in an overfitted model. For this reason, the OR-MLM and FCM-MLM proposals present new approaches that seek to choose a better subset as RPs, but at the cost of external costs. In order to overcome the problem of choosing benchmarks and avoid overfitting, LW-MLM features a regularized version of MLM where each pattern is penalized individually, and modifies the prediction step of new patterns, reducing the number of computations inherent in the multilateration process of the original proposal. The proposals were validated by conducting experiments with real-world and artificial data sets. Computational experiments attest to the satisfactory performance of LW-MLM in terms of generalizability and complexity reduction compared to other versions of MLM and other literature regressors, and experiments for OR-MLM and FCM-MLM also attest to the quality of the model against other versions of the minimum learning machine classifiers.

Palavras-chave: Machine learning. Supervised learning. Minimal learning machine.

Lista de ilustrações

Figura 1 – Exemplo de um problema de regressão.	33
Figura 2 – Efeito da regularização em modelo.	35
Figura 3 – Exemplo de um problema de classificação de duas classes.	41
Figura 4 – Estimação da saída $\mathbf{y}^*$. Esse vetor representa o ponto que se deseja estimar através da multilateração, $\{\mathbf{y}_n\}_{n=1}^N$ são as imagens dos pontos de referência $\mathcal{D}$ e $d(\mathbf{y}_n, \mathbf{y}^*)$ é a distância de saída estimada das imagens dos pontos de referência até o rótulo $\mathbf{y}^*$	47
Figura 5 – Superfícies de decisão (a) subajustada, (b) sobreajustada e (c) ideal. . .	52
Figura 6 – Superfícies de decisão obtidos para a base Ripley.	62
Figura 7 – Funções resultantes para o conjunto Artificial I	69
Figura 8 – Funções resultantes para o conjunto Artificial II	70
Figura 9 – Funções resultantes para o conjunto Artificial III	70
Figura 10 – Funções resultantes para o conjunto Artificial IV	71
Figura 11 – Funções resultantes para o conjunto Mcycle	71
Figura 12 – Variação do RMSE na base Boston Housing.	75
Figura 13 – Variação do RMSE para algumas bases	89
Figura 14 – Variação do RMSE para algumas bases	90

Lista de tabelas

Tabela 1	– Conjuntos de dados utilizados.	60
Tabela 2	– Resultados dos experimentos das versões do MLM. São apresentados os desempenhos da acurácia (ACC) e porcentagem de redução em comparação com os dados de treinamento (RED).	61
Tabela 3	– Resultados dos experimentos das versões do MLM. São apresentados os desempenhos da acurácia (ACC) e porcentagem de redução em comparação com os dados de treinamento (RED).	61
Tabela 4	– Descrição das bases de dados.	64
Tabela 5	– Variantes da MLM e as configurações de seus hiperparâmetros.	65
Tabela 6	– Resultados de RMSE e desvio padrão sobre 30 realizações independentes.	66
Tabela 7	– Resultados de R^2 Score e desvio padrão sobre 30 realizações independentes.	67
Tabela 8	– Valor da norma da matriz $\mathbf{B}$ e desvio padrão sobre 30 realizações independentes. O valor foi normalizado entre 0 e 1 para cada base.	68
Tabela 9	– Descrição dos conjuntos de dados artificiais.	69
Tabela 10	– Valor da norma da matriz $\mathbf{B}$ resultantes do experimento qualitativo. O valor foi normalizado entre 0 e 1 para cada base.	72
Tabela 11	– Hiperparâmetros das redes RBF, ELM e do modelo LSSVM para serem ajustados na validação cruzada.	73
Tabela 12	– Resultados de RMSE e desvio padrão sobre 30 realizações independentes.	74
Tabela 13	– Resultados de R^2 Score e desvio padrão sobre 30 realizações independentes.	74
Tabela 14	– Comparação das variações do MLM em termos de requisitos de memória, custo de treinamento e de predição. Lembramos que N representa a cardinalidade do conjunto de treinamento, enquanto M é o número de RPs ($N \geq M$).	76

Lista de algoritmos

1	Algoritmo k -médias.	43
2	Algoritmo fuzzy c -means	44
3	Processo de treinamento do modelo MLM	47
4	Processo de estimação da saída do modelo MLM	48
5	Processo de treinamento do modelo <i>Random</i> MLM	49
6	Processo de treinamento do modelo <i>Rank</i> MLM	49
7	OR-MLM	53
8	FCM-MLM	55

Lista de abreviaturas e siglas

MLM	Minimal learning machine
LW-MLM	Lightweight minimal learning machine
RPs	Reference points
AM	Aprendizagem de máquina
RSS	Residual sum of squares
OLAM	Optimal linear associative memory
RBF	Radial basis function network
ELM	Extreme learning machine
SLFN	Single-hidden layer feedforward neural network
MLP	Multi-layer perceptron
LS-SVM	Least-squares support vector machine
SVM	Support vector machine
QP	Quadratic programming
NN-MLM	Nearest neighbor MLM
FCM-MLM	Fuzzy c-means MLM
OS-MLM	Optimally selected MLM
MRSR	Multiresponse sparse regression
RMSE	Root mean squared error
R ² Score	Coefficiente de determinação

Lista de símbolos

K	Número de pontos de referência
$\mathbf{P}$	Matriz de regularização
$\{x\}$	Vetor de entrada D -dimensional
t	Variável-alvo contínua
N	Número de amostras do conjunto
$\{\mathbf{x}_n\}$	Conjunto de n vetores
$\{\mathbf{X}\}$	Conjunto de n vetores
$\{y_n\}$	Saídas correspondentes ao conjunto $\{\mathbf{x}_n\}$
$\{\mathbf{y}\}$	Saídas correspondentes ao conjunto $\{\mathbf{X}\}$
$\mathbf{w}$	Vetor de pesos
f	Uma função f qualquer
$\mathbf{x}_i$	i -ésimo vetor de entrada
$\hat{y}_i$	i -ésima estimativa
e_i	i -ésimo resíduo
λ	Parâmetro de regularização
D	Dimensão do vetor de entrada
$\mathbf{X}^{-1}$	Inversa da matriz $\mathbf{X}$
$(\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top$	Pseudo-inversa da matriz $\mathbf{X}$
$\phi()$	Função de base radial
σ	Valor da abertura da função Gaussiana
Φ	Matriz de interpolação
β	Vetor de pesos entre camada oculta e camada de saída da ELM
$h(\mathbf{x})$	Função de ativação da camada oculta da ELM

$\mathbf{H}$	Matriz de saída da camada oculta da ELM
b	Viés
$\varphi(x)$	Mapeamento não-linear
γ	Hiperparâmetro de ajuste do LSSVM
α	Multiplicador de Lagrange
$\mathbf{I}$	Matriz identidade
Ω	Matriz de <i>kernel</i> do LSSVM
$\mathcal{D}$	Conjunto de dados
$\mathcal{X}$	Conjunto de dados de entrada
$\mathcal{Y}$	Conjunto de dados de saída
$\mathbf{D}$	Matriz de distâncias de entrada
Δ	Matriz de distâncias de saída
$d(.,.)$	Função de distância Euclidiana entre dois pontos
$\Phi(\mathbf{x}_i)$	Função de distância de $\mathbf{x}_i$ para os pontos de referência
$\Psi(\mathbf{y}_i)$	Função de distância de $\mathbf{y}_i$ para os pontos de referência
$\mathbf{B}$	Matriz de mapeamento
$\mathbf{B}$	Matriz de resíduos
$\mathcal{R}$	Conjunto de RPs de entrada
$\mathcal{T}$	Conjunto de RPs de saída
C	Parâmetro de regularização do <i>Rank</i> MLM
$\mathcal{N}(0, 1)$	Função de distribuição normal
$\pi val(\cdot, \cdot)$	Função que retorna o valor p do teste KS de duas amostras
$\mathbf{1}(\cdot)$	Função indicadora
t	Limiar de linearidade
$\kappa(\cdot)$	Função de descarte

Sumário

1	INTRODUÇÃO	27
1.1	Objetivo Geral da Dissertação	29
1.2	Objetivos Específicos	30
1.3	Produção científica	30
1.4	Estrutura da Dissertação	30
2	FUNDAMENTAÇÃO TEÓRICA	33
2.1	Regressão	33
2.1.1	Estimação do coeficiente de pesos	34
2.2	Regularização de modelos	34
2.3	Métodos de Aproximação de Função	35
2.3.1	Optimal Linear Associative Memory	35
2.3.2	Rede neural de base radial	36
2.3.3	Máquina de aprendizagem extrema	38
2.3.4	Máquina de vetores-suporte de mínimos quadrados	39
2.4	Classificação	41
2.5	Agrupamento de dados	42
2.5.1	Algoritmo k -médias	42
2.5.2	Algoritmo fuzzy c -means	43
2.6	Conclusão	44
3	MÁQUINA DE APRENDIZAGEM MÍNIMA	45
3.1	Máquinas de aprendizem mínima	45
3.1.1	Formulação básica	45
3.1.2	Treinamento	46
3.1.3	Estimativa da saída	46
3.1.4	Resumo para o algoritmo do MLM	47
3.1.5	Variações da MLM	48
3.1.5.1	<i>Random</i> MLM	48
3.1.5.2	<i>Rank</i> MLM	49
3.1.6	MLM aplicado a problemas de classificação	50
3.1.6.1	MLM de vizinho mais próximo	50
3.1.6.2	Análise empírica da seleção de pontos de referência	51
3.2	<i>Overlap removed</i> MLM (OR-MLM) (Proposta I)	52
3.3	Fuzzy C-means MLM (FCM-MLM) (Proposta II)	53
3.4	Máquina de Aprendizagem Mínima de Peso Leve (Proposta III)	55

3.4.1	Formulação	55
3.4.2	Seleção da matriz de regularização $\mathbf{P}$	56
3.4.2.1	Baseado em uma distribuição normal	56
3.4.2.2	Abordagem não-paramétrica por intervalos não-lineares de f	56
3.4.3	Redução do custo de predição	57
3.4.4	Similaridade com a etapa de treinamento da Rank-MLM	58
3.5	Conclusão	58
4	EXPERIMENTOS E DISCUSSÕES	59
4.1	Simulações e discussões das Propostas I e II	59
4.1.1	Configurações dos experimentos	59
4.1.2	Análise comparativa com outras abordagens do próprio MLM	60
4.1.3	Análise qualitativa	62
4.2	Simulações e discussões da Proposta III	63
4.2.1	Análise comparativa com outras abordagens da própria MLM	63
4.2.1.1	Erro de predição como métrica de desempenho	66
4.2.1.2	Coeficiente de determinação como métrica de desempenho	67
4.2.1.3	A norma como métrica de desempenho	68
4.2.2	Análise qualitativa	68
4.2.3	Análise comparativa entre a proposta e outros modelos	73
4.2.4	Análise da relevância do número de RPs na etapa de predição	75
4.2.5	Análise da complexidade computacional entre as variações do MLM	75
4.3	Conclusão	76
5	CONCLUSÃO	79
5.1	Considerações finais	79
5.2	Contribuições desse trabalho	80
5.3	Trabalhos futuros	80
	REFERÊNCIAS	81
	APÊNDICES	85
	APÊNDICE A – SOLUÇÃO DA EQUAÇÃO (3.14)	87
	APÊNDICE B – RELEVÂNCIA DO NÚMERO DE RPS NA ETAPA DE PREDIÇÃO	89

1 Introdução

O reconhecimento de padrões é essencial em diversas áreas e há muito vem sendo utilizado para encontrar novas informações a partir dos dados. Esta área preocupa-se com a descoberta automática de padrões em dados através do uso de algoritmos e com o uso desses padrões para realizar ações como classificar os dados em diferentes categorias ou estimar um valor contínuo em função do valor de entrada (BISHOP, 2007). O uso destes algoritmos estão presentes na pesquisa científica de diversas áreas de estudo, tais como estatística, processamento e análise de sinais e imagens, visão computacional, bioinformática, automação industrial e no auxílio de diagnósticos médicos (THEODORIDIS, 2015).

Através do reconhecimento de padrões é possível detectar e descobrir uma provável estrutura oculta e padrões de regularidade nos dados. Essa informação, por sua vez, nos permite analisar e compreender a natureza dos dados, que então podem ser utilizados para fazer previsões futuras (THEODORIDIS, 2015). Além disso, é de grande importância para a área de reconhecimento de padrões desenvolver algoritmos eficientes para projetar os modelos e também para análise e previsão, sendo este último muito importante, pois na era do *big data*, temos que lidar com quantidades massivas de dados e que também podem ser representados em espaços de dimensões muito grandes. A análise de dados para tais aplicações define a exigência de que os algoritmos sejam computacionalmente eficientes e, ao mesmo tempo, robustos em seu desempenho, pois alguns desses dados estão contaminados com elevado nível de ruído e também, em alguns casos, os dados podem ter valores ausentes.

O nome aprendizagem de máquina ganhou popularidade há pouco tempo, e esse nome sugere o uso de uma máquina (computador) para aprender algo, em analogia com o modo como o cérebro aprende e prevê. Em alguns casos, os métodos são diretamente inspirados na maneira como o cérebro funciona, como é o caso das redes neurais artificiais. O tipo de aprendizagem pode ser dividido em quatro categorias, a saber: (i) aprendizagem supervisionada, (ii) aprendizagem não supervisionada, (iii) aprendizagem semi-supervisionada e (iv) aprendizagem por reforço.

A aprendizagem supervisionada é a abordagem na qual o modelo encontra uma função a partir de um conjunto de dados rotulados. O objetivo é encontrar os parâmetros ótimos que ajustem um modelo que possa prever rótulos desconhecidos em novas amostras não rotuladas. Esta abordagem pode ser utilizada para dois tipos de problemas. Primeiro, em problemas de regressão, o modelo prevê os resultados em uma saída contínua, o que significa que está mapeando variáveis de entrada para alguma função contínua. E o outro

tipo de tarefa são os problemas de classificação, onde o modelo prevê os resultados em uma saída discreta. Em outras palavras, este tipo de problema mapeia variáveis de entrada em categorias distintas (THEODORIDIS, 2015).

Enquanto isso, a aprendizagem não supervisionada nos permite abordar problemas nos quais não temos conhecimentos prévios sobre os dados, ou seja, sem categorias ou saída contínua esperada. O objetivo dessa abordagem é observar algumas similaridades entre as amostras e identificá-las em grupos apropriados. Algumas amostras podem diferir bastante de todos os grupos e, deste modo, podemos assumir que estas amostras são anomalias do conjunto de dados (DUDA; HART; STORK, 2001).

A aprendizagem semi-supervisionada inclui ambas abordagens descritas acima: tal abordagem utiliza amostras rotuladas e não-rotuladas. A aprendizagem semi-supervisionada pode ser utilizada para tarefas de classificação ou regressão nas amostras rotuladas, ou tarefas de agrupamento quando se utilizam as amostras não rotuladas, sendo as amostras rotuladas responsáveis por auxiliar o processo de formação destes grupos (SANCHES, 2003).

A aprendizagem por reforço é a abordagem de aprendizagem de máquina em que seu treinamento se baseia no oferecimento de recompensas e punições, no qual não precisa especificar como uma tarefa deve ser realizada. Entende-se como o problema encontrado por um modelo que deve aprender como se comportar em uma tarefa dinâmica através de interações do tipo “tentativa e erro” (SUTTON; BARTO, 1998).

Uma complicação frequente na aprendizagem supervisionada é quando o modelo aprendeu as informações somente das amostras vistas no treinamento, sendo incapaz de utilizar seu aprendizado em novos padrões não vistos, ou seja, quando o modelo é incapaz de generalizar bem os dados. Com o intuito de evitar este sobreajuste, é comum utilizar técnicas matemáticas de regularização sobre o modelo. Esta abordagem é capaz de reduzir a complexidade de um modelo penalizando grandes parâmetros individualmente, sem haver a necessidade de remover características ou amostras do conjunto de treinamento (JAMES et al., 2013).

A máquina de aprendizagem mínima (MLM) é uma proposta recente de um algoritmo de aprendizagem supervisionada, no qual pode ser aplicado em problemas de classificação e aproximação de funções (SOUZA et al., 2015). Esta técnica vem sendo aplicada em diversos tipos de problemas do mundo real, tais como a detecção de falhas em motores (COELHO et al., 2014), localização de robôs móveis (MARINHO et al., 2017) e cinemática inversa de robôs (MELO, 2015).

O funcionamento da MLM se baseia na pressuposição da existência de um mapeamento entre as configurações geométricas de pontos do espaço de entrada e as configurações geométricas dos respectivos pontos no espaço de saída. Este mapeamento é obtido na etapa

de treinamento do modelo. Uma das etapas do processo de treinamento da máquina de aprendizagem mínima é a detecção de pontos de referência (*reference points*, RPs). Esses pontos são um subconjunto de padrões que compõem o conjunto de dados de treinamento, a partir dos quais são calculadas as distâncias para todas as outras amostras de treinamento no espaço de entrada (atributos) e de saída (valor desejado).

As matrizes que armazenam as distâncias de cada padrão de treinamento para todos os RPs são chamadas de matrizes de distâncias de entrada e de saída. A aprendizagem da MLM é realizado através da determinação de um modelo de regressão linear entre essas duas matrizes de distância. Dessa forma, dado um ponto no espaço de entrada, pode-se calcular a localização deste ponto no espaço de saída, através do modelo de regressão aprendido (SOUZA et al., 2015).

A formulação original do algoritmo de treinamento de uma máquina de aprendizagem mínima estabelece que a escolha dos RPs seja realizada de maneira aleatória, deixando a cargo do utilizador apenas a definição do número de pontos utilizados (SOUZA et al., 2015). Dessa forma, a escolha do número de pontos necessários é realizada via tentativa e erro.

Por conta disso, este trabalho apresenta duas propostas para contornar este problema. Estas propostas que serão apresentadas a seguir, utilizam algoritmos externos capazes de selecionar o melhor subconjunto para ser utilizado como RPs. Porém, adicionar algoritmos externos podem aumentar consideravelmente os custos de se treinar um bom modelo, principalmente quando se utilizam conjuntos de dados com muitas amostras. Tendo isso em vista, nossa terceira proposta busca diminuir a complexidade do modelo removendo a necessidade escolher o melhor subconjunto para utilizar como RPs. Ou seja, nosso modelo utiliza todos os padrões como RPs, mas nossa formulação é robusta o suficiente para que não ocorra sobreajuste do modelo aos dados de treinamento.

Neste trabalho serão apresentadas simulações acerca de nossas propostas, com o intuito de validar a qualidade e o desempenho daquelas. Para isso, serão realizados vários experimentos distintos, nos quais cada um abordará características específicas das propostas. Também serão levantadas questões sobre a sua similaridade com outras versões da MLM, além de comparar os resultados com outros modelos disponíveis na literatura. As comparações serão realizadas sobre conjuntos de dados reais que estão disponíveis na literatura.

1.1 Objetivo Geral da Dissertação

Este trabalho tem como objetivo geral apresentar novas abordagens para a máquina de aprendizagem mínima, visando resolver algumas de suas limitações em relação à escolha do melhor subconjunto para ser utilizado como pontos de referências, além de propor um

melhoria no método de predição para diminuir a quantidade de cálculos.

1.2 Objetivos Específicos

Em seu sentido mais específico, esta dissertação tem os seguintes objetivos particulares:

- Elaborar uma revisão bibliográfica acerca de métodos de regressão.
- Apresentar duas novas abordagens para a máquina de aprendizagem mínima que utilizam um algoritmo externo para seleção de pontos de referência.
- Além disso, apresentar uma terceira abordagem para a máquina de aprendizagem mínima que dispensa a escolha dos pontos de referência.
- Discutir sobre a influência do número de pontos de referência sobre o erro na etapa de predição.
- Validar a qualidade dos modelos apresentados comparando-os com outras versões da máquina de aprendizagem mínima e outros modelos de aprendizagem de máquina.

1.3 Produção científica

Como base para o trabalho dessa dissertação, os seguintes trabalhos foram desenvolvidos.

- **FLORÊNCIO, J. A. V.**; DIAS, M. L. D.; ROCHA NETO, A. R. Um novo método baseado em protótipos para seleção de pontos de referência em máquinas de aprendizado mínimo (CBIC). Niterói, Rio de Janeiro, Brasil, 2017.
- **FLORÊNCIO, J. A. V.**; DIAS, M. L. D.; ROCHA NETO, A. R.; SOUZA JÚNIOR, A. H. A Fuzzy C-means-based Approach for Selecting Reference Points in Minimal Learning Machines. In: North American Fuzzy Information Processing Society Annual Conference, NAFIPS, Fortaleza, Ceará, Brasil, 2018, Springer, Cham, 2018. p. 398-407.
- **José A. V. Florêncio**; Saulo A. F. Oliveira; João P. P. Gomes; Ajalmar R. Rocha Neto. A new perspective for Minimal Learning Machines: A lightweight approach. Neurocomputing (2019). (Submetido)

1.4 Estrutura da Dissertação

O restante dessa dissertação está dividido em quatro capítulos: fundamentação teórica, máquina de aprendizagem mínima, experimentos e discussões, e conclusão. Todos os conceitos são detalhados e sempre associados à área de aprendizagem de máquinas.

O Capítulo 2 inicia com um detalhamento acerca do problema de aprendizagem supervisionada de aproximação de funções, classificação e agrupamento de dados. Além disso, é discutido sobre a importância da regularização de modelos de aprendizagem de máquina. O capítulo encerra com a apresentação de alguns modelos considerados como estado da arte na literatura de aprendizagem de máquina e que serão utilizados para comparar com nossas propostas.

No Capítulo 3 é apresentado os métodos propostos nesse trabalho. O capítulo inicia com uma explicação sobre as máquinas de aprendizagem mínimas e uma explanação das limitações de seu método padrão de treinamento. Também é discutido sobre algumas variações da MLM que já foram apresentadas na literatura. Após isso, é apresentada a abordagem propostas.

Os experimentos e resultados obtidos com os métodos propostos são apresentados no Capítulo 4. Foram realizados vários experimentos a fim de avaliar a qualidade dos modelos em termos de redução do erro, função resultante, maior capacidade de generalização dos dados e sua complexidade. As simulações estão divididas por metodologias e cada uma aborda a proposta de maneira diferente, além de apresentar a discussão sobre os resultados obtidos.

Por fim, o Capítulo 5 apresenta as conclusões e contribuições acerca dos métodos propostos. Além disso, são demonstradas algumas deficiências dos modelos e algumas sugestões para trabalhos futuros.

2 Fundamentação Teórica

É natural que a partir de uma grande massa de dados, tentemos inferir informações a partir deles. Contudo, como se trata de um grande volume de dados, é inviável analisá-los manualmente. Por conta disso, métodos de aprendizagem de máquina ganharam grande destaque, pois através destes modelos, é possível desenvolver sistemas capazes de adquirir conhecimento de forma automática. Iniciando este capítulo, na Seção 2.1 é apresentada o que é a tarefa de regressão, também chamada de aproximação de função, seus conceitos e aplicações. Seguindo o capítulo, na Seção 2.2 será discutido o porquê e a importância da regularização em modelos de Aprendizagem de Máquina, e por fim, na Seção 2.3 serão apresentados alguns modelos considerados estados da arte e suas definições e que serão utilizados neste trabalho.

2.1 Regressão

Tarefas de regressão são comuns na área de aprendizagem de máquina. O objetivo da regressão é a de prever o valor de uma ou mais variáveis-alvo contínuas t dado uma variável $\mathbf{x}$, no qual $\mathbf{x}$ é um vetor de entrada D -dimensional. Há dois tipos de modelos de regressão: lineares, na qual a dependência da resposta sobre os modelos é definida por uma função linear, e não lineares, onde essa dependência é definida por uma função não linear (HAYKIN et al., 2009).

Dado um conjunto de dados de N amostras de entrada $\{\mathbf{x}_n\}$, onde $n = 1, \dots, N$, juntamente com os valores de suas saídas correspondentes $\{y_n\}$, o objetivo é estimar uma função f , onde $f(\mathbf{x}) = y$. Em sua perspectiva mais simples, isso pode ser feito através da estimação de um vetor de pesos $\mathbf{w}$, na qual a saída y pode ser encontrada através de uma combinação linear entre $\mathbf{X}$ e $\mathbf{w}$, ou seja, $\mathbf{y} = \mathbf{X}\mathbf{w}$.

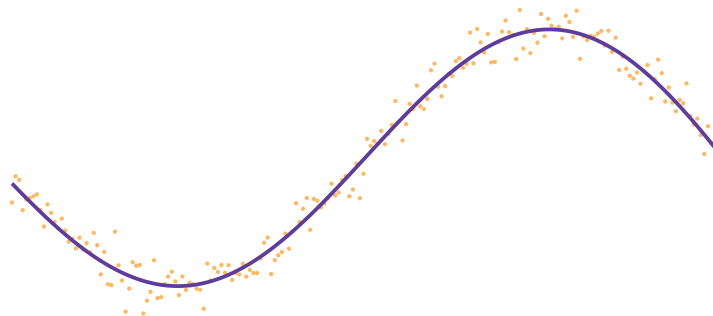


Figura 1 – Exemplo de um problema de regressão.

