

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

Lucas Silva de Sousa

**Estratégia Bayesiana para Treinamento de Máquinas de
Aprendizagem Mínima Esparsas**

Fortaleza – CE

30 de Janeiro de 2018

Lucas Silva de Sousa

Estratégia Bayesiana para Treinamento de Máquinas de Aprendizagem Mínima Esparsas

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

30 de Janeiro de 2018

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)

SOUSA, LUCAS.

Estratégia Bayesiana para Treinamento de Máquinas de Aprendizagem Mínima Esparsas / LUCAS
SOUSA. - 2018.
120 f. : il. color.

Dissertação (Mestrado) - Instituto Federal do Ceará, Mestrado em Ciência da Computação, Campus
Fortaleza, 2018.

Orientação: Prof. Ajalmar Rêgo da Rocha Neto.

1. Máquina de aprendizagem mínima. 2. Esparsidade. 3. Sistemas lineares de múltiplas respostas. 4.
Pontos de referência. I. Título.

CDD 004

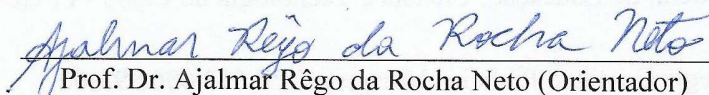
LUCAS SILVA DE SOUSA

Estratégia Bayesiana Esparsa para Treinamento de Máquinas de Aprendizagem Mínima

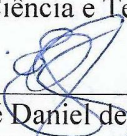
Dissertação submetida à Coordenação do Curso de Pós-graduação em Ciência da Computação do Instituto Federal de Educação, Ciência e Tecnologia do Ceará, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação, área de concentração Ciência da Computação.

Aprovada em 30/01/18.

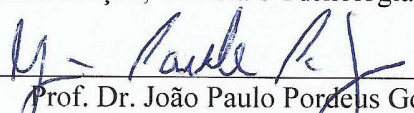
BANCA EXAMINADORA


Prof. Dr. Ajalmar Rêgo da Rocha Neto (Orientador)

Instituto Federal de Educação, Ciência e Tecnologia do Ceará - PPGCC/IFCE


Prof. Dr. José Daniel de Alencar Santos
Instituto Federal de Educação, Ciência e Tecnologia do Ceará - PPGCC/IFCE


Prof. Dr. Regis Cristiano Pinheiro Marques
Instituto Federal de Educação, Ciência e Tecnologia do Ceará - PPGET/IFCE


Prof. Dr. João Paulo Pordus Gomes
Universidade Federal do Ceará - UFC

Este trabalho é dedicado à todos os membros da minha árvore genealógica, do nó raiz até a última folha.

Agradecimentos

Durante o curso de pós-graduação, a palavra chave é dedicação. Não falo de dedicação à família, aos amigos, mas sim à um universo restrito e dominado pela ciência e pela lógica. Um sintoma de que você caiu nesse meio é quando se encontra conversando com sua esposa ou brincando com o filho, somente em corpo presente. Isso ocorre pois sua cabeça está imaginando o comportamento de alguma superfície de decisão após a aplicação de algum método que você elaborou na noite que levou uma multa de trânsito por dirigir distraído. Outro sinal é quando alguém te conta uma história e antes dela terminar você já esqueceu do que se trata e faz aquela velha pergunta: “Sobre o que tu falas mesmo?”. O lado bom disso tudo, é que existem pessoas que mergulham nessa insanidade com você e outras que te resgatam. Aos que me resgataram eu agradeço, aos que me seguiram, nem tanto. De qualquer modo, a seguir faço um agradecimento especial à todos que contribuíram diretamente ou não com este trabalho.

Em primeiro lugar, agradeço aos meus pais Francisco Cleiton e Maria Elisângela por todo o apoio incondicional e por terem me mostrado a teoria de como ser uma pessoa responsável. Também agradeço ao meu filho Leozinho e a minha adorável esposa Aline Duarte pela paciência e por suportarem as viagens adiadas por minha causa. Além disso, eles me fizeram gostar da época em que responsabilidade ficava apenas na teoria. Ao que sobrou da família, eu agradeço por todo amor que sempre me deram e, principalmente à alguns que me serviram de exemplo a não ser seguido.

Tenho certeza de que o período do mestrado não passaria tão rápido se não fosse ao lado de tantos amigos inteligentes e engraçados. Agradeço ao meu orientador Prof. Dr. Ajalmar Rocha pela paciência e por todos os ensinamentos e ao meu amigo Saulo Freitas, pelo tempo e apoio no desenvolvimento dessa dissertação. Também agradeço aos meus amigos de laboratório Madson, Alan, Shara, Elizangela, Ananda, Matias e Alberth pelas trocas de experiências. Agradeço ao PPGCC/IFCE e a todos os professores pela oportunidade de curso de mestrado. Por fim, agradeço a FUNCAP pelo fornecimento da bolsa de mestrado.

“Ainda que eu ande pelo vale da sombra da morte, não temerei mal algum.”
(Salmos 23:4)

Resumo

A máquina de aprendizagem mínima (*Minimal Learning Machine*, MLM) é um método em que seu treinamento ocorre pelo cômputo de um mapeamento entre distâncias de entrada e de saída. Esse mapeamento depende de dois processos, que são a seleção dos pontos de referência e a resolução do sistema linear de múltiplas respostas. Durante a seleção de pontos de referência devem ser satisfeitos dois critérios: (i) redução da quantidade de pontos de referência e (ii) quais são os pontos de referência mais adequados. Porém, na proposta original do modelo MLM, a seleção desses pontos é realizada de maneira aleatória. Os pontos de referência são utilizados para definir as matrizes de distância no espaço de entrada (características) e saída (rótulos). Cada linha dessas matrizes, possui as distâncias de um padrão de treinamento para todos os pontos de referência. Por outro lado, cada ponto de referência define uma coluna das matrizes de distâncias. A solução obtida pela resolução do sistema linear de múltiplas respostas define o mapeamento entre as matrizes de distâncias no espaço de entrada e de saída. A seleção adequada de pontos de referência permite que se obtenha soluções menos complexas para o sistema linear de múltiplas respostas. Além disso, a redução do número de pontos de referência diminui o custo computacional do modelo MLM em sua fase de predição, que depende de um processo de otimização. Soluções esparsas de sistemas lineares de múltiplas respostas possuem a característica de ter a matriz de parâmetros com a maioria das linhas iguais a zero. Cada linha na matriz de pesos com valores iguais a zero, é equivalente a remover uma coluna da matriz de coeficientes. Relacionando isso ao MLM, a seleção de pontos de referência é um processo de tentar estimar as colunas da matriz de distâncias que são equivalentes às linhas da matriz de pesos diferentes de zero. Por isso, o processo de seleção de pontos de referência e cômputo do mapeamento entre as matrizes de distância pode ser tratado como um problema de resolução esparsa de sistemas lineares de múltiplas respostas, que é um problema NP-Difícil. Existem diversos métodos capazes de obter aproximações para esse tipo de problema, mas a extensão do algoritmo de Aprendizado Bayesiano Esperso (*Sparse Bayesian Learning*, SBL) para sistemas lineares de múltiplas respostas, M-SBL, é capaz de obter uma solução esparsa e, por ser uma estratégia Bayesiana, incorpora uma medida de complexidade chamada de navalha de Occam. Neste trabalho, é proposta uma abordagem Bayesiana para treinamento do modelo MLM utilizando o algoritmo M-SBL, chamado de SBMLM (*Sparse Bayesian Minimal Learning Machine*). Com essa abordagem espera-se treinar o modelo MLM selecionando os pontos de referência e estimando a matriz de parâmetros como tarefas integradas, além de se gerar modelos menos complexos.

Palavras-chave: Máquina de aprendizagem mínima, esparsidade, sistemas lineares de múltiplas respostas, pontos de referência.

Abstract

The Minimal Learning Machine (MLM) is a method in which its training process consists of calculating a mapping between input and output distances. This mapping depends on two processes: the reference points selection, and the solution of a multiresponse linear system. During the reference points selection, two criteria must be satisfied: (i) choose a prior number of reference points, and (ii) choose, in a suitable manner, reference points. However, in the original MLM model, the reference point selection occurs randomly. The reference points are used to define the distance matrices in the input space (features) and output spaces (labels). Each row of the matrices represent the distances from a training sample to all reference points. On the other hand, each reference point defines a distance matrix column. The multiresponse linear system solutions defines the mapping between the distance matrices in the input and output spaces. The suitable reference points choice, allow that less complex solution is obtained to the multiresponse linear system. Furthermore, a small number of reference points, reduce the computational cost of the optimization algorithm in the prediction step of the MLM. Sparse solutions of multiresponse linear systems are characterized by the presence of many zero rows in the parameter matrix. Obtain a zero row in the parameter matrix is equivalent to remove a column from the coefficient matrix. Thus, the reference point selection in the MLM is a process that we estimate the columns of the distance matrices that are related to non-zero rows of the parameter matrix. The reference point selection process and the parameter matrix estimation can be embodied in the multiresponse linear system solution process. Unfortunately, such problem is an NP-Hard, thus approximation solutions are obtained. The Sparse Bayesian Learning (SBL) is an algorithm that obtains sparse solutions for multiresponse linear systems and, as it is a Bayesian strategy, it embodies a complexity measure called Occam's razor. Such method was extended to solve multiresponse linear systems and such version was named M-SBL. In this work, we propose a Bayesian strategy based on the M-SBL algorithm in order to train the MLM model. Such method is called Sparse Bayesian Minimal Learning Machine (SBMLM). In this model, the reference point selection and the multiresponse linear system solution are treated by the same algorithm. Furthermore, we intend that the proposed method generates less complex models than the standard MLM algorithm.

Keywords: Minimal Learning Machine, sparsity, multiresponse linear systems, reference points.

Lista de ilustrações

Figura 1 – Exemplos de distribuições normais univariadas.	40
Figura 2 – Exemplos de distribuições normal bivariadas. São apresentadas diversas formas da distribuição com base na disposição da matriz de covariância. Nas Figs. 2a, 2b e 2c são apresentados os casos em que a matriz de covariância é diagonal. No caso ilustrado na Fig. 2d a distribuição normal possui uma matriz de covariância não-diagonal.	42
Figura 3 – Exemplos de distribuições Laplacianas.	43
Figura 4 – Exemplos de distribuições Gamma.	44
Figura 5 – Exemplos de distribuições beta.	45
Figura 6 – Distribuições normais para um problema de duas classes com probabilidades <i>a priori</i> iguais. Nesse caso, as distribuições <i>a posteriori</i> são iguais às verossimilhanças de cada classe. Nesse exemplo, os valores de $\mathbf{x} \in \mathbb{R}$, logo as distribuições são geradas no espaço $\mathbb{R}^2$. O ponto extremo ou limiar de decisão $\mathbf{x}_0$ é o ponto em que $p(\mathbf{x} \mathcal{C}_1) = p(\mathbf{x} \mathcal{C}_2)$. Esse ponto divide os padrões em duas regiões de decisão, R_1 e R_2	47
Figura 7 – Parâmetro estimado pelo método de máxima verossimilhança.	50
Figura 8 – Exemplos de funções de base Gaussiana.	55
Figura 9 – Exemplos de funções de base sigmoidais.	56
Figura 10 – Função de base tangente hiperbólica.	56
Figura 11 – Estimação da saída $\mathbf{y}^*$. Esse vetor representa o ponto que se deseja estimar através da multilateração, $\{\mathbf{t}\}_{k=1}^K$ são as imagens dos pontos de referência $\mathcal{D}_{ref}$ e $\mathbf{d}_{y^*}^k$, é a distância de saída estimada das imagens dos pontos de referência até o rótulo $\mathbf{y}^*$	75
Figura 12 – Superfícies de decisão do modelo MLM utilizando diferentes conjuntos de pontos de referência.	78
Figura 13 – Superfícies de decisão geradas pelo modelo SBMLM e pelo RNMLM em bases de dados artificiais.	85
Figura 14 – Superfícies geradas pelo modelo RNMLM utilizando uma quantidade de pontos de referência com uma quantidade aproximada da obtida pelo método SBMLM.	86
Figura 15 – Tempo de treinamento consumido pelos classificadores nas bases de dados utilizadas nesse experimento.	88
Figura 16 – Quantidade de pontos de referência média obtida pelos modelos SBMLM e RNMLM após o processo de treinamento. Esses valores foram obtidos sobre 30 realizações independentes e utilizando 80% da quantidade de amostras das bases de dados para treinamento.	90

Figura 17 – Redução média alcançada pelos modelos SBMLM e RNMLM após o processo de treinamento. Esses resultados são obtidos sobre 30 realizações independentes e utilizando 80% das amostras das bases de dados para treinamento.	91
Figura 18 – Variação da taxa de erro e número de pontos de referência em função do parâmetro β . Cada valor da taxa de erro e número de pontos de referência é obtido pela média de 30 realizações independentes.	92
Figura 19 – Decaimento do número de pontos de referência durante uma iteração do algoritmo M-SBL com alguns valores do parâmetro β	93
Figura 20 – Pontos selecionados pelo método Mapas Opostos Gaussianos para base de dados artificial com sobreposição.	108
Figura 21 – Superfícies de decisão obtidas com os modelos GOM-RVM, RVM e SVM.	110
Figura 22 – Quantidade de pontos retidos no treinamento dos modelos GOM-RVM, RVM e SVM.	111
Figura 23 – Centros de componentes de misturas de Gaussianas encontrados na base de dados RIP.	118
Figura 24 – Superfícies de decisão geradas pelo modelo GMCRVM e pelo modelo RVM na base dados RIP.	118

Lista de tabelas

Tabela 1	– Bases de dados utilizadas neste trabalho.	84
Tabela 2	– Resultados experimentais do SBMLM e de outros métodos da literatura. São apresentados resultados de acurácia e desvio padrão sobre 30 realizações independentes. Os indicadores (✓) e (✗), representam equivalência ou não-equivalência dos resultados com os do método SBMLM. O teste estatístico de Friedman define os valores dos indicadores de equivalência.	87
Tabela 3	– Resultados comparativos entre o método SBMLM com outras versões do MLM. Os métodos RNMLM e FLMLM, representam versões do MLM com escolha aleatória de pontos de referência e uma versão em que toda a base de dados é utilizada como pontos de referência, respectivamente. O teste estatístico de Friedman é utilizado para comparar os resultados com o modelo SBMLM, de modo que (✓) representa equivalência e (✗) representa não-equivalência.	89
Tabela 4	– Bases de dados utilizadas nos experimentos do método GOM-RVM. . .	109
Tabela 5	– Taxas de acerto obtidas com os métodos GOM-RVM, RVM e SVM. Os três indicadores ao lado dos resultados dos modelos RVM e SVM, (✓) ou (✗), representam equivalência ou não-equivalência com os respectivos resultados do método GOM-RVM. Esses resultados foram obtidos através do teste estatístico de Friedman.	111
Tabela 6	– Taxa de redução obtida com o método de mapas opostos Gaussianos. Para cada valor de λ , são apresentados valores de quantidade de pontos obtidos e o percentual de redução sobre o total da base de dados. . . .	112
Tabela 7	– Lista de bases de dados aplicadas na comparação do modelo GMC-RVM.	119
Tabela 8	– Valores de acurácia obtidas com os modelos GMC-RVM, RVM e SVM. Os resultados apresentados para os modelos RVM e SVM são acompanhados pelos indicadores de equivalência (✓) ou não-equivalência (✗) com os modelos GMC-RVM com $K = 10$, $K = 50$ e $K = 100$	119
Tabela 9	– Quantidade de vetores relevantes obtidos com os modelos GMC-RVM, RVM e SVM. Cada valor foi calculado sobre a média de 30 realizações independentes.	120

Lista de Algoritmos

1	Algoritmo SBL	65
2	Algoritmo M-SBL	71
3	Processo de treinamento do modelo MLM	76
4	Processo de predição do modelo MLM	76
5	Processo de treinamento do modelo SBMLM	81

Lista de abreviaturas e siglas

MLM	Máquinas de aprendizagem mínima
RVM	Máquinas de vetores relevantes
SBL	Aprendizagem Bayesiana esparsa
M-SBL	Aprendizagem Bayesiana esparsa para sistemas lineares de múltiplas respostas.
SBMLM	<i>Sparse Bayesian minimal learning machine</i>
ML	<i>Máxima verossimilhança</i>
MAP	<i>Máxima a posteriori</i>
ELM	<i>Máquina de aprendizagem extrema</i>
SVM	<i>Máquinas de Vetores Suporte</i>
LSSVM	<i>Máquinas de vetores suporte de mínimos quadrados</i>
ARD	<i>Detecção Automática de Relevância</i>
RNMLM	<i>Máquinas de aprendizagem mínima com seleção aleatória de pontos de referência</i>
FLMLM	<i>Máquinas de aprendizagem mínima com toda base de treinamento como ponto de referência</i>
GOM	<i>Mapas Opostos Gaussianos</i>
GOM-RVM	<i>Mapas Opostos Gaussianos para Máquinas de Vetores Relevantes</i>
GMC	<i>Centros de Misturas de Gaussianas</i>
GMC-RVM	<i>Centros de Misturas de Gaussianas para Máquinas de Vetores Relevantes</i>

Lista de símbolos

$p(\cdot)$	Probabilidade
X	Evento aleatório
$\mathbb{R}$	Conjunto dos números reais
x	Variável que pode assumir valores reais
$\boldsymbol{x}$	Vetor pertencente ao $\mathbb{R}^D$
σ^2	Variância
μ	Média de distribuição normal univariada.
$\boldsymbol{\mu}$	Média de distribuição normal multivariada.
$\boldsymbol{\Sigma}$	Covariância da distribuição normal multivariada.
$ \mathbf{M} $	Operador determinante de matriz
$ x $	Operador módulo de números reais
$\mathcal{N}$	Distribuição normal
$\boldsymbol{\theta}$	Conjunto de parâmetros de distribuição de probabilidade.
$\boldsymbol{\theta}_{\text{ML}}$	Parâmetros que maximizam a verossimilhança.
$E(\cdot)$	Esperança
$\mathcal{X}$	Conjunto de padrões
$\mathcal{Y}$	Conjunto de rótulos
$\boldsymbol{w}$	Vetor de pesos
$\boldsymbol{t}$	Saída desejada
$y(\boldsymbol{x}, \boldsymbol{w})$	Função linear com parâmetros $\boldsymbol{w}$ e $\boldsymbol{x}$
ϕ	Função de base
Φ	Conjunto de funções de base
Φ	Matriz de <i>kernel</i>

ϵ	Vetor de resíduo em sistemas lineares de resposta única
$\ \cdot\ _0$	Normal ℓ_0
$\mathcal{I}(\cdot)$	Função indicadora
$\triangleq$	Igual por definição
λ	Compromisso entre esparsidade e erro
$\ \cdot\ _2$	Norma 2
α	Vetor com hiperparâmetros da distribuição a priori
$\mathbf{A}$	Matriz e que a diagonal é formada pelo vetor α .
β	Parâmetro de compromisso entre esparsidade e erro no <i>algoritmo Sparse Bayesian Learning</i> .
$\mathbf{m}$	Média da distribuição a posteriori
$\mathbf{T}$	Matriz de saída em sistemas lineares de múltiplas respostas
$\mathbf{W}$	Matriz de parâmetros do sistema linear de múltiplas respostas
$\mathbf{E}$	Matriz de ruído
$\mathbf{x}_{i\cdot}$	i -ésima linha da matriz $\mathbf{X}$.
$\mathbf{x}_{\cdot j}$	j -ésima coluna da matriz $\mathbf{X}$.
$d(\mathbf{W})$	Medida de esparsidade da matriz de pesos.
$\mathbf{w}_{gen}$	vetor de pesos de geração.
$\mathbf{W}_{gen}$	Matriz de pesos de geração.
$\ \cdot\ _{\mathcal{F}}$	Norma de Frobenius.
$\mathbf{I}$	Matriz identidade
$\mathbf{D}_{\mathcal{X}}$	Matriz de distâncias de entrada
$\mathbf{B}$	Matriz de pesos do sistema linear de múltiplas respostas no modelo Máquina de Aprendizagem Mínima.
$tr(\cdot)$	Função traço
$\mathcal{D}$	Conjunto de amostras de treinamento
$\mathcal{D}_{ref}$	Conjunto de pontos de referência

$\boldsymbol{x}^*$	Padrão de teste
$\boldsymbol{y}^*$	Saída correspondente à um padrão de teste
$\boldsymbol{d}_x^*$	Distâncias de entrada do padrão de teste
$\boldsymbol{d}_y^*$	Distâncias de de saída do padrão de teste

Sumário

1	INTRODUÇÃO	31
1.1	Objetivo Geral e Específicos	34
1.2	Produção científica	34
1.3	Estrutura da Dissertação	35
2	FUNDAMENTAÇÃO TEÓRICA	37
2.1	Conceitos iniciais	37
2.1.1	Função densidade de probabilidade	39
2.2	Distribuições de probabilidade mais comuns	40
2.2.1	Distribuição Normal	40
2.2.2	Distribuição de Bernoulli	42
2.2.3	Distribuição Laplaciana	43
2.2.4	Distribuição Gamma	43
2.2.5	Distribuição Beta	45
2.3	Teoria da decisão de Bayes	45
2.3.1	Funções discriminantes	48
2.4	Estimação da função de densidade de probabilidade e seus parâmetros	48
2.4.1	Estimação de parâmetros por máxima verossimilhança	49
2.4.2	Estimação de parâmetros por máxima posteriori	50
2.5	Conclusão do capítulo	51
3	MODELOS LINEARES	53
3.1	Introdução	53
3.1.1	Funções de base	54
3.1.1.1	Função Gaussiana	55
3.1.1.2	Sigmoide Logística	55
3.1.1.3	Tangente Hiperbólica	56
3.1.2	Estimação de parâmetros	56
3.2	Definição do problema da esparsidade	57
3.2.1	Métodos para solução de sistemas unirespostas	59
3.3	Sparse bayesian Learning	61
3.3.1	Estimação dos hiperparâmetros	63
3.3.2	Resumo do algoritmo para SBL	64
3.4	Modelos lineares de múltiplas respostas	64
3.5	Esparsidade em modelos lineares de múltiplas respostas	66
3.5.1	Métodos para solução de sistemas lineares de múltiplas respostas	67

3.6	Sparse Bayesian Learning para múltiplas respostas	68
3.6.1	Estimação dos hiperparâmetros	70
3.6.2	Sumário do algoritmo	71
3.7	Conclusão do capítulo	72
4	MÁQUINA DE APRENDIZAGEM MÍNIMA BAYESIANA ESPARSA	73
4.1	Máquinas de Aprendizagem Mínima	73
4.1.1	Formulação básica	73
4.1.1.1	Regressão de distâncias	73
4.1.1.2	Estimação da saída	74
4.1.1.3	Resumo para o algoritmo do MLM	76
4.1.1.4	Implicações acerca do treinamento do MLM	77
4.2	Máquina de Aprendizagem Mínima Bayesiana Esparsa (Proposta)	78
4.2.1	M-SBL aplicado ao modelo MLM	78
4.2.2	Estimação dos hiperparâmetros para o SBMLM	80
4.3	Conclusão do capítulo	81
5	SIMULAÇÕES COMPUTACIONAIS	83
5.1	Análise empírica da superfície de decisão	83
5.1.1	Configuração dos experimentos	83
5.1.2	Resultados e discussões	84
5.2	Comparativo do SBMLM com outros métodos da literatura	86
5.2.1	Configuração dos experimentos	86
5.2.2	Resultados e discussões	87
5.3	Comparativo do SBMLM com outras versões do MLM	88
5.3.1	Configuração dos experimentos	88
5.3.2	Resultados e discussões	89
5.4	Análise da variação dos parâmetros do SBMLM	91
5.4.1	Configuração dos experimentos	91
5.4.2	Resultados e discussões	91
5.5	Conclusão do capítulo	92
6	CONCLUSÃO E TRABALHOS FUTUROS	95
6.1	Conclusão	95
6.2	Contribuições desse trabalho	96
6.3	Trabalhos Futuros	96
	REFERÊNCIAS	97

APÊNDICES	103
APÊNDICE A – MINIMIZAÇÃO DA PROBABILIDADE DE ERRO	105
APÊNDICE B – MAPAS OPOSTOS PARA REDUÇÃO DE CON- JUNTOS EM MÁQUINAS DE VETORES RELE- VANTES	107
B.1 Mapas Opostos Gaussianos	107
B.2 GOM aplicado ao treinamento do modelo RVM	108
B.3 Simulações e Discussões	109
B.4 Conclusão	112
APÊNDICE C – MISTURA DE GAUSSIANAS	113
C.0.1 Modelos Gerais de Misturas	113
C.0.2 Modelos de misturas de Gaussianas	114
C.0.3 O Algoritmo EM para mistura de Gaussianas	114
APÊNDICE D – MÁQUINAS DE VETORES RELEVANTES RE- DUZIDAS POR CENTROS DE MISTURAS . . .	117
D.1 Centros de Misturas de Gaussianas para Máquinas de Vetores Re- levantes	117
D.2 Resultados e Discussões	117
D.3 Conclusão	120

1 Introdução

A máquina de aprendizagem mínima (*Minimal Learning Machine*, MLM) (SOUZA et al., 2015) é um método de aprendizagem supervisionado. Esse modelo tem seu treinamento realizado com base no mapeamento entre distâncias no espaço de entrada e de saída. Esse modelo foi aplicado recentemente em diversos problemas como detecção de falha de motores (COELHO et al., 2014), cinemática inversa de robôs (MELO, 2015) e localização de robôs (MARINHO et al., 2017).

Uma das etapas do processo de treinamento do modelo MLM é a detecção de pontos de referência. 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 (características) e de saída (rótulos). As matrizes que armazenam as distâncias de cada padrão de treinamento para todos os pontos de referência são chamadas de matrizes de distâncias de entrada e de saída. O processo de treinamento consiste em estimar um mapeamento entre essas duas matrizes de distâncias. Isso é realizado através da resolução de um sistema linear de múltiplas respostas.

Durante a fase de predição, todos os pontos de referência selecionados no treinamento são utilizados para descrever novos padrões no espaço de distâncias de entrada. Isso é feito através do cálculo da distância do novo padrão a todos os pontos de referência, mas levando em consideração somente as características. As distâncias de saída de um novo padrão são estimadas através do mapeamento calculado durante o treinamento. Na etapa de predição, é aplicado um processo de otimização para a estimação do valor de saída do padrão. Nesse processo são utilizadas as distâncias estimadas no espaço de saída do novo padrão e as distâncias de saída dos pontos de referência.

A fase de seleção de pontos de referência e o método utilizado para resolver o sistema linear de múltiplas respostas, são fatores importantes para a geração de um bom modelo de aprendizagem. Esses fatores impactam diretamente na medida de esparsidade do modelo, complexidade e tempo necessário para a predição.

A tarefa de seleção de pontos de referência deve satisfazer dois critérios: (i) definir a quantidade de pontos de referência e (ii) identificar quais são os pontos de referência corretos para resolução do problema. Foram propostos diversos métodos a fim de identificar esses pontos durante o processo de treinamento, como a aplicação de algoritmos genéticos (DIAS et al., 2017), e remoção de protótipos da região de sobreposição (FLORÊNCIO; DIAS; ROCHA NETO, 2017). Entretanto, essas abordagens ainda tratam o problema de estimação dos pontos de referência através de suposições empíricas sobre suas localizações. Além disso, esses métodos também tratam essa seleção como uma tarefa isolada da estimação

dos parâmetros do sistema linear.

Se todos os padrões de treinamento forem considerados pontos de referência, as matrizes de distância serão quadradas e cada coluna representa a distância de um ponto (ou padrão) de treinamento à todos os pontos de referência. Sabendo disso, qualquer método que selecione pontos de referência, remove colunas das matrizes de distância. Em Wipf (2006), são definidos critérios para soluções esparsas em sistemas lineares de múltiplas respostas na forma

$$\mathbf{T} = \mathbf{\Phi}\mathbf{W} + \mathbf{E}, \quad (1.1)$$

em que $\mathbf{\Phi}$ é a matriz de coeficientes e $\mathbf{W}$ a matriz de pesos ou parâmetros do sistema. Uma característica desse tipo de solução é que a maioria das linhas da matriz de pesos são nulas. Por outro lado, uma linha nula na matriz de pesos, equivale a remover uma coluna da matriz de coeficientes. Portanto, a tarefa de seleção de pontos de referência é uma tentativa de impor esparsidade à solução do sistema linear. A abordagem adotada pelo MLM de selecionar pontos de referência antes da resolução do sistema é interessante, já que a solução por mínimos quadrados, utilizada na formulação padrão do modelo, não gera soluções esparsas. Na formulação padrão do modelo MLM, a seleção de pontos de referência é feita de maneira aleatória, sendo a quantidade de pontos um parâmetro do modelo que deve ser otimizado durante o processo de treinamento. O problema dessa abordagem de seleção é que nenhuma medida, como a minimização da norma ℓ_0 e o erro (WIPF, 2006), são levados em consideração durante a escolha dos pontos de referência.

Na etapa de predição, o processo de otimização realizado para estimação dos valores de saída consiste na aplicação de um algoritmo de multilateração (BACHRACH; TAYLOR, 2005). O custo computacional desse método cresce com base no número de pontos de referência utilizados durante a estimação. Caso o algoritmo de seleção de pontos de referência alcance uma redução elevada do número de pontos, sem afetar a qualidade do modelo, a etapa de predição terá uma redução de tempo considerável.

Alguns métodos de resolução de sistemas lineares, como mínimos quadrados sem regularização, baseiam-se somente na minimização do erro, ou seja, não adotam um critério para controle de complexidade dos modelos gerados (MACKAY, 1992a). Uma forma de limitar a complexidade das soluções de sistemas lineares é através da adoção de um perfil de esparsidade (MACKAY, 1992b). Infelizmente, o problema de resolver um sistema linear de forma esparsa é NP-Difícil (WIPF; RAO, 2007), logo métodos de aproximação podem ser aplicados. Existem várias abordagens para se obter soluções esparsas em sistemas lineares de resposta única e múltiplas respostas. Dentre elas estão as estratégias gulosas como *Orthogonal Matching Pursuit* (TROPP, 2004). Por outro lado, existem estratégias Bayesianas como *basis pursuit* (CHEN; DONOHO; SAUNDERS, 2001) e o método *Focal Underdetermined System Solver* (RAO et al., 2003) que pode ser aplicado em sistemas indeterminados.

Uma alternativa Bayesiana para soluções esparsas em sistemas lineares é o método de Aprendizagem Bayesiana Esparsa (*Sparse Bayesian Learning*, SBL) (MACKAY, 1992b). O método SBL é uma alternativa aos modelos de máxima a posteriori com distribuições a priori fixas. A vantagem da adoção de estratégias Bayesianas para resolução de sistemas lineares é que esses métodos incorporam uma medida de complexidade chamada de navalha de Occam (MACKAY, 1992b).¹ Além disso, essa estratégia incorpora um processo de otimização que obtém os valores dos hiperparâmetros da distribuição a priori sobre cada parâmetro do sistema linear.

Inicialmente, o SBL operava somente sobre sistemas lineares de resposta única, sendo inclusive utilizado para treinamento de redes neurais artificiais (MACKAY, 1992a), bem como para o treinamento de Máquinas de Vetores Relevantes (*Relevance Vector Machines*, RVM) (TIPPING, 2001). No trabalho de Tipping e Faul (2003) foi elaborada uma metodologia incremental, em que colunas são adicionadas à solução do método do sistema a cada iteração do algoritmo. Essa abordagem conseguiu reduzir o tempo de treinamento do modelo RVM de maneira considerável. Em Wipf e Rao (2007) o método SBL foi estendido para operar sobre sistemas lineares de múltiplas respostas. Essa abordagem foi chamada de M-SBL.

O modelo RVM, da mesma forma que uma máquina de vetores suporte (*Support Vector Machine*, SVM), descreve sua superfície de decisão em função de um conjunto de padrões de treinamento. Embora a predição do SVM e RVM sejam idênticas, o modelo RVM tem uma formulação bayesiana enquanto o SVM parte de uma formulação geométrica. A diferença principal entre os dois modelos está na esparsidade, pois o RVM descreve sua superfície de decisão utilizando uma quantidade de padrões de treinamento, em geral, bem inferior a do modelo SVM. Isso é mais um atrativo para soluções de sistemas lineares através do método SBL. Essa característica de descrever a superfície de decisão em função de um conjunto de padrões de treinamento também é encontrada no modelo MLM.