Na Figura 1 temos um exemplo de problema de regressão. Os pontos amarelos

representam a amostra de dados, e a linha roxa é a função que melhor representa aqueles dados, ou seja, a função que minimiza o erro.

2.1.1 Estimação do coeficiente de pesos

Na prática, o vetor de pesos $\mathbf{w}$ é desconhecido. Portanto, antes de podermos utilizá-lo para fazer previsões de novos padrões, devemos utilizar um conjunto dados de treinamento para estimar $\mathbf{w}$. Dado o conjunto de treinamento $\{\mathbf{x}_n, y_n\}_{n=1}^N$, no qual representa N observações de entrada e saída, $\mathbf{x} \in \mathbb{R}^D$ e $y \in \mathbb{R}$. Nosso objetivo é obter estimativas ótimas dos pesos $\mathbf{w}$ de forma que o modelo linear $\mathbf{X}\mathbf{w} = \mathbf{y}$ seja ótimo, isto é, para que $\mathbf{x}_i^\top \mathbf{w} = \hat{y}_i$, $i = 1, \dots, N$.

Seja $\hat{y}_i$ a estimativa ótima do i -ésimo padrão $\mathbf{x}$. Então $e_i = y_i - \hat{y}_i$ representa o i -ésimo resíduo, isto é, a diferença entre o valor da saída esperada e o valor da saída prevista pelo modelo regressor. Definimos como a soma dos quadrados dos resíduos (*residual sum of squares*, RSS) como

$$\text{RSS} = e_1^2 + e_2^2 + \dots + e_N^2, \quad (2.1)$$

que também pode ser descrito por

$$\text{RSS} = \sum_{i=1}^N \left(y_i - \sum_{j=1}^D w_j x_{ij} \right)^2. \quad (2.2)$$

Portanto, a estimativa ótima do vetor de pesos $\mathbf{w}$ é aquela que minimiza a função de custo RSS.

2.2 Regularização de modelos

Um dos principais desafios do treinamento de um modelo de aprendizado de máquina é o de evitar o sobreajuste, também chamado de *overfitting*. Se o houver *overfitting* no modelo resultante do treinamento, ele terá uma baixa acurácia na etapa de predição. Isso se dá porque o modelo está aprendendo todas as informações presentes no conjunto de treinamento, inclusive o ruído de *outliers*, ou seja, padrões que não representam as propriedades reais do conjunto de dados (HAYKIN et al., 2009).

Com o intuito de evitar problemas de *overfitting*, utiliza-se um fator de regularização no modelo. Esta técnica desencoraja a aprendizagem de um modelo mais complexo, ou seja, este fator restringe (regulariza) as estimativas dos coeficientes (pesos) para próximos de zero. Para exemplificar, faremos uma modificação na Equação (2.2):

$$\sum_{i=1}^N \left(y_i - \sum_{j=1}^D w_j x_{ij} \right)^2 + \lambda \sum_{j=1}^D w_j^2 = \text{RSS} + \lambda \sum_{j=1}^D w_j^2, \quad (2.3)$$

onde $\lambda \geq 0$ é o parâmetro de regularização e D é a dimensão o vetor de entrada $\mathbf{x}$. O parâmetro λ decide o quanto queremos penalizar a suavidade do nosso modelo. O aumento

na suavidade do modelo é representado pelo aumento de seus coeficientes e, se quisermos minimizar a função acima, esses coeficientes precisam ser pequenos. É assim que o método de regularização *Ridge* (TIKHONOV, 1943) impede que os coeficientes subam demais.

A escolha de um bom λ é muito importante, pois ele controla a variância do modelo. À medida que o valor de λ aumenta, reduz o valor dos coeficientes e, assim, reduz a variação. Até um ponto, este aumento em λ é benéfico, pois está apenas reduzindo a variância (evitando assim o *overfitting*), sem perder quaisquer propriedades importantes nos dados. Mas depois de certo valor, o modelo começa a perder propriedades importantes, dando origem a um modelo incapaz de generalizar bem os dados e, sendo portanto, um modelo mal ajustado.

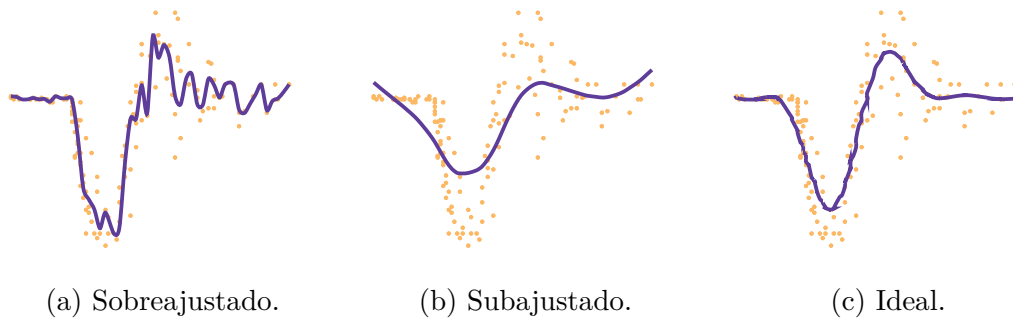


Figura 2 – Efeito da regularização em modelo.

Na Figura 2 temos um exemplo do efeito da regularização de um modelo. Na Figura 2a temos um modelo com *overfitting*, onde ele aprendeu informações irrelevantes para o problema, enquanto a Figura 2b teve muita penalização nos pesos, fazendo com que o modelo resultante não encontre uma função representativa dos dados, ou seja, *underfitting*. E por fim, o modelo da Figura 2c apresenta um modelo que generaliza melhor os dados, consideramos um resultado ideal.

2.3 Métodos de Aproximação de Função

A seguir são apresentados alguns modelos utilizados para tarefas de regressão. Tais modelos serão utilizados posteriormente para comparar os resultados com a proposta deste trabalho.

2.3.1 Optimal Linear Associative Memory

De forma geral, uma memória matricial consiste na geração de um padrão de ativação no conjunto de nós de saída em função da apresentação de um determinado padrão de ativação de entrada. O mapeamento entre o padrão de ativação de entrada e o de saída é feito através de uma transformação matricial, que pode ou não ser acompanhada

de uma operação não-linear, como é o caso do modelo de Willshaw (WILLSHAW, 1971). A arquitetura básica de uma memória matricial é similar à de uma rede neural simples, onde temos um vetor de entrada $\mathbf{x}$ e a saída y conectados através de um vetor de pesos $\mathbf{w}$. Através do vetor de pesos é realizada a transformação entre o padrão de ativação dos neurônios $\mathbf{x}$ para y .

Kohonen e Ruohonen apresentaram uma forma de encontrar a solução ótima para estimação de $\mathbf{w}$ (KOHONEN; RUOHONEN, 1973). Esta solução, conhecida como Memória Associativa Linear Ótima (*Optimal Linear Associative Memory*, OLAM), permite o armazenamento da informação sem armazenar também o *crosstalk*, que é uma medida da forma de representação dos vetores de entrada e de quão ortogonais estes vetores são entre si, e que pode causar perda de generalização e degradação da capacidade de armazenamento das memórias matriciais (BRAGA, 2007).

De maneira geral, o OLAM se baseia na técnica de inversão de matrizes para a solução da equação $\mathbf{X}\mathbf{w} = \mathbf{y}$, onde $\mathbf{w}$ é o vetor de pesos, $\mathbf{X}$ é a matriz com os vetores de entrada $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]$ e $\mathbf{y}$ é o vetor com os valores de saída $\mathbf{y} = [y_1, y_2, \dots, y_n]$, e n é o número de padrões apresentados. Como se deseja obter os parâmetros $\mathbf{w}$, uma vez conhecidos $\mathbf{X}$ e $\mathbf{y}$, a solução ótima para o vetor de pesos é apresentada na Equação (2.4), supondo que a matriz $\mathbf{X}$ é quadrada (o número de amostras é igual ao número de atributos) e seus elementos $\mathbf{x}_n$ são linearmente independentes,

$$\mathbf{w} = \mathbf{X}^{-1}\mathbf{y}, \quad (2.4)$$

onde $\mathbf{X}^{-1}$ é a inversa de $\mathbf{X}$.

No caso em que se tem mais amostras do que atributos, ou seja, a matriz $\mathbf{X}$ não é quadrada, e a independência linear entre os vetores de entrada é preservada, a solução ótima é obtida a partir de

$$\mathbf{w} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}, \quad (2.5)$$

onde $(\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top$ é a pseudoinversa de $\mathbf{X}$.

2.3.2 Rede neural de base radial

As redes neurais de base radial (*radial basis function network*, RBF) (BROOMHEAD; LOWE, 1988) são redes neurais artificiais que possuem como função de ativação a função de base radial. A saída da rede é uma combinação linear de funções de base radial das variáveis de entradas e os neurônios. As redes RBF são utilizadas em várias aplicações, tais como aproximação de função (HEIAT, 2002), previsão de séries temporais (ROJAS et al., 2002) e classificação de padrões (Whei-Min Lin et al., 2001).

Em sua arquitetura mais básica, a rede RBF possui três camadas. A primeira camada é a camada de entrada, na qual possui D neurônios e cada neurônio representa um atributo do vetor de entrada. A segunda camada é chamada de camada oculta, na qual possui K neurônios ocultos, e cada neurônio possui uma função de base radial. Cada neurônio oculto pode ser matematicamente descrito como

$$\phi(\mathbf{x}) = \phi(\|\mathbf{x} - \mathbf{x}_j\|), \quad j = 1, 2, \dots, N, \quad (2.6)$$

onde ϕ é a função de base, $\mathbf{x}_j$ é o j -ésimo centro da função de base radial, e o vetor $\mathbf{x}$ é o padrão de entrada apresentado para a camada de entrada. Por fim, a terceira camada é a camada de saída e para tarefas de regressão, ela possui apenas um neurônio.

Entretanto, nos concentramos no uso de uma função gaussiana como a função de base radial, logo, cada neurônio da camada oculta é definida por

$$\begin{aligned} \phi(\mathbf{x}) &= \phi(\mathbf{x} - \mathbf{x}_j) \\ &= \exp\left(-\frac{1}{2\sigma_j^2}\|\mathbf{x} - \mathbf{x}_j\|^2\right), \quad j = 1, 2, \dots, K, \end{aligned}$$

onde σ_j é o valor da abertura da j -ésima função gaussiana no centro $\mathbf{x}_j$. Geralmente todas as funções gaussianas nos neurônios da camada oculta possuem o mesmo valor de abertura σ , porém não é regra. Em situações desse tipo, o parâmetro que distingue um neurônio oculto do outro é o centro $\mathbf{x}_j$.

Historicamente, as funções de base radial foram introduzidas com o propósito de interpolação de função exata (POWELL, 1987). No entanto, é comum considerarmos que o conjunto de treinamento $\{\mathbf{x}_i, y_i\}_{i=1}^N$ é ruidoso, seja no contexto de classificação de padrões ou de regressão não linear. Infelizmente, o uso de interpolação com base em dados ruidosos pode levar a resultados enganosos, por conta disso, é necessário uma abordagem diferente para o projeto de uma rede RBF.

Um fator importante sobre as redes RBF que precisa de atenção é o tamanho de sua camada oculta. Ter uma camada oculta do mesmo tamanho que a camada de entrada poderia desperdiçar recursos computacionais, principalmente tratando de grande quantidade de amostras de treinamento. Quando cada amostra do conjunto de treinamento é utilizada como centro de cada neurônio da camada oculta, pode ser que qualquer correlação existente entre pontos de dados adjacentes na amostra de treino é correspondentemente transplantado para unidades adjacentes na camada oculta. Ou seja, há redundância de neurônios na camada oculta, em virtude da redundância que pode existir inerentemente na amostra de treinamento. Portanto, é recomendável que K seja menor que o número de padrões do conjunto de treinamento.

Portanto, a função de aproximação realizada pela Rede RBF tem a seguinte forma matemática,

$$F(\mathbf{x}) = \sum_{j=1}^K w_j \phi(\mathbf{x}, \mathbf{x}_j), \quad (2.7)$$

onde a dimensão do vetor $\mathbf{x}$ (e também da camada de entrada) é D enquanto que cada neurônio da cama oculta é descrita como $\phi(\mathbf{x}, \mathbf{x}_j)$, onde $j = 1, 2, \dots, K$, com K menor que N . A camada de saída, assumindo que consiste de apenas uma unidade, possui como coeficiente o vetor de pesos $\mathbf{w}$, cuja dimensionalidade também é K .

A rede RBF utiliza o mesmo método de estimação que o OLAM para a estimação do modelo, isto é, também se baseia na técnica de inversão de matrizes para encontrar uma solução ótima. Portanto, considerando o seguinte sistema de equações lineares $\Phi \mathbf{w} = y$, onde Φ é a matriz de interpolação (e é uma matriz não quadrada), $\mathbf{w}$ é o vetor de pesos e y é o vetor de saída, o vetor de pesos $\mathbf{w}$ pode ser estimado por

$$\mathbf{w} = (\Phi^\top \Phi)^{-1} \Phi^\top y, \quad (2.8)$$

onde $(\Phi^\top \Phi)^{-1} \Phi^\top$ é a pseudoinversa.

2.3.3 Máquina de aprendizagem extrema

A máquina de aprendizagem extrema (*extreme learning machine*, ELM) (HUANG et al., 2004) foi originalmente proposta para as redes neurais *feedforward* de camada única oculta, e posteriormente foi estendida para as redes neurais *feedforward* com uma única camada oculta generalizadas (*generalized single-hidden layer feedforward neural network*, generalized SLFN) (HUANG; CHEN, 2007). Na ELM, a camada oculta não precisa ser ajustada.

A essência da ELM é que se o mapeamento de recurso da camada oculta satisfizer a condição de aproximação universal, a camada oculta da SLFN não precisa ser ajustada de forma iterativa, o que é diferente das SLFNs comuns, como por exemplo, o a rede neural perceptron de multicamadas (*Multi-Layer Perceptron*, MLP). O que isso significa é que no ELM, os pesos da camada oculta podem ser gerados aleatoriamente, porém é preciso que haja uma grande quantidade de neurônios na camada oculta. A função de saída da ELM para SLFNs generalizadas (considerando que há apenas um neurônio na camada de saída) é

$$f_L(\mathbf{x}) = \sum_{i=1}^L \beta_i h_i(\mathbf{x}) = \mathbf{h}(\mathbf{x}) \boldsymbol{\beta}, \quad (2.9)$$

onde $\boldsymbol{\beta} = [\beta_1, \beta_2, \dots, \beta_L]^\top$ é o vetor dos pesos de saída entre a camada oculta de L neurônios e o neurônio de saída e $\mathbf{h}(\mathbf{x}) = [h(\mathbf{x})_1, h(\mathbf{x})_2, \dots, h(\mathbf{x})_L]$ é o vetor de saída da

camada oculta em relação à entrada $\mathbf{x}$. A função $h(\mathbf{x})$ mapeia as amostras do espaço de entrada d -dimensional para o espaço de atributos da camada oculta L -dimensional (espaço de atributos do ELM) $\mathbf{H}$, logo, $h(\mathbf{x})$ é um mapeamento de atributo.

Diferente dos algoritmos tradicionais de treinamento de redes neurais, a ELM tende a obter não somente o menor erro de treinamento, mas também a tende a obter a menor norma da matriz de pesos da saída. De acordo com a teoria de Bartlett (BARTLETT, 1998), para redes neurais *feedforward* que alcançam menor erro de treinamento, quanto menores as normas de pesos, melhor o desempenho de generalização que as redes tendem a ter. Portanto, o ELM busca minimizar o erro de treinamento, bem como a norma dos pesos de saída,

$$\text{minimizar: } \|\mathbf{H}\boldsymbol{\beta} - \mathbf{Y}\| \text{ e } \|\boldsymbol{\beta}\|, \quad (2.10)$$

nos quais

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}(\mathbf{x}_1) \\ \vdots \\ \mathbf{h}(\mathbf{x}_N) \end{bmatrix} = \begin{bmatrix} h_1(\mathbf{x}_1) & \cdots & h_L(\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ h_1(\mathbf{x}_N) & \cdots & h_L(\mathbf{x}_N) \end{bmatrix}, \boldsymbol{\beta} = \begin{bmatrix} \beta_1^\top \\ \vdots \\ \beta_L^\top \end{bmatrix} \text{ e } \mathbf{Y} = \begin{bmatrix} y_1 \\ \vdots \\ y_L \end{bmatrix}, \quad (2.11)$$

onde $\mathbf{H}$ é a matriz de saída da camada oculta, $\boldsymbol{\beta}$ é o vetor de coeficiente de regressão a ser resolvido e $\mathbf{Y}$ é o vetor com as saídas esperadas. O método de mínimos quadrados, em vez do método de otimização padrão, foi usado na implementação original do ELM,

$$\hat{\boldsymbol{\beta}} = \mathbf{H}^\dagger \mathbf{Y}, \quad (2.12)$$

e $\mathbf{H}^\dagger$ é a inversa generalizada de *Moore-Penrose* da matrix $\mathbf{H}$. Portanto, para um novo padrão $\mathbf{x}$, e após calcular $\mathbf{H}$, a estimativa do seu valor de saída $\hat{y}$ pode ser escrita como

$$\hat{y} = \mathbf{H}\hat{\boldsymbol{\beta}}. \quad (2.13)$$

Diferentes métodos podem ser utilizados para calcular a inversa generalizada de *Moore-Penrose* de uma matriz: método de projeção ortogonal, método de ortogonalização, método iterativo e decomposição de valores singulares (*singular value decomposition*, SVD).

2.3.4 Máquina de vetores-suporte de mínimos quadrados

A máquina de vetores-suporte de mínimos quadrados (*least-squares support vector machine*, LS-SVM) é a versão de mínimos quadrados da máquina de vetores-suporte (*support vector machine*, SVM), nos quais ambos são métodos de aprendizagem supervisionada utilizados em tarefas de classificação e aproximação de função. A SVM de mínimos quadrados foi proposta por Suykens (SUYKENS et al., 2002) e assim como as SVM, o LS-SVM é um método de aprendizado baseado em *kernel*.

Originalmente, o LS-SVM foi proposto para solucionar problemas de classificação. Para tarefas de aproximação de função, Saunders, Gammerman e Vovk (1998) apresentaram

uma interpretação do vetor-suporte de regressão *ridge* (GOLUB; LOAN, 2012), que considera restrições de tipo de igualdade em vez de desigualdades da abordagem clássica do SVM. Nesta versão de mínimos quadrados a solução do problema é obtida através da resolução de um conjunto de equações lineares em vez de um problema de programação quadrática (*quadratic programming*, QP) para SVMs clássicos. Enquanto na versão clássica do SVM muitos valores de suporte são iguais a zero (valores diferentes de zero correspondem a vetores-suporte), no LS-SVM os valores de suporte são proporcionais aos erros.

Dado um conjunto de treinamento $\{\mathbf{x}_i, y_i\}_{i=1}^N$, o LS-SVM é obtido pela reformulação do problema de minimização do SVM,

$$\min_{\mathbf{w}, \mathbf{e}} \mathcal{J}(\mathbf{w}, \mathbf{e}) = \frac{1}{2} \mathbf{w}^\top \mathbf{w} + \gamma \frac{1}{2} \sum_{i=1}^N e_i^2, \quad (2.14)$$

sujeito às seguintes restrições de igualdade,

$$y_i = \mathbf{w}^\top \varphi(\mathbf{x}_i) + b + e_i, \quad i = 1, \dots, N, \quad (2.15)$$

onde b é o viés, $\varphi(x)$ é um mapeamento não-linear do espaço original para um espaço de alta dimensão e γ é o hiperparâmetro utilizado para ajustar relação entre a regularização e o erro e do modelo.

Por conta da modificação das restrições de desigualdade para de igualdade, os valores de suporte podem assumir tanto valores positivos como também negativos. Portanto, a solução do LS-SVM para problemas de regressão será obtida após a construção da função lagrangeana,

$$\mathcal{L}(\mathbf{w}, b, \mathbf{e}; \boldsymbol{\alpha}) = \mathcal{J}(\mathbf{w}, \mathbf{e}) - \sum_{i=1}^N \alpha_i \{ \mathbf{w}^\top \varphi(x_i) + b + e_i - y_i \}, \quad (2.16)$$

onde $\{\alpha_i\}_{i=1}^N$ são os multiplicadores de Lagrange, que podem ser positivos ou negativos devido às restrições de igualdade, como segue as condições de Karush-Kuhn-Tucker (FLETCHER, 1987). Portanto, as condições de otimalidade são

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial w} = 0 & \rightarrow & w = \sum_{i=1}^N \alpha_i \varphi(x_i), \\ \frac{\partial \mathcal{L}}{\partial b} = 0 & \rightarrow & \sum_{i=1}^N \alpha_i = 0, \\ \frac{\partial \mathcal{L}}{\partial e_i} = 0 & \rightarrow & \alpha_i = \gamma e_i, \quad i = 1, \dots, N, \\ \frac{\partial \mathcal{L}}{\partial \alpha_i} = 0 & \rightarrow & w^\top \varphi(x_i) + b + e_i - y_i, \quad i = 1, \dots, N. \end{cases}$$

A partir disso, pode-se definir um sistema linear em vez de um problema de programação quadrática, em que

$$\begin{bmatrix} 0 & \mathbf{1}^\top \\ \mathbf{1} & \Omega + \gamma^{-1} \mathbf{I} \end{bmatrix} \begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{y} \end{bmatrix}, \quad (2.17)$$

onde $\mathbf{y} = [y_1, y_2, \dots, y_N]^\top$, $\mathbf{1} = [1, 1, \dots, 1]^\top$ e $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \dots, \alpha_N]^\top$. Neste caso, $\mathbf{I}$ é a matriz identidade de dimensão $N \times N$, e $\Omega \in \mathbb{R}^{N \times N}$ é a matriz de *kernel*, definida por $\Omega_{i,j} = \varphi(\mathbf{x}_i)^\top \varphi(\mathbf{x}_j) = K(\mathbf{x}_i, \mathbf{x}_j)$.

Portanto, após encontrar os valores ótimos de $\boldsymbol{\alpha}$ através do método de mínimos quadrados, o modelo LS-SVM resultante para estimativa de função é

$$y(\mathbf{x}) = \sum_{i=1}^N \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b. \quad (2.18)$$

Solução matematicamente equivalente a redes de regularização e processos gaussianos (geralmente sem termo de viés).

2.4 Classificação

A classificação é outro tipo de problema do aprendizado supervisionado. O objetivo aqui é a de identificar à qual classe o padrão de entrada pertence. Na etapa de treinamento, é apresentado ao algoritmo um conjunto de dados de entrada com seus respectivos rótulos (classes), para que então seja encontrada uma função capaz de identificar as classes de novos padrões não vistos.

Por exemplo, em raios-X de mamografia, recebemos uma imagem em que uma região indica a existência de um tumor. O objetivo de um sistema de diagnóstico auxiliado por computador é prever se esse tumor corresponde à classe benigna ou maligna (THEODORIDIS, 2015).

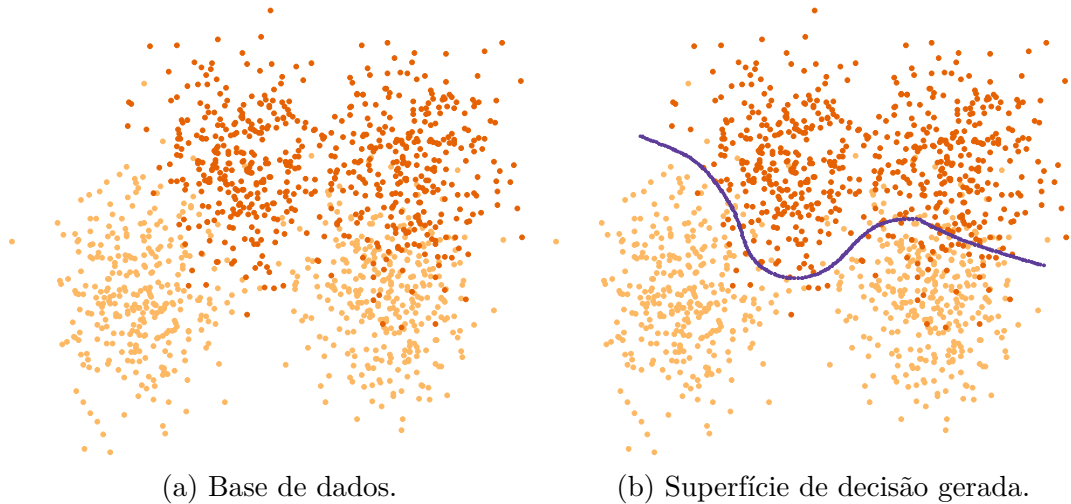


Figura 3 – Exemplo de um problema de classificação de duas classes.

2.5 Agrupamento de dados

Na área de aprendizagem de máquina existe um outro tipo de tarefa para a qual os dados de treinamento não possuem um rótulo associado. Nesse tipo de problema, recebemos um conjunto de dados $\mathbf{X}$ e o objetivo é encontrar as semelhanças subjacentes e grupos (clusters) de padrões "semelhantes". O uso desse processo é apropriado quando se conhece pouco ou nada sobre a estrutura de um conjunto de dados.

Isso é conhecido como aprendizado não supervisionado ou agrupamento (clustering). Tais tarefas surgem em muitas aplicações em redes sociais, ciências e engenharia, como sensoriamento remoto, segmentação de imagens e codificação de fala (THEODORIDIS; KOUTROUMBAS, 2008).

Uma questão importante no aprendizado não supervisionado é a de definir a “similaridade” entre dois vetores de características e a escolha de uma métrica apropriada. Outra questão de importância é a escolha de um algoritmo que agrupe os vetores com base na medida de similaridade adotada. Em geral, diferentes algoritmos podem levar a resultados diferentes, que o especialista deve interpretar.

2.5.1 Algoritmo k -médias

O algoritmo k -médias, proposto por MacQueen (1967), é um algoritmo simples de quantização vetorial e agrupamento iterativo que particiona N pontos de dados em K subconjuntos disjuntos, denotados por $S = \{S_k\}_{k=1}^K$, de modo a minimizar o critério da soma dos quadrados. Formalmente, tal algoritmo minimiza a seguinte função objetivo:

$$J = \sum_{k=1}^K \sum_{\mathbf{x}_n \in S_k} \|\mathbf{x}_n - \mu_k\|^2, \quad (2.19)$$

em que $\mathbf{x}_n$ é o n -ésimo vetor pertencente ao conjunto S_k , μ_k é centróide geométrico dos pontos de dados pertencentes a S_k .

O algoritmo mais comum usa uma técnica de refinamento iterativo. Também referido como algoritmo de Lloyd (LLOYD, 1982), este algoritmo consiste em um simples processo de reestimação em duas etapas:

- i. Atribuição: Os pontos de dados são atribuídos ao grupo cujo centróide é o mais próximo a cada ponto e,
- ii. Atualização: Cada centróide do grupo é recalculado para o centro (média) de todos os pontos de dados atribuídos a ele.

Estas duas etapas são alternadas até que um critério de parada seja alcançado, de modo que não haja mais alterações na atribuição de pontos de dados a algum grupo.

Toda iteração requer $N \times K$ comparações, representando a complexidade do tempo de uma iteração. O Algoritmo 1 apresenta o pseudocódigo de uma possível implementação do método k -médias.

Algorithm 1 Algoritmo k -médias.

Entrada:

- Conjunto de pontos de treinamento de entrada $\{\mathbf{x}_n \in \mathbb{R}^D\}_{n=1}^N$
- K grupos

Saída:

- $\{\mu_k \in \mathbb{R}^D\}_{k=1}^K$: conjunto de centroides

Procedimento:

1. Inicializar os K centroides dos grupos $\{\mu_k\}_{k=1}^K \in \mathbb{R}^D$ aleatoriamente
 2. **Enquanto** não convergir **faça**
 3. **Para** $n \leftarrow 1$ **até** N **faça**
 4. $c_n \leftarrow \arg \min_k \|\mathbf{x}_n - \mu_k\|^2, \quad 1 \leq k \leq K$
 5. **Para** $k \leftarrow 1$ **até** K **faça**
 6. $\mu_k \leftarrow \frac{\sum_{n=1}^N 1\{c_n = k\} \mathbf{x}_n}{\sum_{n=1}^N 1\{c_n = k\}}$
 7. **Retorna** $\{\mu_k\}_{k=1}^K, \{c_n\}_{n=1}^N$
-

2.5.2 Algoritmo fuzzy c -means

O algoritmo *fuzzy c -means*, proposto por Dunn (1973), e posteriormente melhorado por Bezdek (1981), é um método de agrupamento que permite que uma observação pertença a dois ou mais grupos. Este algoritmo é bastante utilizado na área de reconhecimento de padrões. O método baseia-se na minimização da seguinte função objetivo:

$$\arg \min_C \sum_{i=1}^N \sum_{j=1}^C u_{ij}^m \|\mathbf{x}_i - \mathbf{c}_j\|^2, \quad 1 < m < \infty \quad (2.20)$$

onde m é qualquer número real maior do que 1, u_{ij} é o grau de associação de $\mathbf{x}_i$ ao cluster j , $\mathbf{x}_i$ é o i -ésimo padrão d -dimensional, $\mathbf{c}_j$ é centróide d -dimensional do cluster j , e $\|\cdot\|$ é a norma euclidiana, na qual expressa a semelhança entre os dados medidos e o centróide.

O particionamento *fuzzy* é realizado através de uma otimização iterativa da função objetivo mostrada acima, com a atualização da associação u_{ij} e dos centróides $\mathbf{c}_j$ por:

$$u_{ij} = \frac{1}{\sum_{k=1}^C \left(\frac{\|\mathbf{x}_i - \mathbf{c}_j\|}{\|\mathbf{x}_i - \mathbf{c}_k\|} \right)^{\frac{2}{m-1}}} \quad (2.21)$$

e

$$c_k = \frac{\sum_{i=1}^N u_{ij}^m \cdot \mathbf{x}_i}{\sum_{i=1}^N u_{ij}^m}. \quad (2.22)$$

Esta iteração se encerrará quando $\max_{ij} \left\{ |u_{ij}^{(k+1)} - u_{ij}^{(k)}| \right\} < \epsilon$, onde ϵ é o critério de parada entre 0 e 1, considerando que k é cada iteração e este procedimento converge para um mínimo local. O algoritmo é apresentado a seguir.

Algorithm 2 Algoritmo fuzzy c -means

Entrada:

- Conjunto de pontos de treinamento de entrada $\{\mathbf{x}_n \in \mathbb{R}^D\}_{n=1}^N$
- K grupos

Saída:

- $\mathbf{U}$: matriz de associações
- $\{c_j \in \mathbb{R}^D\}_{j=1}^J$: conjunto de centroides

Procedimento:

1. Inicializar a matriz $\mathbf{U} = [u_{ij}]$, $\mathbf{U}^{(0)}$
 2. No k -ésimo passo: calcular os centróides $C^{(k)} = [c_j]$ com $\mathbf{U}^{(k)}$ utilizando a Equação (2.22)
 3. Atualizar $\mathbf{U}^{(k)}$, $\mathbf{U}^{(k+1)}$ utilizando a Equação (2.21)
 4. Se $\|\mathbf{U}^{(k+1)} - \mathbf{U}^{(k)}\| < \epsilon$, então parar; Senão, retornar ao passo 2.
-

2.6 Conclusão

Este capítulo apresenta a teoria básica sobre alguns problemas de aprendizagem supervisionada e não supervisionada. O principal foco aqui é o entendimento sobre os problemas de classificação e regressão de dados, para a aprendizagem supervisionada, e problemas de agrupamento de dados, para a aprendizagem não supervisionada.

Além dos conceitos desses tipos de problemas de aprendizagem de máquinas, foram apresentados alguns algoritmos disponíveis na literatura que serão necessários nos próximos capítulos, tanto para a utilização de alguns deles em nossas propostas, como também para validar a qualidade das propostas em experimentos com bases de dados reais.

3 Máquina de Aprendizagem Mínima

Neste capítulo serão apresentados os modelos propostos desta dissertação, nos quais buscam melhorar em alguns aspectos o treinamento da máquina de aprendizagem mínima (*minimal learning machine*, MLM)(SOUZA et al., 2013). Portanto, neste capítulo serão apresentas os fundamentos do modelo MLM (sua formulação básica, treinamento e estimação da saída) e algumas de suas variações que estão presentes na literatura.

3.1 Máquinas de aprendizagem mínima

A MLM é um método de aprendizagem supervisionada, que busca encontrar o mapeamento linear entre o espaço de distância das entradas para o espaço de distância das saídas. O seu treinamento consiste na resolução de um sistema linear de múltiplas respostas a fim de obter o supracitado mapeamento. A predição de novas amostras se dá usando as distâncias no espaço da entrada e o mapeamento encontrado para estimar as distâncias no espaço da saída, e aplica-se, então, algum método que permita recuperar uma estimativa para a saída a partir das distâncias computadas.

3.1.1 Formulação básica

Vamos definir o problema de aprendizagem como o problema de aproximar uma função alvo $f : \mathcal{X} \rightarrow \mathcal{Y}$ dado por $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i = f(\mathbf{x}_i))\}_{i=1}^N$, onde $\mathcal{X} \in \mathbb{R}^D$ e $\mathcal{Y} \in \mathbb{R}^S$. O objetivo da MLM é de estimar uma aproximação $g(\cdot)$ para a função $f(\cdot)$ a partir das amostras do conjunto $\mathcal{D}$ através do espaço de distância. Por isso, o problema é descrito como as matrizes de distâncias de cada ponto de $\mathcal{D}$, chamadas de $\mathbf{D}$ e Δ . Ambos representam a distância euclidiana (com a notação de $d(\cdot, \cdot)$) de cada ponto de $\mathcal{D}$ para o i -ésimo ponto de referência de $\mathcal{D}$, ou seja, $\mathbf{D}_{i,j} = d(\mathbf{x}_i, \mathbf{x}_j)$ e $\Delta_{i,j} = d(\mathbf{y}_i, \mathbf{y}_j)$, e ambas são matrizes de dimensão $N \times N$. Além disso, nós chamamos $\mathbf{D}$ e Δ de matrizes de distância no espaços de entrada e saída, respectivamente.

Logo, assumindo que o mapeamento entre as matrizes de distância tem uma estrutura linear para cada resposta, o modelo MLM pode ser reescrito na forma:

$$\min_{\mathbf{B}} \mathcal{J}(\mathbf{B}) = \|\mathbf{DB} - \Delta\|_{\mathcal{F}}^2 \quad (3.1)$$

$$\text{sujeito a } \Phi(\mathbf{x}_i)\mathbf{B}_{*,i} = \Psi(\mathbf{y}_i) + \mathbf{E}_{i,*} \quad (3.2)$$

onde $\Phi(\mathbf{x}_i) = \mathbf{D}_{i,*}$, $\Psi(\mathbf{y}_i) = \Delta_{i,*}$, $\mathbf{B}$ é uma matriz $N \times N$ e corresponde aos coeficientes do sistema para as N respostas, $\|\cdot\|_{\mathcal{F}}$ é a norma de Frobenius e $\mathbf{E}$ é uma matriz de resíduos.

3.1.2 Treinamento

A etapa de aprendizagem consiste em estimar o mapeamento entre as distâncias no espaço de entrada e de saída. Supondo que tal mapeamento tenha uma estrutura linear para cada resposta, o modelo de regressão pode ser descrito como:

$$\Delta = \mathbf{D}\mathbf{B} + \mathbf{E}, \quad (3.3)$$

onde $\mathbf{B}$ é uma matriz de dimensão $N \times N$ e corresponde aos coeficientes do sistema para as N respostas. Uma forma de estimar a matriz $\mathbf{B}$ é através da minimização da soma dos quadrados dos resíduos (*residual squared sum*, RSS), dada por:

$$RSS(\mathbf{B}) = \text{tr}((\Delta - \mathbf{D}\mathbf{B})^\top (\Delta - \mathbf{D}\mathbf{B})). \quad (3.4)$$

Como a matriz $\mathbf{D}$ é uma matriz quadrada de dimensão $N \times N$, o sistema linear possui somente uma solução, na qual a matriz $\mathbf{B}$ é obtida por:

$$\hat{\mathbf{B}} = \mathbf{D}^{-1}\Delta. \quad (3.5)$$

3.1.3 Estimativa da saída

Estimar a saída para novos padrões refere-se principalmente a projetar o novo padrão $\mathbf{x}^*$ através do mapeamento $\mathbf{B}$ e estimar a imagem de tal projeção. Portanto, é necessário que o padrão $\mathbf{x}^*$ seja também representado no domínio das distâncias para que possamos representá-lo no espaço de saída. Tal representação é obtida por

$$\mathbf{d}_{y^*} = \phi(\mathbf{x}^*)\mathbf{B}. \quad (3.6)$$

Portanto, o problema se torna em estimar a imagem $\mathbf{y}^* = f(\mathbf{x}^*)$, a partir de $\mathbf{d}_{y^*}$ e das imagens dos pontos de referência de $\mathcal{D}$. Este problema pode ser entendido como um problema de multilateração (NIEWIADOMSKA-SZYNKIEWICZ; MARKS, 2009). O problema de localizar $\mathbf{y}^*$ pertencente ao conjunto $\mathbb{R}^S$ é equivalente a resolver um conjunto sobredeterminado de N (número de pontos de referência em $\mathcal{D}$) equações não-lineares correspondentes à hiper-esferas $(N + 1)$ -dimensionais centradas sobre as imagens dos pontos de referência, denotados por $\{\Psi(\mathbf{y}_n)\}_{n=1}^N$. A localização de $\mathbf{y}^*$ pode ser estimada através da minimização da seguinte função custo:

$$f(\mathbf{x}) = \arg \min_{\mathbf{y}} \|\Psi(\mathbf{y}) - \Phi(\mathbf{x})\mathbf{B}\|_2. \quad (3.7)$$

A Figura 4 representa graficamente o problema para 2 dimensões.

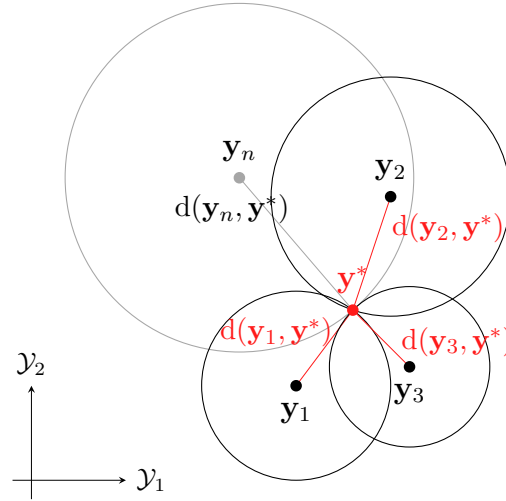


Figura 4 – Estimação da saída $\mathbf{y}^*$. Esse vetor representa o ponto que se deseja estimar através da multilateração, $\{\mathbf{y}_n\}_{n=1}^N$ são as imagens dos pontos de referência $\mathcal{D}$ e $d(\mathbf{y}_n, \mathbf{y}^*)$ é a distância de saída estimada das imagens dos pontos de referência até o rótulo $\mathbf{y}^*$.

Fonte – Proposta original do MLM, apresentada no trabalho de Souza et al. (2015).

Em problemas de classificação, Mesquita, Gomes e Souza Júnior (2017) apresentam em seu trabalho uma variação da MLM, chamada de *Nearest Neighbor* MLM (NN-MLM), onde prova que o ponto de mínimo da Equação (3.7) é o rótulo do ponto de referência mais próximo de $\mathbf{d}_{\mathbf{y}^*}$.

3.1.4 Resumo para o algoritmo do MLM

O modelo MLM é composto por uma fase de treinamento e outra de predição. A primeira é a de treinamento, que é responsável pelo cômputo das matrizes de distâncias $\mathbf{D}$ e Δ e estimação do mapeamento $\hat{\mathbf{B}}$ entre respectivas matrizes. Todo esse processo é realizado a partir do conjunto de dados de treinamento sendo o mesmo descrito no Algoritmo 3. A etapa de predição consiste na estimação do rótulo para uma dado padrão não visto no conjunto de treinamento. Esse processo é apresentado no Algoritmo 4.

Algorithm 3 Processo de treinamento do modelo MLM

Entrada:

- Conjunto de pontos de treinamento de entrada $\mathcal{X}$ e saída $\mathcal{Y}$

Saída:

- Mapeamento $\hat{\mathbf{B}}$

Procedimento:

1. Calcular $\Phi(\mathbf{x}_i)$ (ou seja, $\mathbf{D}$): matriz de distâncias no espaço de entrada entre $\mathcal{X}$ e $\mathcal{X}$.
 2. Calcular $\Psi(\mathbf{y}_i)$ (ou seja, Δ): matriz de distâncias no espaço de saída entre $\mathcal{Y}$ e $\mathcal{Y}$.
 3. Calcular o mapeamento $\hat{\mathbf{B}}$ entre $\Phi(\mathbf{x}_i)$ e $\Psi(\mathbf{y}_i)$ através da Equação (3.5).
-

Algorithm 4 Processo de estimação da saída do modelo MLM

Entrada:

- Novo padrão $\mathbf{x}^*$, pontos de referência de entrada $\mathcal{X}$ e saída $\mathcal{Y}$, mapeamento $\hat{\mathbf{B}}$

Saída:

- Estimação de saída $\mathbf{y}^*$ para o novo padrão apresentado

Procedimento:

1. Calcular $\mathbf{d}_{y^*} = \phi(\mathbf{x}^*)\hat{\mathbf{B}}$
 2. Estimar a saída $\mathbf{y}^*$ resolvendo o problema de otimização da Equação (3.7).
-

3.1.5 Variações da MLM

Na formulação básica apresentada anteriormente, a MLM não possui nenhum hiperparâmetro a ser ajustado, porém, há uma variante apresentada na publicação original (SOUZA et al., 2013) na qual utiliza um subconjunto aleatório de $\mathcal{D}$ como pontos de referência, em que é ajustado pelo hiperparâmetro K , que determina o número de pontos que serão escolhidos, a fim de evitar o sobreajuste do modelo e sua complexidade. Chamaremos esta variante de *Random MLM*. Posteriormente, uma nova variante da MLM foi apresentada por Alencar et al. (2015). Nesta versão, além do hiperparâmetro K para selecionar K padrões aleatórios para serem utilizados como pontos de referência, foi adicionado um parâmetro de regularização C , com o intuito de suavizar a função aprendida e ponderar o erro. Esta versão é chamada de *Rank MLM*.

3.1.5.1 *Random MLM*

O primeiro passo é a seleção aleatória de um subconjunto de dados retirado a partir do conjunto total de dados $\mathcal{D}$. As entradas deste subconjunto de dados é representado por $\mathcal{R} = \{\mathbf{m}_k\}_{k=1}^K$ e suas respectivas saídas são representadas por $\mathcal{T} = \{\mathbf{t}_k\}_{k=1}^K$, em que $\mathcal{R} \subseteq \mathcal{X}$ e $\mathcal{T} \subseteq \mathcal{Y}$.

É definido $\mathbf{D}_x \in \mathbb{R}^{N \times K}$ tal que a k -ésima coluna contenha as distâncias $d(\mathbf{x}_n, \mathbf{m}_k)$ entre os N pontos de entrada $\mathbf{x}_n$ e o k -ésimo ponto de referência $\mathbf{m}_k$. Da mesma forma é definida a matriz $\Delta_y \in \mathbb{R}^{N \times K}$, tal que a k -ésima coluna contém as distâncias $d(\mathbf{y}_n, \mathbf{t}_k)$ entre os N pontos de saída $\mathbf{y}_n$ e a saída $\mathbf{t}_k$ do k -ésimo ponto de referência.

O modelo de regressão multiresposta é escrito da mesma maneira que foi apresentada para a versão original na Equação (3.3). Porém, quando o número de pontos de referência selecionados é menor que o número de pontos disponíveis, $\mathbf{B}$ passa a ser estimada por:

$$\hat{\mathbf{B}} = (\mathbf{D}_x^\top \mathbf{D}_x)^{-1} \mathbf{D}_x^\top \Delta_y. \quad (3.8)$$

Após a etapa de treinamento ser concluída, a estimação da saída é a mesma que foi apresentada anteriormente na Equação (3.7).

Algorithm 5 Processo de treinamento do modelo *Random* MLM

Entrada:

- Conjunto de pontos de treinamento de entrada $\mathcal{X}$ e saída $\mathcal{Y}$, K

Saída:

- Mapeamento $\hat{\mathbf{B}}$, pontos de referência $\mathcal{R}$ e $\mathcal{T}$

Procedimento:

1. Escolher aleatoriamente K pontos de referência $\mathcal{R} \subseteq \mathcal{X}$ e sua respectiva saída $\mathcal{T} \subseteq \mathcal{Y}$
 2. Calcular $\mathbf{D}$: matriz de distâncias no espaço de entrada entre $\mathcal{X}$ e $\mathcal{R}$.
 3. Calcular Δ : matriz de distâncias no espaço de saída entre $\mathcal{Y}$ e $\mathcal{T}$.
 4. Calcular o mapeamento $\hat{\mathbf{B}}$ entre $\mathbf{D}$ e Δ através da Equação (3.8).
-

3.1.5.2 *Rank* MLM

O *Rank* MLM recebe como entrada um vetor $\mathbf{x}_i$ e um valor $\mathbf{y}_i$ associado a este vetor. Com o intuito de controlar a complexidade do modelo MLM e melhorar sua capacidade de generalização, um termo de regularização é adicionado à função custo (Equação (3.1)). Portanto, a função de custo do *Rank* MLM é dada por

$$\min_{\mathbf{B}} \mathcal{J}(\mathbf{B}) = \|\mathbf{D}\mathbf{B} - \Delta\|_{\mathcal{F}} + C\|\mathbf{B}\|_{\mathcal{F}}, \quad (3.9)$$

onde C é o parâmetro de regularização.

Minimizando a Equação (3.9) resulta em uma nova função para estimar o valor de $\mathbf{B}$, dada por:

$$\hat{\mathbf{B}} = (\mathbf{D}^{\top}\mathbf{D} - C\mathbf{I})^{-1}\mathbf{D}^{\top}\Delta, \quad (3.10)$$

onde $\mathbf{I}$ é uma matriz identidade de dimensão $K \times K$.

Por fim, o treinamento do *Rank* MLM consiste na escolha aleatória de K pontos de referência, criando um subconjunto pertencente ao conjunto $\mathcal{D}$ e a estimação de $\mathbf{B}$ a partir da Equação (3.10).

Algorithm 6 Processo de treinamento do modelo *Rank* MLM

Entrada:

- Conjunto de pontos de treinamento de entrada $\mathcal{X}$ e saída $\mathcal{Y}$, K , C

Saída:

- Mapeamento $\hat{\mathbf{B}}$, pontos de referência $\mathcal{R}$ e $\mathcal{T}$

Procedimento:

1. Escolher aleatoriamente K pontos de referência $\mathcal{R} \subseteq \mathcal{X}$ e sua respectiva saída $\mathcal{T} \subseteq \mathcal{Y}$
 2. Calcular $\mathbf{D}$: matriz de distâncias no espaço de entrada entre $\mathcal{X}$ e $\mathcal{R}$.
 3. Calcular Δ : matriz de distâncias no espaço de saída entre $\mathcal{Y}$ e $\mathcal{T}$.
 4. Calcular o mapeamento $\hat{\mathbf{B}}$ entre $\mathbf{D}$ e Δ através da Equação (3.10).
-

3.1.6 MLM aplicado a problemas de classificação

Em problemas de classificação, considera-se um conjunto de dados com N de amostras $\mathcal{X} = \{\mathbf{x}_n\}_{n=1}^N$, em que $\mathbf{x}_n \in \mathcal{X}$ e um conjunto correspondente de rótulos $L = \{l_n\}_{n=1}^N$, com $l_n \in \{C_s\}_{s=1}^S$, com C_s denota a s -ésima classe.

A MLM pode ser estendida a problemas de classificação de maneira direta, através da representação das S classes com a utilização do esquema de codificação 1-dentre- S . Em tal abordagem, uma variável qualitativa com S símbolos é representada por um vetor de S variáveis binárias (ou bits), com somente uma das componentes “ativa” a cada momento (MESQUITA; GOMES; Souza Júnior, 2017).

Na classificação de uma amostra $\mathbf{x}$ com classe $l \in \{C_s\}_{s=1}^S$ desconhecida, a classe estimada $\hat{l}$ associada a saída estimada $\hat{\mathbf{y}}$, encontrado pelo método tradicional do MLM é $\hat{l} = C_{s^*}$, onde

$$s^* = \arg \max_{s=1, \dots, S} \left\{ \hat{y}^{(s)} \right\}, \quad (3.11)$$

e $\hat{y}^{(s)}$ denota a s -ésima componente do vetor $\hat{\mathbf{y}}$.

3.1.6.1 MLM de vizinho mais próximo

O procedimento de estimação de saída do método MLM é a etapa de maior complexidade computacional, uma vez que existe a necessidade de solucionar um problema de otimização. Assim, a redução da complexidade computacional nesta etapa é particularmente valiosa. No que diz respeito ao caso de classificação, pode-se notar que a natureza finita e discreta dos resultados possíveis para a previsão de classe torna o procedimento de estimativa de saída mais fácil.