O custo computacional do método SBL cresce com a quantidade de padrões de treinamento que compõe a base de dados. A fim de limitar o número de padrões de treinamento e, conseqüentemente, o crescimento do custo do algoritmo SBL, foram propostos dois trabalhos paralelos durante o desenvolvimento do método dessa dissertação. O primeiro consiste de uma versão probabilística do método *Opposite Maps* aplicado ao treinamento do RVM (SOUSA; ROCHA NETO, 2017). Esse método, chamado de mapas opostos Gaussianos para máquinas de vetores relevantes (*Gaussian Opposite Maps Relevance Vector Machines*, GOM-RVM), é apresentado no Apêndice B. O segundo trabalho, apresentado no Apêndice D, utiliza os centros das componentes de uma mistura de Gaussianas como padrões de treinamento para o RVM (SOUSA; DIAS; ROCHA NETO,

¹ A Navalha de Occam é o princípio que determina que modelos menos complexos devem ser utilizados para composição de soluções.

2017).

Observando a carência em termos de identificação de pontos de referência e controle de complexidade das soluções no modelo MLM, a proposta desta dissertação consiste em um método de treinamento para o modelo, com base no algoritmo M-SBL. Nessa abordagem, chamada de SBMLM (*Sparse Bayesian Minimal Learning Machine*), todos os padrões de treinamento serão considerados inicialmente como pontos de referência. O método M-SBL será responsável por estimar a matriz de pesos e identificar quais são os pontos de referência mais relevantes para a solução do MLM. Com a aplicação desse algoritmo, espera-se obter soluções esparsas e com baixa complexidade.

1.1 Objetivo Geral e Específicos

Este trabalho tem como objetivo geral a elaboração de uma abordagem de treinamento para as Máquinas de Aprendizagem Mínimas através do método M-SBL. Tal processo de treinamento deve resolver o problema da seleção de pontos de referência e obter soluções com complexidade controlada para o sistema linear de múltiplas respostas.

De uma forma mais específica, este trabalho tem os seguintes objetivos:

- Elaborar uma revisão bibliográfica acerca de métodos de resolução de sistemas lineares de resposta única.
- Realizar um estudo acerca do algoritmo SBL para resolução de sistemas lineares de resposta única.
- Expandir a revisão bibliográfica para métodos de resolução de sistemas lineares de múltiplas respostas.
- Realizar um estudo sobre a extensão do algoritmo SBL para soluções de sistemas lineares de múltiplas respostas.
- Aplicar a extensão do SBL na resolução do sistema linear de múltiplas respostas do treinamento do modelo MLM.
- Analisar e comparar os resultados do método proposto em termos de esparsidade, acurácia, complexidade e tempo de treinamento com outros classificadores da literatura e com as próprias Máquinas de Aprendizagem Mínima.

1.2 Produção científica

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

- **SOUSA, L. S.;** DIAS, M. L. D.; ROCHA NETO, A. R. Máquinas de vetores-suporte de mínimos quadrados esparsas via recozimento simulado. In: Simpósio brasileiro de automação inteligente (SBAI). Porto Alegre, Brasil: [s.n.], 2015.

- **SOUSA, L. S.**; ROCHA NETO, A. R. Gaussian opposite maps for reduced-set relevance vector machines. In: Advances in Computational Intelligence: 14th International Work-Conference on Artificial Neural Networks, IWANN 2017, Cadiz, Spain, June 14-16, 2017, Proceedings, Part I. Cham: Springer International Publishing, 2017. p. 458-468. ISBN 978-3-319-59153-7. Disponível em: <https://doi.org/10.1007/978-3-319-59153-7_40>.
- **SOUSA, L. S.**; DIAS, M. L. D.; ROCHA NETO, A. R. Reduced-set relevance vector machines via gaussian mixture models. In: 4th IEEE Latin American Conference on Computational Intelligence LA-CCI. Guadalajara, México: [s.n.], 2017.
- **SOUSA, L. S.**; ROCHA NETO, A. R. Sparse bayesian minimal learning machine. International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brasil, 2018.(submetido)
- DIAS, M. L. D.; **SOUSA, L. S.**; ROCHA NETO, Ajalmar R; Freire, Ananda L. Fixed-size extreme learning machines through simulated annealing. Neural Processing Letters, Sep 2017. ISSN 1573-773X. Disponível em: <<https://doi.org/10.1007/s11063-017-9700-9>>.

1.3 Estrutura da Dissertação

O restante dessa dissertação está dividido em quatro capítulos: Fundamentação Teórica, Modelos lineares, Sparse Bayesian Minimal Learning Machine (Proposta), Simulações Computacionais e Conclusão e Trabalhos Futuros. Todos os conceitos são detalhados e sempre associados à área de aprendizagem de máquinas.

O Capítulo 2 inicia com uma apresentação dos principais conceitos acerca da teoria da probabilidade. Em seguida, é explicado o conceito de distribuição de probabilidade e são apresentadas as principais distribuições, com exemplos. O capítulo encerra com a apresentação da teoria da decisão de Bayes e as abordagens para estimação dos parâmetros das distribuições de probabilidade.

No Capítulo 3 são apresentados, inicialmente, os conceitos de regressão linear e como esses modelos podem ser expandidos para descrever funções não-lineares. A seguir, são apresentadas algumas funções de bases e uma discussão a respeito da estimação dos parâmetros de um modelo de regressão linear. Além disso, é formalizado o problema de resolução de sistemas lineares de forma esparsa e uma revisão bibliográfica de métodos que podem obter tais soluções. No fim da primeira parte do capítulo, a versão do algoritmo SBL para sistemas lineares de resposta única é explicado. A segunda parte explica todos os conceitos apresentados na primeira parte, mas expandindo os conceitos para sistemas lineares de múltiplas respostas. O Capítulo finaliza com a explicação de uma extensão do

algoritmo SBL para sistemas lineares de múltiplas respostas.

No Capítulo 4, é apresentado o método proposto nesse trabalho. O capítulo inicia com uma explicação sobre as Máquinas de Aprendizagem Mínimas e uma elicitacão das limitações de seu método padrão de treinamento. Após isso, é apresentada a abordagem proposta para substituir o processo de treinamento do MLM.

Os experimentos e resultados obtidos com o método proposto são apresentados no Capítulo 5. Foram realizados quatro tipos de experimentos a fim de avaliar a capacidade do modelo em termos de acurácia, esparsidade, tempo de treinamento e complexidade. Cada uma das simulações é dividida em duas seções nas quais é explicada a metodologia utilizada no experimento e apresentada uma discussão dos resultados obtidos.

Por fim, o Capítulo 6 apresenta as conclusões e contribuições do método proposto. Além disso, são elicitados algumas deficiências do método proposto e algumas possíveis soluções para trabalhos futuros.

2 Fundamentação Teórica

Um ponto que é tratado em aprendizagem de máquina é a incerteza de informações. Essa operação é importante devido a ocorrência de dados ruidosos ou pela insuficiência de informações para indução de superfícies de decisão eficientes. A teoria da probabilidade agrupa um conjunto de ferramentas que são largamente aplicadas na representação e tratamento da incerteza de informações. Muitos fundamentos de probabilidade serão apresentados durante este capítulo. Na Seção 2.1 serão apresentadas as variáveis aleatórias e sua relação com probabilidade, enquanto na Seção 2.2 serão apresentadas algumas distribuições de probabilidade largamente utilizadas na derivação de modelos. Além disso, na Seção 2.3 será explicada a Teoria de decisão de Bayes. Por fim, métodos de estimação de funções de densidade de probabilidade e seus devidos parâmetros, serão apresentados na Seção 2.4. Todos os fundamentos de probabilidade apresentados no presente capítulo foram baseados em Bishop (2006), Theodoridis e Koutroumbas (2008) e Murphy (2012).

2.1 Conceitos iniciais

Quando se fala em probabilidade, se pensa na chance ou possibilidade de algum evento ocorrer. Em aprendizagem de máquina, existe uma relação direta com probabilidade quando se pensa na possibilidade de um padrão ser igual a um determinado valor ou na chance de um padrão pertencer a uma determinada classe. Ao analisar dois eventos, é possível observar a probabilidade dos dois eventos ocorrerem simultaneamente ou a probabilidade de um evento ocorrer dado que um outro ocorreu. Para iniciar o entendimento formal, é possível definir a probabilidade de um evento ocorrer como sendo uma fração entre a quantidade de vezes que um evento ocorreu pelo número de tentativas. conforme a equação a seguir:

$$p(X = x_k) = \frac{n_k}{N}, \quad (2.1)$$

na qual $p(X = x_k)$ representa a probabilidade do evento $X = x_k$ ocorrer e n_k é a quantidade de ocorrências, dentre as N tentativas, que o evento $X = x_k$ ocorreu. Com base na Equação (2.1), se uma moeda é jogada 10 vezes ao acaso e em 5 dessas jogadas ela caiu na posição “cara”, a probabilidade do evento “cara” ocorrer é de 0,5. Pode-se notar que, por definição, a probabilidade de eventos é definida no intervalo $[0, 1]$.

Também pode ser definida a probabilidade em relação a duas variáveis aleatórias. Para isso foram realizadas N observações ao acaso dos valores das variáveis aleatórias X e Y . O quadro a seguir representa a quantidade de ocorrências de cada par de eventos

(x_i, y_j) dentre as N tentativas.

$$\begin{array}{cccccc}
 & & & x_i & & \\
 & \bullet & \bullet & \bullet & \bullet & \bullet \\
 y_j & \bullet & \bullet & n_{ij} & \bullet & \} r_j. \\
 & \bullet & \bullet & \bullet & \bullet & \\
 & & & \underbrace{}_{\mathcal{S}} & &
 \end{array} \quad (2.2)$$

A primeira forma de analisar a ocorrência de dois eventos é pela probabilidade conjunta, que se refere a dois eventos ocorrerem simultaneamente com as duas variáveis aleatórias. A probabilidade conjunta é definida pela equação a seguir:

$$p(X = x_i, Y = y_j) = \frac{n_{i,j}}{N}, \quad (2.3)$$

na qual $p(X = x_i, Y = y_j)$ representa a probabilidade dos eventos $X = x_i$ e $Y = y_j$ ocorrerem simultaneamente, $n_{i,j}$ é a quantidade de tentativas em que os dois eventos ocorreram dentre as N tentativas ao acaso. Também é possível calcular a probabilidade de X receber o valor x_i independentemente dos valores de Y , a saber

$$p(X = x_i) = \frac{c_i}{N}, \quad (2.4)$$

onde $c_i = \sum_j n_{i,j}$. Na Equação (2.4) o cálculo da probabilidade $p(X = x_i)$ necessita do valor do somatório de todos os elementos da coluna em que $X = x_i$, dessa forma todas as possibilidades de valores de y_i são levadas em consideração. A partir das equações (2.3) e (2.4), é possível escrever

$$p(X = x_i) = \sum_j p(X = x_i, Y = y_j), \quad (2.5)$$

que é conhecida como regra da soma. O valor $p(X = x_i)$ também é conhecido como probabilidade marginal. O processo de marginalização calcula o valor de probabilidade através da soma de outras variáveis.

O segundo método de análise é observar na Equação (2.2) somente os valores de Y para os quais o evento $X = x_i$ ocorreu. Os valores possíveis para Y podem ser descritos por $p(Y = y_j | X = x_i)$. Isso é chamado de probabilidade condicional de $Y = y_j$ dado $X = x_i$ e pode ser obtido por

$$p(Y = y_j | X = x_i) = \frac{n_{i,j}}{c_i}. \quad (2.6)$$

A partir das Equações (2.3) e (2.4), é possível escrever

$$p(X = x_i, Y = y_j) = \frac{n_{i,j}}{N} = \frac{n_{i,j}}{c_i} \cdot \frac{c_i}{N} = p(Y = y_j | X = x_i) p(X = x_i). \quad (2.7)$$

Isso é conhecido como regra do produto de probabilidades. Para buscar uma notação mais simples, denota-se a probabilidade de uma variável aleatória simplesmente por

$p(X)$. A partir dessa notação, é possível descrever a regra da soma e a regra do produto, respectivamente, pelas equações a seguir:

$$p(X) = \sum_Y p(X, Y), \quad (2.8)$$

$$p(X, Y) = p(Y|X) p(X). \quad (2.9)$$

Utilizando a regra do produto e a propriedade da simetria $p(X, Y) = p(Y, X)$, obtem-se o relacionamento entre as probabilidades condicionais, a saber

$$p(Y|X) = \frac{p(X|Y) p(Y)}{p(X)}. \quad (2.10)$$

Essa relação é conhecida por Teorema de Bayes, que é uma regra central em reconhecimento de padrões e aprendizagem de máquinas. O termo $p(Y|X)$ é chamado de probabilidade *a posteriori*, $p(X|Y)$ é a verossimilhança, $p(Y)$ é a probabilidade *a priori* e o denominador $p(X)$ é chamado de evidência ou probabilidade marginal. Pode-se reescrever a evidência em função da regra da soma, denotada pela Equação (2.8), e da regra do produto, descrita através da Equação (2.9). Assim,

$$p(Y|X) = \frac{p(X|Y) p(Y)}{\sum_Y p(X|Y) p(Y)} \quad (2.11)$$

A evidência pode ser entendida como um termo de normalização. Isso é necessário para fazer com que a soma da posteriori em relação a todos os valores de Y seja igual a 1.

2.1.1 Função densidade de probabilidade

Além de eventos discretos, também é possível calcular a probabilidade em relação à variáveis contínuas. Se a probabilidade de uma variável real x cair em um intervalo $(x, x + \delta x)$ é dada por $p(x)\delta x$ para $\delta x \rightarrow 0$, então $p(x)$ é chamado de densidade de probabilidade sobre x . A probabilidade de x ocorrer em um intervalo (a, b) é então dada por

$$p(x \in (a, b)) = \int_a^b p(x) dx. \quad (2.12)$$

Pelo fato de que probabilidades possuem valores não-negativos e o valor de x ser um número real, uma função de densidade de probabilidade deve satisfazer duas condições:

$$p(x) \geq 0, \quad (2.13)$$

$$\int_{-\infty}^{+\infty} p(x) dx = 1. \quad (2.14)$$

Considerando duas variáveis reais x e y , as regras da soma e do produto podem ser reescritas como:

$$p(x) = \int p(x, y) dy, \quad (2.15)$$

$$p(x, y) = p(y|x) p(x). \quad (2.16)$$

2.2 Distribuições de probabilidade mais comuns

Na área de aprendizagem de máquina os classificadores baseados na Teoria de Bayes utilizam informações da disposição dos dados de treinamento no espaço para induzir suas superfícies de decisão. A localização de um padrão da base de dados é tratado como um evento dependente do acaso, ou seja, é uma variável aleatória. O comportamento de tais variáveis são descritos por distribuições de probabilidade que podem ser classificadas em contínuas ou discretas com base na natureza dos valores. O conceito formal de distribuição de probabilidade pode ser visto na SubSeção 2.1.1 e mais detalhadamente em Adams e Guillemin (1996). Nas próximas subseções, são apresentadas algumas das distribuições de probabilidade mais conhecidas.

2.2.1 Distribuição Normal

A distribuição normal, também conhecida como distribuição Gaussiana, é um modelo que descreve variáveis contínuas. Com essa distribuição, uma única variável $x \in \mathbb{R}$ pode ser escrita na forma

$$\mathcal{N}(x|\mu, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp \left\{ -\frac{1}{2\sigma^2}(x - \mu)^2 \right\}, \quad (2.17)$$

em que μ é a média e σ^2 é a variância. Na Figura 1, são apresentados dois exemplos do comportamento da distribuição descrita na Equação 2.17.

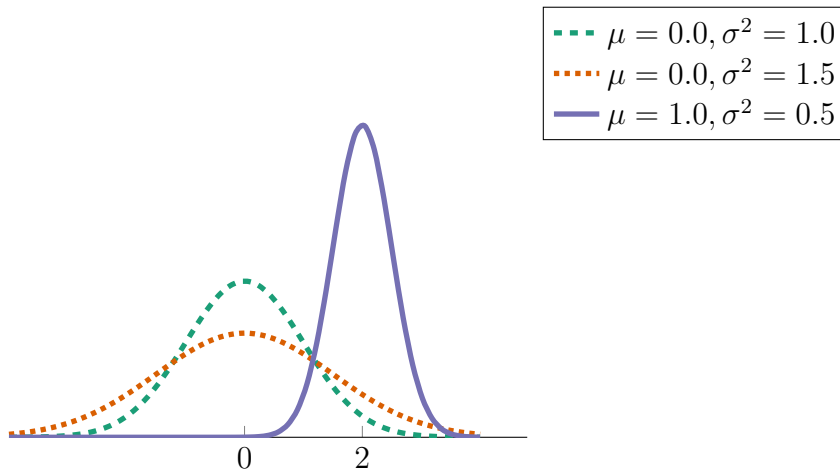


Figura 1 – Exemplos de distribuições normais univariadas.

Fonte – Próprio autor.

Quando $\mathbf{x} \in \mathbb{R}^D$, o ponto pode ser descrito através da forma multivariada da distribuição normal, a saber

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = (2\pi)^{-D/2} |\boldsymbol{\Sigma}|^{-1/2} \exp \left\{ -\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}) \right\}. \quad (2.18)$$

Note que $\boldsymbol{\mu} = E[\mathbf{x}]$ é um vetor pertencente ao $\mathbb{R}^D$ que representa a média, $\boldsymbol{\Sigma}$ é a matriz de covariância $D \times D$ definida por

$$\boldsymbol{\Sigma} = E[(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^\top] \quad (2.19)$$

e $|\boldsymbol{\Sigma}|$ é o determinante da matriz $\boldsymbol{\Sigma}$. É fácil notar que para o caso em que $D = 1$, a distribuição multivariada coincide com a univariada. Uma distribuição normal com média $\boldsymbol{\mu}$ e covariância $\boldsymbol{\Sigma}$ é denotada por $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

A distribuição normal possui uma dimensão a mais em relação à sua entrada. Em outras palavras, quando a distribuição é estimada a partir de dados pertencentes ao $\mathbb{R}^2$, ela pertencerá ao $\mathbb{R}^3$. Sabendo disso, para que alguns casos sejam analisados, serão considerados pontos no $\mathbb{R}^2$ por questões de visualização. Seja $\boldsymbol{\Sigma}$ dado por

$$\boldsymbol{\Sigma} = E \left[\begin{bmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \end{bmatrix} \begin{bmatrix} x_1 - \mu_1, x_2 - \mu_2 \end{bmatrix} \right] = \begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{bmatrix}, \quad (2.20)$$

onde $E[x_i] = \mu_i, i = 1, 2$ e por definição $\sigma_{12} = E[(x_1 - \mu_1)(x_2 - \mu_2)]$, que é conhecido como covariância entre as variáveis aleatórias x_1 e x_2 . Isso é uma medida de correlação estatística mútua. Se as variáveis são estatisticamente independentes, a covariância entre tais variáveis é zero. Obviamente, os elementos da diagonal de $\boldsymbol{\Sigma}$ são as variâncias dos respectivos elementos do vetor aleatório.

Os valores da matriz de covariância influenciam diretamente na disposição da distribuição no espaço. É possível analisar a geometria da distribuição normal considerando quatro situações, apresentadas na Figura 2. Na primeira, Figura 2a, acontece a igualdade $\sigma_1^2 = \sigma_2^2$, logo a distribuição gerada é totalmente simétrica. Quando $\sigma_1^2 \gg \sigma_2^2$, a distribuição resultante é alongada na direção do eixo x_1 , como mostrado através da Figura 2b. A normal é alongada na direção do eixo x_2 quando $\sigma_1^2 \ll \sigma_2^2$, como pode ser visto na Figura 2c. No ultimo caso, Figura 2d, a matriz de covariância é não-diagonal, por isso a distribuição pode assumir diversas formas e orientações no espaço.

A maior razão da popularidade da distribuição Normal é devido sua tratabilidade computacional e o fato do modelo se adequar a um grande número de casos (THEODORIDIS; KOUTROUMBAS, 2008). Para uma única variável, a distribuição que maximiza a entropia é a normal. Além disso, quando considera-se a soma de múltiplas variáveis aleatórias, também tem-se uma distribuição normal. O segundo caso, é justificado pelo *teorema do limite central*. Esse teorema refere-se ao fato de que sujeita à certas restrições, a soma de um conjunto de variáveis aleatórias, que também é uma variável aleatória, tem uma distribuição que tende à uma normal com o aumento do número de termos da soma (BARRON, 1986).

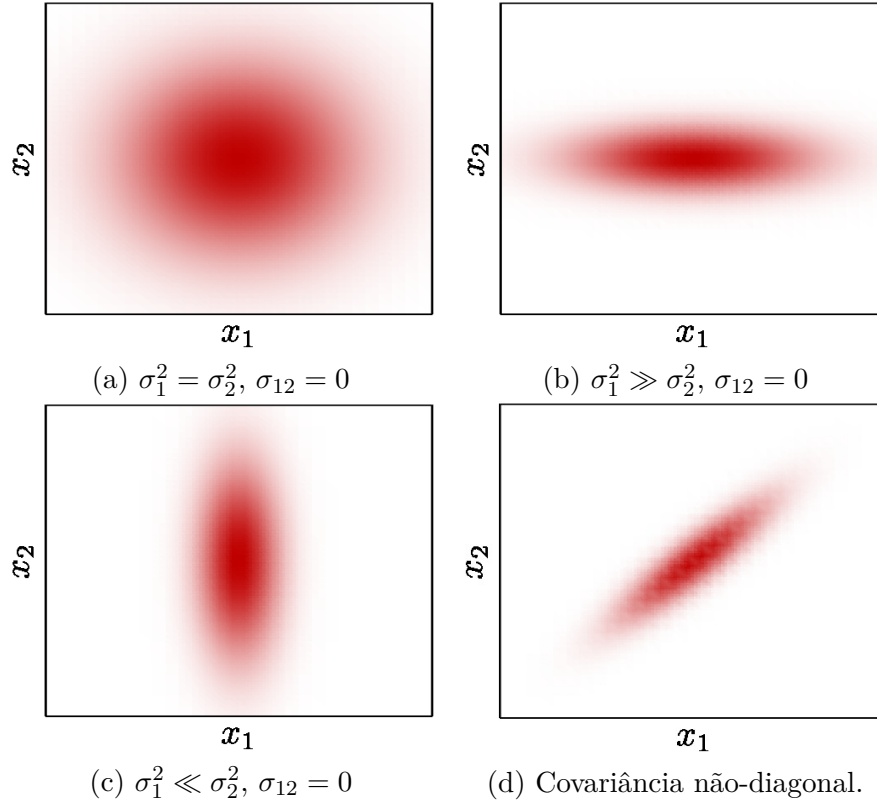


Figura 2 – Exemplos de distribuições normal bivariadas. São apresentadas diversas formas da distribuição com base na disposição da matriz de covariância. Nas Figs. 2a, 2b e 2c são apresentados os casos em que a matriz de covariância é diagonal. No caso ilustrado na Fig. 2d a distribuição normal possui uma matriz de covariância não-diagonal.

Fonte – Próprio autor.

2.2.2 Distribuição de Bernoulli

A distribuição de Bernoulli trata de variáveis aleatórias binárias, que por sua vez também são variáveis discretas. Por isso, considerando uma variável $x \in \{0, 1\}$, a probabilidade de $x = 1$ é denotada pelo parâmetro μ tal que

$$p(x = 1|\mu) = \mu, \quad (2.21)$$

onde $0 \leq \mu \leq 1$. Por outro lado, a probabilidade de $x = 0$ é dada por $p(x = 0|\mu) = 1 - \mu$. A distribuição de probabilidade sobre a variável x pode ser escrita na forma

$$\text{Bern}(x|\mu) = \mu^x(1 - \mu)^{1-x}. \quad (2.22)$$

É fácil observar que essa distribuição é normalizada. Além disso, possui média e covariância dada por

$$E[x] = \mu \quad (2.23)$$

$$\text{Var}[x] = \mu(1 - \mu). \quad (2.24)$$

2.2.3 Distribuição Laplaciana

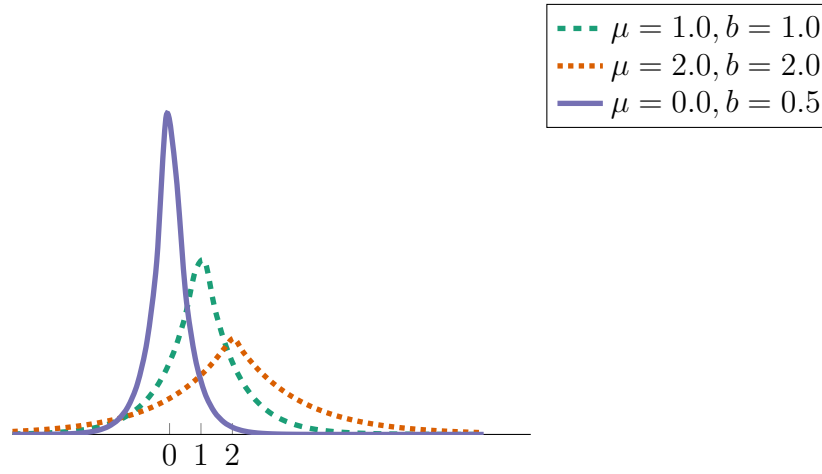


Figura 3 – Exemplos de distribuições Laplacianas.

Fonte – Próprio autor.

A distribuição Laplaciana, também conhecida como exponencial de duplos lados, é uma distribuição de caldas pesadas dada pela seguinte equação, a saber:

$$\text{Lap}(x|\mu, b) \triangleq \frac{1}{2b} \exp\left(-\frac{|x - \mu|}{b}\right). \quad (2.25)$$

Nessa distribuição, μ é um parâmetro de localização e $b > 0$ é um parâmetro de escala. Essa distribuição possui as seguintes propriedades:

- Média: μ ,
- Moda: μ ,
- Variância: $2b^2$.

Exemplos para tal distribuição podem ser vistas na Figura 3. Essa distribuição é robusta quanto à ocorrência de dados discrepantes (*outliers*). Além disso, sua propriedade de ter os maiores valores de probabilidade no ponto médio, favorece a esparsidade em modelos de aprendizagem quando centrada em zero (MURPHY, 2012).

2.2.4 Distribuição Gamma

A distribuição gamma é flexível e definida para valores reais positivos $x > 0$. Essa distribuição depende de dois parâmetros, conhecidos como forma ($a > 0$) e taxa ($b > 0$), e é dada pela equação:

$$\text{Gamma}(a, b) \triangleq \frac{b^a}{\Gamma(a)} x^{a-1} \exp(-xb), \quad (2.26)$$

em que $\Gamma(a)$ é a função gamma:

$$\Gamma(x) \triangleq \int_0^\infty u^{x-1} \exp(-u) du. \quad (2.27)$$

A distribuição Gamma possui as seguintes propriedades:

- Média: a/b ,
- Moda: $(a - 1)/b$,
- Variância: a/b^2 .

Na Figura 4, são apresentados alguns casos dessa distribuição. É possível observar que na medida em que o parâmetro a cresce, o ponto de máximo é reduzido. O parâmetro b modifica o espalhamento da distribuição na direção do eixo.

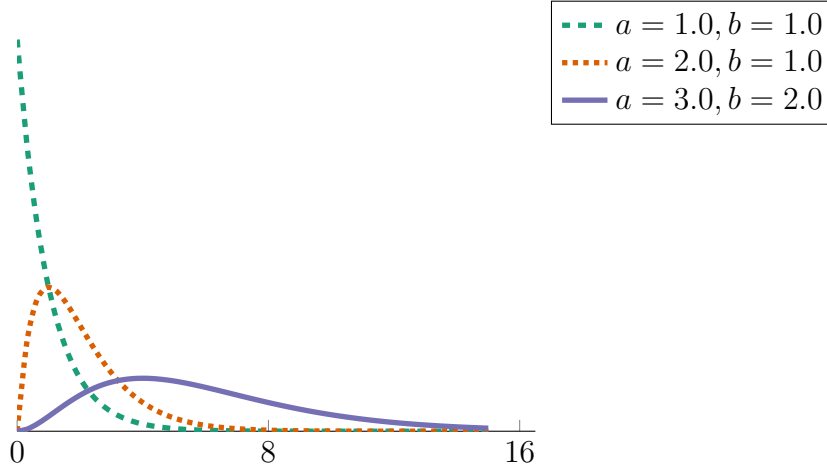


Figura 4 – Exemplos de distribuições Gamma.

Fonte – Próprio autor.

Existem algumas distribuições que são casos especiais da Gamma, a saber:

- **Distribuição Exponencial:** É definida por $\text{Exponencial}(\lambda) \triangleq \text{Gamma}(1, \lambda)$, onde λ é o parâmetro de taxa.
- **Distribuição de Erlang:** É um caso específico da distribuição Gamma, porém com a assumindo somente valores inteiros. Essa distribuição também possui uma formulação com um único parâmetro. Nesse caso, o valor de $a = 2$ é fixado, ou seja, $\text{Erlang}(\lambda) = \text{Gamma}(2, \lambda)$ onde λ é o parâmetro de taxa.
- **Distribuição Chi-quadrado:** Essa distribuição é definida como $\chi^2(v) \triangleq \text{Gamma}(\frac{v}{2}, \frac{1}{2})$. Essa distribuição é a soma dos quadrados de variáveis aleatórias Gaussianas. Mais precisamente, se $Z_i \sim \mathcal{N}(0, 1)$, e $S = \sum_{i=1}^v Z_i^2$, então $S \sim \chi_v^2$.

Se $X \sim \text{Gamma}(a, b)$, então pode ser mostrado que $\frac{1}{X} \sim \text{InvGamma}(a, b)$, onde IGam é a inversa da distribuição Gamma e é definida por:

$$\text{IGam}(a, b) \triangleq \frac{b^a}{\Gamma(a)} x^{-(a+1)} \exp(-b/x). \quad (2.28)$$

Essa distribuição possui as seguintes propriedades:

- Média: $b/(a - 1)$,
- Moda: $b/(a + 1)$,
- Variância: $b^2/((a - 1)^2(a - 2))$.

A média existe somente se $a > 1$ e a variância existe somente se $a > 2$.

2.2.5 Distribuição Beta

A distribuição beta tem suporte no intervalo $[0, 1]$ e é definida a seguir:

$$\text{Beta}(a, b) = \frac{1}{B(a, b)} x^{a-1} (1-x)^{b-1}, \quad (2.29)$$

em que $B(a, b)$ é a função beta, dada por:

$$B(a, b) \triangleq \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}. \quad (2.30)$$

Na Figura 5 podem ser vistos alguns exemplos dessa distribuição com diversos valores de parâmetros a e b .

Para garantir que a distribuição seja integrável, é necessário que $a, b > 0$. Quando $a = b = 1$, a distribuição obtida é uniforme. Se $a, b < 1$, a distribuição resultante é bimodal com pontos de máximo em 0 e 1. Quando $a, b > 1$, a distribuição é unimodal. Além disso, a distribuição possui as seguintes propriedades:

- Média: $a/(a+b)$,
- Moda: $(a-1)/(a+b-2)$,
- Variância: $ab/((a+b)^2(a+b+1))$.

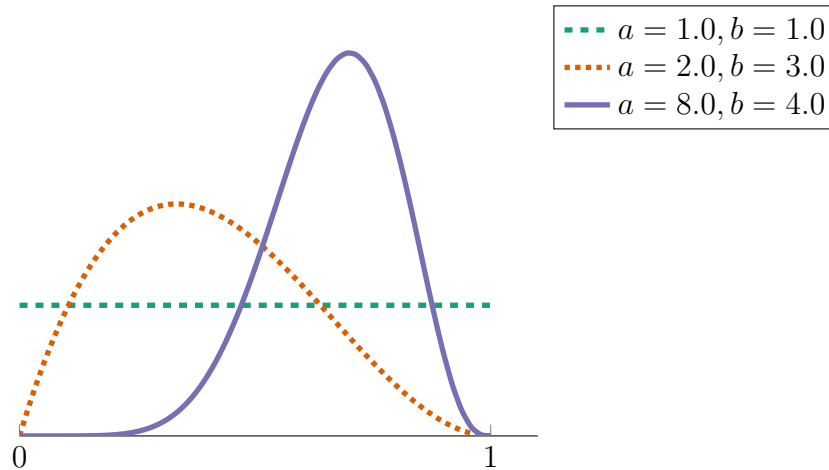


Figura 5 – Exemplos de distribuições beta.

Fonte – Próprio autor.

2.3 Teoria da decisão de Bayes

Uma das formas de definir um sistema de reconhecimento de padrões é utilizando os fundamentos probabilísticos. Para operação de tais sistemas é necessária a geração de

informações estatísticas, a partir de um conjunto de amostras divididas em classes, para estimar a probabilidade de um padrão desconhecido pertencer a uma determinada classe. Essa estimação é necessária devido a variações estatísticas nos dados como ruídos em sensores que capturam as amostras. Sabendo disso, a teoria de decisão de Bayes fornece um conjunto de recursos estatísticos que serão necessários para elaborar classificadores capazes de identificar a qual classe um padrão desconhecido pertence.