A fase de estimação da saída visa encontrar a melhor estimativa para a saída de uma determinada entrada $\mathbf{x}$, a partir das distâncias estimadas entre este ponto e o conjunto de pontos de referência $\{\hat{\delta}(\mathbf{y})\}_{k=1}^K$ no espaço de saída. Em tarefas de classificação, o conjunto de valores possíveis para $\hat{\mathbf{y}}$ é limitado ao número de classes. Dessa forma, uma alternativa à solução do problema de otimização apresentado na Equação (3.7) é adotar a saída (rótulo) associada ao ponto de referência mais próximo no espaço de saída (MESQUITA; GOMES; Souza Júnior, 2017).

A MLM de vizinho mais próximo (*nearest neighbor minimal learning machine*, NN-MLM) estima a saída a partir do rótulo associado ao ponto de referência mais próximo no espaço de saída. A principal vantagem do NN-MLM é a não dependência de métodos baseados em gradiente, acelerando assim a fase de testes do MLM sem comprometer sua acurácia. Dessa forma, a classe estimada $\hat{l}$, a partir de um padrão de entrada $\mathbf{x}$, é dada por $\hat{l} = l_{k^*}$, em que

$$k^* = \arg \min_{k=1, \dots, K} \left\{ \hat{\delta}^2(\mathbf{y}, \mathbf{t}_k) \right\}, \quad (3.12)$$

e l_{k^*} representa o rótulo do k -ésimo ponto de referência.

3.1.6.2 Análise empírica da seleção de pontos de referência

Bem como em máquinas de vetores suporte (*support vector machines*, SVM) (VAPNIK, 1995) e máquinas de vetores de relevância (*relevance vector machines*, RVM) (TIPPING, 2001), na etapa de predição, o MLM requer um subconjunto de pontos selecionados a partir do conjunto de dados. Todavia, diferentemente dos classificadores SVM e RVM, a escolha de tais pontos não é um subproduto direto da sua etapa de treinamento. Neste caso, tais pontos (chamados de pontos de referência) são selecionados na etapa inicial do processo de treinamento. Na proposição original do MLM, a escolha destes pontos é realizada de maneira aleatória, deixando a cargo do usuário apenas a quantidade de pontos a serem utilizados.

Embora a escolha aleatória traga bons resultados numéricos na grande maioria dos casos, essa abordagem não traz garantias de eficácia. No caso em que o valor de K é pequeno, a escolha aleatória pode resultar em um conjunto de pontos de referência não representativo para a base de dados completa. Isso fará com que o modelo criado não seja complexo o bastante, configurando uma condição de subajustamento (*underfitting*). Por outro lado, a utilização de um valor grande para K , ou mesmo a utilização de todo o conjunto de treinamento como pontos de referência, faz com que o modelo criado seja muito complexo, tendo a ocorrência de um superajustamento (*overfitting*). Além disso, o modelo criado não tem nenhuma característica de esparsidade. Na Figura 5 são ilustrados exemplos de superfícies de decisão geradas por modelos subajustados, sobreajustados e modelos com ajustamento ideal.

Como pode ser observado na Figura 5a, utilizar um valor pequeno para K e realizar a escolha de forma aleatória pode acarretar em um subconjunto não representativo, desconsiderando regiões possivelmente importantes. De forma contrária, com a utilização de todos os pontos do conjunto de dados como pontos de referência, cria-se uma superfície de decisão bastante complexa, pode ser observado na Figura 5b. A escolha ideal é fazer com que os pontos de referência não se encontrem em regiões de confusão, ou seja, regiões de sobreposição de dados. Na Figura 5c é apresentada a escolha ideal de um subconjunto de pontos de referência. É interessante observar que o número de pontos nas figuras 5a e 5c é o mesmo.

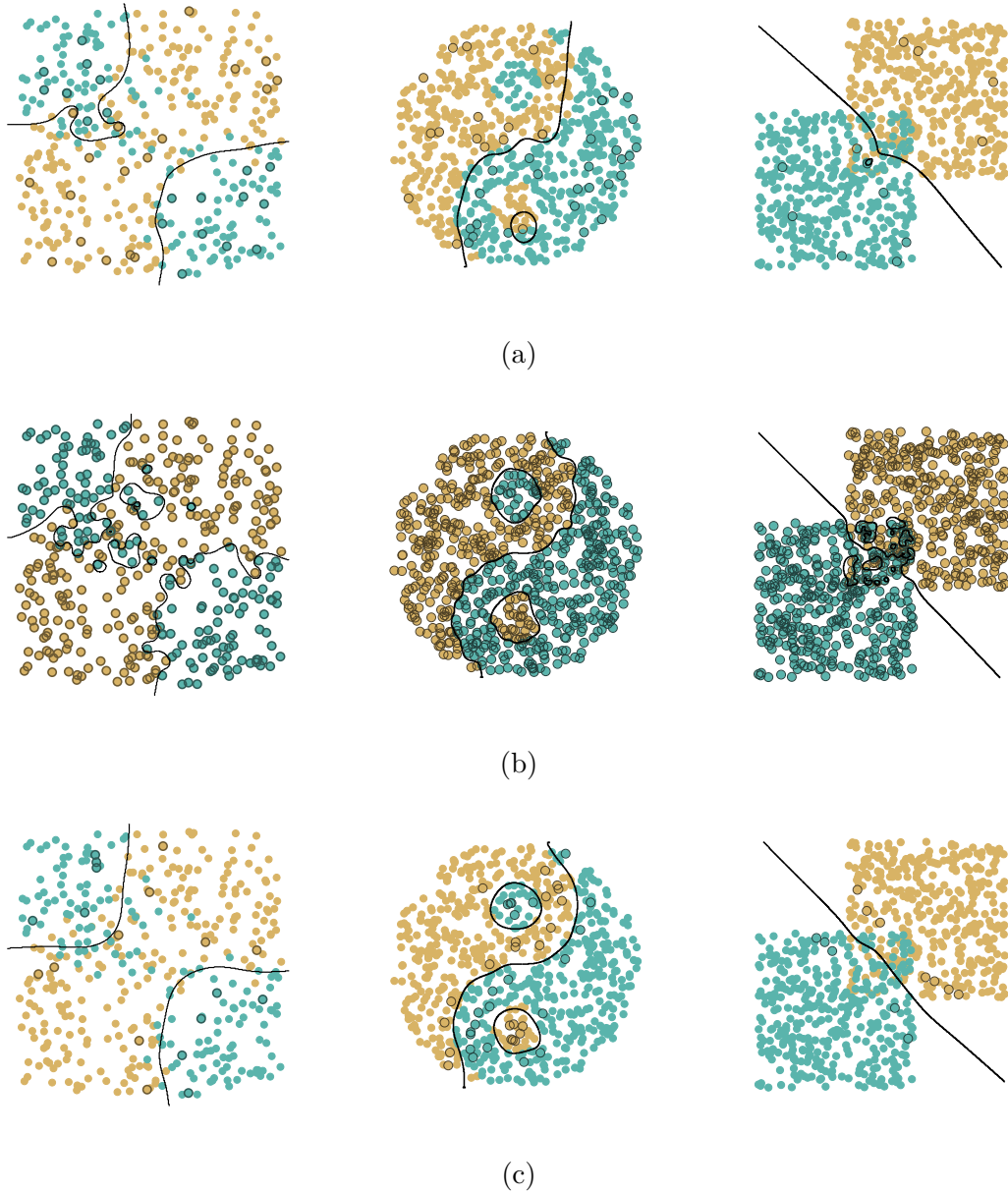


Figura 5 – Superfícies de decisão (a) subajustada, (b) sobreajustada e (c) ideal.

3.2 *Overlap removed* MLM (OR-MLM) (Proposta I)

A proposta *overlap removed* MLM (OR-MLM) utiliza do algoritmo k -médias como ferramenta principal na seleção de pontos de referência para máquinas de aprendizado mínimo. Assim como na proposição original, o OR-MLM necessita de um único parâmetro K , que neste caso é o número máximo de pontos de referência. O número de pontos de referência obtidos pelo OR-MLM será sempre menor ou igual ao parâmetro K (i.e. $|R| \leq K$). Isto é possível graças a execução de uma etapa de remoção de pontos de referência que estejam em regiões heterogêneas¹. Isto é feito através da eliminação dos subconjuntos

¹ Consideram-se regiões heterogêneas, regiões que contenham dados de classes diferentes, isto é, locais com sobreposição de dados.

de dados (oriundos que k -médias) que contenham padrões de classes diferentes.

De maneira sucinta, o OR-MLM é realizado em três etapas. Na primeira etapa o algoritmo k -médias é executado sobre todo o conjunto de dados. Depois disso, a próxima etapa é realizar a remoção de protótipos advindos da execução do k -médias que agrupem padrões de diferentes classes. Por fim, o conjunto de protótipos homogêneos é selecionado como pontos de referência. A ideia principal da proposta é fazer com que os pontos de referência fiquem bem distribuídos sobre todo o conjunto de dados e, conjuntamente, fazer com que tais pontos não se encontrem em regiões de sobreposição. O Algoritmo 7 apresenta o pseudocódigo do método OR-MLM.

Algorithm 7 OR-MLM

Entrada:

- $\mathbf{X}$: conjunto de entradas
- L : conjunto de rótulos
- K : número inicial de pontos de referência

Saída:

- R : conjunto de pontos de referência
- L : conjunto de rótulos dos pontos de referência
- $\hat{\mathbf{B}}$: modelo de regressão

Procedimento:

1. Aplicar o algoritmo k -médias sobre todo o conjunto de dados

$$\{\boldsymbol{\mu}_k\}_{k=1}^K, \{c_n\}_{n=1}^N \leftarrow k\text{-MÉDIAS}(\mathbf{X}, K)$$

2. Criar k conjuntos com os rótulos dos dados

$$L_k \leftarrow \{\mathbf{l}_n \in L \mid c_n = k\}, \quad 1 \leq k \leq K$$

3. Criar conjunto de pontos de referência apenas com centroides homogêneos

$$R \leftarrow \bigcup_k \{\boldsymbol{\mu}_k\} \text{ e } T \leftarrow \bigcup_k \{L_k\}, \quad \forall k : |L_k| = 1$$

4. Calcular as matrizes de distâncias $\mathbf{D}_x$ através de $\mathbf{X}$ e R ; e $\boldsymbol{\Delta}_y$ através de L e T
 5. Calcular o modelo de regressão $\hat{\mathbf{B}}$ através da Equação (3.8)
 6. **Retorna** R , T e $\hat{\mathbf{B}}$
-

3.3 Fuzzy C-means MLM (FCM-MLM) (Proposta II)

A Proposta II, chamada de fuzzy C-means MLM (FCM-MLM), assim como o OR-MLM, utiliza um algoritmo de agrupamento para selecionar pontos de referência em máquinas de aprendizagem mínima, entretanto o FCM-MLM utiliza o algoritmo fuzzy

C -means (BEZDEK, 1981). O único hiperparâmetro do FCM-MLM ajustado pelo usuário é o número máximo de pontos de referência, K . Ou seja, o número de RPs obtidos pelo FCM-MLM sempre será menor ou igual ao hiperparâmetro K ($|\mathcal{R}| \leq K$). Isso é possível devido à execução de uma etapa extra para remover pontos que estão em regiões heterogêneas, ou seja, regiões que contêm dados de diferentes classes. Somente os padrões de regiões que possuem padrões da mesma classe são utilizadas como pontos de referência. Sobre o algoritmo fuzzy C -means, foi utilizado o valor de $m = 2$ para calcular o grau de associação de cada amostra.

Resumidamente, o FCM-MLM compreende três etapas. Na primeira etapa, o algoritmo fuzzy C -means é executado em todo o conjunto de dados. O próximo passo é executar a remoção de protótipos resultantes da execução do fuzzy C -means que agrupam padrões de diferentes classes. Finalmente, o conjunto de padrões mais próximos aos protótipos homogêneos é selecionado como RPs. A idéia principal da proposta é garantir que os pontos de referência estejam bem distribuídos no espaço de entrada e, em conjunto, garantir que esses pontos não estejam localizados em regiões com sobreposição de classe. O algoritmo 8 apresenta o pseudocódigo do método FCM-MLM.

Algorithm 8 FCM-MLM**Entrada:**

- Conjunto de entradas $\mathcal{X}$, conjunto de rótulos $\mathcal{Y}$ e número inicial de pontos de referência K .

Saída:

- Conjunto de pontos de referência $\mathcal{R}$, conjunto de rótulos dos pontos de referência $\mathcal{T}$ e o modelo de regressão $\hat{\mathbf{B}}$

Procedimento:

1. Aplicar o algoritmo fuzzy C -means em todo conjunto de dados

$$\{\mathbf{c}_k\}_{k=1}^K, \{\mu_{nk}\}_{n=1, k=1}^{N, K} \leftarrow C\text{-MEANS}(\mathcal{X}, K),$$

onde $\mu_{nk} \in [0,1]$ é o valor de associação aos clusters.

2. Criar K subconjuntos com o rótulo dos dados

$$\mathcal{Y}^k \leftarrow \{\mathbf{y}_n \in \mathcal{Y} \mid k = \arg \max_{1 \leq i \leq K} \mu_{ni}\}, \quad 1 \leq k \leq K$$

3. Criar o conjunto de pontos de referências apenas com os padrões mais próximos dos centróides homogêneos

$$\mathcal{R} \leftarrow \bigcup_k \left\{ \arg \min_{\mathbf{x}_n \in \mathcal{X}} \|\mathbf{x}_n - \mathbf{c}_k\| \right\}, \quad \forall k : |\mathcal{Y}^k| = 1.$$

O conjunto $\mathcal{T}$ é dado pela saída correspondente dos elementos em $\mathcal{R}$.

4. Calcular as matrizes de distância $\mathbf{D}_x$ utilizando $\mathcal{X}$ e $\mathcal{R}$; e Δ_y utilizando $\mathcal{Y}$ e $\mathcal{T}$
5. Calcular $\hat{\mathbf{B}}$ utilizando a Equação (3.8)
6. **Retorna** $\mathcal{R}$, $\mathcal{T}$ e $\hat{\mathbf{B}}$

3.4 Máquina de Aprendizagem Mínima de Peso Leve (Proposta III)

Nesta seção será apresentada a formulação da terceira proposta deste trabalho. Aqui será descrito qual modificação foi feita sobre a formulação da etapa de treinamento original da máquina de aprendizagem mínima, e três propostas para obtenção da matriz de regularização $\mathbf{P}$, o elemento chave do nosso trabalho. Além disso, também é apresentada a modificação na função de predição do MLM.

3.4.1 Formulação

A nova formulação tem a seguinte função objetivo durante a etapa de treinamento:

$$\min_{\mathbf{B}} \mathcal{J}_{\text{LW}}(\mathbf{B}, \mathbf{P}) = \|\mathbf{DB} - \Delta\|_{\mathcal{F}}^2 + \|\mathbf{PB}\|_{\mathcal{F}}^2 \quad (3.13)$$

na qual resulta na seguinte solução (ver Apêndice A):

$$\hat{\mathbf{B}}_{\text{LW}} = (\mathbf{D}^\top \mathbf{D} + \mathbf{P}^\top \mathbf{P})^{-1} \mathbf{D}^\top \Delta \quad (3.14)$$

onde $\mathbf{P}$ é uma matriz de regularização baseada no fator de regularização de cada amostra. A matriz $\mathbf{P}$ é a principal proposta para o nosso trabalho.

Embora pareça exagerado ter um hiperparâmetro $\mathbf{P}$ como uma matriz $N \times N$, isso se dá para preencher a lacuna devido ao sistema multi-resposta no modelo MLM. De fato, $\mathbf{P}$ pode ser provido por um vetor $\mathbf{p} \in \mathbb{R}^N$, adotando $\mathbf{P}$ como uma matriz diagonal, como mostrado a seguir:

$$\mathbf{P} = \text{diag}(\mathbf{p}). \quad (3.15)$$

3.4.2 Seleção da matriz de regularização $\mathbf{P}$

Nesta seção, descreveremos algumas propostas para produzir matrizes viáveis de regularização.

3.4.2.1 Baseado em uma distribuição normal

Nesta abordagem, busca-se obter a matriz de regularização diagonal $\mathbf{P}$, selecionando aleatoriamente valores para cada elemento da diagonal. Tal abordagem pode ser considerada uma versão padrão do nosso modelo, no qual não precisaria de nenhum método externo para se obter tais valores de regularização. Portanto, a matriz pode ser definida por

$$P_{i,j} = \begin{cases} \mathcal{N}(0, 1), & \text{se } i = j, \\ 0, & \text{caso contrário.} \end{cases} \quad (3.16)$$

3.4.2.2 Abordagem não-paramétrica por intervalos não-lineares de f

Inspirado na tese de que originou a MLM (SOUZA JÚNIOR, 2014), utilizamos a abordagem de seleção baseada em cálculos de distância e testes de hipóteses não-paramétricos. A ideia principal é corresponder localmente as medições de distância nos espaços de entrada e saída. Portanto, para todos os pares de treinamento $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$, calculamos as distâncias para os k pontos mais próximos nos conjuntos de treinamento $\mathcal{X}$ e $\mathcal{Y}$, respectivamente. Denotamos $\phi_k(\mathbf{x}_i)$ e $\psi_k(\mathbf{y}_i)$ os vetores coluna que abrangem as distâncias de seu argumento para seus k vizinhos (pontos mais próximos), também respectivamente.

Por questões de simplicidade e melhor compreensão, considere $\phi_k(\mathbf{x}_i)$ na seguinte descrição e notação, uma vez que uma descrição e notação similar para $\psi_k(\mathbf{y}_i)$ pode ser feita de maneira direta. Portanto,

$$\begin{aligned} \phi_k(\mathbf{x}) &= (d(\mathbf{x}_{i_1}, \mathbf{x}), d(\mathbf{x}_{i_2}, \mathbf{x}), \dots, d(\mathbf{x}_{i_K}, \mathbf{x})) \\ \text{sujeito a } 0 &< d(\mathbf{x}_{i_1}, \mathbf{x}) \leq \dots \leq d(\mathbf{x}_{i_K}, \mathbf{x}) \leq \dots \leq d(\mathbf{x}_{i_N}, \mathbf{x}) \quad \forall \mathbf{x}_{i_n} \in \mathcal{X}, \end{aligned} \quad (3.17)$$

onde os índices $i_1, i_2, \dots, i_N$ representam uma permutação de $\{1, \dots, N\}$.

Assumindo que a função alvo $f(\cdot) : \mathcal{X} \rightarrow \mathcal{Y}$ satisfaz a condição de continuidade de Weierstrass², então as distribuições $\phi_k(\mathbf{x}_i)$ e $\psi_k(\mathbf{y}_i)$ são iguais (correspondentes) até um fator de normalização, se considerarmos uma vizinhança de $f(\cdot)$ tal que f seja linear.

Para garantir a correspondência entre os cálculos de distância de entrada e saída, usamos o p-valor resultante do teste não-paramétrico de hipótese de Kolmogorov-Smirnov (KS) (MASSEY, 1951). A hipótese nula é a de que $\phi_k(\mathbf{x}_i)$ e $\psi_k(\mathbf{y}_i)$ são amostras provenientes da mesma distribuição. Portanto, a ideia é aplicar o teste e utilizar os p-valores do teste KS de duas amostras como critério de classificação, indicando quais pontos correspondem à parte mais linear da função alvo. Naturalmente, as medidas $\phi_k(\mathbf{x}_i)$ e $\psi_k(\mathbf{y}_i)$ são normalizados para ter média zero e variação de uma unidade (denotados por $\overline{\phi}_k(\mathbf{x}_i)$ e $\overline{\psi}_k(\mathbf{y}_i)$) antes de aplicarmos o teste KS.

Dito isso, vamos definir o procedimento para se obter os p-valores do procedimento supracitado por:

$$\mathcal{KS}_{t,\lambda}^k(\mathbf{x}, \mathbf{y}) = \pi val(\overline{\phi}_k(\mathbf{x}), \overline{\psi}_k(\mathbf{y})) + \lambda \mathbf{1}[\pi val(\overline{\phi}_k(\mathbf{x}), \overline{\psi}_k(\mathbf{y})) \leq t], \quad (3.18)$$

onde $\pi val(\cdot, \cdot)$ provê o valor p do teste KS de duas amostras para seus argumentos, $\mathbf{1}[\cdot]$ é a função indicadora que é igual a 1 se o argumento for verdadeiro ou 0 no caso contrário, e λ e t são hiperparâmetros desta função, onde t é um limiar de linearidade e λ é um valor de peso para ser adicionado ao resultado de $\pi val(\cdot, \cdot)$.

De posse de tal definição, derivamos uma matriz diagonal da seguinte forma:

$$P_{i,j} = \begin{cases} \mathcal{KS}_{t,\lambda}^k(\mathbf{x}_i, \mathbf{y}_i), & \text{se } i = j, \\ 0, & \text{caso contrário.} \end{cases} \quad (3.19)$$

Neste trabalho, consideramos duas variações para os parâmetros da nossa proposta. A primeira utiliza um valor $\lambda = 0$, ou seja, a diagonal da matriz $\mathbf{P}$ é diretamente o resultado de $\pi val(\cdot, \cdot)$, e na segunda abordagem utilizamos o método de Otsu (SEZGIN; SANKUR, 2004) para encontrar o valor de t , e $\lambda = 1$ (escolhido como melhor valor via tentativa e erro).

3.4.3 Redução do custo de predição

O *peso leve* em nossa proposta não está relacionado somente aos valores regularizados na matriz $\mathbf{B}$ (coeficientes de mapeamento linear). Também está relacionado ao procedimento de aceleração que adotamos na etapa de predição. Como utilizamos todas as amostras no algoritmo de aprendizado do LW-MLM, acreditamos que toda a estrutura

² $f(x)$ é contínua em $x = x_0$ se $\forall \varepsilon > 0 \exists \delta > 0$, tal que para cada x no domínio de f , $|x - x_0| < \delta \implies |f(x) - f(x_0)| < \varepsilon$

geométrica conhecida é aprendida, ou seja, o conhecimento do domínio é totalmente representado em $\mathbf{B}$. Com isso em mente, nós confiamos que $\mathbf{B}$ por si só, neste cenário, é capaz de mapear a estrutura linear. Tal suposição nos permite a remover alguns componentes (RPs) na etapa de predição para fornecer um procedimento mais compacto, já que a maioria dos RPs estará próxima de zero. Primeiro, vamos definir a função de descarte $\kappa : \mathbb{R}^N \rightarrow \mathbb{R}^K$ como

$$\kappa(\mathbf{a}) = (a_{i_1}, a_{i_2}, \dots, a_{i_K})^\top, \quad (3.20)$$

onde $i_1, i_2, \dots, i_K, \dots, i_N$ formam uma permutação aleatória de $\{1, \dots, N\}$. Agora, a localização de $\hat{\mathbf{y}}$ pode ser estimada através da minimização da seguinte função objetivo

$$\hat{\mathbf{y}} = f(\mathbf{x}) = \arg \min_{\mathbf{y}} \|\kappa(\Psi(\mathbf{y}) - \Phi(\mathbf{x}) \mathbf{B})\|_2. \quad (3.21)$$

Detalhes sobre tal procedimento de aceleração são discutidos nos experimentos, onde mostramos a relação entre o número de componentes utilizados e o erro na predição.

3.4.4 Similaridade com a etapa de treinamento da Rank-MLM

O modelo Rank-MLM, apesar de ser proposto como um modelo de ranqueamento, pode ser visto como uma formulação do modelo MLM original com um parâmetro de regularização ($C \in \mathbb{R}$) incluído na função objetivo. A função regularizada no Rank-MLM é dada na Equação (3.9).

Em contraste com a nossa proposta, a Rank-MLM regulariza o modelo igualmente para todas as amostras, enquanto que, a LW-MLM pode regularizar cada amostra individualmente. Este recurso pode ser útil para identificar valores discrepantes no conjunto de treinamento.

3.5 Conclusão

A máquina de aprendizagem mínima é baseada no mapeamento entre as distâncias de entrada e de saída. Uma etapa importante desse modelo é a seleção de pontos de referência. Com os pontos de referência, é possível montar as matrizes de distâncias que serão utilizadas na composição do sistema linear de múltiplas respostas que define a MLM.

Dois métodos propostos neste trabalho, a OR-MLM e a FCM-MLM, utilizam algoritmos externos para selecionar o melhor subconjunto como pontos de referência em problemas de classificação. Por outro lado, a terceira proposta, LW-MLM, ignora a seleção dos pontos de referência enquanto utiliza a matriz $\mathbf{P}$ para ponderar cada padrão com o objetivo de diminuir a complexidade do modelo. Além disso, foi realizada uma modificação na etapa de predição que visa diminuir a quantidade de cálculos sem perder a qualidade de previsão.

4 Experimentos e Discussões

Neste capítulo são apresentados estão divididos em duas seções: a primeira trata das simulações e resultados obtidos para as Propostas I (OR-MLM) e II (FCM-MLM), em problemas de classificação, comparando-os com outras versões do MLM, e na segunda seção são apresentados os resultados obtidos para a Proposta III (LW-MLM) para tarefas de regressão, comparando-o com outras versões do MLM e outros modelos disponíveis na literatura. Também são feitas discussões com o propósito de demonstrar as qualidades e limitações da Proposta III.

4.1 Simulações e discussões das Propostas I e II

A análise dos modelos OR-MLM e FCM-MLM será baseada em dois experimentos que descrevem algumas de suas características, a saber:

1. **Análise comparativa com outras abordagens do próprio MLM:** Aqui, abordamos a qualidade das propostas através da análise de acurácia e redução de modelo entre as versões do MLM, aplicando os modelos a alguns conjuntos de dados reais e artificiais.
2. **Análise qualitativa:** Neste experimento estamos interessados em analisar visualmente a diferença entre as superfícies de decisão geradas pelos modelos propostos e das variações do MLM a fim de se obter um melhor entendimento sobre o problema de classificação.

4.1.1 Configurações dos experimentos

Foi utilizada uma combinação dos métodos de validação cruzada *holdout* e *k-fold* nos experimentos realizados. O método *holdout* com uma divisão de 80%–treinamento, 20%–teste foi utilizado para estimação das métricas de comparação, que neste caso são a taxa de acerto, ou acurácia (ACC) e percentual de redução de pontos de referência em relação ao conjunto de treinamento (RED).

O ajuste do parâmetro K para os métodos RN-MLM, OR-MLM e FCM-MLM (número de pontos de referência e número máximo de pontos de referência para os dois últimos) foi realizado utilizando o método de busca em grade, combinado com validação cruzada *kfold* de 5 divisões. Os pontos de referência foram selecionados aleatoriamente numa gama de 5–100% (com taxa de variação de 5%) das amostras de treinamento disponíveis. O erro de classificação foi utilizado para a escolha do melhor valor de K . A

formulação do modelo *Optimally selected* MLM (OS-MLM) (MAIA et al., 2018) realiza o leave-one-out internamente com o intuito de definir adequadamente o número K de pontos de referência, então não possui parâmetros a serem otimizados na busca em grade, bem como a FL-MLM, cujos padrões de treinamentos são selecionados como pontos de referência.

Cada experimento foi realizado 30 vezes para cada par *conjunto de dados/método de seleção*, com base na seleção aleatória dos subconjuntos de dados de treinamento e teste. As estimativas globais das métricas de avaliação foram tomadas como uma média geral das 30 realizações para cada conjunto de dados.

Independente do conjunto de dados e do método de seleção, todos os dados são submetidos a uma etapa de normalização, fazendo com que valores medidos em escalas diferentes sejam ajustados para a mesma escala. Nos experimentos realizados utilizou-se a normalização de média zero, dessa forma, os dados estarão dispostos sob uma média 0 e desvio padrão 1.

4.1.2 Análise comparativa com outras abordagens do próprio MLM

A fim de avaliar o comportamento dos métodos de seleção, foram realizadas simulações com conjuntos de dados reais, disponíveis no repositório de aprendizado de máquinas da UCI (LICHMAN, 2013). Além disso, outros dois conjuntos de dados artificiais foram utilizados nas simulações. Informações sobre os conjuntos de dados utilizados neste trabalho são apresentados na Tabela 4.

Tabela 1 – Conjuntos de dados utilizados.

Conjunto de dados	Abreviação	# Padrões	# Treino	# Teste	# Atributos
<i>Liver Disorders</i>	LID	345	276	69	6
<i>Haberman's Survival</i>	HAB	306	245	61	3
<i>Vertebral Column</i>	VCP	310	248	62	6
<i>Hepatitis</i>	HEP	80	64	16	19
<i>Breast Cancer W.</i>	BCW	688	550	138	9
<i>Australian Credit A.</i>	AUS	690	552	138	14
<i>Pima Indians Diabetes</i>	PID	768	614	154	8
<i>Two Moon*</i>	TMN	1001	801	200	2
<i>Ripley*</i>	RIP	1250	1000	250	2

* Conjuntos de dados artificiais.

O desempenho das propostas foram comparados com três variações do MLM, no que dizem respeito a seleção de pontos de referência. A primeira variante é a *full* MLM (FL-MLM), na qual o conjunto de pontos de referência é igual ao conjunto de treinamento ($R = X$). Na segunda variante, *random* MLM (RN-MLM), são selecionados aleatoriamente K pontos de referência, tal como os autores do MLM reportam no artigo original. E

a terceira variante, *Optimally Selected* MLM (OS-MLM) utiliza a técnica de regressão esparsa com múltiplas respostas para a seleção de pontos de referência.

Tabela 2 – Resultados dos experimentos das versões do MLM. São apresentados os desempenhos da acurácia (ACC) e porcentagem de redução em comparação com os dados de treinamento (RED).

<i>Modelo</i>	<i>Métrica</i>	AUS	BCW	HAB	HEP	HIV
FL-MLM	ACC	84,57 $\pm$ 2,80	96,67 $\pm$ 1,56	66,45 $\pm$ 4,32	67,29 $\pm$ 7,46	86,08 $\pm$ 1,12
	RED	–	–	–	–	–
RN-MLM	ACC	85,12 $\pm$ 2,50	96,86 $\pm$ 1,42	73,66 $\pm$ 5,81	69,79 $\pm$ 9,44	86,53 $\pm$ 0,99
	RED	77,07 $\pm$ 18,89	73,37 $\pm$ 17,79	91,81 $\pm$ 4,78	68,65 $\pm$ 15,60	86,70 $\pm$ 15,28
OS-MLM	ACC	85,29 $\pm$ 2,17	95,48 $\pm$ 3,63	73,00 $\pm$ 5,21	68,96 $\pm$ 10,31	86,35 $\pm$ 0,93
	RED	75,12 $\pm$ 4,47	87,68 $\pm$ 13,35	90,17 $\pm$ 1,02	78,28 $\pm$ 4,90	90,44 $\pm$ 2,23
OR-MLM	ACC	85,82 $\pm$ 2,54	96,79 $\pm$ 1,42	72,90 $\pm$ 4,38	71,25 $\pm$ 7,80	86,46 $\pm$ 1,05
	RED	82,14 $\pm$ 12,69	75,10 $\pm$ 14,59	92,20 $\pm$ 5,73	77,55 $\pm$ 13,27	92,83 $\pm$ 9,89
FCM-MLM	ACC	85,63 $\pm$ 2,71	96,88 $\pm$ 1,29	71,86 $\pm$ 5,26	70,00 $\pm$ 10,30	86,42 $\pm$ 0,96
	RED	85,41 $\pm$ 8,63	96,76 $\pm$ 1,61	93,99 $\pm$ 4,88	83,18 $\pm$ 12,49	93,52 $\pm$ 4,77

Tabela 3 – Resultados dos experimentos das versões do MLM. São apresentados os desempenhos da acurácia (ACC) e porcentagem de redução em comparação com os dados de treinamento (RED).

<i>Modelo</i>	<i>Métrica</i>	LID	PID	RIP	TMN	VCP
FL-MLM	ACC	69,13 $\pm$ 5,77	74,61 $\pm$ 3,51	88,73 $\pm$ 1,27	99,85 $\pm$ 0,27	86,77 $\pm$ 4,27
	RED	–	–	–	–	–
RN-MLM	ACC	71,16 $\pm$ 5,39	76,54 $\pm$ 3,77	89,47 $\pm$ 1,87	99,73 $\pm$ 0,39	86,99 $\pm$ 3,90
	RED	63,79 $\pm$ 18,99	84,57 $\pm$ 13,43	85,59 $\pm$ 12,28	73,20 $\pm$ 16,41	58,83 $\pm$ 27,51
OS-MLM	ACC	70,43 $\pm$ 4,91	76,84 $\pm$ 3,30	90,16 $\pm$ 1,42	99,65 $\pm$ 0,45	87,10 $\pm$ 4,28
	RED	87,21 $\pm$ 4,57	83,39 $\pm$ 7,04	89,69 $\pm$ 5,02	89,58 $\pm$ 0,02	71,76 $\pm$ 13,22
OR-MLM	ACC	70,00 $\pm$ 5,69	76,47 $\pm$ 2,96	89,93 $\pm$ 1,73	99,80 $\pm$ 0,34	87,47 $\pm$ 3,87
	RED	61,40 $\pm$ 24,84	84,91 $\pm$ 12,93	86,44 $\pm$ 10,31	80,50 $\pm$ 9,10	55,07 $\pm$ 31,12
FCM-MLM	ACC	70,82 $\pm$ 5,53	76,43 $\pm$ 3,24	90,07 $\pm$ 1,68	99,77 $\pm$ 0,25	87,26 $\pm$ 4,07
	RED	65,72 $\pm$ 10,08	90,89 $\pm$ 6,92	96,64 $\pm$ 3,07	98,19 $\pm$ 0,27	88,83 $\pm$ 7,90

Através da análise das Tabelas 2 e 3 pode-se inferir que os métodos de seleção propostos se mostram suficientemente estáveis, em termos de taxa de acerto, pois ambos obtiveram valores médios de acurácia superiores ou similares aos dos outros métodos. Em termos de número de pontos de referência, o método proposto II, FCM-MLM, obteve resultados consideravelmente melhores que os outros métodos de seleção. Com isso, nossas propostas, em especial a Proposta II, se mostra uma variação da proposta original do MLM viável, no qual apresenta melhores resultados do que as outras versões e com a menor quantidade de RPs que as demais.

4.1.3 Análise qualitativa

Para uma análise qualitativa, os métodos de seleção estudados foram aplicados a um conjunto de dados artificial bastante conhecido na literatura, chamado ripley (RIP). Os parâmetros utilizados neste experimento estão de acordo com a subseção 4.1.1. O conjunto de dados RIP, bem como os resultados deste experimento são apresentados na Figura 6.

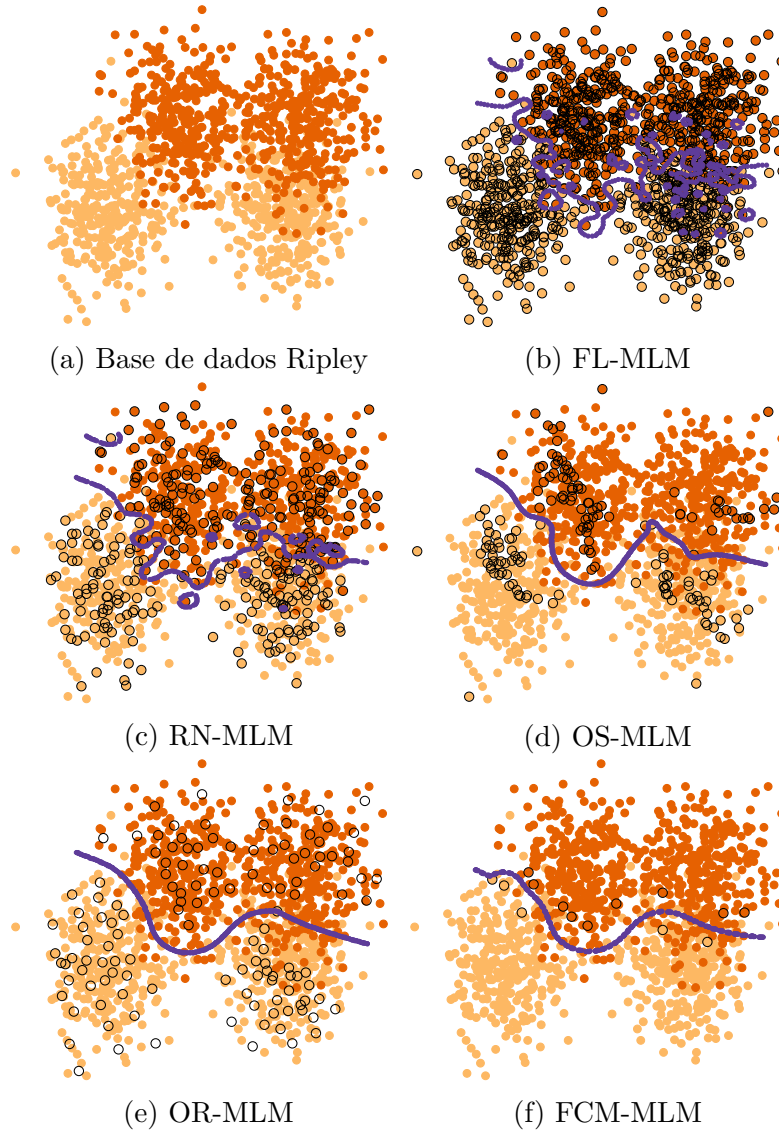


Figura 6 – Superfícies de decisão obtidos para a base Ripley.

Como pode ser observado na Figura 6c, utilizar um valor pequeno para K e realizar a escolha de forma aleatória pode acarretar em um subconjunto não representativo, desconsiderando regiões possivelmente importantes. De forma contrária, com a utilização de todos os pontos do conjunto de dados como pontos de referência, cria-se uma superfície de decisão bastante complexa, pode ser observado na Figura 6b. A escolha ideal é fazer com que os pontos de referência não se encontrem em regiões de confusão, ou seja, regiões de sobreposição de dados.

Nas Figura 6d, 6e e 6f são apresentados superfícies de decisões de melhor qualidade. Mesmo com resultados semelhantes, os três modelos selecionaram subconjuntos diferentes como pontos de referência. É importante analisar que dos métodos propostos, a OR-MLM criou um subconjunto como RP mais disperso, que representa mais o espaço dos dados, enquanto que a FCM-MLM gerou um subconjunto mais reduzido, porém com boa generalização dos resultados, se mostrando como o método mais eficaz.

4.2 Simulações e discussões da Proposta III

A análise da LW-MLM será baseada em quatro pontos que descrevem diversas características do modelo, a saber:

1. **Análise comparativa com outras abordagens do próprio MLM:** Aqui, estamos interessados em avaliar a LW-MLM contra algumas propostas de MLM por meio da raiz do erro quadrático médio (*root mean squared error*, RMSE), coeficiente de determinação (R^2 Score) e a norma da matriz $\mathbf{B}$ em um típico teste de caixa preta em alguns conjuntos de dados. Além disso, este experimento permite analisar as melhorias e problemas que foram causadas pela adoção da matriz $\mathbf{P}$ no processo de treinamento da MLM padrão.
2. **Análise qualitativa:** Neste experimento queremos ver a importância da regularização para tornar a função de predição mais suave e com melhor capacidade de generalização dos dados e, além disso, analisar empiricamente a diferença entre nossa proposta e outras variações da MLM.
3. **Análise comparativa entre a proposta e outros modelos:** Este experimento visa comparar o RMSE e o R^2 Score do método proposto em meio a outros métodos da literatura.
4. **Análise da relevância do número de RPs na etapa de predição:** Aqui estamos interessados em analisar a influência da quantidade de pontos de referência na etapa de predição sobre o valor do RMSE no modelo resultante.

4.2.1 Análise comparativa com outras abordagens da própria MLM

Esta avaliação tem duas perspectivas para fins de análise: a primeira está relacionada ao RMSE e o segundo está relacionado ao R^2 . Nos dois experimentos, usamos conjuntos de dados de *benchmarking* do mundo real a partir de conjuntos de dados dos repositórios UCI Machine Learning (LICHMAN, 2013) e do repositório de Luís Torgo (TORGO, 2005). Nome do conjunto de dados, sigla, número de amostras de treinamento ($\#Tr$), número de amostras de teste ($\#Te$) e dimensionalidade da entrada ($\#Dim$) sobre os conjuntos de

dados mencionados acima são apresentados na Tabela 4. Além disso, destacamos que todos os conjuntos de dados foram pré-processados de forma que amostras contendo valores ausentes foram removidas e todos os atributos foram normalizados pelo desvio padrão.

Tabela 4 – Descrição das bases de dados.

<i>Nome</i>	<i>Acrônimo</i>	<i># Dim</i>	<i># Tr</i>	<i># Te</i>
Abalone	ABA	8	3000	1177
AutoMPG	MPG	7	350	42
Boston housing	BTH	13	400	106
concrete	CON	8	700	330
cpu_act	CPU	12	5000	3192
delta ailerons	DAI	5	5000	2129
delta elevators	DEL	6	6000	3517
kinematics	KIN	8	4500	3692
motor_UPDRS	MUP	20	3000	2875
puma8NH	8NH	8	4500	3692
stock	STO	9	600	350
total_UPDRS	TUP	20	3000	2875
winequality_red	WRE	11	1000	599
winequality_white	WWH	11	3500	1398

Avaliamos o desempenho da LW-MLM (com três mecanismos de geração de $\mathbf{P}$) em comparação a três variações da MLM disponíveis na literatura, detalhadas na Seção 3.1.5. A primeiro, chamada de Full-MLM (FL-MLM), utiliza todo o conjunto de treinamento como pontos de referência. A segunda variante é a Random-MLM (RN-MLM), que seleciona aleatoriamente K pontos de referência a partir dos dados de treinamento. Finalmente, a terceira variante é a Ranking-MLM (Rank-MLM), que além de selecionar aleatoriamente K pontos de referência a partir do conjunto de treinamento, também acrescenta um termo de regularização à sua função de custo (ALENCAR et al., 2015).

Foi utilizada uma combinação dos métodos de validação cruzada com k -folds e o método de *holdout* nos experimentos. O método de *holdout* com a divisão de treinamento e teste utilizada por Zhao et al. (2012) foi empregado para estimar as métricas de desempenho. Para RN-MLM e Rank-MLM, escolhemos os hiperparâmetros através da busca em grade combinada com a validação cruzada de 5-folds, produzindo as melhores combinações com valores mais altos de R^2 a partir de 30 execuções independentes. É possível verificar na Tabela 5 o espaço de valores dos hiperparâmetros utilizados em cada modelo.

Em relação à análise estatística dos resultados, realizamos o teste de Friedman para avaliar todas as métricas de desempenho, a saber, o RMSE, o R^2 e a norma da matriz

Tabela 5 – Variantes da MLM e as configurações de seus hiperparâmetros.

Modelo	Descrição	Valor do hiperparâmetro
Full MLM	K : Número de RPs.	$K = N$.
Random MLM	K : Número de RPs escolhidos aleatoriamente em um intervalo através da busca em grade com validação cruzada.	$K \in \{\lfloor 0.05 \times i \times N \rfloor\}_{i=1}^{10}$.
Rank MLM	K : Número de RPs escolhidos aleatoriamente em um intervalo através da busca em grade com validação cruzada. C : Parâmetro de regularização otimizado por busca em grade com validação cruzada..	$K \in \{\lfloor 0.05 \times i \times N \rfloor\}_{i=1}^{10}$ $C \in \{2^{-4}, 2^{-3}, \dots, 2^1, 2^2\}$.
LW MLM 1	$\mathbf{P}$: Matriz diagonal com valores aleatórios de uma distribuição normal com média zero e variância unitária.	$P_{i,j} = \begin{cases} \mathcal{N}(0, 1), & \text{se } i = j \\ 0, & \text{senão} \end{cases}$.
LW MLM 2	$\mathbf{P}$: Matriz diagonal com valores atribuídos pelo KS-2-Sample-Test descrito em Algoritmo. k : Número de pontos mais próximos para calcular a distância.	$P_{i,j} = \begin{cases} \mathcal{KS}_0^k(\mathbf{x}_i, \mathbf{y}_i), & \text{se } i = j \\ 0, & \text{senão} \end{cases}$. $k = \log_{10}(5 \times N)$.
LW MLM 3	$\mathbf{P}$: Matriz diagonal com valores atribuídos pelo KS-2-Sample-Test descrito em Algoritmo. k : Número de pontos mais próximos para calcular a distância. t : <i>Threshold</i> de linearidade.	$P_{i,j} = \begin{cases} \mathcal{KS}_t^k(\mathbf{x}_i, \mathbf{y}_i), & \text{se } i = j \\ 0, & \text{senão} \end{cases}$. $k = \log_{10}(5 \times N)$. $t = \mathcal{OTSU}_k(\mathcal{X}, \mathcal{Y})$.

$\mathbf{B}$, para melhor distinguir a comparação entre as variações do MLM. O teste de Friedman quantifica a consistência dos resultados obtidos nos modelos quando aplicados em vários conjuntos de dados de acordo com seus rankings de desempenho. Em tal análise, a hipótese nula afirma que não há diferença estatística entre todos os modelos avaliados. Para fazer isso, representamos graficamente essa comparação através do gráfico de Diferença Crítica (CD), como o apresentado em (DEMŠAR, 2006). Nessa plotagem, a linha superior é o eixo no qual as classificações médias de cada modelo são plotadas, enquanto o grupo de

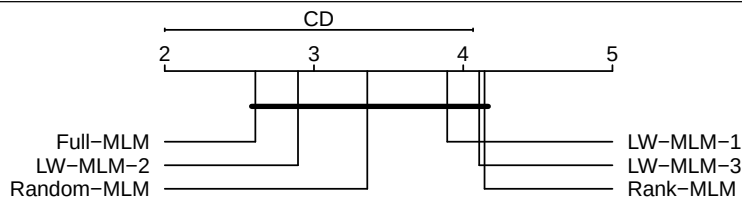
modelos conectados indica que elas não são significativamente diferentes. Além disso, o CD é apresentado acima do plote e adotando o nível de significância, neste trabalho utilizamos $\alpha = 0,05$.

4.2.1.1 Erro de predição como métrica de desempenho

Tabela 6 – Resultados de RMSE e desvio padrão sobre 30 realizações independentes.

Base	Full-MLM	Random-MLM	Rank-MLM	LW-MLM-1	LW-MLM-2	LW-MLM-3
ABA	2,24 $\pm$ 0,07	2,14 $\pm$ 0,06	2,12 $\pm$ 0,06	2,13 $\pm$ 0,06	2,22 $\pm$ 0,07	2,22 $\pm$ 0,07
MPG	2,64 $\pm$ 0,53	2,69 $\pm$ 0,50	2,62 $\pm$ 0,49	2,61 $\pm$ 0,47	2,57 $\pm$ 0,48	2,58 $\pm$ 0,48
BTH	3,11 $\pm$ 0,48	3,47 $\pm$ 0,60	3,57 $\pm$ 0,57	3,54 $\pm$ 0,52	3,31 $\pm$ 0,50	3,80 $\pm$ 0,49
CON	5,76 $\pm$ 0,53	6,40 $\pm$ 0,46	7,25 $\pm$ 0,38	7,21 $\pm$ 0,38	7,37 $\pm$ 0,41	7,94 $\pm$ 0,35
CPU	2,81 $\pm$ 0,06	2,86 $\pm$ 0,07	2,94 $\pm$ 0,07	2,91 $\pm$ 0,06	2,81 $\pm$ 0,06	2,81 $\pm$ 0,06
DAI	0,00 $\pm$ 0,0*	0,00 $\pm$ 0,0*	0,00 $\pm$ 0,0*	0,00 $\pm$ 0,0*	0,00 $\pm$ 0,0*	0,00 $\pm$ 0,0*
DEL	0,00 $\pm$ 0,0*	0,00 $\pm$ 0,0*	0,00 $\pm$ 0,0*	0,00 $\pm$ 0,0*	0,00 $\pm$ 0,0*	0,00 $\pm$ 0,0*
KIN	0,08 $\pm$ 0,00	0,09 $\pm$ 0,00	0,09 $\pm$ 0,00	0,09 $\pm$ 0,00	0,10 $\pm$ 0,00	0,11 $\pm$ 0,00
MUP	2,11 $\pm$ 0,06	2,34 $\pm$ 0,06	2,68 $\pm$ 0,05	2,65 $\pm$ 0,05	2,49 $\pm$ 0,06	2,52 $\pm$ 0,06
8NH	3,41 $\pm$ 0,04	3,36 $\pm$ 0,04	3,33 $\pm$ 0,04	3,33 $\pm$ 0,04	3,32 $\pm$ 0,03	3,37 $\pm$ 0,04
STO	0,66 $\pm$ 0,04	0,74 $\pm$ 0,04	0,83 $\pm$ 0,05	0,83 $\pm$ 0,05	0,79 $\pm$ 0,04	0,85 $\pm$ 0,05
TUP	2,85 $\pm$ 0,08	3,17 $\pm$ 0,07	3,68 $\pm$ 0,06	3,66 $\pm$ 0,07	3,48 $\pm$ 0,07	3,97 $\pm$ 0,07
WRE	0,61 $\pm$ 0,02	0,62 $\pm$ 0,02	0,62 $\pm$ 0,01	0,62 $\pm$ 0,02	0,61 $\pm$ 0,02	0,61 $\pm$ 0,02
WWH	0,61 $\pm$ 0,02	0,65 $\pm$ 0,01	0,67 $\pm$ 0,01	0,67 $\pm$ 0,01	0,61 $\pm$ 0,02	0,61 $\pm$ 0,02

* Os valores não são exatamente zero, mas bem próximos.