Considerando, inicialmente, um problema em que os padrões são divididos em duas classes, $\mathcal{C}_1$ e $\mathcal{C}_2$. Além disso, considerando também que as probabilidades *a priori*, $p(\mathcal{C}_1)$ e $p(\mathcal{C}_2)$, são conhecidas. Tal suposição é bem razoável uma vez que as probabilidades *a priori* podem ser calculadas a partir do conjunto de vetores de treinamento. Ou seja, se o conjunto de treinamento possui N padrões, onde N_1 é a quantidade associada à $\mathcal{C}_1$ e N_2 a quantidade associada à $\mathcal{C}_2$, $p(\mathcal{C}_1) \approx N_1/N$ e $p(\mathcal{C}_2) \approx N_2/N$.

Uma outra medida estatística, que supõe-se que é conhecida, é a função de densidade de probabilidade $p(\mathbf{x}|\mathcal{C}_i)$, $i = 1, 2$. Essa função descreve a distribuição dos vetores de características pertencentes à cada classe. Quando não é conhecida, pode ser estimada a partir dos dados de treinamento. A função de densidade de probabilidade $p(\mathbf{x}|\mathcal{C}_i)$ é referida como a verossimilhança de $\mathcal{C}_i$ com respeito a $\mathbf{x}$. Em casos discretos, $p(\mathbf{x}|\mathcal{C}_i)$ torna-se uma probabilidade.

De posse das informações de verossimilhança e probabilidade *a priori*, é possível calcular a probabilidade *a posteriori* através da *Regra de Bayes*. Essa regra é denotada por:

$$p(\mathcal{C}_i|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_i)p(\mathcal{C}_i)}{p(\mathbf{x})}, \quad (2.31)$$

onde $p(\mathbf{x})$ é chamada de evidência e é dada por:

$$p(\mathbf{x}) = \sum_{i=1}^2 p(\mathbf{x}|\mathcal{C}_i)p(\mathcal{C}_i). \quad (2.32)$$

A classificação através da regra de Bayes pode ser determinada pelas condições a seguir

$$\begin{aligned} \text{Se } p(\mathcal{C}_1|\mathbf{x}) &> p(\mathcal{C}_2|\mathbf{x}), \quad \mathbf{x} \text{ é classificado como } \mathcal{C}_1, \\ \text{Se } p(\mathcal{C}_1|\mathbf{x}) &< p(\mathcal{C}_2|\mathbf{x}), \quad \mathbf{x} \text{ é classificado como } \mathcal{C}_2. \end{aligned} \quad (2.33)$$

Quando ocorrer uma igualdade entre $p(\mathcal{C}_1|\mathbf{x})$ e $p(\mathcal{C}_2|\mathbf{x})$, pode ser atribuída qualquer uma das classes ao padrão $\mathbf{x}$. Observando a Equação (2.31), é possível notar que $p(\mathcal{C}_i|\mathbf{x})$ é diretamente proporcional ao produto entre a verossimilhança e a probabilidade *a priori*. Por isso, a regra de decisão pode ser baseada nas inequações a diante

$$\begin{aligned} p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1) &> p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2), \\ p(\mathbf{x}|\mathcal{C}_1)p(\mathcal{C}_1) &< p(\mathbf{x}|\mathcal{C}_2)p(\mathcal{C}_2). \end{aligned} \quad (2.34)$$

A evidência $p(\mathbf{x})$ não entra no cálculo, pois tem o mesmo valor para todas as classes e isso não afeta a decisão. Além disso, se as probabilidades *a priori* forem iguais, $p(\mathcal{C}_1) =$

$p(\mathcal{C}_2) = 1/2$, a Equação (2.34) torna-se

$$\begin{aligned} p(\mathbf{x}|\mathcal{C}_1) &> p(\mathbf{x}|\mathcal{C}_2), \\ p(\mathbf{x}|\mathcal{C}_1) &< p(\mathbf{x}|\mathcal{C}_2). \end{aligned} \quad (2.35)$$

Assim, no caso de classes em que as probabilidades *a priori* forem iguais, a decisão é baseada somente na avaliação da distribuição de verossimilhança. Na Figura 6 são apresentadas as distribuições para um problema de duas classes com probabilidades *a priori* iguais.

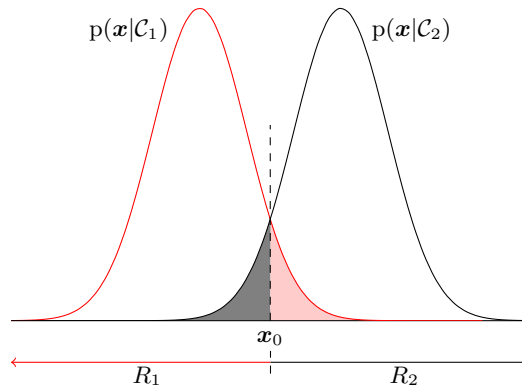


Figura 6 – Distribuições normais para um problema de duas classes com probabilidades *a priori* iguais. Nesse caso, as distribuições *a posteriori* são iguais às verossimilhanças de cada classe. Nesse exemplo, os valores de $\mathbf{x} \in \mathbb{R}$, logo as distribuições são geradas no espaço $\mathbb{R}^2$. O ponto extremo ou limiar de decisão $\mathbf{x}_0$ é o ponto em que $p(\mathbf{x}|\mathcal{C}_1) = p(\mathbf{x}|\mathcal{C}_2)$. Esse ponto divide os padrões em duas regiões de decisão, R_1 e R_2 .

Fonte – Próprio autor.

De acordo com a regra de Bayes e o caso explanado na Figura 6, pode-se realizar uma análise do comportamento do classificador Bayesiano. Se um padrão $\mathbf{x} \in \mathbb{R}$ é classificado como pertencente à classe $\mathcal{C}_1$, significa que ele está localizado em uma região R_1 em que a probabilidade *a posteriori* da classe $\mathcal{C}_1$ possui um valor maior que a da classe $\mathcal{C}_2$. Caso contrário, o padrão $\mathbf{x}$ seria classificado como pertencente à classe $\mathcal{C}_2$ e estaria localizado na região R_2 . Ainda com base na regra de Bayes, existe um ponto extremo $\mathbf{x}_0$ entre as classes $\mathcal{C}_1$ e $\mathcal{C}_2$, no qual o valor das suas distribuições *a posteriori* são iguais. Considerando padrões $\mathbf{x} \in \mathbb{R}$, os valores de $\mathbf{x} < \mathbf{x}_0$ pertencem a classe $\mathcal{C}_1$ ou à classe $\mathcal{C}_2$, caso contrário.

Nesse contexto, é possível notar que podem ocorrer erros de classificação. A probabilidade de erro pode ser calculada com base na probabilidade de um padrão $\mathbf{x} \in \mathcal{C}_1$ cair na região da classe $\mathcal{C}_2$ e de um padrão $\mathbf{x} \in \mathcal{C}_2$ cair na região da classe $\mathcal{C}_1$. Quando $\mathcal{C}_1$ e $\mathcal{C}_2$ são equiprováveis, a probabilidade de erro p_e é dada por

$$p_e = \frac{1}{2} \int_{-\infty}^{\mathbf{x}_0} p(\mathbf{x}|\mathcal{C}_2) d\mathbf{x} + \frac{1}{2} \int_{\mathbf{x}_0}^{+\infty} p(\mathbf{x}|\mathcal{C}_1) d\mathbf{x}. \quad (2.36)$$

Até agora, toda a formulação do modelo de classificação, baseado na regra de Bayes, foi obtido através de uma análise empírica. Para isso, utilizou-se o termo mais provável para guiar a decisão. Porém, existe uma sólida fundamentação matemática que valida o método empírico obtido, por mais simples que seja. O modelo de classificação elaborado no presente capítulo, doravante chamado de *classificador Bayesiano*, é ótimo em relação a minimização da probabilidade de erro de classificação. A prova disso é apresentada no Apêndice A.

O classificador Bayesiano também pode ser facilmente generalizado para problemas com mais de duas classes. Em uma tarefa de classificação com M classes, $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_M$, um padrão desconhecido $\mathbf{x}$ é dito pertencer à uma classe $\mathcal{C}_i$ se

$$p(\mathcal{C}_i|\mathbf{x}) > p(\mathcal{C}_j|\mathbf{x}), \forall j \neq i. \quad (2.37)$$

Mesmo para problemas de múltiplas classes, o classificador Bayesiano ainda minimiza a probabilidade de erro p_e .

2.3.1 Funções discriminantes

Algumas vezes, em um ponto de vista matemático, é mais conveniente trabalhar com funções equivalentes do que utilizar probabilidade diretamente. Por exemplo, $g_i(\mathbf{x}) \equiv f(p(\mathcal{C}_i|\mathbf{x}))$, onde $f(\cdot)$ é monotonicamente crescente. Nesse sentido, $g_i(\mathbf{x})$ é chamada de *função discriminante*.

Com a aplicação de uma função discriminante, a função de decisão descrita na Equação (2.37), pode ser reescrita como:

$$\mathbf{x} \in \mathcal{C}_i \text{ se } g_i(\mathbf{x}) > g_j(\mathbf{x}) \quad \forall j \neq i. \quad (2.38)$$

As superfícies de decisão, que separam o espaço entre classes, são descritas por

$$g_{ij}(\mathbf{x}) \equiv g_i(\mathbf{x}) - g_j(\mathbf{x}) = 0, \quad i, j = 1, 2, \dots, M, \quad i \neq j. \quad (2.39)$$

A aplicação de uma função discriminante não afeta a probabilidade em termos de maximização ou minimização.

2.4 Estimação da função de densidade de probabilidade e seus parâmetros

O funcionamento do classificador Bayesiano, depende da distribuição de probabilidade dos dados. Porém, nem sempre essa distribuição é conhecida. Por isso, em muitos problemas, a função de densidade de probabilidade precisa ser estimada a partir dos dados disponíveis. Várias metodologias podem ser aplicadas na estimação da distribuição, mas a

abordagem escolhida será baseada nas informações que se possui a respeito dos dados. As vezes se conhece a distribuição de probabilidade, mas os parâmetros estatísticos não são conhecidos, como média e variância. Por outro lado, algumas vezes se tem informações estatísticas, mas não se conhece a distribuição de probabilidade. Nesta subseção são apresentados métodos que permitem a estimação de tais distribuições.

2.4.1 Estimação de parâmetros por máxima verossimilhança

Considerando um problema com L classes em que seus respectivos padrões são gerados a partir da distribuição $p(\mathbf{x}|\mathcal{C}_i), i = 1, \dots, L$. Além disso, assumindo que essas funções de verossimilhança estão em uma forma paramétrica em que os parâmetros correspondentes são representados pelos vetores desconhecidos $\boldsymbol{\theta}_i$. Nesse sentido, a distribuição de probabilidade dos padrões de cada classe é reescrita como $p(\mathbf{x}|\mathcal{C}_i; \boldsymbol{\theta}_i)$, em função dos vetores de parâmetros $\boldsymbol{\theta}_i$. O objetivo é estimar os parâmetros desconhecidos utilizando os padrões de cada classe. Supondo também que os dados de uma classe não afetam a estimação de parâmetros de outras classes, é possível formular o problema de maneira simplificada e independentemente de classes.

Sejam $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$ amostras aleatórias obtidas a partir da pdf $p(\mathbf{x}; \boldsymbol{\theta})$. A distribuição conjunta de todos os padrões é dada por $p(\mathcal{X}; \boldsymbol{\theta})$, onde $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ é o conjunto de amostras. Assumindo que existe uma independência entre as amostras, a distribuição conjunta pode ser escrita por:

$$p(\mathcal{X}; \boldsymbol{\theta}) \equiv p(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N; \boldsymbol{\theta}) = \prod_{i=1}^N p(\mathbf{x}_i; \boldsymbol{\theta}). \quad (2.40)$$

Essa é uma função de $\boldsymbol{\theta}$ e também é conhecida como verossimilhança de $\boldsymbol{\theta}$ em relação à $\mathcal{X}$. O método de máxima verossimilhança (ML) estima os valores de $\boldsymbol{\theta}$ de modo que a função de verossimilhança tenha seu valor máximo, a saber

$$\hat{\boldsymbol{\theta}}_{\text{ML}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \prod_{i=1}^N p(\mathbf{x}_i; \boldsymbol{\theta}). \quad (2.41)$$

Uma condição necessária para que $\hat{\boldsymbol{\theta}}_{\text{ML}}$ seja um ponto de máximo, é o gradiente da função de verossimilhança em relação à $\boldsymbol{\theta}$ ser igual a zero, assim

$$\frac{d \prod_{i=1}^N p(\mathbf{x}_i; \boldsymbol{\theta})}{d\boldsymbol{\theta}} = 0. \quad (2.42)$$

Como a função logarítimo é monotonicamente crescente, é possível aplicar essa função à expressão da verossimilhança. Com isso, é definida a seguinte expressão:

$$\mathbf{L}(\boldsymbol{\theta}) = \ln \prod_{i=1}^N p(\mathbf{x}_i; \boldsymbol{\theta}). \quad (2.43)$$

Com isso, a Equação (2.42) é equivalente à

$$\frac{d\mathbf{L}(\boldsymbol{\theta})}{d\boldsymbol{\theta}} = \sum_{i=1}^N \frac{d \ln p(\mathbf{x}_i; \boldsymbol{\theta})}{d\boldsymbol{\theta}} = \sum_{i=1}^N \frac{1}{p(\mathbf{x}_i; \boldsymbol{\theta})} \frac{d p(\mathbf{x}_i; \boldsymbol{\theta})}{d\boldsymbol{\theta}} = 0. \quad (2.44)$$

A estimação por ML corresponde ao ponto de máximo na função de verossimilhança, conforme apresentado na Figura 7.

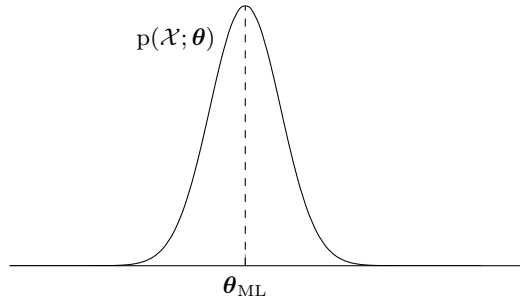


Figura 7 – Parâmetro estimado pelo método de máxima verossimilhança.

Fonte – Próprio autor.

2.4.2 Estimação de parâmetros por máxima posteriori

Quando se estima os hiperparâmetros através do método de máxima verossimilhança, o vetor $\boldsymbol{\theta}$ é desconhecido. Neste caso, ele é tratado como um vetor aleatório e que sua estimação está condicionada às amostras $\mathbf{x}_1, \dots, \mathbf{x}_N$ ocorrerem. Seja $\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$, nosso ponto inicial é $p(\boldsymbol{\theta}|\mathcal{X})$. A posteriori a partir do teorema de Bayes é dada por

$$p(\boldsymbol{\theta}|\mathcal{X}) = \frac{p(\mathcal{X}|\boldsymbol{\theta}) p(\boldsymbol{\theta})}{p(\mathcal{X})}. \quad (2.45)$$

O método de estimação por máxima posteriori (MAP) estima o $\hat{\boldsymbol{\theta}}_{\text{MAP}}$, que representa o ponto em que a distribuição $p(\boldsymbol{\theta}|\mathcal{X})$ é máxima. Ou seja,

$$\frac{d(p(\mathcal{X}|\boldsymbol{\theta}) p(\boldsymbol{\theta}))}{d\boldsymbol{\theta}} = 0. \quad (2.46)$$

Nota-se que o denominador não entra no cálculo do gradiente, já que não possui relação com o valor de $\boldsymbol{\theta}$. Além disso, como o denominador é o mesmo para todos os termos do numerador da fração, é possível retirá-lo do cálculo sem afetar a maximização da função. A principal diferença entre estimação por ML e MAP é a utilização da distribuição a priori $p(\boldsymbol{\theta})$. Quando adota-se uma distribuição uniforme, o valor de $\boldsymbol{\theta}_{\text{MAP}}$ é equivalente ao obtido pelo método $\boldsymbol{\theta}_{\text{ML}}$. Em termos práticos, geralmente é adotada uma distribuição não-uniforme em estimadores MAP.

2.5 Conclusão do capítulo

As funções de densidade de probabilidade são a base para a regra de decisão de Bayes. Para que os dados sejam classificados é necessário que se tenham pelo menos duas distribuições, a priori e verossimilhança dos dados. Em termos práticos, podem ser realizadas suposições de qual é a distribuição dos dados e os parâmetros estimados por máxima verossimilhança ou máxima posteriori. Os pontos de máximo das distribuições estão localizados nos pontos em que o gradiente das distribuições é igual a zero. Em algumas situações, podem ser aplicadas funções discriminantes, como a logaritmo natural, a fim de simplificar o cômputo desses pontos de máximo.

Por outro lado, a própria função de densidade de probabilidade pode ser estimada utilizando modelos de misturas. Especificamente, o modelo de mistura de Gaussianas realiza uma combinação de várias componentes, que são distribuições normais. Nesse processo, os pesos da mistura, ou coeficientes da combinação, e os parâmetros da distribuição são estimados através do algoritmo *Expectation-Maximization*.

3 Modelos lineares

O objetivo da tarefa de aprendizagem é estimar o valor de uma ou mais saídas $t \in \mathbb{R}$, dado um vetor de entrada $\mathbf{x} \in \mathbb{R}^D$. Em outras palavras, se busca uma função que descreva um mapeamento entre os dados de entrada e suas respectivas saídas. Uma forma de aproximar funções é através de uma combinação linear entre as variáveis de entrada e um vetor de coeficientes $\mathbf{w} \in \mathbb{R}^D$. Uma limitação dessa abordagem é que as funções obtidas sempre são lineares. Para superar isso, são aplicadas funções de base nos dados de entrada. Com isso, o modelo se torna uma combinação linear de funções não-lineares e muitos outros modelos podem ser obtidos. Neste capítulo serão apresentados os modelos lineares e as principais funções de base que podem ser aplicadas. Além disso, serão discutidos métodos para obtenção dos valores dos coeficientes $\mathbf{w}$. Os conceitos explicados no presente capítulo foram baseados nos trabalhos de Bishop (2006), Wipf (2006) e (WILLIAMS; RASMUSSEN, 2006).

3.1 Introdução

O modelo linear mais simples para regressão, envolve uma combinação linear de variáveis de entrada, a saber

$$y(\mathbf{x}, \mathbf{w}) = w_0 + w_1x_1 + \dots + w_Dx_D, \quad (3.1)$$

onde $\mathbf{x} = (x_1, x_2, \dots, x_D)^\top$ é o vetor de entrada e $\mathbf{w} = (w_1, w_2, \dots, w_D)^\top$ é conhecido como vetor de pesos. Tal modelo é conhecido como *regressão linear* e sua principal propriedade é tratar-se de uma função linear com parâmetros $(w_0, w_1, \dots, w_D)$. Por outro lado, também é uma função linear nas variáveis de entrada $(x_1, x_2, \dots, x_D)$, o que impõe limitações significativas aos modelos que seguem esse formato. Por isso, a classe de modelos é estendida considerando uma combinação linear de funções não-lineares fixas sobre as variáveis de entrada. Com isso, o modelo torna-se

$$y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{j=1}^{M-1} w_j \phi_j(\mathbf{x}), \quad (3.2)$$

onde $\phi_j(\mathbf{x})$ são conhecidas como *funções de base*. Nesse caso, a quantidade total de parâmetros do sistema é M .

O parâmetro w_0 , também é conhecido como *bias* ou *intercepto*. Para escrever o modelo na forma de produto interno, é conveniente definir uma função de base constante adicional $\phi_0(\mathbf{x}) = 1$ para multiplicar o parâmetro w_0 , tal que

$$y(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}) \quad (3.3)$$

onde $\mathbf{w} = (w_0, w_1, \dots, w_{M-1})^\top$ e $\phi(\mathbf{x}) = (\phi_0(\mathbf{x}), \phi_1(\mathbf{x}), \dots, \phi_{M-1}(\mathbf{x}))^\top$. Geralmente, existe uma etapa de pré-processamento nos dados de entrada antes de se aplicar as funções não-lineares.

Em termos práticos, é dado um conjunto de dados de treinamento na forma $\{(\mathbf{x}_i, t_i)\}_{i=1}^N$, onde $\mathbf{x}_i \in \mathbb{R}^D$ são os vetores de entrada e os valores t_i são as saídas desejadas correspondentes. A natureza do valores t_i definem o problema que será tratado. Quando t_i pode assumir somente valores discretos e o conjunto desses possíveis valores é finito, o problema é de classificação. A discretização da saída é realizada através de funções de limiar. Quando t_i pode assumir qualquer valor em $\mathbb{R}$, o problema é de regressão. De qualquer forma, o objetivo é estimar os parâmetros $\mathbf{w}$, de modo que $y(\mathbf{x}_i, \mathbf{w})$ sejam os mais próximos possíveis de t_i . Tal processo é chamado de treinamento. Uma vez que o vetor $\mathbf{w}$ foi devidamente estimado, o modelo pode ser utilizado para obter os valores das saídas t^* , desconhecidas, a partir de vetores de entrada $\mathbf{x}^* \in \mathbb{R}^D$. Vale observar que os padrões $\mathbf{x}^*$ não pertencem ao conjunto de treinamento. Essa fase é chamada de predição ou teste.

Quando são aplicadas funções de base aos N vetores de treinamento, é definida uma matriz $N \times M$, conhecida por *matriz de kernel*, denotada por

$$\Phi = \begin{pmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \cdots & \phi_{M-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \cdots & \phi_{M-1}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \cdots & \phi_{M-1}(\mathbf{x}_N) \end{pmatrix} = \begin{pmatrix} \phi(\mathbf{x}_1) \\ \phi(\mathbf{x}_2) \\ \vdots \\ \phi(\mathbf{x}_N) \end{pmatrix}. \quad (3.4)$$

Além disso, também é definido o vetor $\mathbf{t} = (t_1, \dots, t_N)^\top$, que engloba todas as saídas correspondentes aos padrões de treinamento. O modelo generativo a seguir, pode representar o mapeamento entre a saída desejada e a saída do modelo a partir dos dados de entrada, a saber,

$$\mathbf{t} = \Phi \mathbf{w} + \epsilon. \quad (3.5)$$

Um ponto importante à ser observado é que cada linha da matriz de kernel, quando multiplicada pelo vetor de pesos, formam uma combinação linear da forma descrita na Equação (3.3). Seguindo esse modelo generativo, o objetivo é encontrar o vetor de pesos desconhecido $\mathbf{w}$, dados $\mathbf{t}$ e Φ . O vetor ϵ representa um vetor de ruído aditivo.

3.1.1 Funções de base

Existem muitas possibilidades para a escolha das funções de base. Nesta Subseção, serão apresentadas algumas das principais.

3.1.1.1 Função Gaussiana

A função de base Gaussiana não requer nenhuma interpretação probabilística. Ela é definida através da equação a seguir:

$$\phi_j(x) = \exp \left\{ -\frac{(x - \mu_j)^2}{2\sigma^2} \right\}, \quad (3.6)$$

em que μ_j define a localização do ponto médio da função no espaço de entrada e σ^2 é um parâmetro que define a escala da função. Como a função será multiplicada pelos parâmetros adaptativos w_j , nos modelos lineares definidos pela Equação (3.3), um termo de normalização não é importante. Na Figura 8 são apresentados exemplos da função de base Gaussiana variando o parâmetro de localização.

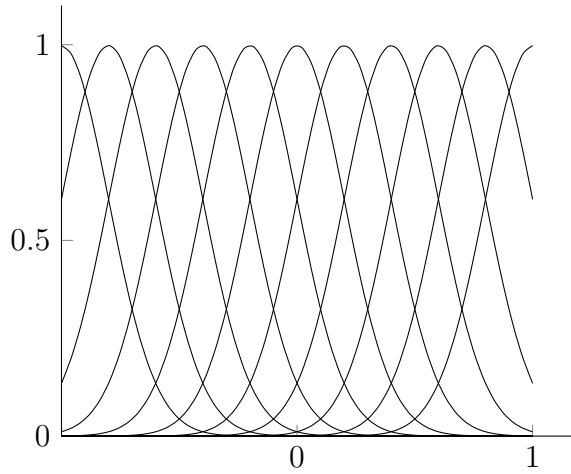


Figura 8 – Exemplos de funções de base Gaussiana.

Fonte – Próprio autor.

3.1.1.2 Sigmoides Logística

Outra possibilidade para função de base é a sigmoide. Tal função é denotada por:

$$\phi_j(x) = \sigma \left(\frac{x - \mu_j}{s} \right), \quad (3.7)$$

onde $\sigma(a)$ é uma função sigmoide logística definida por

$$\sigma(a) = \frac{1}{1 + \exp(-a)}. \quad (3.8)$$

O parâmetro μ_j define a localização do ponto de inflexão da função e o parâmetro s é responsável por modificar a inclinação. Quanto maior é o valor de s , mais suave é o crescimento da função. Na Figura 9 são apresentados exemplos dessa função.

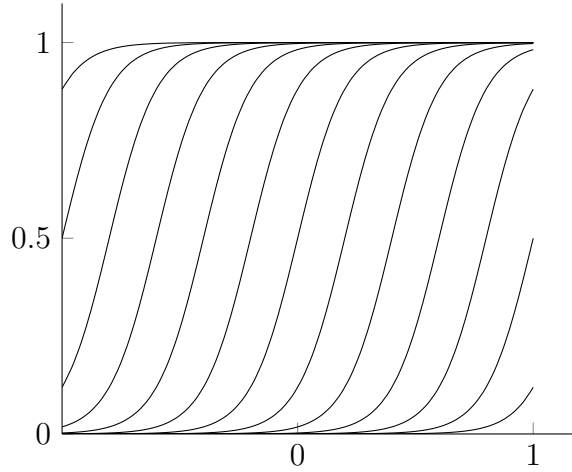


Figura 9 – Exemplos de funções de base sigmoidais.

Fonte – Próprio autor.

3.1.1.3 Tangente Hiperbólica

Similarmente a função sigmoide logística, a função tangente hiperbólica também pode ser utilizada como função de base, sendo descrita por

$$\tanh(a) = 2\sigma(a) - 1, \quad (3.9)$$

onde $\sigma(a)$ é uma função sigmoide logística definida na Equação (3.8). Uma combinação linear de funções logísticas sigmoidais é equivalente à uma combinação linear de funções $\tanh(\cdot)$ (BISHOP, 2006). Na Figura 10 é dado um exemplo da função tangente hiperbólica.

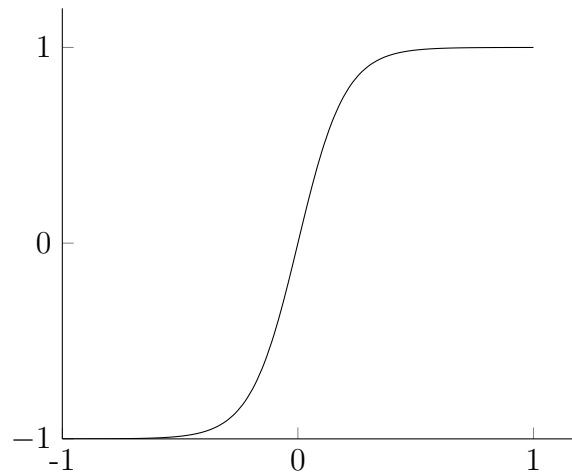


Figura 10 – Função de base tangente hiperbólica.

Fonte – Próprio autor.

3.1.2 Estimação de parâmetros

A forma mais direta de estimar os valores de $\mathbf{w}$ é por Máxima Verossimilhança (Seção 2.4.1) dos dados $p(\mathbf{t}|\mathbf{w})$. Segundo Bishop (2006), Essa abordagem é equivalente

à solução por mínimos quadrados. Quando a dimensão de $\mathbf{w}$ é menor em relação à quantidade de padrões, $M \ll N$, então a solução de ML é bastante efetiva. Porém, em uma grande quantidade de problemas, a dimensão do vetor de pesos $\mathbf{w}$ excede a quantidade de padrões (WIPF, 2006). Nessa situação, o mapeamento inverso de $\mathbf{t}$ para $\mathbf{w}$ é indeterminado. Além disso, consiste em uma tarefa de estimação mais complicada, já que existe um número infinito de soluções produzidas com a mesma verossimilhança.

Uma alternativa Bayesiana para a solução do problema assume que $\mathbf{w}$ foi obtida a partir de alguma distribuição $p(\mathbf{w})$. Isso permite que o espaço de soluções seja restrito a partir de informações específicas do problema. A solução, nesse caso, é obtida através do método MAP (Seção 2.4.2) conforme a equação a seguir:

$$\mathbf{w}_{\text{MAP}} = \underset{\mathbf{w}}{\operatorname{argmax}} \{p(\mathbf{t}|\mathbf{w}) p(\mathbf{w})\}. \quad (3.10)$$

Um problema dessa abordagem é a grande quantidade de coeficientes w_i com valores diferentes de zero (WIPF, 2006).

Outra forma de tratar o problema é considerando que em $\mathbf{w}$ existem muitos elementos iguais a zero. Tais soluções são conhecidas como esparsas e o nível de esparsidade cresce proporcionalmente à quantidade de elementos iguais a zero em $\mathbf{w}$. Tal característica é enfatizada com a incorporação de distribuições *a priori* de caudas longas, possivelmente infinitas e centradas em zero (RAO et al., 2003). Um caminho alternativo para esparsidade é usar as chamadas distribuições *a priori* empíricas, que são caracterizadas por parâmetros flexíveis que podem ser estimados a partir dos próprios dados (NEAL, 1996). Uma das características de ambas as abordagens é que a estimativa de $\mathbf{w}$ a partir de $\mathbf{t}$ e Φ é não-linear. Além disso, embora o valor de M cresça, existe uma grande possibilidade de que uma representação altamente esparsa exista. Em geral, a tarefa de estimação de $\mathbf{w}$ torna-se exponencialmente mais difícil, mesmo em problemas de tamanho moderado. Por isso, a resolução de alguns problemas chega a ser impraticável (WIPF, 2006).

Na próxima seção, o problema de estimar uma solução esparsa para um sistema linear conforme na Equação (3.5) é apresentado.

3.2 Definição do problema da esparsidade

A fim de simplificar os cálculos, é útil definir

$$\|\mathbf{w}\|_0 \triangleq \sum_{i=1}^M \mathcal{I}(|w_i| > 0), \quad (3.11)$$

em que $\mathcal{I}(\cdot)$ é uma função indicadora, $\|\cdot\|_0$ é uma medida de diversidade pois tal medida conta a quantidade de elementos em $\mathbf{w}$ que não são iguais a zero. É comum se referir a isso como norma ℓ_0 , mesmo não sendo uma norma verdadeira. O contrário de diversidade

é a esparsidade, que conta o número de elementos em $\mathbf{w}$ que são exatamente iguais a zero. As duas medidas são relacionadas através da equação:

$$\text{diversidade} = M - \text{esparsidade}. \quad (3.12)$$

Os elementos de $\mathbf{w}$ diferentes de zero são chamados de fontes ativas.

Em Elad (2010) são estabelecidos alguns critérios para que o conjunto de parâmetros encontrados, seja o mais próximo possível de um vetor de parâmetros geradores $\mathbf{w}_{gen}$ do sistema. Em ambientes não ruidosos, uma possível solução para tal problema é pela minimização da norma zero do vetor $\mathbf{w}$. Isso é equivalente à resolver o problema de otimização a seguir:

$$\begin{aligned} \min \quad & \|\mathbf{w}\|_0, \\ \text{s.a.} \quad & \mathbf{t} = \Phi \mathbf{w}. \end{aligned} \quad (3.13)$$

Esse problema algumas vezes é chamado de recuperação exata de esparsidade, já que qualquer solução força a igualdade. Como dito, em geral, problemas como o apresentado na Equação (3.13) é NP-Difícil (NATARAJAN, 1995) e, assim, soluções são encontradas através de métodos de aproximação. Tais métodos foram estudados em profundidade nos trabalhos Donoho e Elad (2003), Fuchs (2004), Gribonval e Nielsen (2003), Tropp (2004).

Em bases de dados reais, como as que serão tratadas nessa dissertação, dificilmente não há ocorrência de ruído. Por isso, somente a norma ℓ_0 não é uma medida que representa tais problemas com precisão. Como o ruído está presente, tipicamente não é possível representar $\mathbf{t}$ com exatidão. Nesses casos, o problema de otimização relaxado a seguir é sugerido:

$$\mathbf{w}_0(\lambda) \triangleq \arg\min_{\mathbf{w}} \|\mathbf{t} - \Phi \mathbf{w}\|_2^2 + \lambda \|\mathbf{w}\|_0, \quad (3.14)$$

onde λ é uma taxa de balanceamento entre a qualidade da estimação e a esparsidade. Do mesmo modo que nos problemas apresentados pelas equações (3.13) e (3.14) também se trata de um problema NP-Difícil, mas o que não é claro é como selecionar o valor de λ (WIPF, 2006). Além disso, não se tem a garantia de uma solução global, já que mesmo encontrando o valor ótimo de λ é necessário ter um bom estimador para $\mathbf{w}_{gen}$ ou pelo menos um perfil de esparsidade correspondente. O perfil de esparsidade de $\mathbf{w}$ é altamente importante, pois irá ditar quais colunas de Φ são mais relevantes.

De um ponto de vista conceitual, o problema descrito na Equação (3.14) pode ser reformulado adicionando constantes e aplicando uma transformação $\exp(-(\cdot))$. Isso leva a uma função de verossimilhança Gaussiana $p(\mathbf{t}|\mathbf{w})$ com uma variância dependente de λ

$$p(\mathbf{t}|\mathbf{w}) \propto \exp\left(-\frac{1}{\lambda} \|\mathbf{t} - \Phi \mathbf{w}\|_2^2\right) \quad (3.15)$$

e uma distribuição *a priori* dada por

$$p_0(\mathbf{w}) \propto \exp(-\|\mathbf{w}\|_0). \quad (3.16)$$

No espaço dos pesos, a distribuição imprópria descrita pela Equação 3.16 mantém um pico afiado quando um peso é igual a zero e caldas uniformes em qualquer outro lugar. O problema descrito na Equação (3.14) pode ser reescrito como

$$\mathbf{w}_0(\lambda) \equiv \operatorname{argmax}_{\mathbf{w}} p(\mathbf{t}|\mathbf{w})p_0(\mathbf{w}) = \operatorname{argmax}_{\mathbf{w}} \frac{p(\mathbf{t}|\mathbf{w})p_0(\mathbf{w})}{p(\mathbf{t})} = \operatorname{argmax}_{\mathbf{w}} p_0(\mathbf{w}|\mathbf{t}). \quad (3.17)$$

Além disso, o problema pode ser visto como uma tarefa de estimação MAP com uma *posteriori* caracterizada por numerosas soluções ótimas locais.

Em resumo, a esparsidade em problemas dessa natureza, atua como um mecanismo de regularização para evitar um sobreajuste (*overfitting*) aos dados de treinamento $\mathbf{t}$. Além disso, permite a geração de modelos com estruturas mais interpretáveis (WIPF, 2006).

3.2.1 Métodos para solução de sistemas unirespostas

Existem vários modelos de aprendizagem com treinamento baseado na solução de um sistema linear de resposta única. Dentre eles estão as Máquinas de Aprendizado Extremo (ELMs)(HUANG; ZHU; SIEW, 2006), em situações que se tem somente um neurônio na camada de saída, e as Máquinas de Vetores Suporte de Mínimos Quadrados (LSSVMs) (SUYKENS; VANDEWALLE, 1999). No caso do ELM, que é uma rede neural, o conhecimento é retido através do armazenamento de pesos de neurônios. Geralmente, esses modelos possuem uma quantidade elevada de neurônios em sua camada oculta. Nesse sentido, soluções esparsas permitem a remoção dos neurônios que são menos importantes para a solução. No modelo LSSVM, o conhecimento obtido durante o processo de treinamento é representado por um vetor de multiplicadores de lagrange e por toda a base de treinamento. Diversos métodos também foram propostos a fim de identificar quais são os padrões de treinamento que são realmente importantes para representação do conhecimento. Essas amostras são conhecidas por vetores suporte e no modelo precursor do LSSVM, conhecido como Máquinas de Vetores Suporte (SVMs) (HEARST et al., 1998), há a identificação automática desses padrões. Ou seja, o modelo SVM gera modelos esparsos após seu processo de treinamento.

Para obtenção de soluções esparsas segundo as formulações apresentadas nas equações (3.13) e (3.14), existem, basicamente, duas estratégias que são mais aplicadas. A primeira engloba métodos gulosos e por outro lado existem estratégias Bayesianas. Os métodos gulosos são abordagens que permitem obter soluções ótimas locais. Nessa categoria, existem algoritmos de seleção sequencial baseados em perseguição correspondente (*Matching Pursuit*) (TROPP; GILBERT, 2007). Esses algoritmos constroem uma pequena coleção de colunas da matriz de coeficientes do sistema linear, com cada nova adição correspondente à solução residual atual. A primeira estratégia Bayesiana é o método ML, que é discutida na Seção 3.1.2, sendo equivalente à solução por mínimos quadrados. Além

disso, existem os métodos MAP e o método Bayes Empírico (*Empirical Bayes*) (MURPHY, 2012).

Os métodos bayesianos baseados em MAP, adotam novas ideias para distribuições a priori. Para o método *Basis Pursuit* (CHEN; DONOHO; SAUNDERS, 2001), também conhecido como LASSO (TIBSHIRANI, 1996) ou norma ℓ_1 regularizada, é adotada uma distribuição de probabilidade a priori Laplaciana sobre $\mathbf{w}$, dada por:

$$p_1(\mathbf{w}) \propto \exp\left(-\sum_{i=1}^M |w_i|\right). \quad (3.18)$$

Com isso, a função de custo a seguir é otimizada para buscar o conjunto de parâmetros $\mathbf{w}$, tal como,

$$\mathbf{w} = \underset{\mathbf{w}}{\operatorname{argmin}} \|\mathbf{t} - \Phi\mathbf{w}\|_2^2 + \lambda \sum_{i=1}^M |w_i|. \quad (3.19)$$

Existem diversas propriedades e algoritmos que permitem a otimização dessa função de custo (DONOHO; ELAD, 2003; RAO et al., 2003; TROPP, 2006). Em geral, esses métodos geram soluções pouco esparsas (WIPF, 2006).

Alternativamente, é possível adotar uma distribuição a priori de Jeffreys

$$p_j(\mathbf{w}) \propto \prod_{i=1}^M \frac{1}{|w_i|}, \quad (3.20)$$

a qual é invariante à escala. Embora seja uma distribuição a priori imprópria (BERGER, 2013), possui características de distribuições esparsas, já que tem caldas pesadas e um pico em zero. Com isso, o problema de otimização torna-se:

$$\mathbf{w} = \underset{\mathbf{w}}{\operatorname{argmin}} \|\mathbf{t} - \Phi\mathbf{w}\|_2^2 + \lambda \sum_{i=1}^M \log |w_i|, \quad (3.21)$$

em que o termo de regularização λ algumas vezes é chamado de entropia Gaussiana (RAO; KREUTZ-DELGADO, 1999). A função de custo baseada na distribuição a priori de Jeffreys possui muitos mínimos locais, porém, se o algoritmo possui uma boa iniciação, podem ser encontradas soluções mais próximas de $\mathbf{w}_{gen}$ do que o método *Basis Pursuit*.

Uma terceira abordagem MAP usa como distribuição de probabilidade a priori uma Gaussiana generalizada, dada por:

$$p(\mathbf{w}) \propto \exp\left(-\sum_{i=1}^M |w_i|^r\right), \quad (3.22)$$

em que $r \in [0, 1]$ é um parâmetro definido pelo usuário. O problema de otimização correspondente, chamado por *Focal Underdetermined System Solver* (FOCUSS), envolve a seguinte equação:

$$\mathbf{w} = \underset{\mathbf{w}}{\operatorname{argmin}} \|\mathbf{t} - \Phi\mathbf{w}\|_2^2 + \lambda \sum_{i=1}^M |w_i|^r. \quad (3.23)$$

Esse procedimento foi originalmente proposto por Leahy e Jeffs (1991). Quando o valor de $r \rightarrow 0$, o problema de otimização se aproxima da norma ℓ_0 , descrito na Equação (3.14). Por outro lado, quando $r \rightarrow 1$, o problema é equivalente à aplicar o método *Basis Pursuit*, já que minimiza a norma ℓ_1 .

O método Bayes empírico, é uma alternativa às estratégias baseadas em máxima a posteriori com distribuições a priori fixas. Nesse método são adotadas distribuições a priori flexíveis durante o processo de otimização. O algoritmo que descreve esse procedimento é descrito na Seção 3.3.

3.3 Sparse bayesian Learning

O método *Sparse Bayesian Learning* (SBL), também conhecido como *Empirical Bayes* (MURPHY, 2012) ou *Automatic Relevance Determination* (ARD)(WIPF; NAGARAJAN, 2008), inicialmente foi aplicado ao treinamento de redes neurais artificiais (MACKAY, 1992a). Posteriormente, no trabalho de Tipping (2001), o SBL foi aplicado em métodos de kernel a fim de obter padrões de treinamento relevantes (*relevant vectors*), sendo tal denominado de Máquinas de Vetores Relevantes (RVM). Devido sua capacidade de obter soluções esparsas na resolução de sistemas lineares, o SBL será descrito no presente trabalho e é a base para a proposta dessa dissertação.

Ao se expandir o modelo generativo descrito na Equação (3.5), pode-se apresentar o sistema linear da seguinte forma:

$$\begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_N \end{bmatrix} = \begin{bmatrix} \phi_0(\mathbf{x}_1) & \phi_1(\mathbf{x}_1) & \cdots & \phi_{M-1}(\mathbf{x}_1) \\ \phi_0(\mathbf{x}_2) & \phi_1(\mathbf{x}_2) & \cdots & \phi_{M-1}(\mathbf{x}_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(\mathbf{x}_N) & \phi_1(\mathbf{x}_N) & \cdots & \phi_{M-1}(\mathbf{x}_N) \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_N \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_N \end{bmatrix}. \quad (3.24)$$

Observa-se que cada elemento t_i é obtido através do produto interno da linha $\phi(\mathbf{x}_i)$ com o vetor $\mathbf{w}$ somado de um resíduo ϵ_i , tal como

$$t_i = \phi(\mathbf{x}_i)\mathbf{w} + \epsilon_i. \quad (3.25)$$

Supondo que ϵ_i é obtido a partir de um processo contaminado com um ruído aditivo gerado a partir de uma distribuição normal com média zero, isto é, $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$, a distribuição de verossimilhança sobre os dados é dada por:

$$\begin{aligned} p(t_i|\mathbf{x}_i, \mathbf{w}, \sigma^2) &= \mathcal{N}(y(\mathbf{x}_i, \mathbf{w}), \sigma^2), \\ &= (2\pi\sigma^2)^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2\sigma^2} (t_i - y(\mathbf{x}_i, \mathbf{w}))^2 \right\}, \\ &= (2\pi\sigma^2)^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2\sigma^2} (t_i - \mathbf{w}^\top \phi(\mathbf{x}_i))^2 \right\}. \end{aligned} \quad (3.26)$$

Observando as N padrões de treinamento simultaneamente, a distribuição de probabilidade sobre todos os dados pode então ser calculada com base do produtório a seguir

$$\begin{aligned} p(\mathbf{t}|\mathcal{X}, \mathbf{w}, \sigma^2) &= \prod_{i=1}^N \mathcal{N}(\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i), \sigma^2), \\ &= \prod_{i=1}^N (2\pi\sigma^2)^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2\sigma^2} (t_i - \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i))^2 \right\}, \\ &= (2\pi\sigma^2)^{-\frac{N}{2}} \exp \left\{ -\frac{1}{2\sigma^2} \|\mathbf{t} - \Phi \mathbf{w}\|^2 \right\}. \end{aligned} \quad (3.27)$$

Em problemas de aprendizado supervisionado, como regressão e classificação, é possível notar que não se busca para o modelo a distribuição das variáveis de entrada diretamente, e sim a distribuição dessas variáveis em relação a $\mathbf{t}$ (BISHOP, 2006), uma vez que o conjunto de todas as variáveis de entrada $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^N$ sempre estão dentre as variáveis condicionantes do problema. Então, para se obter uma notação mais concisa, o conjunto que representa todas as variáveis de entrada foi removido do conjunto de parâmetros explícitos da distribuição de probabilidade. Por isso, a verossimilhança é escrita como $p(\mathbf{t}|\mathbf{w}, \sigma^2)$.

Quando se deseja relacionar as variáveis de entrada e saída através de um modelo generativo, é necessário que a complexidade do modelo seja controlada. Ou seja, é necessário que o crescimento dos parâmetros $\mathbf{w}$ seja limitado. Isso é feito pela definição de uma distribuição a priori explicitamente sobre $\mathbf{w}$. É preferível que seja escolhida uma distribuição suave e com baixa complexidade (WIPF, 2006). No presente trabalho, será seguida a abordagem utilizada na formulação do modelo RVM em que se aplica uma distribuição normal com média zero (TIPPING, 2001). Assim, para uma componente do vetor $\mathbf{w}$, a distribuição é dada por:

$$p(w_i|\alpha_i) = \mathcal{N}(0, \alpha_i^{-1}), \quad (3.28)$$

em que α_i é o hiperparâmetro da distribuição. Observando os M valores de $\mathbf{w}$ simultaneamente, obtem-se a distribuição de $\mathbf{w}$, sujeita aos hiperparâmetros $\boldsymbol{\alpha}$, a saber

$$\begin{aligned} p(\mathbf{w}|\boldsymbol{\alpha}) &= \prod_{i=1}^M \mathcal{N}(0, \alpha_i^{-1}), \\ &= \prod_{i=1}^M (2\pi\alpha_i^{-1})^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2\alpha_i^{-1}} (w_i - 0)^2 \right\}, \\ &= (2\pi)^{-\frac{M}{2}} \left(\prod_{i=1}^M \alpha_i^{\frac{1}{2}} \right) \exp \left\{ -\frac{1}{2} \sum_{i=1}^M \alpha_i w_i^2 \right\}. \end{aligned} \quad (3.29)$$

Para simplificar, é possível substituir o termo $\sum_{i=1}^M \alpha_i w_i^2$ por $\mathbf{w}^\top \mathbf{A} \mathbf{w}$, em que $\mathbf{A} = \text{diag}(\boldsymbol{\alpha})$. Assim,

$$p(\mathbf{w}|\boldsymbol{\alpha}) = (2\pi)^{-\frac{M}{2}} \left(\prod_{i=1}^M \alpha_i^{\frac{1}{2}} \right) \exp \left\{ -\frac{1}{2} \mathbf{w}^\top \mathbf{A} \mathbf{w} \right\}. \quad (3.30)$$

Isso significa que todo hiperparâmetro α_i está associado à um peso w_i . Tais hiperparâmetros são responsáveis por dar importância aos componentes de $\mathbf{w}$.

Como as distribuições de verossimilhança e a priori são distribuições normais, então a distribuição de probabilidade a posteriori também será uma normal $\mathcal{N}(\mathbf{m}, \mathbf{\Sigma})$. Para simplificação dos cálculos, foi assumido que $\beta^{-1} = \sigma^2$. Assim,

$$p(\mathbf{w}|\mathbf{t}, \boldsymbol{\alpha}, \beta) = p(\mathbf{t}|\mathbf{w}, \beta) p(\mathbf{w}|\boldsymbol{\alpha}) \sim \mathcal{N}(\mathbf{m}, \mathbf{\Sigma}), \quad (3.31)$$

em que

$$\mathbf{m} = \beta \mathbf{\Sigma} \mathbf{\Phi}^\top \mathbf{t}, \text{ e} \quad (3.32)$$

$$\mathbf{\Sigma} = (\mathbf{A} + \beta \mathbf{\Phi}^\top \mathbf{\Phi})^{-1}. \quad (3.33)$$

A esparsidade do modelo, é naturalmente alcançada sempre que os valores dos hiperparâmetros α_i tendem ao infinito. Quando o valor de α_i é muito grande, a distribuição que representa a i -ésima componente de $\mathbf{w}$ tem uma abertura muito grande, logo seu valor será muito próximo de zero. Isso estimula o surgimento de vários componentes em $\mathbf{w}$ iguais a zero.

3.3.1 Estimação dos hiperparâmetros

A estimação dos hiperparâmetros é realizada utilizando o processo criado em MacKay (1999). Alguns autores também chamam tal procedimento de *type-II maximum likelihood* (JAMIL; BRAAK, 2012) ou *evidence framework* (MACKAY, 1999). Nesse processo, a evidência $p(\mathbf{t}|\boldsymbol{\alpha}, \beta)$ é considerada como a verossimilhança de um outro modelo de inferência com distribuição a posteriori dada por $p(\boldsymbol{\alpha}, \beta|\mathbf{t})$. A posteriori desse segundo modelo deve ser maximizada a fim de se obter os hiperparâmetros $\boldsymbol{\alpha}$ para a distribuição a priori do primeiro modelo de inferência. Portanto, para calcular $\mathbf{m}$ e $\mathbf{\Sigma}$, a seguinte distribuição a posteriori deve ser maximizada:

$$p(\boldsymbol{\alpha}, \beta|\mathbf{t}) \propto p(\mathbf{t}|\boldsymbol{\alpha}, \beta) p(\boldsymbol{\alpha}) p(\beta). \quad (3.34)$$

Ao se adotar distribuições uniformes e assim as distribuições $p(\boldsymbol{\alpha})$ e $p(\beta)$ são ignoradas. Com isso, o problema se resume a maximizar a evidência do primeiro modelo de inferência, ou seja

$$p(\mathbf{t}|\boldsymbol{\alpha}, \beta) = \int p(\mathbf{t}|\mathbf{w}, \beta) p(\mathbf{w}|\boldsymbol{\alpha}) d\mathbf{w}. \quad (3.35)$$

Substituindo a Equação (3.27) e Equação (3.30) para integração, a evidência é dada por

$$p(\mathbf{t}|\boldsymbol{\alpha}, \beta) = \left(\frac{\beta}{2\pi}\right)^{\frac{N}{2}} \left(\frac{1}{2\pi}\right)^{\frac{M}{2}} \left(\prod_{i=1}^M \alpha_i^{\frac{1}{2}}\right) \exp\{-E(\mathbf{t})\} (2\pi)^{\frac{M}{2}} |\mathbf{\Sigma}|^{\frac{1}{2}}, \quad (3.36)$$

em que

$$E(\mathbf{t}) = \frac{1}{2}(\beta \mathbf{t}^\top \mathbf{t} - \mathbf{m}^\top \mathbf{\Sigma}^{-1} \mathbf{m}). \quad (3.37)$$

Para encontrar o ponto de máximo da função, é aplicada a função discriminante logarítimo natural, assim

$$\ln p(\mathbf{t}|\boldsymbol{\alpha}, \beta) = \frac{N}{2} \ln \beta - E(\mathbf{t}) - \frac{1}{2} \ln |\boldsymbol{\Sigma}| - \frac{N}{2} \ln(2\pi) + \frac{1}{2} \sum_{i=1}^M \ln \alpha_i. \quad (3.38)$$

Essa função precisa ser maximizada em relação à $\boldsymbol{\alpha}$ e β . Assim, é calculado o ponto em que o gradiente da função é zero, em relação a $\boldsymbol{\alpha}$ e β . Resolvendo tais operações, são obtidas as regras de atualização desses parâmetros que são dadas por

$$\alpha_i^{(new)} = \frac{\gamma_i}{m_i^2}, e \quad (3.39)$$

$$\beta^{(new)} = \frac{N - \sum_i \gamma_i}{\|\mathbf{t} - \boldsymbol{\Phi} \mathbf{m}\|^2}, \quad (3.40)$$

em que $\gamma_i = 1 - \alpha_i \Sigma_{ii}$. Os valores de $\boldsymbol{\alpha}$ e β maximizam a distribuição marginal, apresentada na Equação (3.38).

Na prática, são dados valores iniciais para os parâmetros $\boldsymbol{\alpha}$ e β . Com isso, os parâmetros da média e covariância da distribuição de probabilidade a posteriori é calculada utilizando a estimativa inicial de $\boldsymbol{\alpha}$ e β através das equações (3.32) e (3.33). Iterativamente, os valores de $\boldsymbol{\alpha}$ e β são recalculados através das equações (3.39) e (3.40), até que algum critério de convergência seja satisfeito.

Os valores de $\boldsymbol{\alpha}$ e β que maximizam a evidência são utilizados para calcular a distribuição preditiva $p(t|\mathbf{t}, \boldsymbol{\alpha}, \beta) = \mathcal{N}(\mathbf{m}^\top \phi(\mathbf{x}^*), \sigma^2(\mathbf{x}^*))$ para novas entradas $\mathbf{x}^*$, em que $\mathbf{t} = \{t_i\}_{i=1}^N$ e $\sigma^2(\mathbf{x}^*) = \beta^{-1} + \phi(\mathbf{x}^*)^\top \boldsymbol{\Sigma} \phi(\mathbf{x}^*)$. O valor de $\mathbf{t}$ é a média da distribuição acima $\mathbf{m}^\top \phi(\mathbf{x}^*)$. A confiança na predição é determinada por $\sigma^2(\mathbf{x}^*)$.

3.3.2 Resumo do algoritmo para SBL

Dadas as informações $\mathbf{t}$ e uma matriz de kernel $\boldsymbol{\Phi}$, o algoritmo SBL é apresentado pelos passos descritos no Algoritmo 1.

3.4 Modelos lineares de múltiplas respostas

Nesse caso são considerados L modelos generativos tal como apresentado na Equação (3.5). Isso é escrito como um modelo linear de múltiplas respostas apresentado a seguir:

$$\mathbf{T} = \boldsymbol{\Phi} \mathbf{W} + \mathbf{E}, \quad (3.41)$$

em que $\mathbf{T} = (\mathbf{t}_1, \dots, \mathbf{t}_L)$, $\mathbf{W} = (\mathbf{w}_1, \dots, \mathbf{w}_L)$, e $\mathbf{E}$ é a matriz de ruído. Para simplificar a notação, $\mathbf{x}_j$ é a j -ésima coluna de uma matriz $\mathbf{X}$ enquanto $\mathbf{x}_i$ representa a i -ésima linha.

Expandindo o modelo descrito na Equação (3.41), é obtida a equação matricial a seguir:

$$\begin{bmatrix} t_{11} & \dots & t_{1L} \\ \vdots & \ddots & \vdots \\ t_{N1} & \dots & t_{NL} \end{bmatrix} = \begin{bmatrix} \phi_1(\mathbf{x}_1) & \dots & \phi_M(\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ \phi_1(\mathbf{x}_N) & \dots & \phi_M(\mathbf{x}_N) \end{bmatrix} \begin{bmatrix} w_{11} & \dots & w_{1L} \\ \vdots & \ddots & \vdots \\ w_{M1} & \dots & w_{ML} \end{bmatrix} + \begin{bmatrix} \epsilon_{11} & \dots & \epsilon_{1L} \\ \vdots & \ddots & \vdots \\ \epsilon_{N1} & \dots & \epsilon_{NL} \end{bmatrix}. \quad (3.42)$$

Nesse sistema, pode ser observado que a coluna $\mathbf{t}_i$ da matriz $\mathbf{T}$ é gerada a partir do produto da matriz Φ pela coluna $\mathbf{w}_i$ somada com a coluna ϵ_i . Além disso é possível notar que um vetor de base, uma linha da matriz Φ , é responsável pela criação de mais de um vetor de resposta, uma coluna da matriz $\mathbf{T}$. Diante disso, quando se deseja obter soluções esparsas, assume-se que a matriz de pesos $\mathbf{W}$ deve possuir um número mínimo de linhas diferentes de zero, o que é equivalente a remover colunas da matriz Φ e, por sua vez, colunas da matriz $\mathbf{T}$. O objetivo da inferência torna-se obter simultaneamente cada vetor de pesos $\mathbf{w}_j$ adotando a todos um perfil de esparsidade comum.

Algoritmo 1 Algoritmo SBL

Entrada:

- Matriz de kernel Φ .
- vetor de rótulos de treinamento $\mathbf{t}$.

Saída:

- Vetor de pesos $\mathbf{w}$ esparso.

Procedimento:

1. Atribua valores para α , de tal maneira que sejam iguais a 1 ou faça uma escolha aleatória de números não-negativos.
 2. Atribua um valor para β . Defina uma estimativa inicial para a variável.
 3. Calcule os parâmetros da distribuição a posteriori $\mathbf{m}$ e Σ através das equações (3.31) e (3.33).
 4. Atualize o valor de α através da Equação (3.39).
 5. Atualize o valor do parâmetro de abertura da distribuição de verossimilhança β através da Equação (3.40).
 6. Remova da matriz Φ e do vetor $\mathbf{t}$, as colunas e as linhas correspondentes aos hiperparâmetros α que são maiores que um valor limiar muito grande. (tendem ao infinito)
 7. Itere nos passos 3, 4, 5 e 6, até que algum critério de convergência seja satisfeito.
 8. Utilize a média da distribuição a posteriori obtida após a execução dos passos anteriores como uma estimativa de $\mathbf{w}_{gen}$.
-

3.5 Esparsidade em modelos lineares de múltiplas respostas

Para simplificar os cálculos, é útil definir uma notação para medidas de esparsidade em soluções de sistemas lineares de múltiplas respostas, dada por

$$d(\mathbf{W}) \triangleq \sum_{i=1}^M \mathcal{I}(\|\mathbf{w}_i\| > 0). \quad (3.43)$$

De maneira similar à definição do problema de esparsidade em sistemas lineares unirespostas, a função $\mathcal{I}(\cdot)$ é uma função indicadora e $\|\cdot\|$ é uma norma vetorial arbitrária. $d(\cdot)$ é uma medida de diversidade linear já que conta o número de linhas em $\mathbf{W}$ que não são iguais a zero. Por outro lado, a medida de esparsidade linear mede a quantidade de linhas em $\mathbf{W}$ que são iguais a zero. Além disso, para um vetor coluna $\mathbf{w}_i$, tem-se que $d(\mathbf{w}_i) = \|\mathbf{w}_i\|_0$ e que $d(\cdot)$ é uma extensão da norma ℓ_0 para matrizes. As linhas diferentes de zero na matriz de pesos $\mathbf{W}$ também são conhecidas como fontes ativas.

Em Wipf (2006) é definida uma matriz de pesos gerados $\mathbf{W}_{gen}$ e critérios que garantem sua unicidade em termos de esparsidade. Os métodos pra resolução de sistemas lineares de múltiplas respostas tem como objetivo estimar soluções mais próximas possíveis de $\mathbf{W}_{gen}$. Inicialmente, será considerado o caso mais simples em que não existe a presença de ruído. Nesse caso, a matriz $\mathbf{E}$ é nula e uma das formas de estimar a matriz $\mathbf{W}_{gen}$ é através da resolução do problema de otimização

$$\mathbf{W}_0 \triangleq \underset{\mathbf{W}}{\operatorname{argmin}}(d(\mathbf{W})), \quad (3.44)$$

$$s.a. \quad \mathbf{T} = \Phi \mathbf{W}.$$

Em geral isso é um problema NP-Difícil, por isso métodos aproximados são utilizados para se obter tal solução (WIPF, 2006).

Quando o ruído é levado em consideração, ou seja, a matriz $\mathbf{E}$ não é nula, o problema é ainda mais complexo. Tipicamente, $\mathbf{T}$ não é representado com exatidão e uma forma de obter esse valor é buscando solução para o problema de otimização

$$\mathbf{W}_0(\lambda) \triangleq \underset{\mathbf{W}}{\operatorname{argmin}}(\|\mathbf{T} - \Phi \mathbf{W}\|_{\mathcal{F}}^2 + \lambda d(\mathbf{W})), \quad (3.45)$$

em que λ é um parâmetro de balanceamento entre qualidade e esparsidade. Uma característica essencial de utilizar $d(\mathbf{W})$ como um termo regularizador é que sempre que um único elemento em uma linha de $\mathbf{W}$ for diferente de zero, não existe penalidade a mais pela existência de outros elementos diferentes de zero na mesma linha. Isso promove um perfil de esparsidade comum, como desejado. Infelizmente, resolver o problema apresentado na Equação (3.45) também é um problema NP-Difícil e não é claro como obter o valor de λ . Além disso, a minimização da função de custo não é garantia de se encontrar uma solução global ou pelo menos uma solução com perfil de esparsidade correspondente a

$\mathbf{W}_{gen}$, ainda que se tenha o valor ótimo de λ (WIPF, 2006). A condição de se obter um perfil de esparsidade equivalente é crucial já que esse fator determina quais colunas de Φ são relevantes. Em termos físicos, essa propriedade é altamente desejável.

A partir de um ponto de vista conceitual, o problema apresentado na Equação (3.45) pode ser reformulado em termos Bayesianos aplicando uma transformação $\exp(-(\cdot))$. Isso gera uma função de verossimilhança normal $p(\mathbf{T}|\mathbf{W})$ com variância λ e uma distribuição a priori dada por $p(\mathbf{W}) \propto \exp(-d(\mathbf{W}))$. No espaço dos pesos, tal distribuição possui um pico afiado e caudas pesadas sempre que uma norma de linha é igual a zero e são uniformes em qualquer outro lugar. De maneira equivalente o problema de otimização apresentado na Equação (3.45) pode ser reescrito como:

$$\mathbf{W}_0(\lambda) \equiv \underset{\mathbf{W}}{\operatorname{argmax}}(p(\mathbf{T}|\mathbf{W}) p(\mathbf{W})) = \underset{\mathbf{W}}{\operatorname{argmax}}(p(\mathbf{W}|\mathbf{T})). \quad (3.46)$$

Desse modo, o problema descrito na Equação (3.46) pode ser visto como um problema de estimação por MAP, caracterizado pela ocorrência de soluções ótimas locais.

3.5.1 Métodos para solução de sistemas lineares de múltiplas respostas

Para resolução de sistemas lineares com múltiplas respostas, abordagens utilizadas na resolução de sistemas de resposta única, descritas na Seção 3.2.1, foram extendidas. Da mesma forma, os métodos são divididos basicamente em estratégias gulosas e métodos Bayesianos e todas visam a resolução dos problemas apresentados nas equações (3.44) e (3.45).

Dentre as estratégias gulosas, se destaca o algoritmo, baseado em busca correspondente (*matching pursuit*), e é a extensão do método *Orthogonal Matching Pursuit* (COTTER et al., 2005). Esse método, de forma sequencial, constrói uma pequena coleção de colunas da matriz de *kernel*, em que cada nova adição ao conjunto continua sendo equivalente ao atual.

As estratégias de soluções Bayesianas são divididas em métodos baseados em MAP e o Bayes empírico (MACKAY, 1992a). Os métodos MAP modificam a distribuição a priori a fim de obter novas funções para serem otimizadas. A primeira abordagem baseada em MAP é uma extensão do método *Basis Pursuit* (CHEN; DONOHO; SAUNDERS, 2001) ou LASSO (TIBSHIRANI, 1996) para sistemas lineares de múltiplas respostas. Esse algoritmo realiza um relaxamento convexo no problema apresentado na Equação (3.45) e tenta resolver o seguinte problema

$$\mathbf{W} = \underset{\mathbf{W}}{\operatorname{argmin}} \|\mathbf{T} - \Phi \mathbf{W}\|_{\mathcal{F}}^2 + \lambda \sum_{i=1}^M \|\mathbf{w}_i\|_2. \quad (3.47)$$

Para a otimização dessa função de custo, muitos métodos podem ser aplicados. Observando em uma perspectiva Bayesiana, o problema descrito na Equação (3.47) é equivalente à

uma estimação MAP com uma distribuição a priori Laplaciana sobre a norma ℓ_2 para cada linha. As propriedades desse método são discutidas em Cotter et al. (2005), Malioutov, Cetin e Willsky (2005).

Uma segunda estratégia baseada em MAP, é a extensão do método que utiliza a distribuição a priori de Jeffrey¹, onde a penalidade baseada na norma ℓ_1 é substituída por um termo regularizado baseado na negativa do logaritmo da distribuição a priori de Jeffreys sobre a norma das linhas da matriz de pesos. Com isso, o problema de otimização se torna

$$\mathbf{W} = \underset{\mathbf{W}}{\operatorname{argmin}} \|\mathbf{T} - \Phi \mathbf{W}\|_{\mathcal{F}}^2 + \lambda \sum_{i=1}^M \log \|\mathbf{w}_i\|_2. \quad (3.48)$$

Essa função de custo sofre com a existência de muitos mínimos locais, mas com uma boa iniciação pode encontrar estimativas para $\mathbf{W}_{gen}$ melhores que a extensão do método *Basis Pursuit*.

O método M-FOCUSS, descrito em Cotter et al. (2005), Rao e Kreutz-Delgado (1998), Rao, Engan e Cotter (2004), é uma extensão do método FOCUSS para sistemas lineares de resposta única. Essa abordagem emprega uma norma ℓ_p como medida de diversidade, onde $p \in [0, 1]$ é um parâmetro definido pelo usuário, para estimular a produção de modelos com poucas linhas em $\mathbf{W}$ diferentes de zero. No contexto de estimação por MAP, esse método pode ser derivado utilizando uma distribuição a priori Gaussiana generalizada sobre a norma das linhas, como ocorrido com a distribuição Laplaciana e a priori de Jeffreys. O algoritmo M-FOCUSS busca o ponto de mínimo da função

$$\mathbf{W} = \underset{\mathbf{W}}{\operatorname{argmin}} \|\mathbf{T} - \Phi \mathbf{W}\|_{\mathcal{F}}^2 + \lambda \sum_{i=1}^M (\|\mathbf{w}_i\|_2)^p. \quad (3.49)$$

Quando $p \rightarrow 0$, a função de custo aproxima-se do problema descrito na Equação (3.45). Por outro lado, quando $p = 1$, a função de custo é equivalente à função de custo da extensão do método *Basis Pursuit* para sistemas lineares de múltiplas respostas.