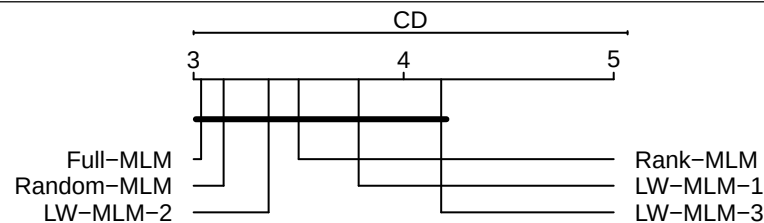


Como pode-se verificar na Tabela 6, relatamos as métricas do RMSE referentes aos experimentos para todos os conjuntos de dados e modelos. Nesse experimento, depois de realizar o teste de Friedman, notamos que todos os modelos são equivalentes, pois todos estão conectados. Veja a plotagem de CD localizada na parte inferior da Tabela 6. Esta análise é interessante pois constata que a nova formulação não sacrificou o desempenho da generalização.

4.2.1.2 Coeficiente de determinação como métrica de desempenho

Tabela 7 – Resultados de R^2 Score e desvio padrão sobre 30 realizações independentes.

Base	Full-MLM	Random-MLM	Rank-MLM	LW-MLM-1	LW-MLM-2	LW-MLM-3
ABA	$0,53 \pm 0,02$	$0,57 \pm 0,02$	$0,58 \pm 0,02$	$0,57 \pm 0,02$	$0,54 \pm 0,02$	$0,54 \pm 0,02$
MPG	$0,89 \pm 0,05$	$0,88 \pm 0,04$	$0,89 \pm 0,04$	$0,89 \pm 0,04$	$0,89 \pm 0,04$	$0,89 \pm 0,04$
BTH	$0,88 \pm 0,04$	$0,85 \pm 0,05$	$0,85 \pm 0,05$	$0,85 \pm 0,05$	$0,87 \pm 0,04$	$0,83 \pm 0,05$
CON	$0,88 \pm 0,02$	$0,85 \pm 0,02$	$0,81 \pm 0,02$	$0,81 \pm 0,02$	$0,81 \pm 0,02$	$0,78 \pm 0,02$
CPU	$0,98 \pm 0,00$	$0,98 \pm 0,00$	$0,97 \pm 0,00$	$0,98 \pm 0,00$	$0,98 \pm 0,00$	$0,98 \pm 0,00$
DAI	$0,68 \pm 0,01$	$0,71 \pm 0,01$	$0,71 \pm 0,01$	$0,71 \pm 0,01$	$0,69 \pm 0,01$	$0,69 \pm 0,01$
DEL	$0,59 \pm 0,01$	$0,64 \pm 0,01$	$0,64 \pm 0,01$	$0,64 \pm 0,01$	$0,59 \pm 0,01$	$0,59 \pm 0,01$
KIN	$0,90 \pm 0,00$	$0,89 \pm 0,00$	$0,88 \pm 0,00$	$0,88 \pm 0,00$	$0,87 \pm 0,00$	$0,83 \pm 0,01$
MUP	$0,93 \pm 0,00$	$0,92 \pm 0,00$	$0,89 \pm 0,00$	$0,90 \pm 0,00$	$0,91 \pm 0,00$	$0,91 \pm 0,01$
8NH	$0,63 \pm 0,01$	$0,64 \pm 0,01$	$0,65 \pm 0,01$	$0,65 \pm 0,01$	$0,65 \pm 0,01$	$0,64 \pm 0,01$
STO	$0,99 \pm 0,00$	$0,99 \pm 0,00$	$0,98 \pm 0,00$	$0,98 \pm 0,00$	$0,99 \pm 0,00$	$0,98 \pm 0,00$
TUP	$0,93 \pm 0,00$	$0,91 \pm 0,00$	$0,88 \pm 0,01$	$0,89 \pm 0,01$	$0,90 \pm 0,00$	$0,87 \pm 0,01$
WRE	$0,44 \pm 0,03$	$0,40 \pm 0,03$	$0,40 \pm 0,03$	$0,41 \pm 0,03$	$0,44 \pm 0,03$	$0,44 \pm 0,03$
WWH	$0,53 \pm 0,02$	$0,47 \pm 0,02$	$0,43 \pm 0,02$	$0,43 \pm 0,02$	$0,53 \pm 0,02$	$0,53 \pm 0,02$



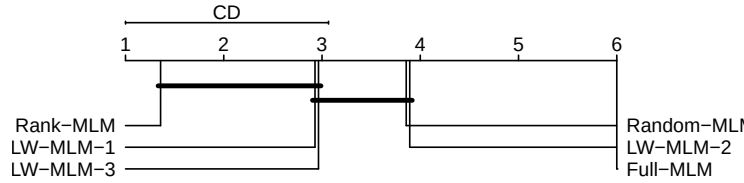
Uma outra métrica interessante para comparar a qualidade de modelos regressores é o coeficiente de determinação R^2 , pois ele seu resultado não escala de acordo com a ordem de magnitude da saída. A Tabela 7 mostra esses resultados para todos os modelos. Como se pode ver, os resultados são muito semelhantes e, novamente, todos os modelos podem ser considerados equivalentes, pois todos estão conectados.

4.2.1.3 A norma como métrica de desempenho

Tabela 8 – Valor da norma da matriz $\mathbf{B}$ e desvio padrão sobre 30 realizações independentes. O valor foi normalizado entre 0 e 1 para cada base.

Base	Full-MLM	Random-MLM	Rank-MLM	LW-MLM-1	LW-MLM-2	LW-MLM-3
ABA	1,00 ± 0,02	0,03 ± 0,01	0,01 ± 0,00	0,07 ± 0,00	0,84 ± 0,02	0,84 ± 0,02
MPG	1,00 ± 0,06	0,28 ± 0,12	0,07 ± 0,01	0,14 ± 0,02	0,24 ± 0,02	0,24 ± 0,03
BTH	1,00 ± 0,04	0,49 ± 0,06	0,20 ± 0,02	0,32 ± 0,04	0,36 ± 0,02	0,14 ± 0,08
CON	1,00 ± 0,27	0,33 ± 0,32	0,00 ± 0,0*	0,00 ± 0,0*	0,00 ± 0,0*	0,00 ± 0,0*
CPU	1,00 ± 0,04	0,43 ± 0,02	0,03 ± 0,00	0,11 ± 0,01	0,82 ± 0,01	0,82 ± 0,01
DAI	1,00 ± 0,01	0,04 ± 0,02	0,02 ± 0,01	0,10 ± 0,01	0,92 ± 0,01	0,92 ± 0,01
DEL	1,00 ± 0,02	0,02 ± 0,00	0,01 ± 0,00	0,09 ± 0,00	0,96 ± 0,02	0,96 ± 0,02
KIN	1,00 ± 0,01	0,53 ± 0,01	0,31 ± 0,00	0,45 ± 0,01	0,31 ± 0,01	0,12 ± 0,00
MUP	1,00 ± 0,02	0,59 ± 0,02	0,23 ± 0,00	0,44 ± 0,01	0,51 ± 0,02	0,50 ± 0,02
8NH	1,00 ± 0,01	0,23 ± 0,06	0,15 ± 0,02	0,32 ± 0,01	0,21 ± 0,01	0,07 ± 0,00
STO	1,00 ± 0,02	0,51 ± 0,02	0,13 ± 0,00	0,26 ± 0,01	0,29 ± 0,02	0,15 ± 0,01
TUP	1,00 ± 0,02	0,60 ± 0,02	0,23 ± 0,00	0,45 ± 0,02	0,45 ± 0,01	0,20 ± 0,00
WRE	1,00 ± 0,03	0,43 ± 0,10	0,09 ± 0,02	0,20 ± 0,02	0,96 ± 0,03	0,96 ± 0,03
WWH	1,00 ± 0,02	0,47 ± 0,02	0,07 ± 0,00	0,19 ± 0,01	0,99 ± 0,02	0,99 ± 0,02

* Os valores não são exatamente zero, mas bem próximos.



Por fim, a Tabela 8 mostra a norma da matriz $\mathbf{B}$. Os valores foram redimensionados entre 1 e 0, onde 1 representa o maior valor da norma da matriz $\mathbf{B}$ da Full-MLM no conjunto de dados¹. A partir disso, é possível concluir que nossa proposta pode reduzir e, quando combinada às demais métricas, obtém resultados viáveis sem, mais uma vez, sacrificar a generalização do modelo.

4.2.2 Análise qualitativa

Este experimento é realizado para nos auxiliar no entendimento visual de como nossa proposta se comporta em alguns conjuntos de dados (quatro artificiais e um real) de problemas de regressão $f : \mathbb{R} \rightarrow \mathbb{R}$ em comparação aos outros modelos de MLM. O conjunto de dados do mundo real, chamado por *mcycle*, contém 133 observações de um acidente de motocicleta simulado, usado para testar capacetes de impacto (SILVERMAN, 1985). As informações sobre os conjuntos de dados artificiais são mostradas na Tabela 9, na qual temos a função que foi utilizada para gerar os conjuntos de dados, uma descrição e seu domínio.

¹ Adotamos o fator de normalização usando $\mathbf{B}$ da Full-MLM pois em outros modelos, $\mathbf{D}$ e/ou Δ não são matrizes quadradas nem apresentam fator de regularização, portanto, $\|\mathbf{B}\|_{\mathcal{F}} = \|\mathbf{D}^{-1}\Delta\|_{\mathcal{F}} \geq \|\mathbf{D}^{-1}\|_{\mathcal{F}}\|\Delta\|_{\mathcal{F}}$

Tabela 9 – Descrição dos conjuntos de dados artificiais.

Descrição	Ruído uniforme $\mathcal{U}(a, b)$	Função
Artificial I	$\mathcal{U}(0,500, 0,500)$	$y_i = x_i(1,5x_i + 3 + rand_i)$
Artificial II	$\mathcal{U}(-0,16, 0,080)$	$y_i = \tanh(x_i) + rand_i$
Artificial III	$\mathcal{U}(0,250, 0,500)$	$y_i = i \times (\sin(1,5x_i) + rand_i)$
Artificial IV	$\mathcal{U}(0,025, 0,050)$	$y_i = \sum_{j=1,3,5,\dots}^{\infty} \frac{\sin(jx_i)}{j}$

A busca em grade com validação cruzada foi utilizada para selecionar a melhor combinação de hiperparâmetros dos modelos. Como a Random-MLM e a Rank-MLM utilizam um subconjunto das amostras de treinamento como pontos de referência, optamos por utilizar uma borda roxa nestes pontos para diferenciar os pontos de referência do restante do conjunto de dados, a fim de melhorar o entendimento e visualização. Enquanto isso, mantivemos os pontos de referência definidos com a mesma cor para o treinamento definido nos outros modelos.

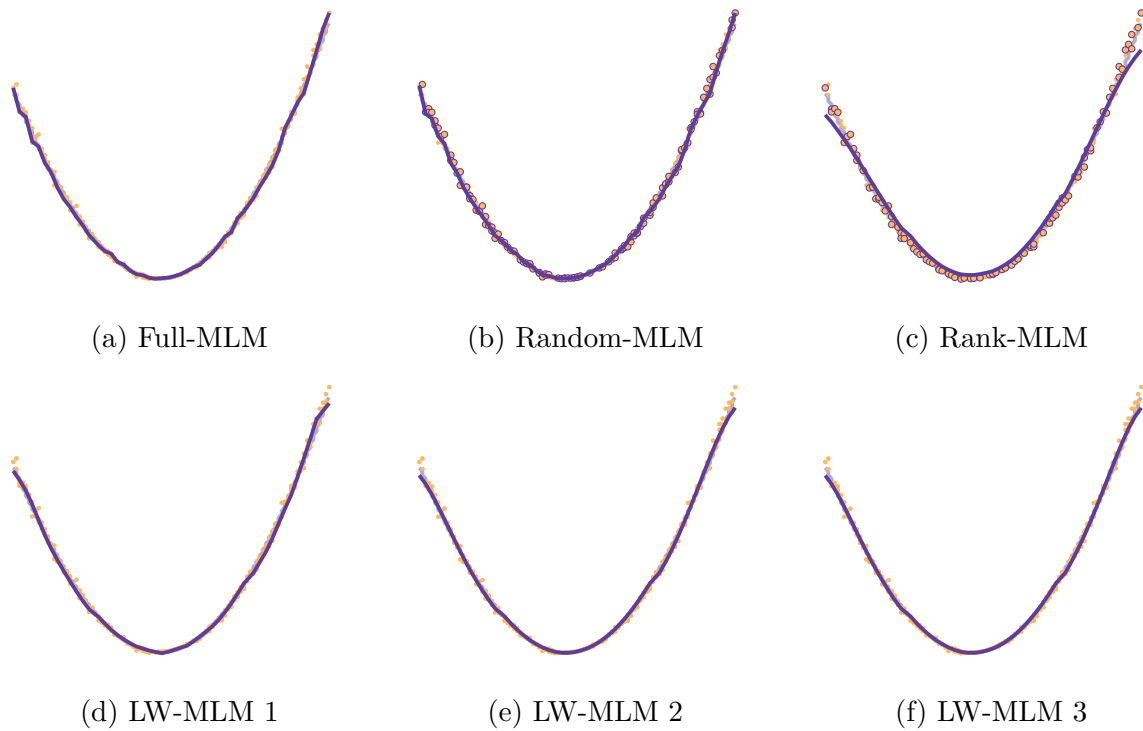


Figura 7 – Funções resultantes para o conjunto Artificial I

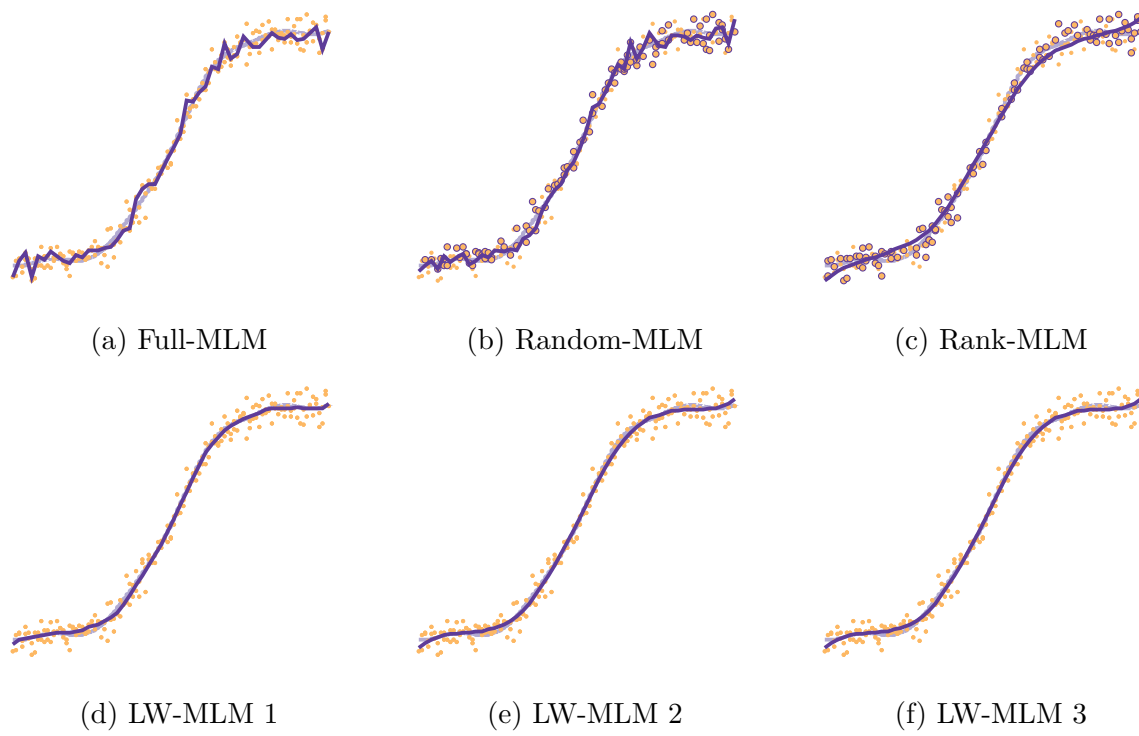


Figura 8 – Funções resultantes para o conjunto Artificial II

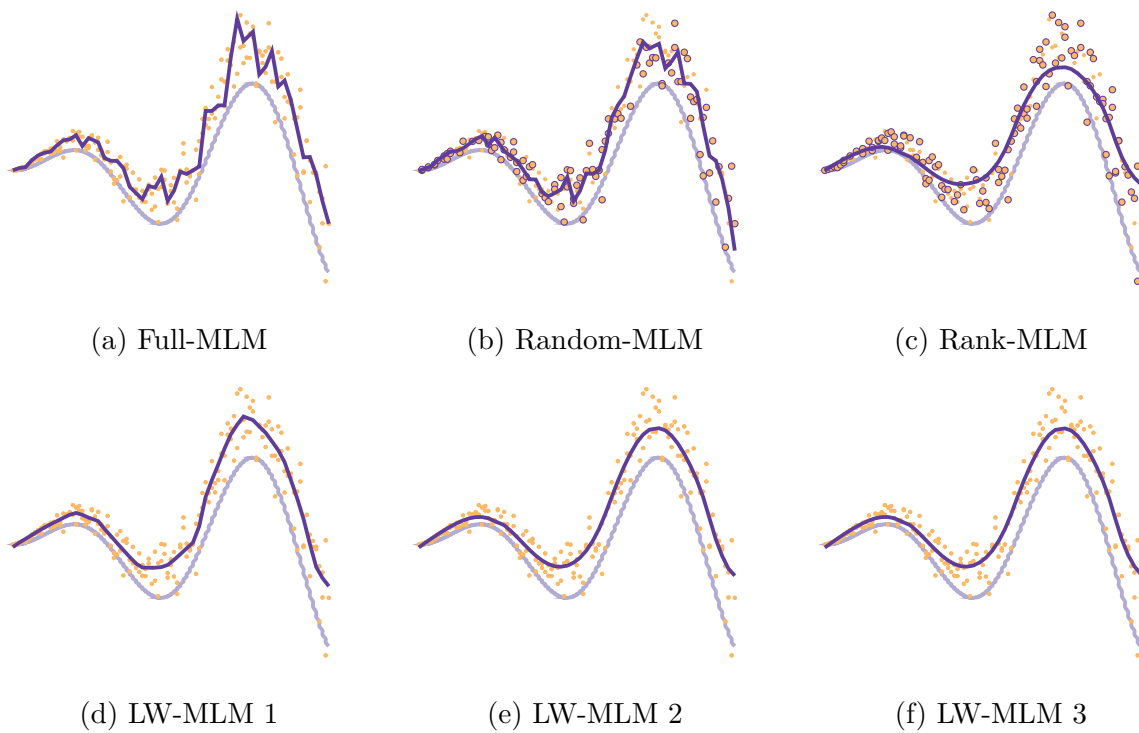


Figura 9 – Funções resultantes para o conjunto Artificial III

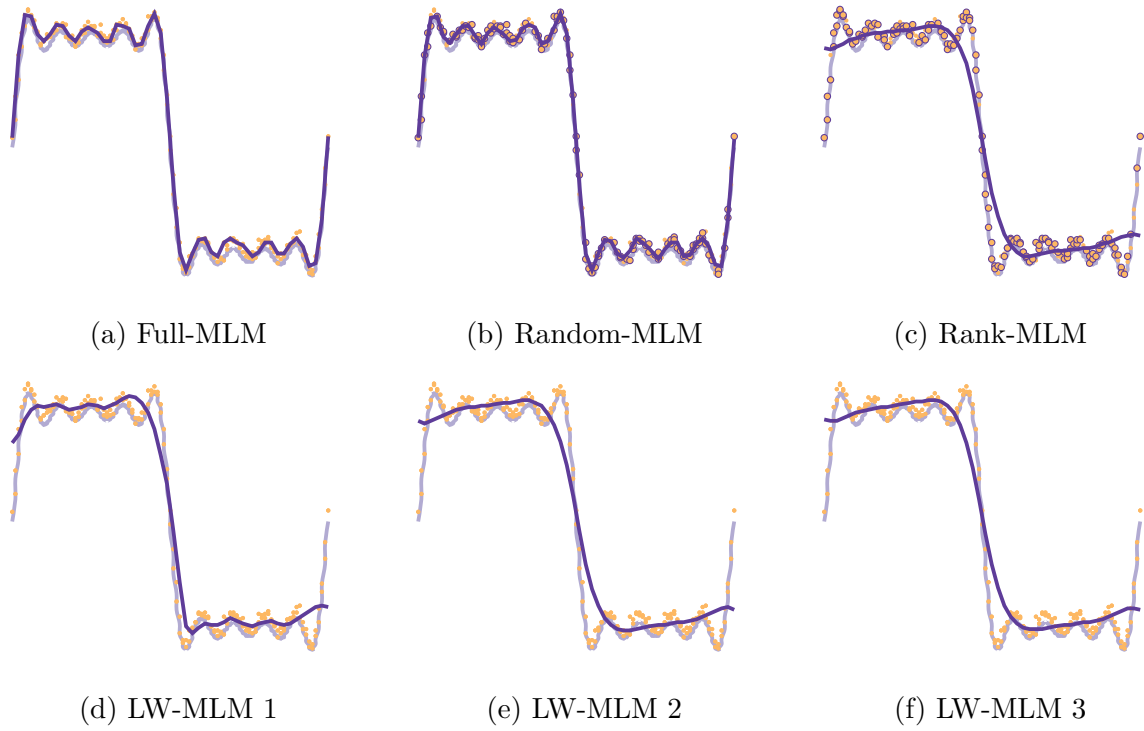


Figura 10 – Funções resultantes para o conjunto Artificial IV

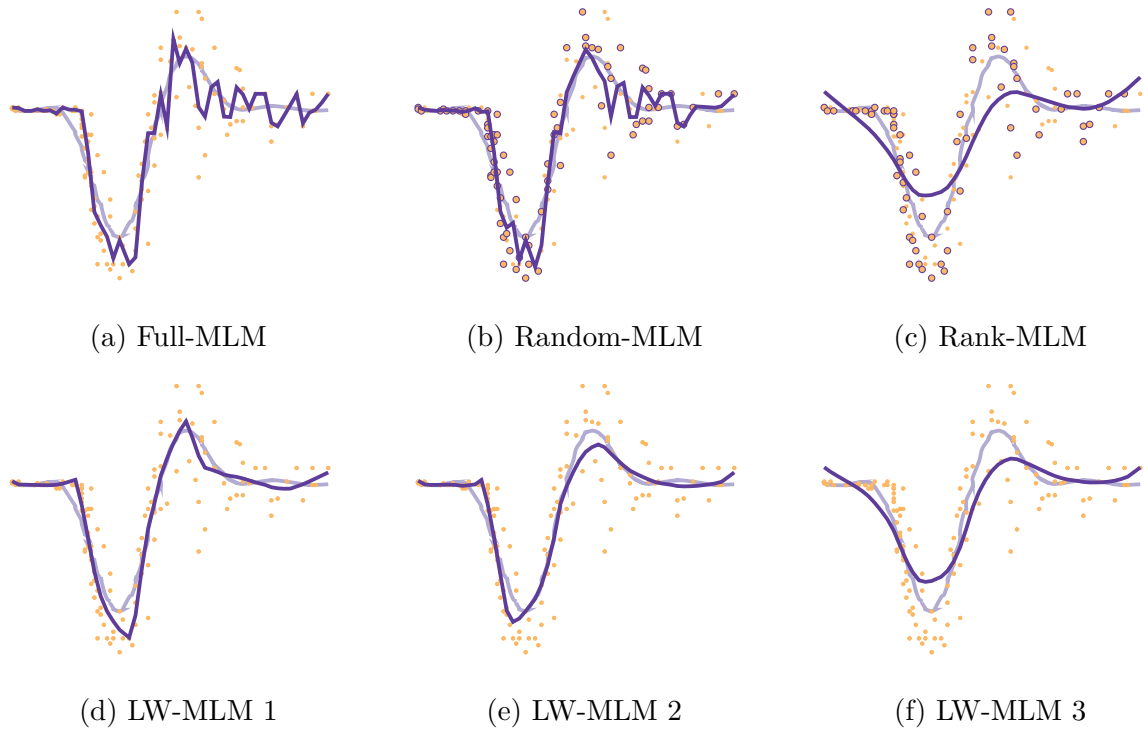


Figura 11 – Funções resultantes para o conjunto Mcycle

Nas Figuras 7 e 8, podemos ver que apesar das funções dos conjuntos de dados Artificiais I e II serem não-lineares, elas não apresentam muita complexidade para os modelos utilizados, obtendo resultados similares entre todas as seis variações.