O método SBL também foi estendido para trabalhar com sistemas lineares de múltiplas respostas. Sua abordagem é uma alternativa aos métodos que utilizam distribuições a priori fixas. Como esse método faz parte da proposta principal dessa dissertação, ele é apresentado em mais detalhes a seguir.

3.6 Sparse Bayesian Learning para múltiplas respostas

Em Wipf e Rao (2007), o método SBL foi estendido para trabalhar com sistemas lineares de múltiplas respostas. Tal melhoria, que será referida como *Multiresponse Sparse*

¹ A distribuição a priori de Jeffreys é uma distribuição imprópria e segue a forma $p(x) = 1/x$ (BERGER, 2013).

Bayesian Learning (M-SBL), permitiu a aplicação de tal modelo em métodos de aprendizagem como as máquinas de aprendizagem mínimas, conforme será proposto no presente trabalho. A descrição do M-SBL é baseada em Wipf (2006).

Considerando o modelo de múltiplas respostas apresentado na Equação (3.41), é definida uma distribuição de probabilidade de verossimilhança sobre os pesos $p(\mathbf{T}|\mathbf{W})$. É adotada uma distribuição Normal com média zero e variância β . Assim, para cada par $\mathbf{t}_{\cdot j}, \mathbf{w}_{\cdot j}$,

$$p(\mathbf{t}_{\cdot j}|\mathbf{w}_{\cdot j}, \beta) = (2\pi\beta)^{-\frac{N}{2}} \exp \left\{ -\frac{1}{2\beta} \|\mathbf{t}_{\cdot j} - \Phi \mathbf{w}_{\cdot j}\|_2^2 \right\}. \quad (3.50)$$

Isso é equivalente à distribuição de verossimilhança dos sistemas lineares de resposta única, descrita pela Equação (3.27).

No caso de múltiplas respostas, quando se quer estimar os parâmetros $\mathbf{W}$, é preferível que se tenham muitas linhas iguais a zero. Sabendo disso, visando restringir o crescimento dos valores das linhas de $\mathbf{W}$, são associadas às suas linhas distribuições de probabilidade a priori com dimensão L . Nesse caso, são adotadas distribuições Normais multivariadas com média zero. Portanto, a i -ésima linha de $\mathbf{W}$ é associada à uma distribuição Normal com média zero e covariância $\alpha_i \mathbf{I}$, tal que $\alpha_i \in \mathbb{R}_+$. A saber,

$$p(\mathbf{w}_{i\cdot}; \alpha_i) \triangleq \mathcal{N}(\mathbf{0}, \alpha_i \mathbf{I}). \quad (3.51)$$

Observando simultaneamente as M linhas de $\mathbf{W}$, é obtida a probabilidade a priori sobre toda a matriz de pesos

$$p(\mathbf{W}; \boldsymbol{\alpha}) = \prod_{i=1}^M \mathcal{N}(\mathbf{0}, \alpha_i \mathbf{I}), \quad (3.52)$$

em que $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_M)^\top$.

A distribuição de probabilidade a posteriori é obtida pelo produto da verossimilhança com a distribuição a priori. Além disso, como foram adotadas distribuições Normais para a distribuição de verossimilhança e também para a priori, a posteriori também será uma Normal. Assim, a distribuição a posteriori da j -ésima coluna de $\mathbf{W}$ é dada por:

$$p(\mathbf{w}_{\cdot j}|\mathbf{t}_{\cdot j}; \boldsymbol{\alpha}) = \frac{p(\mathbf{w}_{\cdot j}, \mathbf{t}_{\cdot j}; \boldsymbol{\alpha})}{\int p(\mathbf{w}_{\cdot j}, \mathbf{t}_{\cdot j}; \boldsymbol{\alpha}) d\mathbf{w}_{\cdot j}} \sim \mathcal{N}(\boldsymbol{\mu}_{\cdot j}, \boldsymbol{\Sigma}), \quad (3.53)$$

em que a média e covariância é dada por:

$$\boldsymbol{\Sigma} \triangleq \text{Cov}(\mathbf{w}_{\cdot j}|\mathbf{t}_{\cdot j}; \boldsymbol{\alpha}) = \mathbf{A} - \mathbf{A} \Phi^\top \Sigma_t^{-1} \Phi \mathbf{A}, \forall j = 1, \dots, L, \quad (3.54)$$

$$\mathbf{m} = (\boldsymbol{\mu}_{\cdot 1}, \dots, \boldsymbol{\mu}_{\cdot L}) \triangleq \mathbf{E}(\mathbf{W}|\mathbf{T}; \boldsymbol{\alpha}) = \mathbf{A} \Phi^\top \Sigma_t^{-1} \mathbf{T}, \quad (3.55)$$

tal que $\mathbf{A} \triangleq \text{diag}(\boldsymbol{\alpha})$ e $\Sigma_t \triangleq \beta \mathbf{I} + \Phi \mathbf{A} \Phi^\top$.

De maneira similar ao caso unidimensional, a média da distribuição a posteriori $\mathbf{m}$ é uma estimativa para $\mathbf{W}_{gen}$. A esparsidade é naturalmente atingida quando

os hiperparâmetros α_i tendem a zero. Isso força a distribuição a posteriori satisfazer $p(\mathbf{w}_{i\cdot} = \mathbf{0} | \mathbf{T}; \alpha_i = 0) = 1$, fazendo com que a média da i -ésima linha μ_i seja zero, como desejado. Com isso, o problema de estimar os parâmetros $\mathbf{W}_{gen}$ é transferido para a tarefa de estimação dos valores adequados e localidação dos valores diferentes de zero para os hiperparâmetros $\boldsymbol{\alpha}$. Isso pode ser realizado através do processo de aproximação da evidência, descrito a seguir.

3.6.1 Estimação dos hiperparâmetros

O processo de estimação dos hiperparâmetros, da mesma forma que no caso unidimensional, é realizado através da maximização da evidência em relação aos hiperparâmetros $\boldsymbol{\alpha}$ e β . Isso é feito buscando o ponto em que o gradiente da evidência é nulo através de um processo iterativo. Antes do cálculo do gradiente, é aplicada a função discriminante logarítmica à evidência. Com isso, a função que se deseja maximizar é dada por:

$$-2 \ln p(\mathbf{T} | \boldsymbol{\alpha}) = -2 \ln \int p(\mathbf{T} | \mathbf{W}) p(\mathbf{W}; \boldsymbol{\alpha}) d\mathbf{W} \quad (3.56)$$

$$= -2 \ln p(\mathbf{T}; \boldsymbol{\alpha}) \quad (3.57)$$

$$\equiv L \ln |\boldsymbol{\Sigma}_t| + \sum_{j=1}^L \mathbf{t}_{\cdot j}^T \boldsymbol{\Sigma}_t^{-1} \mathbf{t}_{\cdot j}. \quad (3.58)$$

O produto por -2 foi realizado em termos de simplificação da função. Isso não modifica o ponto em que o gradiente é zero, somente o problema de maximização para minimização.

A operação de marginalização para otimização dos hiperparâmetros é praticada por Bayesianos, como uma forma natural de incorporar uma medida de complexidade à tarefa de comparação de modelos. A medida de complexidade que se incorpora é a navalha de Occam (MACKAY, 1992a). Existem, pelo menos duas maneiras de minimizar a função $-2 \ln p(\mathbf{T} | \boldsymbol{\alpha})$. A primeira é através do algoritmo *Expectation-Maximization* (EM), proposto em Dempster, Rubin e Tsutakawa (1981), Harville (1977). A regra de atualização dos hiperparâmetros através do método EM é dada por:

$$\alpha_i^{(new)} = \frac{1}{L} \|\boldsymbol{\mu}_{i\cdot}\|_2^2 + \Sigma_{ii}, \forall i = 1, \dots, M. \quad (3.59)$$

O algoritmo EM possui garantia de convergência comprovada, mas em termos práticos, a convergência é lenta. Alternativamente, os hiperparâmetros podem ser atualizados utilizando a regra proposta por MacKay (1992a). Essa metodologia também possui convergência comprovada e é bem mais rápida do que o algoritmo EM. A regra de atualização é dada por:

$$\alpha_i^{(new)} = \frac{\frac{1}{L} \|\boldsymbol{\mu}_{i\cdot}\|_2^2}{1 - \alpha_i^{-1} \Sigma_{ii}}, \forall i = 1, \dots, M. \quad (3.60)$$

Em termos práticos, a qualidade dos resultados obtidos através da regra de atualização baseada no EM é mais elevada, mas possui um desempenho mais baixo. A regra que

permite a atualização do parâmetro β é dada por:

$$\beta^{(new)} = \frac{\frac{1}{L} \|\mathbf{T} - \Phi \mathbf{m}\|_{\mathcal{F}}^2}{N - M + \sum_{i=1}^M \frac{\Sigma_{ii}}{\alpha_i}}. \quad (3.61)$$

Esse processo é originalmente discutido em Tipping (2001) e MacKay (1992a).

3.6.2 Sumário do algoritmo

Algoritmo 2 Algoritmo M-SBL

Entrada:

- Matriz de kernel Φ .
- matriz de saídas $\mathbf{T}$.

Saída:

- Matriz de pesos $\mathbf{W}$ esparsa.

Procedimento:

1. Atribua valores para α . Defina todos os valores iguais a 1 ou faça uma escolha aleatória de números não-negativos.
 2. Calcule os parâmetros da distribuição a posteriori $\mathbf{m}$ e Σ com base nas equações (4.13) e (3.54).
 3. Atualize o valor de α utilizando a regra EM, apresentada na Equação (3.59), ou através da regra rápida, mostrada na Equação (3.60).
 4. Atualize o valor do parâmetro de abertura da distribuição de verossimilhança β com base na Equação (3.61).
 5. Remova da matriz Φ e da matriz $\mathbf{T}$, as colunas correspondentes aos hiperparâmetros α que são menores que um limiar muito pequeno. (tendem a zero)
 6. Itere nos passos 2, 3, 4 e 5, até que algum critério de convergência seja satisfeito.
 7. Utilize a média da distribuição a posteriori obtida após a execução dos passos anteriores como uma estimativa de $\mathbf{W}_{gen}$.
-

O Algoritmo M-SBL consiste de um processo iterativo. No início de sua execução, são dadas estimativas iniciais para os valores de α e λ . O objetivo é calcular os parâmetros da distribuição a posteriori, mais precisamente sua média $\mathbf{m}$, que representa a estimativa de $\mathbf{W}_{gen}$ que se busca. A cada iteração, novos valores para α e λ são estimados e as colunas de Φ podem ser removidas quando se observam valores em α que tendem a zero, ou seja, menores que um limiar muito pequeno ($\alpha_i < 10^{-12}$). Isso faz com que o algoritmo tenha iterações cada vez mais rápidas a medida em que elas ocorrem, pois os computos dos parâmetros se tornam menos custosos, já que as matrizes se tornam cada vez mais esparsas. Esse processo é realizado até que algum critério de convergência seja satisfeito. Um sumário do algoritmo M-SBL é apresentado no Algoritmo 2, em que se tem como entrada as matrizes $\mathbf{T}$ e uma matriz de *kernel* Φ .

3.7 Conclusão do capítulo

Um sistema linear de múltiplas respostas é composto por um conjunto de sistemas lineares de resposta única. Nesses problemas, soluções em que a matriz de pesos possui a maioria das linhas com norma igual a zero, são preferidas. Na literatura, as abordagens para resolver problemas de múltiplas respostas são extensões das metodologias convencionais, para resolução de sistemas de resposta única. O método SBL também foi estendido para trabalhar com problemas de múltiplas respostas e adota uma estratégia em que os parâmetros da distribuição a priori sobre os pesos são modificados durante a execução do algoritmo. Essa abordagem é uma alternativa à métodos que utilizam distribuições de probabilidade a priori fixas.

4 Máquina de Aprendizagem Mínima Bayesiana Esparsa

Neste capítulo será apresentado o modelo proposto nessa dissertação. Esse método consiste em uma abordagem Bayesiana para superar algumas limitações da Máquina de Aprendizagem Mínima (*Minimal Learning Machine*, MLM)(SOUZA et al., 2015) no tocante ao treinamento do modelo. Neste capítulo serão apresentados os fundamentos do modelo MLM, serão levantados alguns questionamentos acerca do modelo e será descrita a proposta principal dessa dissertação.

4.1 Máquinas de Aprendizagem Mínima

Uma MLM é um método supervisionado no qual o processo de treinamento consiste na resolução de um sistema linear de múltiplas respostas a fim de encontrar o mapeamento existente entre as distâncias de entrada e de saída. Nesse modelo, a saída para padrões não vistos durante o treinamento são calculadas através da resolução de um sistema linear, juntamente de um processo de otimização no espaço de saídas possíveis.

4.1.1 Formulação básica

Considando um conjunto de pontos de uma função contínua $f : \mathcal{X} \rightarrow \mathcal{Y}$ dado por $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i = f(\mathbf{x}_i))\}_{i=1}^N$, em que $\mathcal{X} \subset \mathbb{R}^D$ e $\mathcal{Y} \subset \mathbb{R}^S$. O objetivo do método MLM é estimar uma aproximação para a função f , a partir dos pontos do conjunto $\mathcal{D}$. Porém, essa estimativa é realizada no espaço das distâncias dos pontos de $\mathcal{D}$. Por isso, considere o conjunto de K pontos de referência, denotado por $\mathcal{D}_{ref}$, tal que $\mathcal{D}_{ref} \subseteq \mathcal{D}$. A matriz de distâncias no espaço de entrada $\mathbf{D}_{\mathcal{X}}$ representa as distâncias do domínio de cada ponto de $\mathcal{D}$ ao domínio dos pontos de $\mathcal{D}_{ref}$. De maneira similar, a matriz $\mathbf{D}_{\mathcal{Y}}$ representa as distâncias das imagens dos pontos de $\mathcal{D}$ até às imagens dos pontos do conjunto $\mathcal{D}_{ref}$. Em ambas as matrizes, $\mathbf{D}_{\mathcal{X}}$ e $\mathbf{D}_{\mathcal{Y}}$, cada coluna i representa a distância Euclidiana de cada ponto de $\mathcal{D}$ ao i -ésimo ponto de referência de $\mathcal{D}_{ref}$, logo possuem dimensões $N \times K$. Além disso, $\mathbf{D}_{\mathcal{X}}$ e $\mathbf{D}_{\mathcal{Y}}$ são chamadas de matrizes de distância no espaço de entrada e de saída, respectivamente.

4.1.1.1 Regressão de distâncias

O processo de treinamento consiste em recuperar o mapeamento existente entre as distâncias no espaço de entrada e de saída. Isso pode ser feito através do modelo de

regressão de múltiplas respostas a seguir:

$$\mathbf{D}_Y = g(\mathbf{D}_X) + \mathbf{E}, \quad (4.1)$$

em que g é uma função cujo domínio é representado pelo conjunto induzido pelas distâncias de entrada e a imagem é definida pelo conjunto induzido pelas distâncias no espaço de saída. Além disso, $\mathbf{E}$ é uma matriz de resíduos. Assumindo que o mapeamento possui uma estrutura linear para cada resposta, o modelo de regressão pode ser reescrito na forma:

$$\mathbf{D}_Y = \mathbf{D}_X \mathbf{B} + \mathbf{E}, \quad (4.2)$$

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

$$\text{RSS}(\mathbf{B}) = \text{tr}((\mathbf{D}_Y - \mathbf{D}_X \mathbf{B})^\top (\mathbf{D}_Y - \mathbf{D}_X \mathbf{B})). \quad (4.3)$$

No caso em que $K < N$, o sistema é sobredeterminado e não possui soluções diretamente. Isso corresponde ao caso em que $\mathcal{D}_{ref}$ é um subconjunto próprio de $\mathcal{D}$. Nessa situação, $\mathbf{B}$ pode ser aproximada utilizando o método de mínimos quadrados, a saber

$$\mathbf{B} = (\mathbf{D}_X^\top \mathbf{D}_X)^{-1} \mathbf{D}_X^\top \mathbf{D}_Y. \quad (4.4)$$

Observando o caso em que a matriz $\mathbf{D}_X$ utiliza todas as amostras em $\mathcal{D}$ como pontos de referência, isto é, $K = N$, o sistema possui uma única solução, dada por:

$$\mathbf{B} = \mathbf{D}_X^{-1} \mathbf{D}_Y. \quad (4.5)$$

A terceira situação que pode ocorrer é a quantidade de pontos de referência ser maior que a quantidade de amostras utilizadas na criação das matrizes de distância, ou seja, ($K > N$). Isso pode acontecer quando K pontos de referência são selecionados e na criação das matrizes de distância, somente Q elementos de $\mathcal{D}$, tal que $Q < N$, são utilizados para compor as linhas de $\mathbf{D}_X$ e $\mathbf{D}_Y$, ou seja, as matrizes de distância terão dimensão $Q \times K$. Nesse caso, o sistema linear é indeterminado e possui infinitas soluções possíveis. Uma observação importante é que outras metodologias podem ser utilizadas na estimação de $\mathbf{B}$, como RLS (GRANT, 1987).

4.1.1.2 Estimação da saída

De posse da matriz $\mathbf{B}$ e do conjunto de pontos de referência $\mathcal{D}_{ref}$, é possível estimar a imagem de pontos de teste cujos elementos do domínio são representados por $\mathbf{x}^* \in \mathbb{R}^D$. Durante o treinamento, foi encontrada uma estimativa do mapeamento entre os espaços de distâncias no espaço de entrada e saída. Por isso, é necessário que o padrão $\mathbf{x}^*$ também seja representado no domínio das distâncias. Essa representação é dada pelo vetor $\mathbf{d}_{x^*}$ de

dimensões $1 \times K$, que possui as distâncias euclidianas do ponto de teste $\mathbf{x}^*$ ao domínio de todos pontos de referência $\mathcal{D}_{ref}$. A partir de $\mathbf{d}_{x^*}$, é possível estimar as distâncias no espaço de saída através do mapeamento $\mathbf{B}$, a saber

$$\mathbf{d}_{y^*} = \mathbf{d}_{x^*} \mathbf{B}. \quad (4.6)$$

O problema se torna estimar a imagem $\mathbf{y}^* = f(\mathbf{x}^*)$, a partir de $\mathbf{d}_{y^*}$ e das imagens dos pontos de referência em $\mathcal{D}_{ref}$. Isso pode ser tratado como um problema de multilateração (NIEWIADOMSKA-SZYNKIEWICZ; MARKS, 2009). Em um ponto de vista geométrico, estimar $\mathbf{y}^*$ pertencente ao conjunto $\mathbb{R}^S$ é equivalente a resolver um conjunto sobre-determinado de K equações não-lineares correspondentes à hiper-esferas S -dimensionais centradas sobre as imagens dos pontos de referência, denotadas por $\{\mathbf{t}_k\}_{k=1}^K$. Isso é apresentado na Figura 11. A localização de $\mathbf{y}^*$ pode ser estimada através da minimização da função de objetivo abaixo:

$$\mathbf{y}^* = \underset{\mathbf{y}}{\operatorname{argmin}} \sum_{k=1}^K \left((\mathbf{y} - \mathbf{t}_k)^\top (\mathbf{y} - \mathbf{t}_k) - d_y^{(k)} \right)^2, \quad (4.7)$$

em que $d_y^{(k)}$ é a distância euclidiana do rótulo $\mathbf{y}$ até a imagem do k -ésimo ponto de referência.

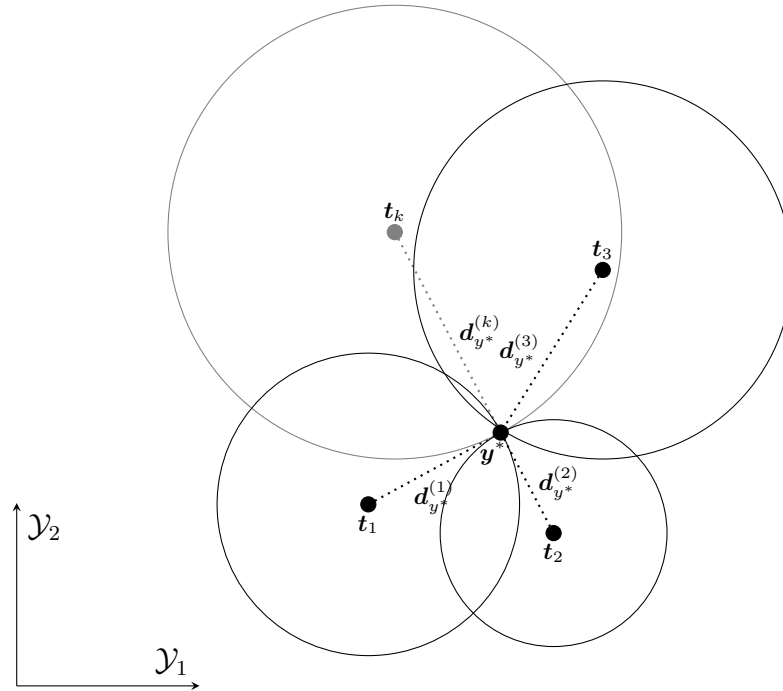


Figura 11 – Estimação da saída $\mathbf{y}^*$. Esse vetor representa o ponto que se deseja estimar através da multilateração, $\{\mathbf{t}\}_{k=1}^K$ são as imagens dos pontos de referência $\mathcal{D}_{ref}$ e $d_{y^*}^k$, é a distância de saída estimada das imagens dos pontos de referência até o rótulo $\mathbf{y}^*$.

No tocante a problemas de classificação, foi provado que o ponto de mínimo da Equação (4.7) é o rótulo do ponto de referência de menor distância. Essa abordagem foi chamada de MLM-NN (MESQUITA; GOMES; JUNIOR, 2017).

4.1.1.3 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 pela identificação dos pontos de referência e estimação do mapeamento das distâncias de entrada e de saída. 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.

Algoritmo 3 Processo de treinamento do modelo MLM

Entrada:

- Conjunto de pontos de treinamento $\mathcal{D}$

Saída:

- Conjunto de pontos de referência $\mathcal{D}_{ref}$
- Mapeamento entre distâncias de entrada e de saída $\mathbf{B}$

Procedimento:

1. Selecione o conjunto de pontos de referência $\mathcal{D}_{ref} \subseteq \mathcal{D}$. Na formulação padrão, essa escolha é aleatória.
 2. Calcule as matrizes de distâncias no espaço de entrada e de saída $\mathbf{D}_X$ e $\mathbf{D}_Y$ utilizando a base de dados $\mathcal{D}$ e os pontos de referência selecionados $\mathcal{D}_{ref}$.
 3. Calcule o mapeamento entre as distâncias de entrada e de saída através da Equação (4.4) ou Equação (4.5), com base na quantidade de pontos de referência selecionados.
-

Algoritmo 4 Processo de predição do modelo MLM

Entrada:

- Padrão de teste $\mathbf{x}^*$
- Conjunto de pontos de referência $\mathcal{D}_{ref}$
- Matriz de mapeamento de distâncias de entrada e de saída $\mathbf{B}$

Saída:

- Rótulo $\mathbf{y}^*$

Procedimento:

1. Calcule o vetor de distâncias $\mathbf{d}_{x^*}$ utilizando o domínio dos pontos de referência $\mathcal{D}_{ref}$.
 2. Utilize o mapeamento $\mathbf{B}$ para estimar as distâncias no espaço de saída $\mathbf{d}_{y^*}$ com base na Equação (4.6).
 3. Estime o rótulo $\mathbf{y}^*$ resolvendo o problema de otimização apresentado pela Equação (4.7).
-

4.1.1.4 Implicações acerca do treinamento do MLM

O MLM consiste em um método que estima uma função $f : \mathcal{X} \rightarrow \mathcal{Y}$ levando o problema para o espaço das distâncias. Nesse segundo plano, é resolvido um sistema linear de múltiplas respostas a fim de encontrar o mapeamento entre as distâncias de entrada e saída. Tal problema é dado pela Equação (4.2) e dependendo das dimensões das matrizes que compõem o problema, ele pode ser resolvido com base na Equação (4.4) ou Equação (4.5). A qualidade da solução desse problema é um dos principais fatores que definem o desempenho do modelo gerado, já que $\mathbf{B}$, mesmo que indiretamente, representa o mapeamento entre os padrões de entrada e saída.

Ainda no contexto do MLM, o primeiro fator determinante para uma boa capacidade de generalização do modelo, é a escolha dos pontos de referência. Eles definem as colunas das matrizes $\mathbf{D}_{\mathcal{X}}$ e $\mathbf{D}_{\mathcal{Y}}$ e isso impacta diretamente na qualidade da resolução do sistema linear descrito na Equação (4.2). Durante a escolha de tais pontos de referência, dois pontos devem ser analisados: (i) a quantidade de pontos e (ii) qual subconjunto de pontos de treinamento será escolhido. Na formulação padrão somente a quantidade é considerada, já que a escolha de quais amostras de treinamento a serem utilizadas é feita de maneira aleatória.

Um segundo ponto que define a qualidade do modelo durante o processo de treinamento é o método de resolução do sistema linear de múltiplas respostas. No Capítulo 3, são discutidas várias abordagens para resolver esse tipo de problema e suas implicações. A complexidade é um fator importante que deve ser analisado durante a concepção de um modelo. Uma das formas de controlar isso é pela adoção de um perfil de esparsidade à solução. Seguindo a discussão apresentada na Seção 3.5, quando se deseja soluções esparsas para sistemas lineares de múltiplas respostas, se busca uma matriz de parâmetros em que a maioria das linhas é nula. Isso é equivalente a eliminar colunas da matriz de coeficientes. Relacionando isso ao modelo MLM, a seleção de pontos de referência pode ser vista como uma tentativa de impor esparsidade a um modelo linear em que todas as amostras de treinamento são consideradas como pontos de referência. Isso ocorre devido as colunas da matriz $\mathbf{D}_{\mathcal{X}}$ serem definidas pelos pontos de referência do modelo.

A busca por soluções esparsas durante o treinamento do método MLM é uma característica desejável não só em termos de redução da complexidade do modelo. Também se deve ao fato de que em sua etapa de predição, a complexidade do algoritmo de multilateração cresce com o número de pontos de referência. Portanto, soluções com poucos pontos de referência, favorecem o desempenho do modelo durante essa etapa.

Na Figura. 12, são apresentados modelos gerados a partir do processo de treinamento do MLM utilizando diferentes grupos de pontos de referência. Na superfície apresentada na Figura 12a, foi escolhida uma quantidade maior de pontos de referência de maneira

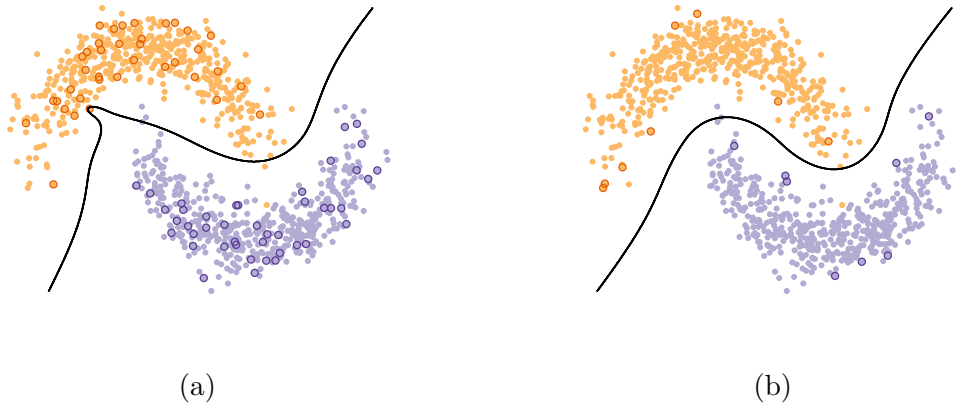


Figura 12 – Superfícies de decisão do modelo MLM utilizando diferentes conjuntos de pontos de referência.

Fonte – Próprio autor.

aleatória. Por outro lado, na Figura 12b, foi adotado um certo critério na determinação dos pontos e na quantidade. É possível notar que a superfície apresentada na Figura 12a é menos suave do que a superfície apresentada na Figura 12b. Isso suporta a hipótese de que a quantidade e a escolha correta dos pontos de referência é um fator determinante da complexidade do modelo gerado.

4.2 Máquina de Aprendizagem Mínima Bayesiana Esparsa (Proposta)

Conforme descrito no Capítulo 3, o modelo M-SBL é capaz de resolver sistemas lineares de múltiplas respostas impondo esparsidade à solução. Devido a essa característica e a necessidade de um critério para impor esparsidade ao MLM, o M-SBL irá substituir seu processo de treinamento padrão. Com isso, a busca pelos pontos de referência e o cálculo dos parâmetros será feito através do algoritmo M-SBL. Nesse processo, todo o conjunto de treinamento é considerado, inicialmente, como pontos de referência e o próprio M-SBL é responsável por identificar os padrões (ou pontos) que irão ser mantidos como pontos de referência no modelo final.

4.2.1 M-SBL aplicado ao modelo MLM

A partir do conjunto de amostras de treinamento $\mathcal{D}$, são montadas as matrizes de distâncias considerando que $\mathcal{D}_{ref} = \mathcal{D}$. Com isso, as matrizes de distâncias $\mathbf{D}_X$ e $\mathbf{D}_Y$ possuem dimensões $N \times N$. De posse dessas matrizes, o problema é estimar o mapeamento entre as distâncias de entrada e de saída. Isso é feito através da resolução do sistema linear descrito pela Equação (4.2). A resolução desse sistema linear de múltiplas respostas será

realizado com base no método M-SBL. Com isso, serão obtidas as matrizes de distância $\mathbf{D}_X$ e $\mathbf{D}_Y$ de forma esparsa, assim como a matriz $\mathbf{B}$.

Como discutido no Capítulo 3, quando se deseja obter soluções esparsas para o sistema linear na forma da Equação 4.2, assume-se que a matriz de pesos $\mathbf{B}$ deve possuir um número mínimo de linhas diferentes de zero, o que é equivalente a remover colunas da matriz $\mathbf{D}_X$ e, por sua vez, colunas da matriz $\mathbf{D}_Y$. Nesse caso, o objetivo da inferência torna-se obter simultaneamente cada vetor de pesos $\mathbf{b}_i$ da matriz $\mathbf{B}$ adotando um perfil de esparsidade comum. Por questões de notação, considere $\mathbf{d}_{\cdot j}^x$ e $\mathbf{d}_{\cdot j}^y$ como sendo o j -ésimo vetor coluna da matriz de distâncias no espaço de entrada e saída, respectivamente. Considerando esse modelo, é definida uma distribuição de probabilidade de verossimilhança sobre os pesos $p(\mathbf{D}_Y|\mathbf{B})$. Nesse caso, foi adotada uma distribuição Normal com média zero e variância β . Assim, para cada par $\mathbf{d}_{\cdot j}^y$, $\mathbf{b}_{\cdot j}$, temos:

$$p(\mathbf{d}_{\cdot j}^y|\mathbf{b}_{\cdot j}, \beta) = (2\pi\beta)^{-\frac{N}{2}} \exp \left\{ -\frac{1}{2\beta} \|\mathbf{d}_{\cdot j}^y - \mathbf{D}_X \mathbf{b}_{\cdot j}\|_2^2 \right\}. \quad (4.8)$$

Para adoção de um perfil de esparsidade para a solução do sistema linear de múltiplas respostas descrito na Equação 4.2, é desejável que se tenham muitas linhas iguais a zero na matriz $\mathbf{B}$. Para restringir o crescimento dessas linhas, são associadas à cada linha da matriz, distribuições de probabilidade *a priori* com média zero e covariância $\alpha_i \mathbf{I}$, tal que $\alpha_i \in \mathbb{R}_+$. A saber,

$$p(\mathbf{b}_i; \alpha_i) \triangleq \mathcal{N}(\mathbf{0}, \alpha_i \mathbf{I}). \quad (4.9)$$

Observando simultaneamente as K linhas de $\mathbf{B}$, é obtida a probabilidade a priori sobre toda a matriz de pesos

$$p(\mathbf{B}; \boldsymbol{\alpha}) = \prod_{i=1}^K \mathcal{N}(\mathbf{0}, \alpha_i \mathbf{I}), \quad (4.10)$$

em que $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_K)^\top$. Como o método proposto também é uma forma de definir quem serão os pontos de referência, inicialmente todos os dados de treinamento são tomados como candidatos a pontos de referência. Com isso, $K = N$ no início do processo.

De posse das expressões para as distribuições *a priori* e verossimilhança, a distribuição posteriori é obtida através do produto entre as duas. Assim, a distribuição a posteriori da j -ésima coluna de $\mathbf{B}$ é dada por:

$$p(\mathbf{b}_{\cdot j}|\mathbf{t}_{\cdot j}; \boldsymbol{\alpha}) = \frac{p(\mathbf{b}_{\cdot j}, \mathbf{t}_{\cdot j}; \boldsymbol{\alpha})}{\int p(\mathbf{b}_{\cdot j}, \mathbf{t}_{\cdot j}; \boldsymbol{\alpha}) d\mathbf{b}_{\cdot j}} \sim \mathcal{N}(\boldsymbol{\mu}_{\cdot j}, \boldsymbol{\Sigma}), \quad (4.11)$$

em que a média e covariância é dada por:

$$\boldsymbol{\Sigma} \triangleq \text{Cov}(\mathbf{b}_{\cdot j}|\mathbf{t}_{\cdot j}; \boldsymbol{\alpha}) = \mathbf{A} - \mathbf{A} \mathbf{D}_X^\top \boldsymbol{\Sigma}_t^{-1} \mathbf{D}_X \mathbf{A}, \forall j = 1, \dots, L, \quad (4.12)$$

$$\mathbf{m} = (\boldsymbol{\mu}_{\cdot 1}, \dots, \boldsymbol{\mu}_{\cdot L}) \triangleq \mathbf{E}(\mathbf{B}|\mathbf{D}_Y; \boldsymbol{\alpha}) = \mathbf{A} \mathbf{D}_X^\top \boldsymbol{\Sigma}_t^{-1} \mathbf{D}_Y, \quad (4.13)$$

tal que $\mathbf{A} \triangleq \text{diag}(\boldsymbol{\alpha})$ e $\boldsymbol{\Sigma}_t \triangleq \beta \mathbf{I} + \mathbf{D}_X \mathbf{A} \mathbf{D}_X^\top$.

Como descrito no Capítulo 3, a esparsidade do modelo ocorre com o alto crescimento dos valores de α_i .

4.2.2 Estimação dos hiperparâmetros para o SBMLM

A estimação dos hiperparâmetros do modelo segue conforme descrito na Seção 3.6.1. Ou seja, a evidência é maximizada em relação aos hiperparâmetros α e β . Isso é feito através de um processo iterativo que caminha em direção ao gradiente nulo da evidência. Antes do cálculo do gradiente, consideramos a aplicação de uma função discriminante logarítmica sobre a função custo. A saber,

$$-2 \ln p(\mathbf{D}_Y | \alpha) = -2 \ln \int p(\mathbf{D}_Y | \mathbf{B}) p(\mathbf{B}; \alpha) d\mathbf{B} \quad (4.14)$$

$$= -2 \ln p(\mathbf{D}_Y; \alpha) \quad (4.15)$$

$$\equiv L \ln |\Sigma_t| + \sum_{j=1}^L (\mathbf{d}_j^y)^\top \Sigma_t^{-1} \mathbf{d}_j^y. \quad (4.16)$$

O produto por -2 foi realizado em termos de simplificação da função. Isso não modifica o ponto em que o gradiente é zero, somente o problema de maximização para minimização.

Com isso, a primeira forma de atualização dos hiperparâmetros é através do método EM, assim:

$$\alpha_i^{(new)} = \frac{1}{L} \|\boldsymbol{\mu}_i\|_2^2 + \Sigma_{ii}, \forall i = 1, \dots, K. \quad (4.17)$$

A comprovação da convergência da solução utilizando o EM é comprovada, porém é um processo demorado. Alternativamente, podemos utilizar uma segunda regra para atualização dos hiperparâmetros com convergência comprovada e desempenho mais elevado que o método EM:

$$\alpha_i^{(new)} = \frac{\frac{1}{L} \|\boldsymbol{\mu}_i\|_2^2}{1 - \alpha_i^{-1} \Sigma_{ii}}, \forall i = 1, \dots, K. \quad (4.18)$$

A regra que permite a atualização do parâmetro β é dada por:

$$\beta^{(new)} = \frac{\frac{1}{L} \|\mathbf{D}_Y - \mathbf{D}_X \mathbf{m}\|_{\mathcal{F}}^2}{N - K + \sum_{i=1}^K \frac{\Sigma_{ii}}{\alpha_i}}. \quad (4.19)$$

Esse processo é discutido em detalhes no Capítulo 3.

Devido a solução esparsa, nem todas as amostras de $\mathcal{D}_{ref}$ são utilizadas no processo de predição. Ou seja, devem ser mantidos nesse conjunto somente as amostras que são pontos de referência, que por sua vez, são as amostras correspondentes as colunas que foram mantidas nas matrizes $\mathbf{D}_X$ e $\mathbf{D}_Y$. Para identificar os pontos de referência, uma modificação no algoritmo padrão do M-SBL foi realizada para que ele também retorne os índices das colunas das matrizes de distâncias que foram removidas. Com isso, as amostras correspondentes a esses índices também podem ser removidas do conjunto $\mathcal{D}_{ref}$. Com essas modificações o processo de treinamento do método, doravante denominado SBMLM, é descrito no Algoritmo 5.

Algoritmo 5 Processo de treinamento do modelo SBMLM

Entrada:

- Conjunto de pontos de treinamento $\mathcal{D}$

Saída:

- Conjunto de pontos de referência $\mathcal{D}_{ref}$
- Mapeamento entre distâncias de entrada e de saída $\mathbf{B}$

Procedimento:

1. Inicialmente, considere todos os pontos de treinamento como pontos de referência, ou seja, $\mathcal{D}_{ref} = \mathcal{D}$.
 2. Calcule as matrizes de distâncias $\mathbf{D}_x$ e $\mathbf{D}_y$ utilizando $\mathcal{D}_{ref}$ e $\mathcal{D}$.
 3. Utilize o método descrito na Seção 4.2.1 para resolver o sistema linear $\mathbf{D}_y = \mathbf{D}_x \mathbf{B} + \mathbf{E}$.
 4. Remova as amostras de $\mathbf{D}_{ref}$ que não são pontos de referência, utilizando os índices das colunas que foram removidas das matrizes $\mathbf{D}_x$ e $\mathbf{D}_y$ durante a resolução do sistema.
-

O processo de predição do MLM não foi modificado na proposição do método SBMLM. Porém, essa é uma etapa beneficiada pela esparsidade obtida no processo de treinamento, já que o algoritmo de multilateração tem crescimento de custo com base no número de pontos de referência. Um ponto importante a ser ressaltado é que o modelo SBMLM tem seus pontos de referência encontrados automaticamente aproveitando-se dos critérios de complexidade incorporados naturalmente pelo M-SBL. Além disso, a matriz de parâmetros $\mathbf{B}$ também é um produto do processo M-SBL. Ou seja, no SBMLM, a seleção de pontos de referência e cálculo da matriz de parâmetros não são tarefas independentes, como na formulação padrão.

No algoritmo M-SBL, podem ser adotadas diversas distribuições de probabilidade. Porém, no SBMLM é adotada uma distribuição de verossimilhança normal com média zero e abertura β . Além disso, foi utilizada uma distribuição de probabilidade a priori normal com média zero e abertura $\alpha_i \mathbf{I}$ (hiperparâmetros) sobre cada linha da matriz $\mathbf{B}$. Por mais que a tarefa utilizada seja de classificação, o M-SBL é baseado na maximização da evidência para estimar os hiperparâmetros do modelo, ou seja, é necessário que se utilize distribuições contínuas devido a integração da evidência. Por esse motivo, foi adotada uma distribuição normal como verossimilhança do modelo. Com isso, também é evitado o uso de outros algoritmos de aproximação para calcular a distribuição a posteriori e a evidência, como ocorrido no RVM.

4.3 Conclusão do capítulo

O modelo MLM é baseado no mapeamento entre as distâncias de entrada e de saída. Uma etapa importante desse modelo é a seleção de pontos de referência, que originalmente

é feita de forma aleatória. 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 o MLM. A seleção aleatória de pontos de referência, é uma tentativa de impor esparsidade ao sistema linear, mas não possui um critério para a escolha de quais pontos são os melhores candidatos à pontos de referência. O método proposto, chamado de SBMLM, resolve o sistema linear através do método M-SBL, que impõe esparsidade automaticamente ao sistema linear durante sua resolução. Com isso, o SBMLM visa preencher a necessidade de critérios na escolha dos pontos de referência do MLM padrão e na abordagem de resolução do sistema linear.

5 Simulações Computacionais

Neste capítulo são apresentados os resultados obtidos para o modelo proposto, SBMLM, comparativamente a outros métodos. Além disso, são feitas discussões a fim de enfatizar as qualidades e limitações do método proposto. A análise do modelo será baseada em quatro pontos que descrevem diversas características do modelo, a saber:

1. **Análise empírica da superfície de decisão:** Esta é uma estratégia empírica para verificar a complexidade do modelo (*overfitting*) e a quantidade de pontos de referência.
2. **Comparativo do SBMLM com outros métodos da literatura:** Este experimento visa comparar acurácia e tempo de treinamento do método proposto em meio a outros métodos da literatura.
3. **Análise comparativa com outras abordagens do próprio MLM:** Este experimento permite analisar as melhorias e problemas que foram causadas pela adoção do algoritmo M-SBL no processo de treinamento do MLM padrão. Tal abordagem experimental visa a comparação entre SBMLM e o MLM em termos de acurácia e esparsidade.
4. **Análise da variação dos parâmetros do SBMLM:** Este experimento visa verificar a influência da variação do valor inicial do parâmetro de abertura da distribuição de verossimilhança sobre a taxa de erro, esparsidade e decaimento do número de pontos de referências.

Nos experimentos realizados neste trabalho, foram utilizadas as bases de dados do UCI Machine Learning (LICHMAN, 2013). Além disso, foi utilizada a base de dados artificial *Ripley*. Todas são descritas na Tabela 1.

5.1 Análise empírica da superfície de decisão

Neste experimento, são comparadas as superfícies de decisão geradas pelo SBMLM, com a versão convencional do MLM, em que ocorre a seleção de pontos de referência aleatória, que nesse caso é chamada de RNMLM.

5.1.1 Configuração dos experimentos

Nessa análise, foram utilizadas duas base de dados artificiais. A primeira delas é a Ripley, que já é largamente utilizada na literatura. O segundo problema consiste de

Tabela 1 – Bases de dados utilizadas neste trabalho.

<i>Base de dados</i>	<i>Abreviação</i>	<i>#Amostras</i>	<i>#Características</i>
Sonar, Mines vs. Rocks	SON	208	60
Heart Disease	HEA	270	13
Breast Cancer Winconsin	BCW	683	9
Haberman's Survival	HAB	306	3
Pima Indians Diabets	PID	768	8
Ripley	RIP	1250	2
Vertebral Column Pathologies	VCP	310	6
Australian Credit Approval	AUS	690	14
Hepatitis	HEP	80	19

uma base artificial binária em que os padrões de cada classe tem seus dados organizados em formato de quadrados. Além disso, os dois quadrados são sobrepostos em uma de suas extremidades. Os resultados utilizando esse conjunto de dados são apresentados na Figura 13c e Figura 13d. Os modelos foram treinados utilizando 80% das amostras de cada base de dados. O parâmetro β do modelo SBMLM foi otimizado através de validação cruzada com busca em grade. Em tal processo, a base de dados de treinamento foi particionada em 10-*folds* e um intervalo de 20 valores linearmente crescentes de 2^{-7} a 2^3 foram utilizados. O parâmetro K do RNMLM, também foi obtido através de validação cruzada com busca em grade. Porém, os valores de K variaram de 5% a 50% do tamanho total da base de dados com incrementos de 5%.

5.1.2 Resultados e discussões

As superfícies de decisão nas bases de dados são apresentadas na Figura 13. Observando os resultados para a base de dados Ripley, Figura 13a e Figura 13b, percebe-se que o modelo SBMLM utilizou menos pontos de referências que o método RNMLM. Além disso, a complexidade da superfície de decisão do RNMLM é bem maior que a superfície gerada pelo SBMLM. Um outro ponto importante é a discrepância entre a quantidade de pontos de referência utilizados pelo RNMLM e o SBMLM. Enquanto o RNMLM utilizou 302 padrões de treinamento como conjunto de pontos de referência, o SBMLM utilizou apenas 8.

O comportamento apresentado pelas superfícies de decisão nas figuras 13c e 13d, também é similar ao caso da base de dados Ripley. O modelo RNMLM apresentou uma quantidade mais elevada de pontos de referência do que o modelo SBMLM, além disso observa-se na superfície gerada pelo modelo RNMLM um desvio na superfície de decisão sobre a região de sobreposição. Ou seja, a superfície gerada pelo modelo RNMLM ainda é mais complexa do que a superfície gerada pelo SBMLM. Em termos de esparsidade, enquanto o modelo RNMLM gera sua superfície de decisão utilizando 83 pontos de referência, o SBMLM utiliza 5.

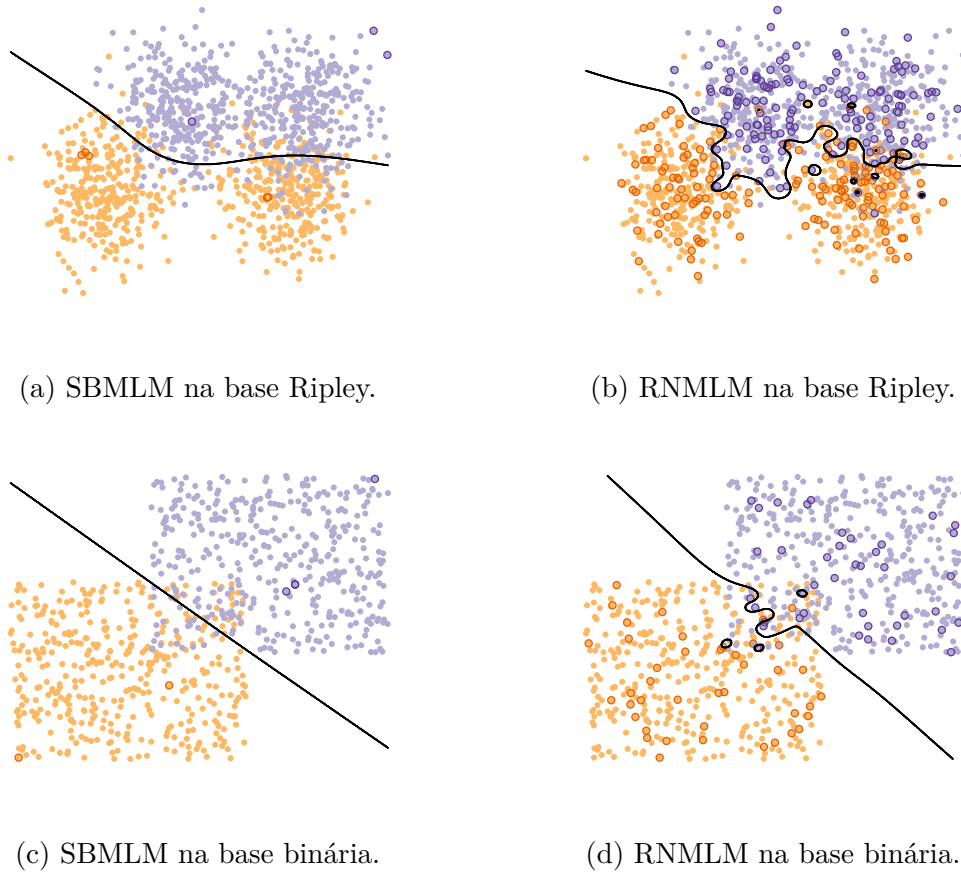
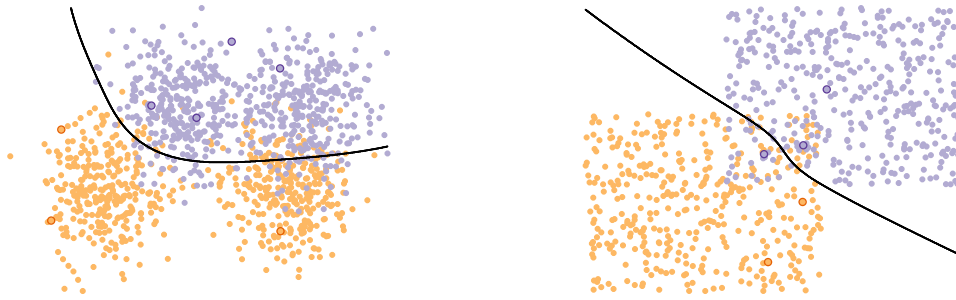


Figura 13 – Superfícies de decisão geradas pelo modelo SBMLM e pelo RNMLM em bases de dados artificiais.

Fonte – Próprio autor.

Empiricamente, os resultados da Figura 13 confirmam que a proposição do método SBMLM gera soluções menos complexas e mais esparsas. Mas para mostrar que a escolha da quantidade e escolha dos pontos corretos são critérios importantes na geração da superfície de decisão, na Figura 14, são apresentadas mais superfícies de decisão geradas pelo modelo RNMLM. Nesses resultados foram utilizadas quantidades de pontos de referência iguais às alcançadas pelo SBMLM, nos resultados da Figura 13. É possível observar que a complexidade foi reduzida em relação aos resultados do RNMLM na Figura 13. Porém, acontecem pequenas distorções nos resultados. Tal acontecimento se deve ao fato de que o método de mínimos quadrados, que é a abordagem adotada para resolver o RNMLM, não possui um critério para controle de complexidade. Isso é discutido em detalhes no Capítulo 3.



(a) RNMLM na base Ripley.

(b) RNMLM na base binária.

Figura 14 – Superfícies geradas pelo modelo RNMLM utilizando uma quantidade de pontos de referência com uma quantidade aproximada da obtida pelo método SBMLM.

Fonte – Próprio autor.

5.2 Comparativo do SBMLM com outros métodos da literatura

Nesta seção, são analisados resultados de acurácia e tempo de execução do SBMLM em relação à outros classificadores encontrados na literatura.

5.2.1 Configuração dos experimentos

Neste experimento, o modelo SBMLM é comparado com as máquinas de vetores suporte de mínimos quadrados (*Least Squares Support Vector Machines*, LSSVM) (SUYKENS; VANDEWALLE, 1999), Máquinas de Aprendizado Extremo (*Extreme Learning Machines*, ELM) (HUANG; ZHU; SIEW, 2006), rede RBF (*Radial Basis Function*, RBF) (PARK; SANDBERG, 1991), Máquinas de Vetores Suporte (*Support Vector Machines*, SVM) (STEINWART; CHRISTMANN, 2008) e as Máquinas de Vetores Relevantes (*Relevance Vector Machines*, RVM) (TIPPING, 2001). Todos os modelos foram aplicados aos problemas descritos na Tabela 1. Além disso, os resultados foram gerados a partir de 30 realizações independentes, de modo que em cada uma, as bases de dados são divididas em duas partições. A primeira delas representa o conjunto de treinamento e possui 80% das amostras do conjunto de dados total. O conjunto de teste é a segunda partição e possui 20% dos dados. Em cada rodada, todos os modelos são avaliados sobre a mesma partição da base de dados.

Para obtenção dos valores dos parâmetros dos modelos, um processo de validação cruzada com busca em grade foi realizada em cada rodada (ARLOT; CELISSE, 2010). Nesse processo, os intervalos de valores dos parâmetros são os mesmos para todas as bases de dados. Para a estimativa inicial do parâmetro β do modelo SBMLM, 20 valores linearmente crescentes de 2^{-7} a 2^3 foram utilizados. Esse intervalo foi definido através

de resultados experimentais. Para modelo RVM, a estimativa inicial para o parâmetro da abertura da verossimilhança também foi definido no intervalo $[2^{-7}, 2^3]$ com 20 valores linearmente crescentes. Além disso, o intervalo de busca do parâmetro de abertura da função de *kernel* Gaussiana, foi definido como 20 valores linearmente crescentes no intervalo $[2^{-15}, 2^3]$. Os intervalos de busca para os parâmetros C e abertura da função de *kernel* Gaussiana, foram definidos no trabalho de Hsu et al. (2003). São respectivamente 20 valores linearmente crescentes entre $[2^{-5}, 2^{15}]$ e 20 valores linearmente crescentes entre $[2^{-15}, 2^3]$. Os mesmos intervalos de busca dos parâmetros do SVM foram aplicados aos parâmetros γ e abertura do kernel no LSSVM. Para a rede RBF, a quantidade de funções Gaussianas aplicadas varia entre 5% à 100% com incrementos de 5%. O parâmetro de abertura das funções foi estimado em 20 valores linearmente crescentes no intervalo de 2^{-15} e 2^3 . Finalmente, o classificador ELM teve a quantidade de neurônios da camada oculta variando de 10 a 100 com incrementos de 5.

5.2.2 Resultados e discussões

Tabela 2 – Resultados experimentais do SBMLM e de outros métodos da literatura. São apresentados resultados de acurácia e desvio padrão sobre 30 realizações independentes. Os indicadores (✓) e (✗), representam equivalência ou não-equivalência dos resultados com os do método SBMLM. O teste estatístico de Friedman define os valores dos indicadores de equivalência.

<i>base</i>	SBMLM	RVM	SVM	RBF	ELM	LSSVM
RIP	92, 34 ± 0, 29	90, 00 ± 0, 00 ✗	92, 32 ± 0, 43 ✓	91, 93 ± 0, 36 ✗	91, 16 ± 0, 21 ✗	91, 54 ± 0, 29 ✗
HEP	74, 79 ± 6, 02	68, 54 ± 3, 06 ✗	64, 16 ± 4, 62 ✗	69, 16 ± 5, 42 ✓	58, 75 ± 6, 27 ✗	63, 95 ± 4, 24 ✗
BCW	96, 52 ± 1, 52	95, 55 ± 1, 65 ✗	96, 08 ± 1, 56 ✓	96, 15 ± 1, 52 ✓	96, 18 ± 1, 47 ✓	96, 18 ± 1, 44 ✓
AUS	83, 55 ± 2, 26	84, 71 ± 1, 17 ✓	84, 13 ± 1, 06 ✓	84, 54 ± 2, 50 ✗	85, 79 ± 0, 40 ✗	83, 98 ± 2, 59 ✓
HAB	74, 37 ± 7, 27	73, 27 ± 6, 16 ✓	74, 59 ± 2, 64 ✓	71, 53 ± 2, 63 ✓	78, 19 ± 3, 47 ✗	73, 55 ± 6, 42 ✓
VCP	84, 08 ± 1, 17	85, 69 ± 1, 17 ✓	86, 93 ± 0, 88 ✗	81, 02 ± 2, 06 ✓	87, 15 ± 0, 29 ✗	87, 20 ± 0, 58 ✗
HEA	82, 83 ± 2, 99	80, 12 ± 2, 37 ✗	80, 49 ± 1, 73 ✗	81, 85 ± 2, 72 ✓	79, 32 ± 5, 71 ✗	84, 62 ± 3, 04 ✗
SON	87, 85 ± 2, 52	85, 55 ± 2, 41 ✗	88, 01 ± 2, 37 ✓	90, 63 ± 0, 86 ✓	76, 19 ± 7, 26 ✗	90, 39 ± 0, 43 ✓
PID	74, 80 ± 0, 71	73, 57 ± 1, 06 ✗	77, 16 ± 0, 59 ✗	74, 82 ± 0, 82 ✓	73, 52 ± 0, 82 ✗	72, 85 ± 0, 71 ✗

Os resultados da comparação do modelo SBMLM com outras abordagens da literatura podem ser visualizados na Tabela 2. Observando os resultados na base RIP, nota-se que somente o classificador SVM atigiu resultados equivalentes ao modelo SBMLM e na base de dados HEP, somente a rede RBF atingiu a equivalencia. Nessas duas bases de dados, o modelo proposto obteve seus melhores resultados. Quanto à base de dados BCW, o SBMLM alcançou uma acurácia média maior que os outros modelos, mas em termos estatísticos, somente o RVM não obteve resultado equivalente. Na base de dados AUS, o método SBMLM obteve seu resultado mais baixo, mas estatisticamente, os modelos RVM, LSSVM e SVM apresentam resultados equivalentes. Relacionando o modelo RVM

com o SBMLM, observa-se que o RVM alcançou resultados equivalentes ao SBMLM somente nas bases de dados HAB, VCP e AUS. Nas demais bases de dados, o RVM obteve resultados inferiores ao SBMLM. Isso mostra que a aplicação do método SBL ao MLM obteve resultados melhores, na maioria das bases de dados utilizadas, que o método adotado no RVM.

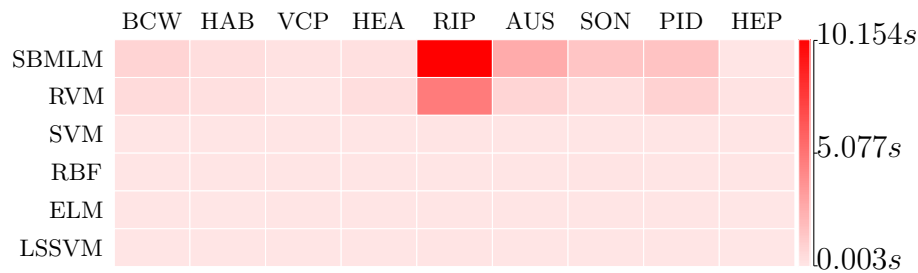


Figura 15 – Tempo de treinamento consumido pelos classificadores nas bases de dados utilizadas nesse experimento.

Fonte – Próprio autor.

Na Figura 15, são apresentadas as informações de tempo médio de treinamento do método SBMLM e dos outros classificadores da literatura nesse experimento. A maioria dos métodos consumiram tempo médio de aproximadamente três milésimos de segundo. Porém, os métodos SBMLM e RVM, na base de dados RIP, obtiveram tempos mais discrepantes. O método SBMLM obteve um maior consumo de tempo dentre todos os outros modelos, na base de dados RIP. Uma justificativa disso é o elevado custo computacional do método M-SBL e o tamanho da base de dados RIP em relação às outras.

5.3 Comparativo do SBMLM com outras versões do MLM

Nesta seção, serão analisadas as melhorias do método SBMLM em relação às outras versões do método MLM. Nessa comparação, será utilizada uma versão do MLM com seleção de pontos de referência aleatória, chamada de RNMLM. Além disso, será utilizada uma versão em que todos os pontos de treinamento são considerados pontos de referência, neste experimento é chamada de FLMLM.

5.3.1 Configuração dos experimentos

Neste experimento, cada base de dados foi particionada de modo que 80% das amostras são utilizadas para treinamento e 20% para testes. Os resultados são calculados sobre 30 realizações independentes. Em cada uma delas, os três classificadores são submetidos à mesma partição da base de dados.

A seleção dos parâmetros dos modelos utilizados neste experimento, foi feita através de validação cruzada com busca em grade. Para a estimativa inicial do parâmetro β do

modelo SBMLM, 20 valores linearmente crescentes de 2^{-7} a 2^3 foram utilizados. Esse intervalo foi definido através de resultados experimentais. O modelo FLMLM, não possui parâmetros a serem otimizados, já que todos os padrões de treinamento são selecionados como pontos de referência. Na versão do MLM com seleção aleatória de pontos de referência, RNMLM, a quantidade de pontos de referência varia de 5% a 50% do número de padrões da base de dados de treinamento, com incrementos de 5%.

5.3.2 Resultados e discussões

Na Tabela 3, são apresentados os resultados de acurácia dos modelos SBMLM, RNMLM e FLMLM. Analisando essa tabela, nota-se que o modelo FLMLM obteve os resultados mais baixos, já que em seis bases de dados, os resultados foram inferiores ao SBMLM e ao RNMLM. Por outro lado, o RNMLM atingiu resultados estatisticamente equivalentes em quase todas as bases de dados testadas. Uma informação importante disso, é que o SBMLM manteve os resultados atingidos pelo RNMLM e melhorou em relação à versão FLMLM.

Tabela 3 – Resultados comparativos entre o método SBMLM com outras versões do MLM. Os métodos RNMLM e FLMLM, representam versões do MLM com escolha aleatória de pontos de referência e uma versão em que toda a base de dados é utilizada como pontos de referência, respectivamente. O teste estatístico de Friedman é utilizado para comparar os resultados com o modelo SBMLM, de modo que (✓) representa equivalência e (✗) representa não-equivalência.

<i>base</i>	SBMLM	RNMLM	FLMLM
VCP	85, 32 ± 4, 00	84, 51 ± 3, 45 ✓	85, 59 ± 3, 46 ✓
RIP	90, 90 ± 1, 53	90, 50 ± 1, 41 ✓	89, 61 ± 1, 56 ✗
AUS	86, 15 ± 2, 85	85, 21 ± 2, 98 ✗	83, 21 ± 3, 06 ✗
SON	87, 46 ± 5, 33	86, 19 ± 5, 46 ✓	87, 06 ± 5, 78 ✓
BCW	96, 49 ± 1, 27	96, 71 ± 1, 24 ✓	96, 49 ± 1, 24 ✓
HAB	73, 98 ± 3, 59	75, 13 ± 3, 73 ✓	67, 97 ± 4, 78 ✗
HEA	82, 77 ± 4, 81	80, 55 ± 7, 94 ✓	79, 13 ± 4, 48 ✗
PID	77, 05 ± 3, 28	76, 71 ± 3, 73 ✓	74, 45 ± 3, 34 ✗
HEP	74, 37 ± 12, 53	70, 62 ± 11, 62 ✓	68, 95 ± 12, 22 ✗

As informações no tocante à quantidade de pontos de referência são apresentados na figuras 16 e 17. Em todas as bases de dados, o SBMLM gerou modelos mais esparsos que o RNMLM. As quantidades de pontos de referência mais discrepantes entre o SBMLM e o RNMLM ocorrem nas bases de dados VCP, RIP e AUS. Nessas bases, o método SBMLM atingiu seus resultados utilizando menos de 30 pontos de referência, enquanto o RNMLM utilizava em torno de 90. A base de dados SON os dois modelos utilizaram uma

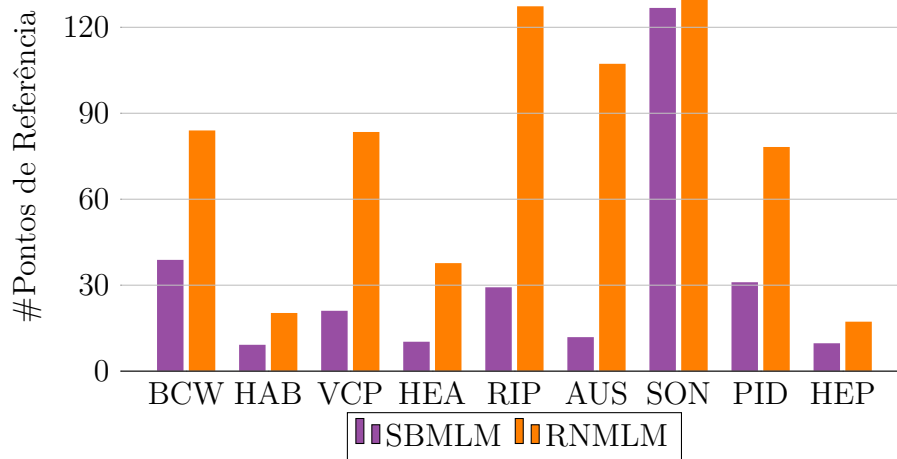


Figura 16 – Quantidade de pontos de referência média obtida pelos modelos SBMLM e RNMLM após o processo de treinamento. Esses valores foram obtidos sobre 30 realizações independentes e utilizando 80% da quantidade de amostras das bases de dados para treinamento.

Fonte – Próprio autor.

quantidade de pontos de referência aproximada. Nesse caso, os dois modelos obtiveram aproximadamente 120 padrões de treinamento como pontos de referência.

Uma segunda forma de ler os resultados em termos de esparsidade, é através da Tabela 17. Esses resultados apontam que em relação à quantidade total de amostras de treinamento, os modelos SBMLM e o RNMLM atingiram reduções sempre próximas à 80%, com exceção do que ocorreu na base de dados SON, em que os dois modelos atingiram reduções próximas à 40%. Essa informação é interessante, pois mostra que embora diferença entre a quantidade de pontos de referência seja elevada, em termos de redução em relação à quantidade total de amostras da base de dados, é bem pequena. Consequentemente, o modelo SBMLM atingiu reduções mais elevadas que o RNMLM em todos os cenários.

Analisando os resultados de acurácia, apresentados na Tabela 3, e esparsidade, descritos nas figuras 16 e 17, em conjunto, nota-se que o modelo SBMLM alcançou resultados equivalentes ou superiores em termos de taxa de acerto, utilizando uma quantidade menor de pontos de referência. Observando o exemplo da base dados RIP, em termos de taxa de acerto, os resultados são equivalentes entre o modelo SBMLM e RNMLM. Entretanto, enquanto o RNMLM utiliza mais de 120 pontos de referência, o SBMLM utiliza menos de 30. Esse perfil se repete na maioria das bases de dados, com exceção da SON, em que as taxas de acerto do SBMLM e RNMLM são equivalentes e as quantidades de pontos de referência utilizadas pelos modelos também são aproximadas.