Enquanto isso, na Figura 9, pode-se notar que todos os modelos produziram uma função estimada viável. No entanto, aqueles que de alguma forma utilizaram um fator de regularização (Rank-MLM e LW-MLM) foram os que apresentaram resultados melhores resultados e menos sobreajustados. Esta análise é crítica pois, se usarmos apenas as métricas obtidas pelo RMSE ou R^2 para avaliar esses modelos, não identificaríamos facilmente quais modelos têm sobreajuste. Um fato interessante em relação à Random-MLM é que a escolha de pontos de referência causa um pouco de suavidade, atuando como um fator de regularização no modelo. No entanto, encontrar esse subconjunto é considerado um problema NP-Hard e, portanto, é impraticável calcular em algumas configurações.

Quanto ao conjunto de dados Artificial IV, apresentado em Figura 10, pode-se notar que a regularização (provenientes das variantes Rank-MLM e LW-MLM) prejudicou o desempenho do modelo, especialmente próximo aos padrões da região ondulada. No entanto, o comportamento geral das funções aprendidas foi mantido.

Finalmente, em relação ao último conjunto de dados, o conjunto de dados *mcycle*, representado na Figura 11, pode-se notar um caso em que a regularização e o sobreajuste adicionam dificuldades em relação ao processo de aprendizagem. A Full-MLM e a Random-MLM apresentam aspectos de sobreajuste, adotando todas as entradas do conjunto como pontos de referência (Full-MLM) ou somente alguns destes (Random-MLM), indicando a necessidade de um fator de regularização. No entanto, na direção oposta, a Rank-MLM apresentou uma função mais complexa, com erros mais altos, pois todas as entradas foram regularizadas igualmente. Por fim, as variações da LW-MLM que empregam diferentes estruturas de $\mathbf{P}$ e, além disso, regularização por padrão, aprenderam funções mais suaves com o ajuste adequado. A partir de tal constatação, apoiamos verdadeiramente que essa abordagem de regularização por padrão é benéfica.

Tabela 10 – Valor da norma da matriz $\mathbf{B}$ resultantes do experimento qualitativo. O valor foi normalizado entre 0 e 1 para cada base.

Modelos	Artificial I	Artificial II	Artificial III	Artificial IV	Mcycle
Full-MLM	1,000	1,000	1,000	1,000	1,000
Random-MLM	0,413	0,576	0,614	0,500	0,634
Rank-MLM	0,008	0,002	0,003	0,034	0,003
LW-MLM-1	0,066	0,018	0,018	0,170	0,027
LW-MLM-2	0,013	0,003	0,003	0,021	0,034
LW-MLM-3	0,012	0,003	0,003	0,016	0,003

Também é possível verificar como se comporta a norma da matriz $\mathbf{B}$ em cada modelo. É possível inferir na Tabela 10 a importância do fator de regularização (tanto o C do Rank-MLM como a matriz $\mathbf{P}$ da nossa proposta) e sua influência sobre o modelo. Quando a norma é elevada, significa que há muita informação na estimativa do mapeamento $\mathbf{B}$, o que pode prejudicar na predição. Através da regularização é possível ajustar para que

o treinamento seja suavizado, garantindo uma melhor qualidade no modelo. Os valores da nossa proposta estão bem próximos ao do Rank-MLM, e enquanto isso, a Random-MLM possui valores maiores, porém, ainda abaixo da Full-MLM.

4.2.3 Análise comparativa entre a proposta e outros modelos

Nesta seção é realizada uma análise similar à feita na seção anterior, entretanto, utilizando somente a proposta deste trabalho contra os outros modelos de aprendizado de máquina que foram apresentados na Seção 2.3. Porém, esta comparação se limita à análise do RMSE e do coeficiente de determinação.

Assim como no experimento anterior, os modelos foram aplicados aos problemas da Tabela 4. Além disso, os resultados foram gerados a partir de 30 realizações independentes, de modo que em cada realização, os conjuntos de dados foram divididos em duas partes, uma de treino e outra de teste, assim como descrito em Zhao et al. (2012). Em cada rodada, todos os modelos são avaliados sobre a mesma partição da base de dados.

Para obtenção dos valores dos hiperparâmetros dos modelos, um processo de validação cruzada com busca em grade foi realizada em cada rodada. Nesse processo, os intervalos de valores dos parâmetros são os mesmos para todas as bases de dados. Os hiperparâmetros da proposta já foram apresentados na Tabela 5. O OLAM não precisa ajustar nenhum hiperparâmetro, e para os outros modelos, seus hiperparâmetros estão descritos na Tabela 11.

Tabela 11 – Hiperparâmetros das redes RBF, ELM e do modelo LSSVM para serem ajustados na validação cruzada.

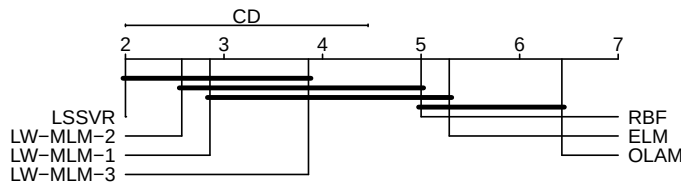
Modelo	Descrição	Valor do hiperparâmetro
RBF	M : Número de neurônios na camada oculta.	$M = 10 - 80\%$ de $ \mathbf{D} $.
	σ : Abertura da gaussiana.	$\sigma = \{r : r = 0.01 + 0.2475 \times n, n \in \{0, \dots, 4\}\}$.
ELM	M : Número de neurônios na camada oculta.	$M \in \{10, 20, \dots, 90, 100\}$.
LSSVM	K : Matriz kernel	$K(\mathbf{x}, \mathbf{x}_i) = \exp(-\ \mathbf{x} - \mathbf{x}_i\ ^2 / \sigma^2)$
	σ : Abertura da Gaussiana.	$\sigma \in \{2^{-5}, 2^{-3}, \dots, 2^3, 2^5\}$.
	γ : Variável de ajuste.	$\gamma \in \{2^{-5}, 2^{-3}, \dots, 2^3, 2^5\}$.

Na Tabela 12 é apresentada a comparação do RMSE para todos os conjuntos de dados e modelos. É possível verificar que a nossa proposta apresenta ótimos resultados, tão bons quanto os modelos de estado da arte disponíveis na literatura. E em muitos casos apresenta os menores valores de erro. Além da análise do RMSE, temos na Tabela 13 os resultados obtidos na métrica R^2 Score. Analisando ambas tabelas, podemos inferir

Tabela 12 – Resultados de RMSE e desvio padrão sobre 30 realizações independentes.

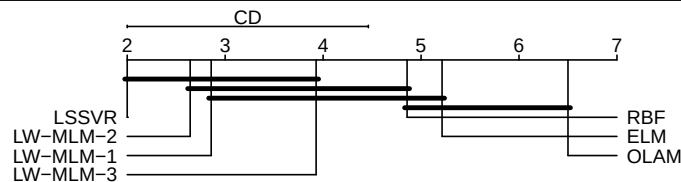
Base	LW-MLM-1	LW-MLM-2	LW-MLM-3	OLAM	RBF	ELM	LSSVR
ABA	2,13 $\pm$ 0,06	2,22 $\pm$ 0,07	2,22 $\pm$ 0,07	2,24 $\pm$ 0,07	7,62 $\pm$ 14,14	2,23 $\pm$ 0,20	2,12 $\pm$ 0,07
MPG	2,61 $\pm$ 0,47	2,57 $\pm$ 0,48	2,58 $\pm$ 0,48	3,37 $\pm$ 0,38	2,69 $\pm$ 0,45	2,86 $\pm$ 0,42	2,63 $\pm$ 0,52
BTH	3,54 $\pm$ 0,52	3,31 $\pm$ 0,50	3,80 $\pm$ 0,49	4,89 $\pm$ 0,49	3,41 $\pm$ 0,51	4,08 $\pm$ 0,69	3,39 $\pm$ 0,53
CON	7,21 $\pm$ 0,38	7,37 $\pm$ 0,41	7,94 $\pm$ 0,35	10,50 $\pm$ 0,38	6,95 $\pm$ 0,67	7,93 $\pm$ 2,10	6,39 $\pm$ 0,43
CPU	2,91 $\pm$ 0,06	2,81 $\pm$ 0,06	2,81 $\pm$ 0,06	9,95 $\pm$ 0,27	3,93 $\pm$ 0,74	3,43 $\pm$ 0,21	3,07 $\pm$ 0,07
DAI	0,00 $\pm$ 0,0*	0,00 $\pm$ 0,0*	0,00 $\pm$ 0,0*	0,00 $\pm$ 0,0*	0,00 $\pm$ 0,0*	0,00 $\pm$ 0,0*	0,00 $\pm$ 0,0*
DEL	0,00 $\pm$ 0,0*	0,00 $\pm$ 0,0*	0,00 $\pm$ 0,0*	0,00 $\pm$ 0,0*	0,00 $\pm$ 0,0*	0,00 $\pm$ 0,0*	0,00 $\pm$ 0,0*
KIN	0,09 $\pm$ 0,00	0,10 $\pm$ 0,00	0,11 $\pm$ 0,00	0,20 $\pm$ 0,00	0,08 $\pm$ 0,00	0,16 $\pm$ 0,01	0,08 $\pm$ 0,00
MUP	2,65 $\pm$ 0,05	2,49 $\pm$ 0,06	2,52 $\pm$ 0,06	7,23 $\pm$ 0,06	3,37 $\pm$ 0,18	5,72 $\pm$ 0,53	2,23 $\pm$ 0,05
8NH	3,33 $\pm$ 0,04	3,32 $\pm$ 0,03	3,37 $\pm$ 0,04	4,46 $\pm$ 0,05	3,34 $\pm$ 0,03	3,99 $\pm$ 0,14	3,28 $\pm$ 0,03
STO	0,83 $\pm$ 0,05	0,79 $\pm$ 0,04	0,85 $\pm$ 0,05	2,36 $\pm$ 0,10	0,87 $\pm$ 0,07	0,97 $\pm$ 0,06	0,74 $\pm$ 0,05
TUP	3,66 $\pm$ 0,07	3,48 $\pm$ 0,07	3,97 $\pm$ 0,07	9,28 $\pm$ 0,07	4,64 $\pm$ 0,29	7,75 $\pm$ 0,34	3,03 $\pm$ 0,07
WRE	0,62 $\pm$ 0,02	0,61 $\pm$ 0,02	0,61 $\pm$ 0,02	0,64 $\pm$ 0,01	0,65 $\pm$ 0,02	0,65 $\pm$ 0,02	0,63 $\pm$ 0,01
WWH	0,67 $\pm$ 0,01	0,61 $\pm$ 0,02	0,61 $\pm$ 0,02	0,76 $\pm$ 0,02	0,78 $\pm$ 0,09	0,73 $\pm$ 0,02	0,69 $\pm$ 0,01

* Os valores não são exatamente zero, mas bem próximos.

Tabela 13 – Resultados de R^2 Score e desvio padrão sobre 30 realizações independentes.

Base	LW-MLM-1	LW-MLM-2	LW-MLM-3	OLAM	RBF	ELM	LSSVR
ABA	0,57 $\pm$ 0,02	0,54 $\pm$ 0,02	0,54 $\pm$ 0,02	0,52 $\pm$ 0,02	0,33 $\pm$ 0,02	0,54 $\pm$ 0,02	0,57 $\pm$ 0,02
MPG	0,89 $\pm$ 0,05	0,89 $\pm$ 0,04	0,89 $\pm$ 0,04	0,82 $\pm$ 0,04	0,88 $\pm$ 0,04	0,87 $\pm$ 0,04	0,88 $\pm$ 0,05
BTH	0,85 $\pm$ 0,04	0,87 $\pm$ 0,05	0,83 $\pm$ 0,05	0,71 $\pm$ 0,05	0,86 $\pm$ 0,04	0,80 $\pm$ 0,05	0,86 $\pm$ 0,04
CON	0,81 $\pm$ 0,02	0,81 $\pm$ 0,02	0,78 $\pm$ 0,02	0,60 $\pm$ 0,02	0,83 $\pm$ 0,02	0,78 $\pm$ 0,02	0,85 $\pm$ 0,02
CPU	0,98 $\pm$ 0,0*	0,98 $\pm$ 0,0*	0,98 $\pm$ 0,0*	0,71 $\pm$ 0,0*	0,95 $\pm$ 0,0*	0,97 $\pm$ 0,0*	0,97 $\pm$ 0,0*
DAI	0,71 $\pm$ 0,01	0,69 $\pm$ 0,01	0,69 $\pm$ 0,01	0,68 $\pm$ 0,01	0,66 $\pm$ 0,01	0,70 $\pm$ 0,01	0,70 $\pm$ 0,01
DEL	0,64 $\pm$ 0,01	0,59 $\pm$ 0,01	0,59 $\pm$ 0,01	0,63 $\pm$ 0,01	0,62 $\pm$ 0,01	0,63 $\pm$ 0,01	0,64 $\pm$ 0,01
KIN	0,88 $\pm$ 0,0*	0,87 $\pm$ 0,0*	0,83 $\pm$ 0,0*	0,41 $\pm$ 0,0*	0,90 $\pm$ 0,0*	0,64 $\pm$ 0,01	0,92 $\pm$ 0,0*
MUP	0,90 $\pm$ 0,0*	0,91 $\pm$ 0,0*	0,91 $\pm$ 0,0*	0,21 $\pm$ 0,0*	0,83 $\pm$ 0,0*	0,51 $\pm$ 0,01	0,93 $\pm$ 0,0*
8NH	0,65 $\pm$ 0,01	0,65 $\pm$ 0,01	0,64 $\pm$ 0,01	0,37 $\pm$ 0,01	0,65 $\pm$ 0,01	0,50 $\pm$ 0,01	0,66 $\pm$ 0,01
STO	0,98 $\pm$ 0,0*	0,99 $\pm$ 0,0*	0,98 $\pm$ 0,0*	0,87 $\pm$ 0,0*	0,98 $\pm$ 0,0*	0,98 $\pm$ 0,0*	0,99 $\pm$ 0,0*
TUP	0,89 $\pm$ 0,0*	0,90 $\pm$ 0,0*	0,87 $\pm$ 0,01	0,25 $\pm$ 0,01	0,82 $\pm$ 0,0*	0,48 $\pm$ 0,01	0,92 $\pm$ 0,0*
WRE	0,41 $\pm$ 0,03	0,44 $\pm$ 0,03	0,44 $\pm$ 0,03	0,36 $\pm$ 0,03	0,35 $\pm$ 0,03	0,35 $\pm$ 0,03	0,39 $\pm$ 0,03
WWH	0,43 $\pm$ 0,02	0,53 $\pm$ 0,02	0,53 $\pm$ 0,02	0,27 $\pm$ 0,02	0,31 $\pm$ 0,02	0,33 $\pm$ 0,02	0,40 $\pm$ 0,02

* O desvio padrão não é exatamente zero, mas bem próximo.



a capacidade que nossa proposta tem de resolver diversos problemas de características distintas, se mostrando um modelo eficaz.

Podemos verificar na Tabela 12 que o resultado das bases de dados *delta ailerons* (DAI) e *delta elevators* (DEL) são apresentados como 0. Na realidade, o RMSE mostrados nestas duas bases de dados é bem próxima de zero, mas ainda são maiores. Para melhor

interpretação dos resultados para estas bases (e todas as outras), é preciso analisar também os resultados apresentados na Tabela 13 e na Figura 13 presente no Apêndice B.

E em ambas tabelas podemos verificar o plot de diferença crítica das duas métricas. É possível atestar que nossa proposta apresenta bons resultados, equivalente aos modelos de estado da arte disponíveis na literatura e é uma nova abordagem interessante do MLM que se destaca entre as outras variações.

4.2.4 Análise da relevância do número de RPs na etapa de predição

Um dos pilares da nossa versão peso leve da MLM é que assumimos que $\mathbf{B}$ totalmente estimada (significa que $\mathbf{B}$ é estimada de acordo com a Equação (3.5)) pode aprender toda a estrutura geométrica conhecida dos dados. Tal suposição nos encorajou a descartar alguns pontos de referência, já que sua projeção por $\mathbf{B}$ também alcançaria erro próximo de zero, ou seja, $\|\Psi(\mathbf{y}) - \Phi(\mathbf{x}) \mathbf{B}\|_2 \approx \|\mathcal{K}(\Psi(\mathbf{y}) - \Phi(\mathbf{x}) \mathbf{B})\|_2 \rightarrow 0$. Assim, neste experimento, apresentamos como a quantidade de pontos de referência na etapa de predição influencia o erro ao usar uma matriz $\mathbf{B}$ totalmente estimada.

Como estamos focando em tarefas de regressão cuja saída $\in \mathbb{R}$, precisamos de pelo menos dois pontos de referência devido à configuração de multilateração na MLM. Assim, variamos a quantidade de RPs de 2 pontos e, em seguida, aumentamos em uma etapa de múltiplos de 5% para mostrar como o modelo se comporta em relação ao RMSE.

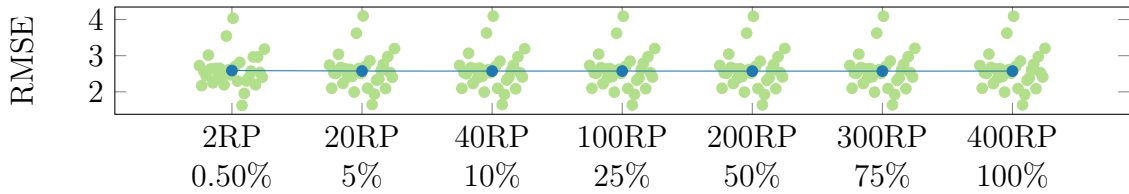


Figura 12 – Variação do RMSE na base Boston Housing.

Em cada agrupamento de pontos da Figura 12 mostramos o RMSE para 30 execuções independentes (pontos verdes) e as conectadas (azuis) representam a média para cada conjunto de 30 execuções. Além disso, destacamos que, de acordo com a quantidade de PRs, pode-se observar que, para valores entre 5% e 50%, a LW-MLM pode ser considerado como Random-MLM ou Rank-MLM, enquanto os 100% em a LW-MLM pode ser considerado como uma Full-MLM, sempre levando em consideração a regularização. Este experimento com o restante dos conjuntos de dados está presente no Apêndice B.

4.2.5 Análise da complexidade computacional entre as variações do MLM

Como nossa proposta consiste em duas etapas: a seleção dos pontos de referência e o cálculo da solução de um sistema linear, a complexidade geral da LW-MLM é dada

Tabela 14 – Comparação das variações do MLM em termos de requisitos de memória, custo de treinamento e de predição. Lembramos que N representa a cardinalidade do conjunto de treinamento, enquanto M é o número de RPs ($N \geq M$).

MODELO	MEMÓRIA	TREINAMENTO	PREDIÇÃO
Full-MLM	$\mathcal{O}(N^2)$	$\mathcal{O}(N^3)$	$\mathcal{O}(N)$
Random-MLM	$\mathcal{O}(NM)$	$\mathcal{O}(NM^2)$	$\mathcal{O}(M)$
Rank-MLM	$\mathcal{O}(NM)$	$\mathcal{O}(NM^2)$	$\mathcal{O}(M)$
LW-MLM	$\mathcal{O}(N^2)$	$\mathcal{O}(N^3)$	$\mathcal{O}(M)$

pelo soma da complexidade dessas duas etapas.

Como utilizamos todo o conjunto de dados como pontos de referência na etapa de treinamento, a memória usada é da ordem de $\mathcal{O}(N)$ para $\mathbf{D}$ e $\mathbf{P}$, e a solução do sistema linear é fornecida por $\mathcal{O}(N^3)$, que é praticamente o mesmo que a Full-MLM. No entanto, a etapa de predição tem um custo menor pois menos pontos são utilizados. Como mostramos nos resultados anteriores, a LW-MLM pode aprender toda a estrutura geométrica dos dados durante o treinamento, e utilizar somente alguns deles na predição. Para simplificar nossa notação, adotamos M como o número de RPs usados na etapa de predição, e já discutimos como o M influencia os resultados dos experimentos. Dito isso, obtemos o custo de predição de $\mathcal{O}(M)$. Negligenciamos o custo relacionado à criação de $\mathbf{P}$ e ao processo de otimização, pois eles são assintoticamente menores que a resolução do sistema.

Como se pode ver na Tabela 14, a LW-MLM é uma variante competitiva, pois mantém os custos polinomiais relativos aos custos de memória, treinamento e predição. Embora a LW-MLM tenha um custo semelhante à Full-MLM, o treinamento é realizado uma vez, enquanto a etapa de predição (a mais crítica) foi reduzida, permanecendo semelhante às outras variantes. Essa descoberta é empolgante, pois, quando combinada com as taxas de erro mais altas apresentadas, número de RPs e o valor da norma de $\mathbf{B}$ em relação a outras variantes de MLM, indica que nossa formulação é benéfica para o modelo de MLM.

4.3 Conclusão

Neste capítulo é apresentada a metodologia utilizada para validação dos modelos propostos, bem como as características dos conjuntos de dados utilizados. Os experimentos são divididos em duas seções, a primeira levando em consideração a OR-MLM e a FCM-MLM para problemas de classificação, e a segunda com a LW-MLM para problemas de regressão.

Foram realizados vários experimentos através de várias perspectivas para melhor

compreender os modelos propostos. As propostas foram comparadas com outros modelos disponíveis na literatura e conseguiram apresentar ótimos resultados, se mostrando modelos de qualidade equivalente ou superior aos modelos da literatura além de obter resultados competitivos.

5 Conclusão

O objetivo deste último capítulo é apresentar as conclusões sobre as propostas deste trabalho e também a sua contribuição para a máquina de aprendizagem mínima, tendo em vista que se trata de um modelo de aprendizagem de máquina recente que ainda carece de mais estudos. Por fim, algumas sugestões de trabalhos futuros que podem enriquecer o conhecimento teórico sobre a proposta apresentada.

5.1 Considerações finais

Primeiramente, este trabalho de mestrado contribuiu com duas novas versões da máquina de aprendizagem mínima para classificação, a OR-MLM e a FCM-MLM. Ambas propostas foram validadas através de simulações em problemas do mundo real e conjuntos de dados artificiais contra outras versões da MLM. Os resultados se mostraram satisfatórios, principalmente as da Proposta II, que alcançou melhores resultados de redução de modelo, além de ter alcançado uma boa acurácia em diversos conjuntos de dados, e na análise qualitativa se mostrou muito eficiente.

Além disso, também propusemos mais uma nova formulação para a MLM, esta com base no fator de regularização de amostras, denominada máquina de aprendizagem mínima de peso leve (LW-MLM). Ampliamos principalmente a definição da função de custo na versão original da MLM e, em seguida, incorporamos um procedimento de aceleração na etapa de predição.

Validamos a LW-MLM através de diversos tipos de experimentos, avaliando diferentes aspectos: erro de predição, qualidade dos valores estimados versus medidos, complexidade do modelo (via norma da matriz $\mathbf{B}$ quando comparado a outras variações da MLM) e, finalmente, o influência do número de PRs na previsão fora da amostra. No primeiro conjunto de experimentos, avaliamos nossa proposta e algumas variantes da MLM por meio do erro de previsão e da qualidade do ajuste dos valores estimados versus medidos em uma maneira típica de teste de caixa preta em alguns conjuntos de dados, e depois repetimos os experimentos entre nossa proposta e outros modelos da literatura. Nesta avaliação típica de caixa preta, nossa proposta alcançou taxas de precisão semelhantes ou mais altas em relação a outras variantes, todas com baixa variação, sendo todas consideradas estatisticamente equivalentes.

No segundo conjunto de experimentos, avaliamos visualmente as variantes da MLM em alguns problemas de regressão. A partir de uma análise empírica, constatamos que, devido a nossa formulação na LW-MLM, alguns problemas possam tirar proveito da

abordagem de regularização por padrão, enquanto outros modelos não executam com precisão. No terceiro conjunto de experimentos, no qual abordamos a relevância dos PRs durante a etapa de predição, destacamos a propriedade de estabilidade da LW-MLM referente a essa seleção, mesmo quando menos pontos são empregados. Através dos resultados, notamos que o aspecto de dpeso leve na LW-MLM é benéfico para o modelo MLM, pois o mantém como uma variante competitiva com um custo polinomial referente à memória, treinamento, enquanto a etapa de predição (a mais crítica) é reduzido em termos de complexidade computacional, permanecendo semelhante às outras variantes em relação ao erro de predição.

5.2 Contribuições desse trabalho

As contribuições desse trabalho estão relacionadas principalmente à máquina de aprendizagem mínima, já que é apresentada uma novas abordagens, duas delas para problemas de classificação que utiliza algoritmos externos para selecionar o melhor conjunto de RPs, e uma terceira versão na qual dispensa a necessidade de utilizar outro algoritmo para seleção de RPs, e que além disso, trabalha na redução dos cálculos na etapa de predição.

Uma segunda contribuição é a própria revisão bibliográfica acerca de métodos de aproximação de função, e sobre a importância da regularização de modelos, com o intuito de gerar modelos capazes de representar bem os dados, evitando que a aprendizagem esteja limitada somente ao conjunto de treinamento.