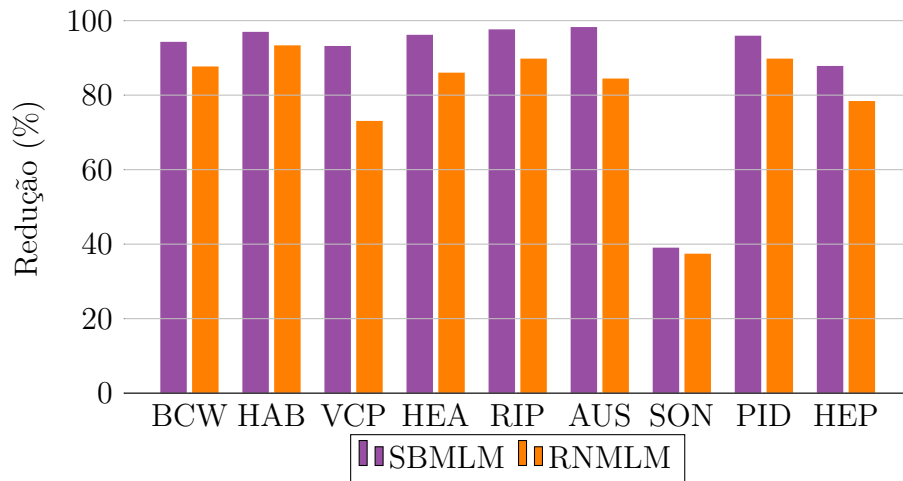


Figura 17 – Redução média alcançada pelos modelos SBMLM e RNMLM após o processo de treinamento. Esses resultados são obtidos sobre 30 realizações independentes e utilizando 80% das amostras das bases de dados para treinamento.

Fonte – Próprio autor.

5.4 Análise da variação dos parâmetros do SBMLM

Neste experimento, é analisado o comportamento do SBMLM com diferentes valores iniciais do parâmetro β . Serão analisados critérios de esparsidade, taxa de erro e decaimento do número de pontos de referência.

5.4.1 Configuração dos experimentos

Este experimento é dividido em duas etapas. Na primeira, serão analisados os valores de taxa de erro e número de pontos de referência para valores de β , variando no intervalo $[2^{-7}, 2^3]$. Cada taxa de erro e quantidade de pontos de referência, serão obtidos através da média de 30 realizações independentes. Serão aplicadas duas bases de dados, uma artificial e outra real. A base real utilizada é a VCP e a artificial consiste da base binária utilizada nos experimentos descritos nas Figura 13c e Figura 13d. A segunda etapa deste experimento, consiste de uma análise da velocidade de decaimento do número de pontos de referência no decorrer das iterações do algoritmo M-SBL. São apresentados decaimentos dos pontos de referência para valores de β iguais a $\{2^{-7}, 2^0, 2^3\}$. As bases de dados utilizadas nessa análise são a RIP e VCP.

5.4.2 Resultados e discussões

Na Figura 18, são apresentados os gráficos de variação da taxa de erro e do número de pontos de referência. Analisando a taxa de erro, a base artificial apresentou uma baixa variação com o aumento do parâmetro β . Por outro lado, na base de dados real, a taxa de erro aumenta consideravelmente com o crescimento do parâmetro β . A base de dados

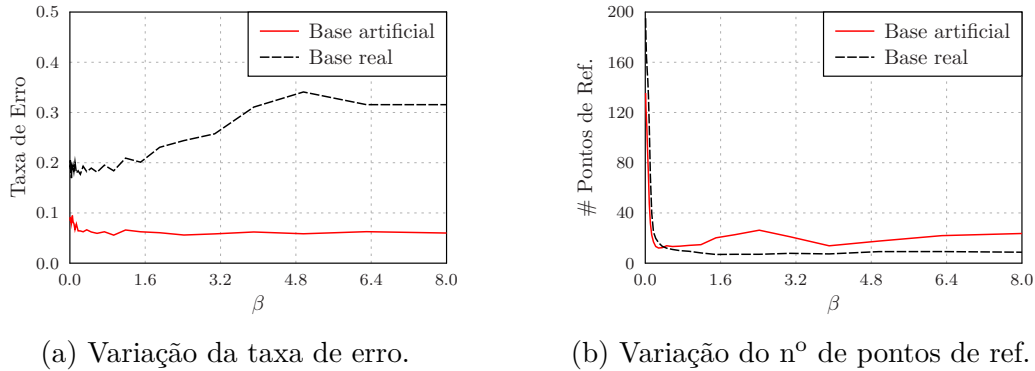


Figura 18 – Variação da taxa de erro e número de pontos de referência em função do parâmetro β . Cada valor da taxa de erro e número de pontos de referência é obtido pela média de 30 realizações independentes.

Fonte – Próprio autor.

artificial consiste de um problema bem mais simples que a base real, por isso, não sofre tanto com a variação do parâmetro. A variação do número de pontos de referência entre as duas bases é muito próxima, mas com uma maior discrepância com valores de β entre 1.6 e 3.2.

No tocante aos resultados apresentados na Figura 19, nota-se que nas duas bases de dados, há uma tendência à uma queda de pontos de referência mais rápida quando se tem valores de β maiores. além disso, para valores pequenos de β , o algoritmo não encerra com uma quantidade pequena de pontos de referência, como pode ser observado nos casos em que $\beta = 2^{-7}$.

Fazendo uma relação entre os resultados da Figura 18 e Figura 19, nota-se que para valores pequenos de β , o erro tende a diminuir e a poda de pontos de referência é mais suave. Por outro lado, para valores grandes de β , o decaimento do número de pontos de referência é mais acentuado, o modelo final é mais esparsos, mas possui uma maior taxa de erro. Ou seja, o parâmetro β funciona como um compromisso entre qualidade do modelo e esparsidade.

5.5 Conclusão do capítulo

Nesse Capítulo, foram apresentados os experimentos e os resultados obtidos com o modelo SBMLM. Os experimentos foram divididos em quatro pontos de modo a analisar diversas características do modelo SBMLM. Inicialmente, é feita uma análise empírica da superfície de decisão gerada pelo modelo SBMLM e pelo MLM. Na segunda análise, o MLM é comparado com outros classificadores da literatura em termos de acurácia. Para analisar esparsidade e acurácia, também foi feita uma comparação entre o SBMLM e abordagens

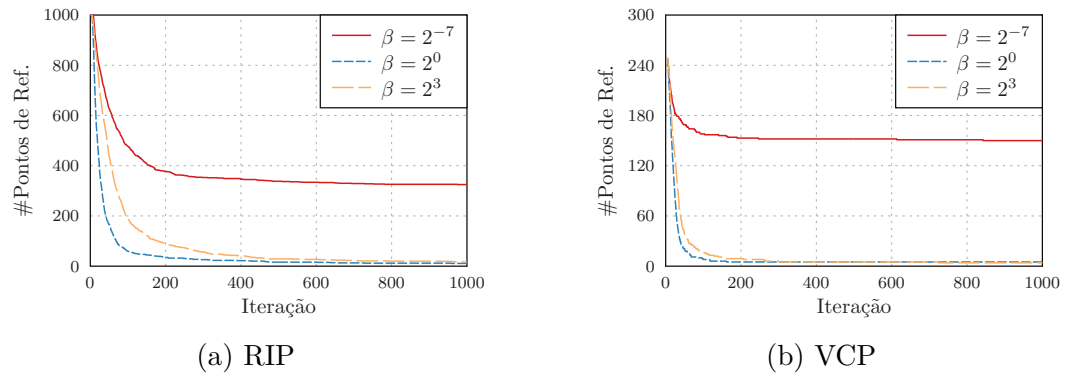


Figura 19 – Decaimento do número de pontos de referência durante uma iteração do algoritmo M-SBL com alguns valores do parâmetro β .

Fonte – Próprio autor.

do próprio MLM. Por fim, são analisadas as implicações da variação do parâmetro β do modelo SBMLM. Todos os experimentos foram realizados sobre as bases de dados do UCI Machine Learning e sobre a base artificial Ripley.

6 Conclusão e Trabalhos Futuros

O objetivo deste capítulo é a apresentação das conclusões acerca do método proposto no presente trabalho. Além disso, serão elicitadas as principais contribuições da proposta. Por fim, serão levantados os principais problemas do método proposto e serão sugeridas metodologias para resolução desses problemas como trabalhos futuros.

6.1 Conclusão

O principal ponto dessa dissertação é tratar duas limitações do modelo MLM. A primeira delas, é a adoção de um critério para seleção de pontos de referência e a segunda é uma metodologia para solução do sistema linear de múltiplas respostas que adote algum critério de controle complexidade. Para contornar isso, é proposto o método SBMLM, que consiste de uma abordagem Bayesiana para treinamento do modelo MLM. Uma das principais características do método proposto, é a solução esparsa do sistema linear de múltiplas respostas, que define o mapeamento entre as distâncias de entrada e de saída no treinamento do modelo MLM. Essa solução é obtida pelo método M-SBL, que é baseado no *Evidence Framework* e é capaz de encontrar uma solução para o sistema linear de múltiplas respostas juntamente dos pontos de referência associados. Além disso, o M-SBL incorpora um critério de complexidade chamada de navalha de Occam para obtenção de suas soluções.

O método proposto foi submetido à uma série de quatro tipos de experimentos para avaliação de diversas características. Nesses experimentos foram aplicadas bases de dados do *UCI Machine Learning* e a base de dados artificial *Ripley*. Com base na análise do primeiro experimento, que avaliava de forma empírica a complexidade do modelo e esparsidade, conclui-se que o SBMLM gera superfícies menos complexas que a versão original do MLM, além de obter soluções mais esparsas. O segundo grupo de experimentos permite concluir que o modelo atingiu taxas de acertos próximas de outros classificadores da literatura, apesar do elevado tempo de treinamento. Em comparação com versões do MLM, o SBMLM atingiu resultados equivalentes ou superiores em termos de taxas de acerto e com uma esparsidade superior em todas as bases de dados. Por fim, no último experimento foram observados resultados do modelo com a variação do parâmetro de abertura da distribuição de verossimilhança. Com isso, notou-se que esse parâmetro é um compromisso entre qualidade do aprendizado e esparsidade.

Em resumo, o método proposto atingiu os resultados esperados em termos de controle de complexidade, esparsidade e taxa de acerto.

6.2 Contribuições desse trabalho

As contribuições desse trabalho estão relacionadas principalmente ao modelo MLM, já que é apresentada uma nova abordagem de atacar o problema de seleção de pontos de referência em conjunto com a resolução do sistema linear. Além disso, os resultados obtidos pelo método proposto em termos de esparsidade favorece, por exemplo, a utilização do modelo MLM por dispositivos embarcados, onde existe uma carência de memória.

Uma segunda contribuição é a própria revisão bibliográfica acerca de métodos para resolução de sistemas lineares de resposta única e de múltiplas respostas. Isso pode guiar várias metodologias de redução de modelos de aprendizagem baseados na resolução de sistemas lineares.

6.3 Trabalhos Futuros

Como primeiro futuro, é sugerida a aplicação do método SBMLM a bases de dados de regressão. Essa sugestão parte da própria formulação do método M-SBL utilizando distribuição normal como verossimilhança dos dados.

A adoção de uma distribuição de verossimilhança normal no modelo SBMLM foi feita para se obter uma solução analítica para a evidência e para a própria distribuição a posteriori. Porém, isso é uma limitação do método proposto, já que outras distribuições possuem suporte mais adequado para se trabalhar com dados discretos, que é o caso da matriz de distâncias de saída em problemas de classificação. Uma sugestão de trabalho futuro, é a adoção de uma distribuição de Bernoulli como verossimilhança e obtenção da distribuição a posteriori através de métodos de aproximação, como o de Laplace (AZEVEDO-FILHO; SHACHTER, 1994).

Um outro problema que pode ocorrer é a situação em que o sistema linear de múltiplas respostas é indeterminado. Nesse caso, as matrizes de distâncias possuem mais colunas do que linhas e o sistema pode ter infinitas soluções. Uma proposta para resolução desse tipo de problema, é a utilização do método de resolução de sistemas lineares indeterminados FOCUSS (GORODNITSKY; RAO, 1997), pois ele também obtém soluções esparsas e possui um controle de complexidade.

Referências

ADAMS, M. R.; GUILLEMIN, V. *Measure theory and probability*. 1. ed. [S.l.]: Birkhäuser Basel, 1996. ISBN 978-1-4612-0779-5. Citado na página 40.

ARLOT, S.; CELISSE, A. A survey of cross-validation procedures for model selection. *Statist. Surv.*, The American Statistical Association, the Bernoulli Society, the Institute of Mathematical Statistics, and the Statistical Society of Canada, v. 4, p. 40–79, 2010. Citado na página 86.

AZEVEDO-FILHO, A.; SHACHTER, R. D. Laplace's method approximations for probabilistic inference in belief networks with continuous variables. In: MORGAN KAUFMANN PUBLISHERS INC. *Proceedings of the Tenth international conference on Uncertainty in artificial intelligence*. [S.l.], 1994. p. 28–36. Citado na página 96.

BACHRACH, J.; TAYLOR, C. Localization in sensor networks. *Handbook of sensor networks: Algorithms and Architectures*, John Wiley & Sons Hoboken, NJ, USA, v. 1, 2005. Citado na página 32.

BARRON, A. R. Entropy and the central limit theorem. *The Annals of Probability*, Institute of Mathematical Statistics, v. 14, n. 1, p. 336–342, 1986. ISSN 00911798. Disponível em: <<http://www.jstor.org/stable/2244098>>. Citado na página 41.

BERGER, J. O. *Statistical decision theory and Bayesian analysis*. [S.l.]: Springer Science & Business Media, 2013. Citado 2 vezes nas páginas 60 e 68.

BISHOP, C. *Pattern Recognition and Machine Learning*. Springer, 2006. (Information Science and Statistics). ISBN 9780387310732. Disponível em: <<https://books.google.com.br/books?id=kTNoQgAACAAJ>>. Citado 4 vezes nas páginas 37, 53, 56 e 62.

CHEN, S. S.; DONOHO, D. L.; SAUNDERS, M. A. Atomic decomposition by basis pursuit. *SIAM review*, SIAM, v. 43, n. 1, p. 129–159, 2001. Citado 3 vezes nas páginas 32, 60 e 67.

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.n.], 2014. p. 42–48. Disponível em: <<https://doi.org/10.1109/CIES.2014.7011829>>. Citado na página 31.

COTTER, S. F. et al. Sparse solutions to linear inverse problems with multiple measurement vectors. *IEEE Transactions on Signal Processing*, IEEE, v. 53, n. 7, p. 2477–2488, 2005. Citado 2 vezes nas páginas 67 e 68.

DEMPSTER, A. P.; LAIRD, N. M.; RUBIN, D. B. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, JSTOR, p. 1–38, 1977. Citado na página 113.

DEMPSTER, A. P.; RUBIN, D. B.; TSUTAKAWA, R. K. Estimation in covariance components models. *Journal of the American Statistical Association*, Taylor & Francis, v. 76, n. 374, p. 341–353, 1981. Citado na página 70.

DIAS, M. L. et al. Uma abordagem de selecao de pontos de referência para maquinas de aprendizado mínimo via algoritmos genéticos. Porto Alegre, Brasil, 2017. Citado na página 31.

DONOHU, D. L.; ELAD, M. Optimally sparse representation in general (nonorthogonal) dictionaries via l1 minimization. *Proceedings of the National Academy of Sciences*, National Acad Sciences, v. 100, n. 5, p. 2197–2202, 2003. Citado 2 vezes nas páginas 58 e 60.

ELAD, M. *Sparse and Redundant Representations - From Theory to Applications in Signal and Image Processing*. Springer, 2010. ISBN 978-1-4419-7010-7. Disponível em: <<https://doi.org/10.1007/978-1-4419-7011-4>>. Citado na página 58.

FLORÊNCIO, J. A. V.; DIAS, M. L. D.; ROCHA NETO, A. R. da. Um novo método baseado em protótipos para seleção de pontos de referência em máquinas de aprendizado mínimo. Niterói, Brasil, 2017. Citado na página 31.

FUCHS, J. J. On sparse representations in arbitrary redundant bases. *IEEE Transactions on Information Theory*, v. 50, n. 6, p. 1341–1344, June 2004. ISSN 0018-9448. Citado na página 58.

GORODNITSKY, I. F.; RAO, B. D. Sparse signal reconstruction from limited data using focuss: a re-weighted minimum norm algorithm. *IEEE Transactions on Signal Processing*, v. 45, n. 3, p. 600–616, Mar 1997. ISSN 1053-587X. Citado na página 96.

GRANT, I. H. Recursive least squares. *Teaching Statistics*, Wiley Online Library, v. 9, n. 1, p. 15–18, 1987. Citado na página 74.

GRIBONVAL, R.; NIELSEN, M. Sparse representations in unions of bases. *IEEE Transactions on Information Theory*, v. 49, n. 12, p. 3320–3325, Dec 2003. ISSN 0018-9448. Citado na página 58.

HARVILLE, D. A. Maximum likelihood approaches to variance component estimation and to related problems. *Journal of the American Statistical Association*, Taylor & Francis, v. 72, n. 358, p. 320–338, 1977. Citado na página 70.

HEARST, M. A. et al. Support vector machines. *IEEE Intelligent Systems and their Applications*, v. 13, n. 4, p. 18–28, July 1998. ISSN 1094-7167. Citado na página 59.

HSU, C.-W. et al. A practical guide to support vector classification. Taipei, Taiwan, 2003. Citado na página 87.

HUANG, G.-B.; ZHU, Q.-Y.; SIEW, C.-K. Extreme learning machine: Theory and applications. *Neurocomputing*, v. 70, n. 1, p. 489 – 501, 2006. ISSN 0925-2312. Neural Networks. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0925231206000385>>. Citado 2 vezes nas páginas 59 e 86.

JAMIL, T.; BRAAK, C. J. ter. Selection properties of type ii maximum likelihood (empirical bayes) in linear models with individual variance components for predictors. *Pattern Recognition Letters*, v. 33, n. 9, p. 1205 – 1212, 2012. ISSN 0167-8655. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0167865512000074>>. Citado na página 63.

- LEAHY, R. M.; JEFFS, B. D. On the design of maximally sparse beamforming arrays. *IEEE Transactions on antennas and propagation*, IEEE, v. 39, n. 8, p. 1178–1187, 1991. Citado na página 61.
- LICHMAN, M. *UCI Machine Learning Repository*. 2013. Citado 2 vezes nas páginas 83 e 109.
- MACKAY, D. J. Bayesian interpolation. *Neural computation*, MIT Press, v. 4, n. 3, p. 415–447, 1992. Citado 6 vezes nas páginas 32, 33, 61, 67, 70 e 71.
- MACKAY, D. J. *Bayesian methods for adaptive models*. Tese (Doutorado) — California Institute of Technology, 1992. Citado 2 vezes nas páginas 32 e 33.
- MACKAY, D. J. Comparison of approximate methods for handling hyperparameters. *Neural computation*, MIT Press, v. 11, n. 5, p. 1035–1068, 1999. Citado na página 63.
- MALIOUTOV, D.; CETIN, M.; WILLSKY, A. S. A sparse signal reconstruction perspective for source localization with sensor arrays. *IEEE transactions on signal processing*, IEEE, v. 53, n. 8, p. 3010–3022, 2005. Citado na página 68.
- 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 31.
- MELO, D. B. d. *Algoritmos de aprendizagem para aproximação da cinemática inversa de robôs manipuladores: um estudo comparativo*. Tese (Doutorado), 2015. Citado na página 31.
- MESQUITA, D. P.; GOMES, J. P.; JUNIOR, A. H. S. Ensemble of efficient minimal learning machines for classification and regression. *Neural Processing Letters*, Springer, p. 1–16, 2017. Citado na página 76.
- MOON, T. K. The expectation-maximization algorithm. *IEEE Signal Processing Magazine*, v. 13, n. 6, p. 47–60, Nov 1996. ISSN 1053-5888. Citado na página 114.
- MURPHY, K. P. *Machine learning: a probabilistic perspective*. [S.l.]: MIT press, 2012. Citado 4 vezes nas páginas 37, 43, 60 e 61.
- NATARAJAN, B. K. Sparse approximate solutions to linear systems. *SIAM Journal on Computing*, v. 24, n. 2, p. 227–234, 1995. Disponível em: <<https://doi.org/10.1137/S0097539792240406>>. Citado na página 58.
- NEAL, R. M. *Bayesian Learning for Neural Networks*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1996. ISBN 0387947248. Citado na página 57.
- 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 75.

- PARK, J.; SANDBERG, I. W. Universal approximation using radial-basis-function networks. *Neural Computation*, v. 3, n. 2, p. 246–257, 1991. Disponível em: <<https://doi.org/10.1162/neco.1991.3.2.246>>. Citado na página 86.
- RAO, B. D.; ENGAN, K.; COTTER, S. F. Diversity measure minimization based method for computing sparse solutions to linear inverse problems with multiple measurement vectors. In: *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2004, Montreal, Quebec, Canada, May 17-21, 2004*. [s.n.], 2004. p. 369–372. Disponível em: <<https://doi.org/10.1109/ICASSP.2004.1326271>>. Citado na página 68.
- RAO, B. D. et al. Subset selection in noise based on diversity measure minimization. *IEEE transactions on Signal processing*, IEEE, v. 51, n. 3, p. 760–770, 2003. Citado 3 vezes nas páginas 32, 57 e 60.
- RAO, B. D.; KREUTZ-DELGADO, K. Basis selection in the presence of noise. In: *Conference Record of Thirty-Second Asilomar Conference on Signals, Systems and Computers (Cat. No.98CH36284)*. [S.l.: s.n.], 1998. v. 1, p. 752–756 vol.1. ISSN 1058-6393. Citado na página 68.
- RAO, B. D.; KREUTZ-DELGADO, K. An affine scaling methodology for best basis selection. *IEEE Transactions on signal processing*, IEEE, v. 47, n. 1, p. 187–200, 1999. Citado na página 60.
- ROCHA NETO, A. R.; BARRETO, G. A. Opposite Maps: Vector Quantization Algorithms for Building Reduced-Set SVM and LSSVM Classifiers. *NEURAL PROCESSING LETTERS*, 37, n. 1, SI, p. 3–19, FEB 2013. ISSN 1370-4621. Citado na página 107.
- SOUSA, L. S.; DIAS, M. L. D.; ROCHA NETO, A. R. Reduced-set relevance vector machines via gaussian mixture models. In: *4th IEEE Latin American Conference on Computational Intelligence LA-CCI*. Guadalajara, México: [s.n.], 2017. Citado 2 vezes nas páginas 34 e 117.
- SOUSA, L. S. de; ROCHA NETO, A. R. da. Gaussian opposite maps for reduced-set relevance vector machines. In: _____. *Advances in Computational Intelligence: 14th International Work-Conference on Artificial Neural Networks, IWANN 2017, Cadiz, Spain, June 14-16, 2017, Proceedings, Part I*. Cham: Springer International Publishing, 2017. p. 458–468. ISBN 978-3-319-59153-7. Disponível em: <https://doi.org/10.1007/978-3-319-59153-7_40>. Citado 2 vezes nas páginas 33 e 107.
- 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 31, 73 e 75.
- STEINWART, I.; CHRISTMANN, A. *Support vector machines*. New York, USA: Springer, 2008. ISBN 978-0-387-77242-4. Citado na página 86.
- SUYKENS, J.; VANDEWALLE, J. Least squares support vector machine classifiers. *Neural Processing Letters*, v. 9, n. 3, p. 293–300, Jun 1999. ISSN 1573-773X. Disponível em: <<https://doi.org/10.1023/A:1018628609742>>. Citado 2 vezes nas páginas 59 e 86.

THEODORIDIS, S.; KOUTROUMBAS, K. *Pattern Recognition*. Elsevier Science, 2008. ISBN 9780080949123. Disponível em: <<https://books.google.it/books?id=QgD-3Tcj8DkC>>. Citado 2 vezes nas páginas 37 e 41.

TIBSHIRANI, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, JSTOR, p. 267–288, 1996. Citado 2 vezes nas páginas 60 e 67.

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 6 vezes nas páginas 33, 61, 62, 71, 86 e 109.

TIPPING, M. E.; FAUL, A. C. Fast marginal likelihood maximisation for sparse bayesian models. In: *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics, AISTATS 2003, Key West, Florida, USA, January 3-6, 2003*. [s.n.], 2003. Disponível em: <<http://research.microsoft.com/en-us/um/cambridge/events/aistats2003/proceedings/123.pdf>>. Citado na página 33.

TROPP, J. A. Greed is good: algorithmic results for sparse approximation. *IEEE Transactions on Information Theory*, v. 50, n. 10, p. 2231–2242, Oct 2004. ISSN 0018-9448. Citado 2 vezes nas páginas 32 e 58.

TROPP, J. A. Just relax: Convex programming methods for identifying sparse signals in noise. *IEEE transactions on information theory*, IEEE, v. 52, n. 3, p. 1030–1051, 2006. Citado na página 60.

TROPP, J. A.; GILBERT, A. C. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on Information Theory*, v. 53, n. 12, p. 4655–4666, Dec 2007. ISSN 0018-9448. Citado na página 59.

WILLIAMS, C. K.; RASMUSSEN, C. E. Gaussian processes for machine learning. *the MIT Press*, v. 2, n. 3, p. 4, 2006. Citado na página 53.

WIPF, D. P. Bayesian methods for finding sparse representations. In: . [S.l.]: University of California, San Diego, 2006. Citado 10 vezes nas páginas 32, 53, 57, 58, 59, 60, 62, 66, 67 e 69.

WIPF, D. P.; NAGARAJAN, S. S. A new view of automatic relevance determination. In: PLATT, J. C. et al. (Ed.). *Advances in Neural Information Processing Systems 20*. Curran Associates, Inc., 2008. p. 1625–1632. Disponível em: <<http://papers.nips.cc/paper/3372-a-new-view-of-automatic-relevance-determination.pdf>>. Citado na página 61.

WIPF, D. P.; RAO, B. D. An empirical bayesian strategy for solving the simultaneous sparse approximation problem. *IEEE Trans. Signal Processing*, v. 55, n. 7-2, p. 3704–3716, 2007. Disponível em: <<https://doi.org/10.1109/TSP.2007.894265>>. Citado 3 vezes nas páginas 32, 33 e 68.

Apêndices

APÊNDICE A – Minimização da probabilidade de erro

Para iniciar a prova de que o classificador Bayesiano minimiza a probabilidade de erro de classificação, é necessário partir da premissa de que se o ponto extremo $\mathbf{x}_0$ for movido para qualquer local, a região de interseção entre as distribuições aumenta. Em outras palavras, a probabilidade de um padrão $\mathbf{x}$ ser classificado de forma errada cresce. A prova formal é apresentada a seguir.

Prova: Seja R_1 a região do espaço de características em que se decide que padrões pertencem à classe $\mathcal{C}_1$ e R_2 a região em que se decide que padrões pertencem à classe $\mathcal{C}_2$. Um erro ocorre quando um padrão que pertence à classe $\mathcal{C}_1$ é dito pertencer à classe $\mathcal{C}_2$ ou quando um padrão que pertence à classe $\mathcal{C}_2$ é classificado como pertencente à classe $\mathcal{C}_1$. Sabendo disso, a probabilidade de um erro ocorrer pode ser expressa através de probabilidade conjunta, a saber

$$p_e = p(\mathbf{x} \in R_2, \mathcal{C}_1) + p(\mathbf{x} \in R_1, \mathcal{C}_2). \quad (\text{A.1})$$

Seguindo a Equação (2.6), pode-se reescrever a probabilidade de erro p_e em função da probabilidade condicional, conforme descrito a seguir

$$\begin{aligned} p_e &= p(\mathbf{x} \in R_2 | \mathcal{C}_1) p(\mathcal{C}_1) + p(\mathbf{x} \in R_1 | \mathcal{C}_2) p(\mathcal{C}_2) \\ &= p(\mathcal{C}_1) \int_{R_2} p(\mathbf{x} | \mathcal{C}_1) d\mathbf{x} + p(\mathcal{C}_2) \int_{R_1} p(\mathbf{x} | \mathcal{C}_2) d\mathbf{x}. \end{aligned} \quad (\text{A.2})$$

Uma outra maneira de escrever a probabilidade de erro, é através da própria regra de Bayes. Assim,

$$p_e = \int_{R_2} p(\mathcal{C}_1 | \mathbf{x}) p(\mathbf{x}) d\mathbf{x} + \int_{R_1} p(\mathcal{C}_2 | \mathbf{x}) p(\mathbf{x}) d\mathbf{x}. \quad (\text{A.3})$$

É possível observar que a probabilidade de erro é minimizada quando as partições R_1 e R_2 são selecionadas de modo que:

$$\begin{aligned} R_1 &: p(\mathcal{C}_1 | \mathbf{x}) > p(\mathcal{C}_2 | \mathbf{x}) \\ R_2 &: p(\mathcal{C}_2 | \mathbf{x}) > p(\mathcal{C}_1 | \mathbf{x}). \end{aligned} \quad (\text{A.4})$$

Pela definição de função de densidade de probabilidade, para analisar a probabilidade total de uma determinada classe ocorrer, é necessário verificar todo o espaço de características. Tal espaço é coberto pela união das regiões R_1 e R_2 . Sabendo disso, pode-se escrever a probabilidade de $\mathcal{C}_1$ ocorrer como

$$\int_{R_1} p(\mathcal{C}_1 | \mathbf{x}) p(\mathbf{x}) d\mathbf{x} + \int_{R_2} p(\mathcal{C}_1 | \mathbf{x}) p(\mathbf{x}) d\mathbf{x} = p(\mathcal{C}_1). \quad (\text{A.5})$$

Combinando as Equações (A.3) e (A.5), é possível obter

$$\begin{aligned}
 p_e &= p(\mathcal{C}_1) - \int_{R_1} p(\mathcal{C}_1|\mathbf{x}) p(\mathbf{x}) d\mathbf{x} + \int_{R_1} p(\mathcal{C}_2|\mathbf{x}) p(\mathbf{x}) d\mathbf{x}, \\
 &= p(\mathcal{C}_1) - \int_{R_1} [p(\mathcal{C}_1|\mathbf{x}) p(\mathbf{x}) - p(\mathcal{C}_2|\mathbf{x}) p(\mathbf{x})] d\mathbf{x}, \\
 &= p(\mathcal{C}_1) - \int_{R_1} p(\mathbf{x}) [p(\mathcal{C}_1|\mathbf{x}) - p(\mathcal{C}_2|\mathbf{x})] d\mathbf{x}. \tag{A.6}
 \end{aligned}$$

É possível notar, a partir da Equação A.6, que p_e é minimizado quando a região R_1 tem $p(\mathcal{C}_1|\mathbf{x}) > p(\mathcal{C}_2|\mathbf{x})$. Por outro lado, para a região R_2 o oposto é verdadeiro. Logo, o classificador Bayesiano minimiza a probabilidade de erro p_e . $\square$

APÊNDICE B – Mapas opostos para redução de conjuntos em Máquinas de Vetores Relevantes

Neste apêndice são apresentados os resultados obtidos com o método Mapas Opostos Gaussianos para conjuntos reduzidos em Máquinas de Vetores Relevantes (*Gaussian Opposite Maps for Reduced-set Relevance Vector Machine*, GOM-RVM), proposto no trabalho Sousa e Rocha Neto (2017). Esse método consiste de uma abordagem para redução de padrões de treinamento antes da aplicação do SBL no treinamento do RVM. Durante o restante deste apêndice, é apresentado o método de Mapas Opostos Gaussianos (*Gaussian Opposite Maps*, GOM) e como tal abordagem foi agregada ao treinamento do modelo RVM. Em seguida são apresentados os resultados obtidos com o método GOM-RVM.

B.1 Mapas Opostos Gaussianos

O método de Mapas Opostos (*Opposite Maps*, OM) é uma técnica baseada nos Mapas Auto-Organizáveis (*Self-Organized Map*, SOM) e foi utilizada para obtenção de conjuntos reduzidos no treinamento dos modelos SVM e LSSVM (ROCHA NETO; BARRETO, 2013). O método OM encontra os pontos na região entre duas classes em problemas de classificação. Com isso, esse método fornece ao SVM os vetores suporte, que geralmente estão próximos à superfície de decisão. Na abordagem proposta neste trabalho, o OM foi modificado para trabalhar com modelos de misturas de Gaussianas ao invés de SOM na tarefa de seleção de pontos.