5.3 Trabalhos futuros

Como sugestão de trabalhos futuros, a primeira atividade a ser feita é o de testar a qualidade da Proposta III, LW-MLM, para problemas de classificação. Em tarefas de classificação, a abordagem na MLM é diferente na etapa de predição. Além disso, há algumas propostas da MLM que realizam a seleção esparsa de pontos de referências através de algum critério e que precisam ser comparadas ao nosso modelo. Ademais, é importante que haja um estudo investigativo sobre a matriz $\mathbf{P}$, com o intuito de buscar uma formulação definitiva para a proposta.

Referências

ALENCAR, A. S. C. et al. Mlm-rank: A ranking algorithm based on the minimal learning machine. In: *2015 Brazilian Conference on Intelligent Systems, BRACIS 2015, Natal, Brazil, November 4-7, 2015*. [s.n.], 2015. p. 305–309. Disponível em: <<https://doi.org/10.1109/BRACIS.2015.39>>. Citado 2 vezes nas páginas 48 e 64.

BARTLETT, P. L. The sample complexity of pattern classification with neural networks: The size of the weights is more important than the size of the network. *IEEE Trans. Information Theory*, v. 44, n. 2, p. 525–536, 1998. Disponível em: <<https://doi.org/10.1109/18.661502>>. Citado na página 39.

BEZDEK, J. C. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Springer, 1981. ISBN 978-1-4757-0452-5. Disponível em: <<https://doi.org/10.1007/978-1-4757-0450-1>>. Citado 2 vezes nas páginas 43 e 54.

BISHOP, C. M. *Pattern recognition and machine learning, 5th Edition*. Springer, 2007. (Information science and statistics). ISBN 9780387310732. Disponível em: <<http://www.worldcat.org/oclc/71008143>>. Citado na página 27.

BRAGA, A. de P. *Redes neurais artificiais: teoria e aplicações*. LTC Editora, 2007. ISBN 9788521615644. Disponível em: <<https://books.google.com.br/books?id=R-p1GwAACAAJ>>. Citado na página 36.

BROOMHEAD, D. S.; LOWE, D. Multivariable functional interpolation and adaptive networks. *Complex Systems*, v. 2, n. 3, 1988. Disponível em: <http://www.complex-systems.com/abstracts/v02_i03_a05.html>. Citado na página 36.

COELHO, D. N. et al. Performance comparison of classifiers in the detection of short circuit incipient fault in a three-phase induction motor. In: *2014 IEEE Symposium on Computational Intelligence for Engineering Solutions, CIES 2014, Orlando, FL, USA, December 9-12, 2014*. [S.l.: s.n.], 2014. p. 42–48. Citado na página 28.

DEMŠAR, J. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, v. 7, p. 1–30, 2006. Cited By 5091. Disponível em: <<https://www.scopus.com/inward/record.uri?eid=2-s2.0-29644438050&partnerID=40&md5=81992d7f5603cc9729bb9c2c1dcd4f8c>>. Citado na página 65.

DUDA, R. O.; HART, P. E.; STORK, D. G. *Pattern classification, 2nd Edition*. Wiley, 2001. ISBN 9780471056690. Disponível em: <<http://www.worldcat.org/oclc/41347061>>. Citado na página 28.

DUNN, J. C. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, Taylor & Francis, v. 3, n. 3, p. 32–57, 1973. Citado na página 43.

FLETCHER, R. *Practical methods of optimization*. [S.l.]: John Wiley & Sons, 1987. Citado na página 40.

- GOLUB, G. H.; LOAN, C. F. V. *Matrix computations*. [S.l.]: JHU press, 2012. v. 3. Citado na página 40.
- HAYKIN, S. S. et al. *Neural networks and learning machines/Simon Haykin*. [S.l.]: New York: Prentice Hall,, 2009. Citado 2 vezes nas páginas 33 e 34.
- HEIAT, A. Comparison of artificial neural network and regression models for estimating software development effort. *Information and Software Technology*, v. 44, n. 15, p. 911 – 922, 2002. ISSN 0950-5849. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0950584902001283>>. Citado na página 36.
- HUANG, G.; CHEN, L. Convex incremental extreme learning machine. *Neurocomputing*, v. 70, n. 16-18, p. 3056–3062, 2007. Disponível em: <<https://doi.org/10.1016/j.neucom.2007.02.009>>. Citado na página 38.
- HUANG, G.-B. et al. Extreme learning machine: a new learning scheme of feedforward neural networks. *Neural networks*, v. 2, p. 985–990, 2004. Citado na página 38.
- JAMES, G. et al. *An introduction to statistical learning*. [S.l.]: Springer, 2013. v. 112. Citado na página 28.
- KOHONEN, T.; RUOHONEN, M. Representation of associated data by matrix operators. *IEEE Trans. Computers*, v. 22, n. 7, p. 701–702, 1973. Disponível em: <<https://doi.org/10.1109/TC.1973.5009138>>. Citado na página 36.
- LICHMAN, M. *UCI Machine Learning Repository*. 2013. Citado 2 vezes nas páginas 60 e 63.
- LLOYD, S. P. Least squares quantization in PCM. *IEEE Trans. Information Theory*, v. 28, n. 2, p. 129–136, 1982. Disponível em: <<https://doi.org/10.1109/TIT.1982.1056489>>. Citado na página 42.
- MACQUEEN, J. Some methods for classification and analysis of multivariate observations. In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*. Berkeley, Calif.: University of California Press, 1967. p. 281–297. Disponível em: <<http://projecteuclid.org/euclid.bsmmsp/1200512992>>. Citado na página 42.
- MAIA, Á. N. et al. Optimally selected minimal learning machine. In: *Intelligent Data Engineering and Automated Learning - IDEAL 2018 - 19th International Conference, Madrid, Spain, November 21-23, 2018, Proceedings, Part I*. [s.n.], 2018. p. 670–678. Disponível em: <https://doi.org/10.1007/978-3-030-03493-1_70>. Citado na página 60.
- MARINHO, L. B. et al. A novel mobile robot localization approach based on topological maps using classification with reject option in omnidirectional images. *Expert Systems with Applications*, v. 72, p. 1 – 17, 2017. ISSN 0957-4174. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0957417416306790>>. Citado na página 28.
- MASSEY, F. J. The kolmogorov-smirnov test for goodness of fit. *Journal of the American Statistical Association*, [American Statistical Association, Taylor & Francis, Ltd.], v. 46, n. 253, p. 68–78, 1951. ISSN 01621459. Disponível em: <<http://www.jstor.org/stable/2280095>>. Citado na página 57.

MELO, D. B. d. *Algoritmos de aprendizagem para aproximação da cinemática inversa de robôs manipuladores: um estudo comparativo*. Tese (Doutorado), 2015. Citado na página 28.

MESQUITA, D. P. P.; GOMES, J. P. P.; Souza Júnior, A. H. de. Ensemble of efficient minimal learning machines for classification and regression. *Neural Processing Letters*, v. 46, n. 3, p. 751–766, 2017. Disponível em: <<https://doi.org/10.1007/s11063-017-9587-5>>. Citado 2 vezes nas páginas 47 e 50.

NIEWIADOMSKA-SZYNKIEWICZ, E.; MARKS, M. Optimization schemes for wireless sensor network localization. *Applied Mathematics and Computer Science*, v. 19, n. 2, p. 291–302, 2009. Disponível em: <<https://doi.org/10.2478/v10006-009-0025-3>>. Citado na página 46.

PETERSEN, K. B.; PEDERSEN, M. S. et al. The matrix cookbook. *Technical University of Denmark*, v. 7, n. 15, p. 510, 2008. Citado na página 87.

POWELL, M. J. D. Algorithms for approximation. In: MASON, J. C.; COX, M. G. (Ed.). New York, NY, USA: Clarendon Press, 1987. cap. Radial Basis Functions for Multivariable Interpolation: A Review, p. 143–167. ISBN 0-19-853612-7. Disponível em: <<http://dl.acm.org/citation.cfm?id=48424.48433>>. Citado na página 37.

ROJAS, I. et al. Time series analysis using normalized pg-rbf network with regression weights. *Neurocomputing*, v. 42, n. 1, p. 267 – 285, 2002. ISSN 0925-2312. Evolutionary neural systems. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0925231201003381>>. Citado na página 36.

SANCHES, M. K. *Aprendizado de máquina semi-supervisionado: proposta de um algoritmo para rotular exemplos a partir de poucos exemplos rotulados*. Dissertação (Mestrado em Ciências de Computação e Matemática Computacional) — Universidade de São Paulo, São Carlos, 2003. Citado na página 28.

SAUNDERS, C.; GAMMERMAN, A.; VOVK, V. Ridge regression learning algorithm in dual variables. In: *Proceedings of the Fifteenth International Conference on Machine Learning (ICML 1998), Madison, Wisconsin, USA, July 24-27, 1998*. [S.l.: s.n.], 1998. p. 515–521. Citado na página 39.

SEZGIN, M.; SANKUR, B. Survey over image thresholding techniques and quantitative performance evaluation. *J. Electronic Imaging*, v. 13, n. 1, p. 146–168, 2004. Disponível em: <<https://doi.org/10.1117/1.1631315>>. Citado na página 57.

SILVERMAN, B. W. Some aspects of the spline smoothing approach to non-parametric regression curve fitting. *Journal of the Royal Statistical Society: Series B (Methodological)*, Wiley Online Library, v. 47, n. 1, p. 1–21, 1985. Citado na página 68.

SOUZA, A. H. de et al. Minimal learning machine: a novel supervised distance-based approach for regression and classification. *Neurocomputing*, Elsevier, v. 164, p. 34–44, 2015. Citado 3 vezes nas páginas 28, 29 e 47.

SOUZA, A. H. de et al. Minimal learning machine: A new distance-based method for supervised learning. In: *Advances in Computational Intelligence - 12th International Work-Conference on Artificial Neural Networks, IWANN 2013, Puerto de la Cruz, Tenerife, Spain, June 12-14, 2013, Proceedings, Part I*. [s.n.], 2013. p. 408–416. Disponível

em: <https://doi.org/10.1007/978-3-642-38679-4_40>. Citado 2 vezes nas páginas 45 e 48.

SOUZA JÚNIOR, A. H. d. *Regional Models and Minimal Learning Machines for Nonlinear Dynamic System Identification*. Tese (Doutorado) — Federal University of Ceará, 2014. Citado na página 56.

SUTTON, R. S.; BARTO, A. G. *Reinforcement learning - an introduction*. MIT Press, 1998. (Adaptive computation and machine learning). ISBN 0262193981. Disponível em: <<http://www.worldcat.org/oclc/37293240>>. Citado na página 28.

SUYKENS, J. A. K. et al. *Least Squares Support Vector Machines*. WORLD SCIENTIFIC, 2002. Disponível em: <<https://www.worldscientific.com/doi/abs/10.1142/5089>>. Citado na página 39.

THEODORIDIS, S. *Machine learning: a Bayesian and optimization perspective*. [S.l.]: Academic Press, 2015. Citado 3 vezes nas páginas 27, 28 e 41.

THEODORIDIS, S.; KOUTROUMBAS, K. Pattern recognition. *IEEE Trans. Neural Networks*, v. 19, n. 2, p. 376, 2008. Disponível em: <<https://doi.org/10.1109/TNN.2008.929642>>. Citado na página 42.

TIKHONOV, A. N. On the stability of inverse problems. In: *Dokl. Akad. Nauk SSSR*. [S.l.: s.n.], 1943. v. 39, p. 195–198. Citado na página 35.

TIPPING, M. E. Sparse bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, v. 1, p. 211–244, 2001. Disponível em: <<http://www.ai.mit.edu/projects/jmlr/papers/volume1/tipping01a/abstract.html>>. Citado na página 51.

TORGO, L. *Regression datasets*. 2005. Citado na página 63.

VAPNIK, V. N. *The Nature of Statistical Learning Theory*. New York: Springer, 1995. Citado na página 51.

Whei-Min Lin et al. A fault classification method by rbf neural network with ols learning procedure. *IEEE Transactions on Power Delivery*, v. 16, n. 4, p. 473–477, Oct 2001. ISSN 0885-8977. Citado na página 36.

WILLSHAW, D. *Models of distributed associative memory*. Tese (Doutorado) — University of Edinburgh, UK, 1971. Disponível em: <<http://hdl.handle.net/1842/8130>>. Citado na página 36.

ZHAO, Y. et al. An improved recursive reduced least squares support vector regression. *Neurocomputing*, v. 87, p. 1–9, 2012. Disponível em: <<https://doi.org/10.1016/j.neucom.2012.01.015>>. Citado 2 vezes nas páginas 64 e 73.

Apêndices

APÊNDICE A – Solução da Equação (3.14)

Em primeiro lugar, reescrevemos a Equação (3.1)

$$\frac{\partial}{\partial \mathbf{B}} \mathcal{J}(\mathbf{B}) = \frac{\partial}{\partial \mathbf{B}} \|\mathbf{DB} - \Delta\|_{\mathcal{F}}^2 + \|\mathbf{PB}\|_{\mathcal{F}}^2 = \frac{\partial}{\partial \mathbf{B}} \|\mathbf{DB} - \Delta\|_{\mathcal{F}}^2 + \frac{\partial}{\partial \mathbf{B}} \|\mathbf{PB}\|_{\mathcal{F}}^2. \quad (\text{A.1})$$

Utilizando as seguintes propriedades retiradas do livro Matrix Cookbook (PETERSEN; PEDERSEN et al., 2008):

- Propriedades n°103 e n°104: $\frac{\partial}{\partial \mathbf{X}} \text{Tr}\{\mathbf{AX}^\top\} = \frac{\partial}{\partial \mathbf{X}} \text{Tr}\{\mathbf{X}^\top \mathbf{A}\} = \mathbf{A}$;
- Propriedade n°108: $\frac{\partial}{\partial \mathbf{X}} \text{Tr}\{\mathbf{X}^\top \mathbf{BX}\} = \mathbf{BX} + \mathbf{B}^\top \mathbf{X}$;
- Propriedade n°115: $\frac{\partial}{\partial \mathbf{X}} \|\mathbf{X}\|_{\mathcal{F}}^2 = \frac{\partial}{\partial \mathbf{X}} \text{Tr}\{\mathbf{XX}^H\} = \frac{\partial}{\partial \mathbf{X}} \text{Tr}\{\mathbf{X}^H \mathbf{X}\} = 2\mathbf{X}$.

onde $\mathbf{X}^H$ é a transposta e a matriz complexa conjugada (Hermitian) de $\mathbf{X}$, obtém-se:

$$\frac{\partial}{\partial \mathbf{B}} \mathcal{J}(\mathbf{B}) = \frac{\partial}{\partial \mathbf{B}} \text{Tr}\{(\mathbf{DB} - \Delta)^\top (\mathbf{DB} - \Delta)\} + \frac{\partial}{\partial \mathbf{B}} \text{Tr}\{(\mathbf{PB})^\top \mathbf{PB}\}. \quad (\text{A.2})$$

$$= \frac{\partial}{\partial \mathbf{B}} \text{Tr}\{\mathbf{B}^\top \mathbf{D}^\top \mathbf{DB} - \mathbf{B}^\top \mathbf{D}^\top \Delta - \Delta^\top \mathbf{DB} + \Delta^\top \Delta\} + \frac{\partial}{\partial \mathbf{B}} \text{Tr}\{\mathbf{B}^\top \mathbf{P}^\top \mathbf{PB}\}. \quad (\text{A.3})$$

Como $\partial(\cdot)$ e $\text{Tr}\{\cdot\}$ são operadores lineares, nós reescrevemos a Eq () como:

$$= \frac{\partial}{\partial \mathbf{B}} \text{Tr}\{\mathbf{B}^\top \mathbf{D}^\top \mathbf{DB}\} - \frac{\partial}{\partial \mathbf{B}} \text{Tr}\{\mathbf{B}^\top \mathbf{D}^\top \Delta\} - \frac{\partial}{\partial \mathbf{B}} \text{Tr}\{\Delta^\top \mathbf{DB}\} + \frac{\partial}{\partial \mathbf{B}} \text{Tr}\{\mathbf{B}^\top \mathbf{P}^\top \mathbf{PB}\}. \quad (\text{A.4})$$

$$= \mathbf{D}^\top \mathbf{DB} + (\mathbf{D}^\top \mathbf{D})^\top \mathbf{B} - \mathbf{D}^\top \Delta - (\Delta^\top \mathbf{D})^\top + \mathbf{P}^\top \mathbf{PB} + (\mathbf{P}^\top \mathbf{P})^\top \mathbf{B}. \quad (\text{A.5})$$

Desde que $\mathbf{D}$, Δ , e $\mathbf{P}$ são simétricos, nós expressamos $\frac{\partial}{\partial \mathbf{B}}$ como segue:

$$\frac{\partial}{\partial \mathbf{B}} \mathcal{J}(\mathbf{B}) = 2\mathbf{D}^\top \mathbf{DB} - 2\mathbf{D}^\top \Delta + 2\mathbf{P}^\top \mathbf{PB}. \quad (\text{A.6})$$

Finalmente, definindo $\frac{\partial}{\partial \mathbf{B}} \mathcal{J}(\mathbf{B}) = \mathbf{0}$ para alcançar a solução ideal, obtém-se:

$$\hat{\mathbf{B}} = (\mathbf{D}^\top \mathbf{D} + \mathbf{P}^\top \mathbf{P})^{-1} \mathbf{D}^\top \Delta.$$

APÊNDICE B – Relevância do número de RPs na etapa de predição

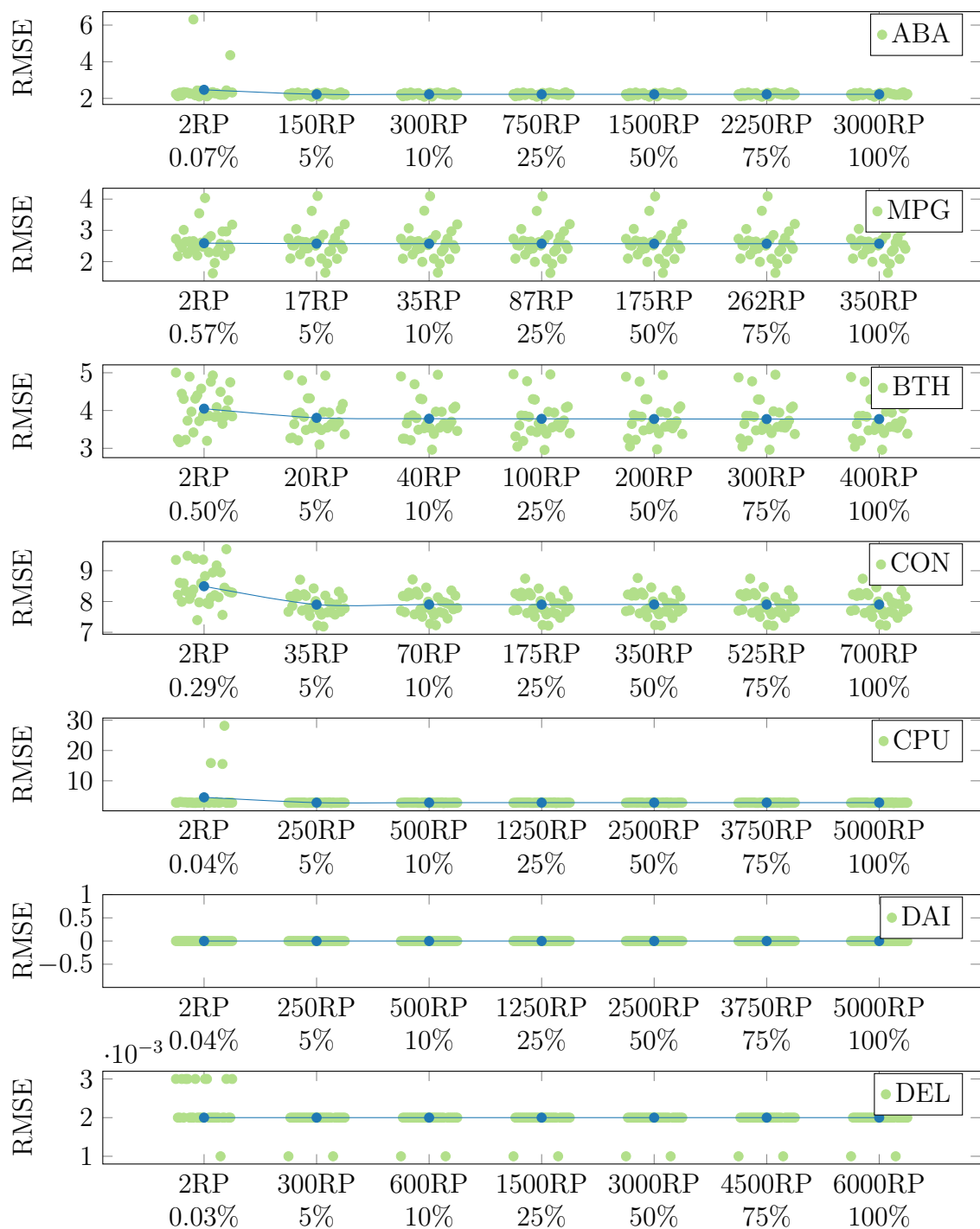


Figura 13 – Variação do RMSE para algumas bases

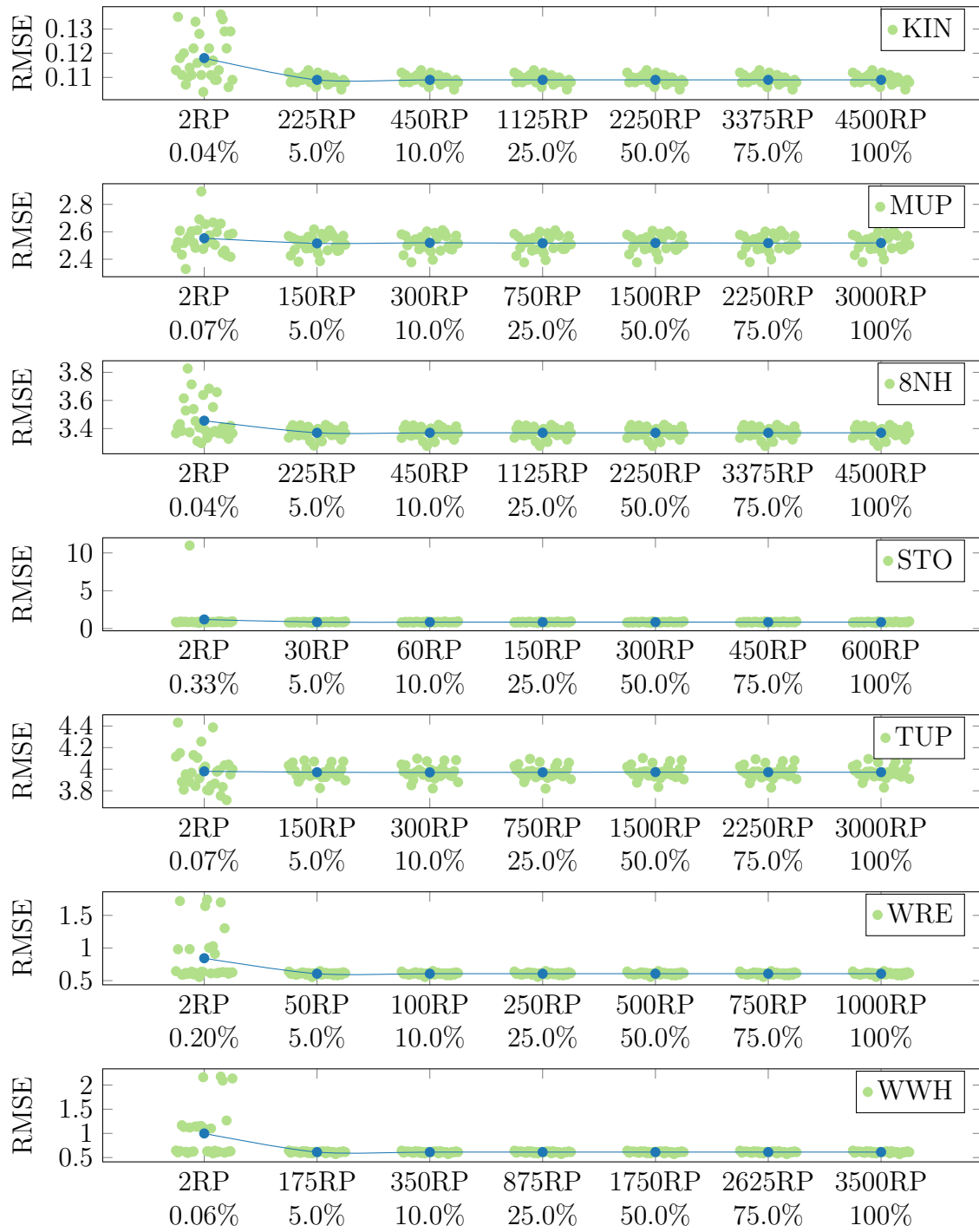


Figura 14 – Variação do RMSE para algumas bases