Dado um conjunto de dados de treinamento $\mathcal{D} = (\mathbf{x}_i, t_i)_{i=1}^N$, o processo do GOM é iniciado com a divisão dos padrões de treinamento em dois subconjuntos $\mathcal{D}_1$ e $\mathcal{D}_2$, tal que:

$$\begin{aligned}\mathcal{D}_1 &= \{(\mathbf{x}_i, t_i) | t_i = 1\}, i = 1, \dots, n_1, \\ \mathcal{D}_2 &= \{(\mathbf{x}_i, t_i) | t_i = 0\}, i = 1, \dots, n_2,\end{aligned}$$

em que n_1 e n_2 são as cardinalidades dos conjuntos $\mathcal{D}_1$ e $\mathcal{D}_2$, respectivamente.

Seja $\mathcal{X}_1$ e $\mathcal{X}_2$, o conjunto de entrada dos padrões pertencentes aos conjuntos $\mathcal{D}_1$ e $\mathcal{D}_2$, respectivamente. Também considere $\mathcal{T}_1$ e $\mathcal{T}_2$ o conjunto de saídas de cada padrão dos conjuntos $\mathcal{D}^{(1)}$ e $\mathcal{D}^{(2)}$, respectivamente. É aplicado um processo de mistura de Gaussianas com δ componentes em cada conjunto de padrões, $\mathcal{X}_1$ e $\mathcal{X}_2$, a fim de se obter as distribuições

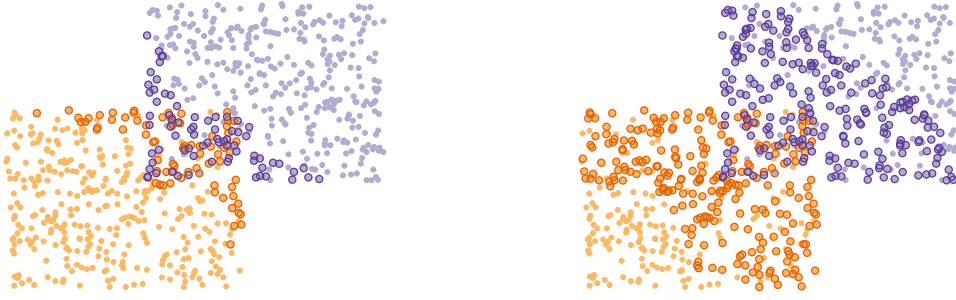
(a) $\lambda = 0, 1$ (b) $\lambda = 0, 5$

Figura 20 – Pontos selecionados pelo método Mapas Opostos Gaussianos para base de dados artificial com sobreposição.

Fonte – Próprio autor.

de probabilidade $p(\mathcal{X}_1)$ e $p(\mathcal{X}_2)$. Essas distribuições são utilizadas para encontrar o *mapa oposto* dos padrões. Isso é feito através da avaliação de cada padrão $\mathbf{x}_i \in \mathcal{X}_2$ sob a distribuição $p(\mathcal{X}_1)$ para se obter o vetor $\mathbf{c}^{(1)}$, tal que

$$c_i^{(1)} = p(\mathcal{X}_1, \mathbf{x}_i), i = 1, \dots, n_2, \quad (\text{B.1})$$

em que $p(\mathcal{X}_1, \mathbf{x}_i)$ é o valor da probabilidade $p(\mathcal{X}_1)$ no ponto $\mathbf{x}_i$. Por outro lado, é avaliado cada padrão $\mathbf{x}_j \in \mathcal{X}_1$ sob a distribuição $p(\mathcal{X}_2)$ para se obter o vetor $\mathbf{c}^{(2)}$, tal que

$$c_j^{(2)} = p(\mathcal{X}_2, \mathbf{x}_j), j = 1, \dots, n_1, \quad (\text{B.2})$$

em que $p(\mathcal{X}_2, \mathbf{x}_j)$ é o valor da probabilidade $p(\mathcal{X}_2)$ no ponto $\mathbf{x}_j$.

Seja $\mathbf{g}^{(1)} = [c_1^{(1)}, \dots, c_{n_1}^{(1)}]$ um vetor ordenado de forma decrescente com os valores de $\mathbf{c}^{(1)}$ e $\mathbf{g}^{(2)} = [c_1^{(2)}, \dots, c_{n_2}^{(2)}]$ um vetor ordenado de forma decrescente com os valores de $\mathbf{c}^{(2)}$. O conjunto reduzido é dado por $\mathcal{D}^{(rs)} = \mathcal{D}_1^{(rs)} \cup \mathcal{D}_2^{(rs)}$, em que $\mathcal{D}_1^{(rs)}$ são os padrões $(\mathbf{x}_i \in \mathcal{X}_1, t_i \in \mathcal{T}_1)$ correspondentes aos $\lceil \lambda n_1 \rceil$ primeiros elementos de $\mathbf{g}^{(1)}$ e $\mathcal{D}_2^{(rs)}$ são os padrões $(\mathbf{x}_j \in \mathcal{X}_2, t_j \in \mathcal{T}_2)$ correspondentes aos $\lceil \lambda n_2 \rceil$ primeiros elementos de $\mathbf{g}^{(2)}$.

O parâmetro λ define a quantidade de padrões que serão encontradas pelo método GOM e seus valores variam no intervalo $[0, 1]$. Quando $\lambda = 1$, toda a base de dados de treinamento é considerada como padrão pertencente ao mapa oposto. Por outro lado, quando $\lambda \rightarrow 0$, o GOM seleciona pontos mais próximos da região de sobreposição entre as classes. Esse comportamento pode ser visto na Figura 20.

B.2 GOM aplicado ao treinamento do modelo RVM

O treinamento do modelo RVM, consiste de um processo iterativo para se obter os parâmetros da distribuição a posteriori, dados pelas equações (3.32) e (3.33). Nota-se

que o parâmetro Σ é obtido através de uma inversão de matriz de ordem N , pois na proposta original do modelo RVM (TIPPING, 2001), todos os padrões de treinamento são utilizados na fase de inicialização do algoritmo. Entretanto, em cada iteração do algoritmo uma porção dos valores de α tendem ao infinito, isto é, as colunas da matriz de *kernel* correspondentes a esses elementos são removidas. O problema é que as primeiras iterações do algoritmo tem custo elevado.

O método GOM-RVM substitui o conjunto de treinamento $\mathcal{D}$ pelo conjunto $\mathcal{D}^{(rs)}$, obtido pelo método de Mapas Opostos Gaussianos, durante a fase inicial do algoritmo de treinamento do modelo RVM. Além da redução do custo inicial de treinamento, o método é uma tentativa de fazer com que o RVM encontre os vetores relevantes mais próximos da superfície de decisão.

B.3 Simulações e Discussões

Para avaliação do modelo, foram utilizadas bases de dados do UCI Machine Learning (LICHMAN, 2013) e uma base de dados artificial chamada Ripley, as quais são descritas na Tabela 4. Os modelos de aprendizagem GOM-RVM, RVM e o SVM foram comparados nos experimentos em termos de acurácia, esparsidade e tempo de treinamento.

Tabela 4 – Bases de dados utilizadas nos experimentos do método GOM-RVM.

Base de dados	Abreviação	# Amostras	# Características
Breast Cancer Winconsin	BCW	683	9
Haberman's Survival	HAB	306	3
Pima Indians Diabets	PID	768	8
Ripley	RIP	1250	2
Vertebral Column Pathologies	VCP	310	6

Durante os experimentos 80% dos padrões das bases de dados foram utilizados para treinamento e 20% foram aplicados durante os testes. Cada resultado é obtido a partir de 30 realizações independentes sobre cada base de dados. Em todos os métodos foram utilizadas funções de *kernel* Gaussianas cujos parâmetros de abertura, bem como o parâmetro C no SVM e o parâmetro λ do GOM-RVM, foram otimizados por validação cruzada com busca em grade com 10-*folds*. O parâmetro δ do modelo GOM-RVM foi fixado em 5 neste experimento.

Na Figura 21, são apresentadas as superfícies de decisão obtidas pelos modelos GOM-RVM, RVM e SVM. Essas figuras permitirão uma análise qualitativa a respeito do método proposto. Nota-se que a superfície de decisão do RVM, pouco foi afetada pela aplicação do GOM na inicialização do treinamento. Além disso, a quantidade de vetores relevantes é exatamente a mesma obtida pelo modelo RVM. Tanto o método proposto como

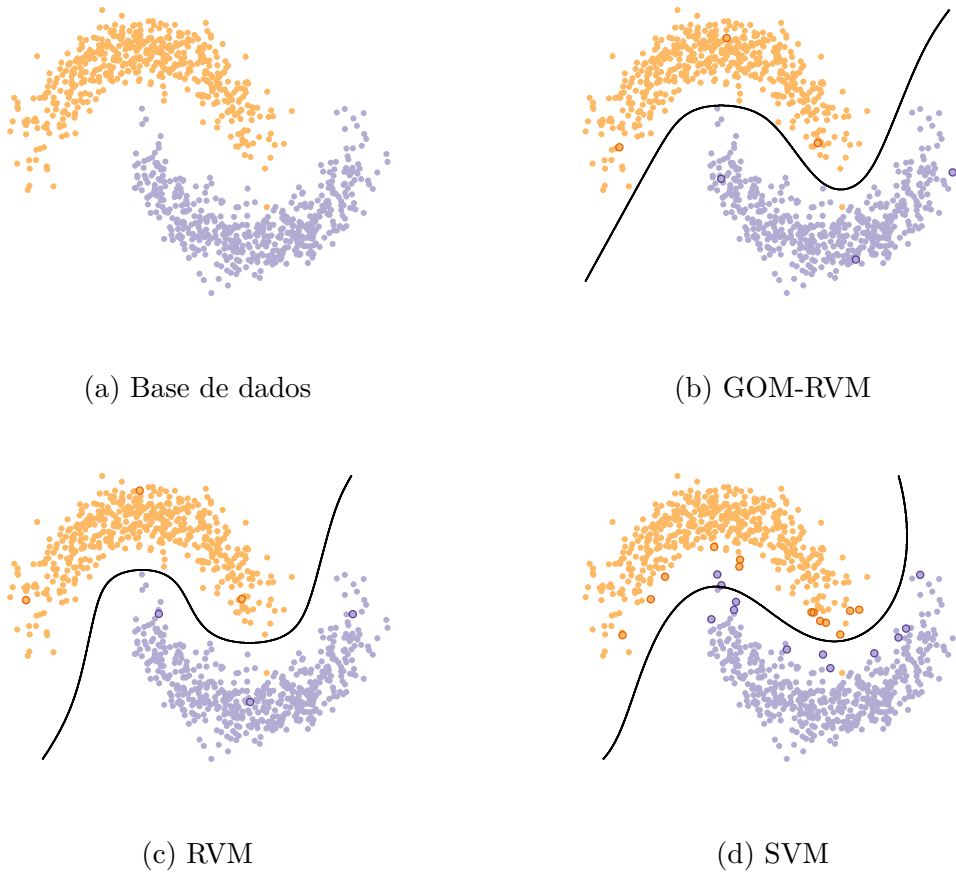


Figura 21 – Superfícies de decisão obtidas com os modelos GOM-RVM, RVM e SVM.

Fonte – Próprio autor.

o próprio RVM obtiveram uma quantidade de pontos relevantes menor que a quantidade de vetores suporte do método SVM.

Na Tabela 5 são apresentados resultados de acurácia do modelo proposto, GOM-RVM em meio aos modelos RVM e SVM. Os valores apresentados são obtidos através da média de 30 realizações independentes. Observa-se que somente na base de dados VCP, nenhuma abordagem do GOM-RVM alcançou resultados equivalentes ao RVM. Um caso interessante é o apresentado na base BCW, em que o método GOM-RVM com $\lambda = 0,4$, obteve um resultado maior que o obtido pelo modelo RVM. Em relação ao modelo SVM, somente nas bases BCW e PID, o modelo SVM não atingiu resultados superiores ao SVM.

Observa-se na Figura 22 que o modelo GOM-RVM, além de atingir bons resultados em termos de acurácia, apresentou uma quantidade de vetores relevantes bem pequena. Quando se compara a quantidade de vetores suporte obtidos pelo SVM com a quantidade de vetores relevantes dos demais modelos, nota-se uma discrepância elevada. Em geral, o parâmetro λ influencia não só a quantidade de padrões de treinamento inicial, mas também observa-se que após o processo de treinamento os modelos gerados são mais esparsos com

Tabela 5 – Taxas de acerto obtidas com os métodos GOM-RVM, RVM e SVM. Os três indicadores ao lado dos resultados dos modelos RVM e SVM, (✓) ou (✗), representam equivalência ou não-equivalência com os respectivos resultados do método GOM-RVM. Esses resultados foram obtidos através do teste estatístico de Friedman.

base	$\lambda = 0,4$ GOM-RVM	$\lambda = 0,3$ GOM-RVM	$\lambda = 0,1$ GOM-RVM	RVM	SVM
BCW	$0,95 \pm 0,01$	$0,95 \pm 0,02$	$0,93 \pm 0,02$	$0,93 \pm 0,00$ ✗ ✓ ✓	$0,96 \pm 0,01$ ✗ ✗ ✗
HAB	$0,71 \pm 0,01$	$0,71 \pm 0,01$	$0,71 \pm 0,01$	$0,71 \pm 0,01$ ✓ ✓ ✓	$0,69 \pm 0,01$ ✗ ✗ ✗
PID	$0,71 \pm 0,02$	$0,73 \pm 0,02$	$0,63 \pm 0,01$	$0,74 \pm 0,03$ ✗ ✓ ✓	$0,73 \pm 0,02$ ✓ ✗ ✗
RIP	$0,93 \pm 0,01$	$0,92 \pm 0,01$	$0,51 \pm 0,07$	$0,93 \pm 0,01$ ✓ ✗ ✗	$0,92 \pm 0,00$ ✗ ✗ ✗
VCP	$0,84 \pm 0,02$	$0,84 \pm 0,02$	$0,84 \pm 0,01$	$0,86 \pm 0,01$ ✗ ✗ ✗	$0,80 \pm 0,03$ ✗ ✗ ✗

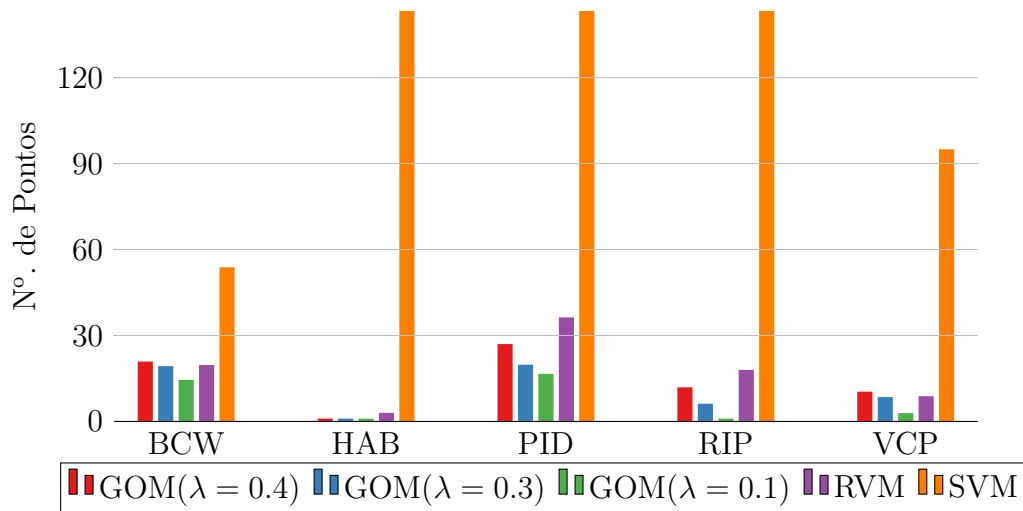


Figura 22 – Quantidade de pontos retidos no treinamento dos modelos GOM-RVM, RVM e SVM.

Fonte – Próprio autor.

a redução do valor do parâmetro. Relacionando a quantidade de vetores relevantes obtidos pelo método GOM-RVM com o RVM, observa-se que o método GOM-RVM com $\lambda = 0.4$ gera uma quantidade final de vetores relevantes mais aproximada da quantidade obtida com o modelo RVM.

Na Tabela 6 é apresentada uma análise da redução obtida através da aplicação do método de mapas opostos Gaussianos antes do processo de treinamento ser realizado. Com isso, pode ser realizada uma análise de quanto o método GOM reduz em relação ao total da base de dados. Para cada variação do parâmetro λ , é apresentada a quantidade de padrões de treinamento encontrados pelo método GOM e o percentual de redução que esse valor representa em relação ao total da base de dados. Com o valor de $\lambda = 0,4$, as taxas de redução chegam próximas à 90%. O menor percentual de redução ocorreu na base de dados RIP.

Tabela 6 – Taxa de redução obtida com o método de mapas opostos Gaussianos. Para cada valor de λ , são apresentados valores de quantidade de pontos obtidos e o percentual de redução sobre o total da base de dados.

<i>base</i>	<i>Nº Padrões</i>	GOM ($\lambda = 0, 4$)	GOM ($\lambda = 0, 3$)	GOM ($\lambda = 0, 1$)
BCW	683	115, 1 – 83, 2%	229, 6 – 66, 4%	272, 5 – 60, 1%
HAB	306	22, 1 – 92, 8%	70, 0 – 77, 1%	93, 0 – 69, 6%
PID	768	105, 6 – 86, 2%	281, 0 – 63, 4%	356, 7 – 53, 6%
RIP	1254	252, 4 – 79, 9%	611, 1 – 51, 3%	740, 2 – 41, 0%
VCP	310	22, 0 – 92, 9%	63, 1 – 79, 7%	83, 8 – 73, 0%

B.4 Conclusão

Neste trabalho é proposta uma variação das Máquinas de Vetores Relevantes utilizando o método de mapas opostos Gaussianos, chamada de GOM-RVM. O ponto chave desse método é a substituição da base de dados de treinamento pelo conjunto de padrões encontrados pelo método GOM. A utilização desses pontos força o algoritmo de treinamento do modelo RVM a buscar os vetores relevantes em uma região mais próxima da superfície de decisão. Essa suposição de que os vetores relevantes estão mais próximos da superfície de decisão é baseada na localização dos vetores suporte do modelo SVM.

Como pode ser observado nos resultados, o modelo GOM-RVM alcançou taxas de acertos equivalentes e em alguns casos superiores aos modelos SVM e RVM. Além disso, a quantidade de vetores relevantes foi reduzida. Conclui-se que a utilização de mais padrões de treinamento gera modelos com taxas de acerto mais elevadas e isso é confirmado pelo aumento do parâmetro λ . Valores pequenos de λ dificultam a convergência do algoritmo de treinamento do RVM, consequentemente esses modelos possuem uma taxa de acerto mais baixa. Por fim, a modificação do algoritmo padrão de mapas opostos utilizando as distribuições normais encontraram pontos próximos a superfície de decisão, conforme o método de mapas opostos convencional.

APÊNDICE C – Mistura de Gaussianas

Uma forma de determinar uma distribuição de probabilidade desconhecida é através de uma combinação linear de funções de densidade de probabilidade. Cada parâmetro das componentes da combinação linear é ajustado através do algoritmo de maximização da esperança (*Expectation Maximization*, EM). A prova da convergência do algoritmo EM pode ser encontrada no trabalho de Dempster, Laird e Rubin (1977).

C.0.1 Modelos Gerais de Misturas

Considerando o conjunto de amostras $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$, em que $\mathbf{x}_i \in R^D$ e assumindo que os pontos do conjunto $\mathcal{X}$ são gerados de forma independente a partir de uma distribuição $p(\mathbf{x})$. Além disso, supondo que $p(\mathbf{x})$ é definido como um modelo de mistura finita com K componentes, tal que

$$p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{k=1}^K \alpha_k p_k(\mathbf{x}, \boldsymbol{\theta}_k), \quad (\text{C.1})$$

onde:

- Os valores $\alpha_k, 1 \leq k \leq K$ são os pesos da mistura e representam a probabilidade de que uma amostra aleatória $\mathbf{x}$ foi gerada pela componente k . Como α_k representa a probabilidade de um evento ocorrer em um espaço amostral de tamanho K , então $\sum_{k=1}^K \alpha_k = 1$.
- As componentes da mistura são as distribuições $p_k(\mathbf{x}, \boldsymbol{\theta}_k), 1 \leq k \leq K$. Cada p_k é uma função de densidade ou distribuição definida sobre $p(\mathbf{x})$ com parâmetros $\boldsymbol{\theta}_k$.

O conjunto completo de parâmetros para o modelo de mistura com K componentes é:

$$\boldsymbol{\theta} = \{\alpha_1, \dots, \alpha_K, \boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_K\}. \quad (\text{C.2})$$

Dada uma amostra $\mathbf{x}_i$ e o conjunto de parâmetros do modelo da mistura $\boldsymbol{\theta}$, a incerteza de qual das K componentes geraram o vetor $\mathbf{x}_i$ pode ser calculada através da equação:

$$w_{ik} = \frac{p_k(\mathbf{x}_i, \boldsymbol{\theta}_k) \cdot \alpha_k}{\sum_{m=1}^K p_m(\mathbf{x}_i, \boldsymbol{\theta}_m) \cdot \alpha_m}, 1 \leq k \leq K, 1 \leq i \leq N. \quad (\text{C.3})$$

w_{ik} também é conhecida como peso de pertinência.

C.0.2 Modelos de misturas de Gaussianas

Em um modelo de mistura de Gaussianas, para $\mathbf{x} \in \mathbb{R}^D$, cada componente $p_k(\mathbf{x}|\boldsymbol{\theta}_k)$ da Equação (C.1) é uma distribuição Gaussianas multivariada que segue a forma:

$$p_k(\mathbf{x}|\boldsymbol{\theta}_k) = \frac{1}{(2\pi)^{d/2} |\boldsymbol{\Sigma}_k|^{1/2}} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} (\mathbf{x} - \boldsymbol{\mu}_k) \right\}, \quad (\text{C.4})$$

onde $\boldsymbol{\theta}_k = \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}$.

C.0.3 O Algoritmo EM para mistura de Gaussianas

O algoritmo EM, detalhado no trabalho de Moon (1996), pode ser aplicado ao problema de estimação de parâmetros das componentes em modelos de mistura. O método consiste de um processo iterativo que inicia com uma estimação, geralmente aleatória, do vetor $\boldsymbol{\theta}$. O algoritmo atualiza as componentes desse vetor até que um critério de convergência seja satisfeito. Cada iteração consiste de um passo E (*Expectation*) e um passo M (*Maximization*).

- **Passo E:** Utilizando o conjunto de parâmetros $\boldsymbol{\theta}$ atual, calcula-se w_{ik} através da Equação (C.3) para todas as amostras $\mathbf{x}_i, 1 \leq i \leq N$ e todas as componentes da mistura $p_k(\mathbf{x}|\boldsymbol{\theta}_k), 1 \leq k \leq K$. Note que $\sum_{k=1}^K w_{ik} = 1$ para cada ponto $\mathbf{x}_i$. Tal passo gera uma matriz com dimensões $N \times K$, na qual a soma de cada linha é igual a 1.
- **Passo M:** Agora utilizam-se os pesos w_{ik} e os dados para calcular os novos valores de parâmetros. Seja $N_k = \sum_{i=1}^N w_{ik}$, ou seja, a soma dos pesos de pertinência para a k -ésima componente. Em outras palavras, N_k representa o número de amostras associadas à componente k .

Especificamente, os novos pesos da mistura são calculados através da equação

$$\alpha_k^* = \frac{N_k}{N}, 1 \leq k \leq K. \quad (\text{C.5})$$

A média atualizada $\boldsymbol{\mu}_k^*$, Equação (C.6), é calculada de maneira similar à uma média empírica padrão, exceto que a i -ésima amostra $\mathbf{x}_i$ tem um peso w_{ik} . Note que isso é uma equação vetorial já que $\boldsymbol{\mu}_k^*$ e $\mathbf{x}_i$ são vetores d -dimensionais.

$$\boldsymbol{\mu}_k^* = \left(\frac{1}{N_k} \right) \sum_{i=1}^N w_{ik} \mathbf{x}_i, 1 \leq k \leq K. \quad (\text{C.6})$$

A matriz de covariância atualizada, dada pela Equação C.7, é calculada de maneira muito parecida com o cálculo padrão da covariância. Porém, cada amostra é ponderada pelo peso de pertinência w_{ik} . A matriz de covariância possui dimensões $D \times D$.

$$\boldsymbol{\Sigma}_k^* = \left(\frac{1}{N_k} \right) \sum_{i=1}^N w_{ik} (\mathbf{x}_i - \boldsymbol{\mu}_k^*)(\mathbf{x}_i - \boldsymbol{\mu}_k^*)^\top, 1 \leq k \leq K. \quad (\text{C.7})$$

As equações do passo M precisam ser computadas em ordem. Primeiro os K novos valores de α_k , então as K novas médias $\boldsymbol{\mu}_k$ e por fim as K novas matrizes de covariância $\boldsymbol{\Sigma}_k$.

APÊNDICE D – Máquinas de Vetores Relevantes Reduzidas por Centros de Misturas

Neste Apêndice é apresentada uma abordagem desenvolvida durante o desenvolvimento dessa dissertação, originalmente proposta no trabalho SOUSA, DIAS e ROCHA NETO (2017), para redução do conjunto de treinamento em Máquinas de Vetores Relevantes. Esse método baseia-se no uso de centros das componentes de um processo de mistura de Gaussianas como conjunto de treinamento para o modelo RVM. Com essa abordagem se deseja reduzir o custo computacional das primeiras iterações do algoritmo SBL. Na próxima seção é apresentado o método proposto, doravante denominado GMC-RVM, e a seguir os resultados e conclusões.

D.1 Centros de Misturas de Gaussianas para Máquinas de Vetores Relevantes

Esse método é uma estratégia de pré-processamento dos padrões de treinamento com o intuito de reduzir o custo inicial do algoritmo SBL durante o treinamento do modelo RVM. Isso é realizado através da aplicação de um processo de mistura de Gaussianas, descrito no Apêndice C, com K componentes. O algoritmo de mistura de Gaussianas gera um conjunto Θ com K subconjuntos θ_k contendo os parâmetros das distribuições Gaussianas que compõem a mistura. Os centros μ_k de cada subconjunto de parâmetros θ_k são utilizados como padrões de treinamento para iniciar o algoritmo de treinamento do modelo RVM. Os pontos encontrados pelo processo podem ser vistos na Figura 23.

D.2 Resultados e Discussões

Inicialmente é realizado um experimento para análise qualitativa da superfície de decisão obtida pelo modelo GMC-RVM. Para isso, foram geradas superfícies de decisão com os modelos GMC-RVM e RVM sobre a base de dados RIP. A estimação do parâmetro de abertura do *kernel* Gaussiano utilizado nos dois classificadores foi realizada por validação cruzada com busca em grade com 10-*folds*. O parâmetro K do modelo GMC-RVM foi fixado em 100, isto é, foram utilizadas 100 componentes Gaussianas na mistura. O resultado obtido nesse experimento é apresentado na Figura 24.

Observa-se que a utilização dos centros da mistura de Gaussianas no modelo GMC-RVM não afeta a qualidade da superfície de decisão obtida, já que é bem próxima da

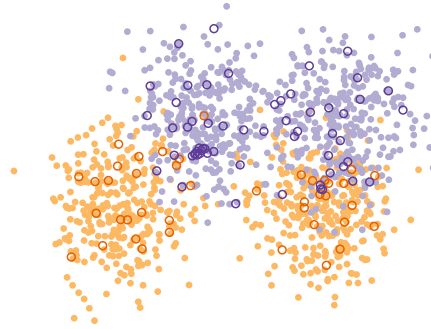


Figura 23 – Centros de componentes de misturas de Gaussianas encontrados na base de dados RIP.

Fonte – Próprio autor.

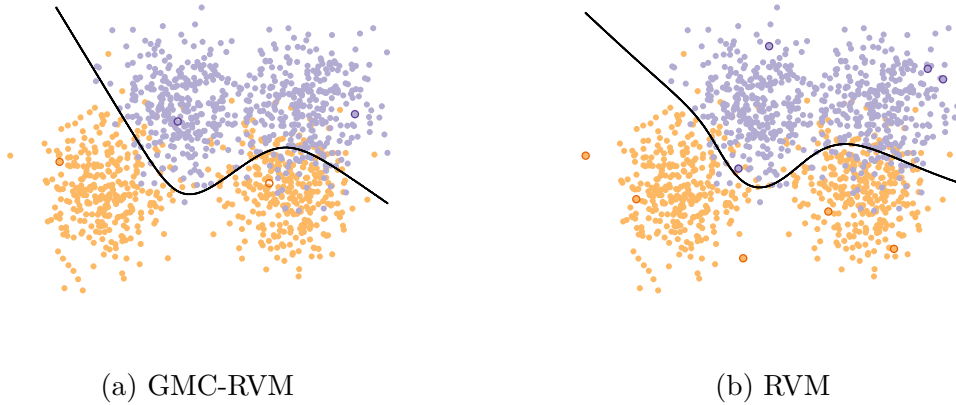


Figura 24 – Superfícies de decisão geradas pelo modelo GMC-RVM e pelo modelo RVM na base dados RIP.

Fonte – Próprio autor.

superfície de decisão gerada pelo modelo RVM padrão. Um segundo ponto que pode ser observado pela figura é que o modelo GMC-RVM obteve menos vetores relevantes que o modelo RVM padrão. Empiricamente, pode-se afirmar que o modelo GMC-RVM reduz o número de vetores relevantes sem afetar a qualidade do modelo.

Um segundo grupo de experimentos visa comparar os resultados obtidos pelo modelo GMC-RVM com os classificadores RVM e SVM. Esses testes foram realizados utilizando as bases de dados do UCI Machine Learning e a base de dados artificial Ripley e Banana, as quais são descritas na Tabela 7. Nesse experimento, as bases de dados foram divididas em 80% para treinamento e 20% para testes. Os parâmetros dos modelos foram otimizados por validação cruzada com busca em grade com 10-*folds*, com exceção do parâmetro K que recebeu os valores fixos 10, 50 e 100. Cada resultado é calculado sobre a

média de 30 realizações independentes.

Tabela 7 – Lista de bases de dados aplicadas na comparação do modelo GMC-RVM.

Base de dados	Abreviação	# Amostras	# Características
Haberman's Survival	HAB	306	3
Vertebral Column Pathologies	VCP	310	6
Pima Indians Diabets	PID	768	8
Australian Credit Approval	AUS	690	14
Ripley	RIP	1250	2
Banana	BNA	5300	2

Observando os resultados apresentados na Tabela 8, nota-se que na base de dados HAB, o método GMC-RVM com $K = 10$ atingiu resultados superiores aos modelos RVM e SVM. Na base de dados VCP o GMC-RVM atingiu valores menores em média, mas devido o alto desvio padrão, obteve resultados equivalentes estatisticamente. Os resultados mais baixos em termos de acurácia do modelo ocorreram nas bases de dados PID e AUS, em que os classificadores RVM e SVM atingiram resultados superiores ao GMC-RVM em todas as variações do parâmetro K .

Tabela 8 – Valores de acurácia obtidas com os modelos GMC-RVM, RVM e SVM. Os resultados apresentados para os modelos RVM e SVM são acompanhados pelos indicadores de equivalência (✓) ou não-equivalência (✗) com os modelos GMC-RVM com $K = 10$, $K = 50$ e $K = 100$.

<i>base</i>	$K = 10$ GMC-RVM	$K = 50$ GMC-RVM	$K = 100$ GMC-RVM	RVM	SVM
HAB	75,19 ± 1,27	70,66 ± 4,90	73,28 ± 4,89	70,93 ± 1,42 ✗✓✓	70,77 ± 1,06 ✗✓✓
VCP	78,87 ± 13,18	80,54 ± 12,39	80,16 ± 12,80	85,11 ± 4,33 ✓✓✓	84,25 ± 4,88 ✓✓✓
PID	63,74 ± 7,52	71,62 ± 2,86	70,56 ± 4,13	75,80 ± 2,55 ✗✗✗	75,67 ± 2,26 ✗✗✗
AUS	74,69 ± 4,12	71,06 ± 15,24	67,58 ± 13,36	80,65 ± 1,32 ✗✗✗	84,73 ± 2,56 ✗✗✗
RIP	90,64 ± 1,62	91,04 ± 1,17	90,56 ± 1,24	90,59 ± 1,77 ✓✓✓	90,59 ± 1,82 ✓✓✓
BNA	90,93 ± 1,11	89,88 ± 0,92	89,87 ± 0,27	90,69 ± 1,36 ✓✓✓	91,11 ± 0,83 ✓✗✗

A Tabela 9 apresenta resultados em termos de quantidade de vetores relevantes obtidos com os modelos GMC-RVM e RVM e também em termos da quantidade de vetores suporte obtidos com o modelo SVM. Nota-se que em todas as bases de dados, o número de vetores relevantes dos modelos GMC-RVM e RVM são inferiores ao número de vetores suporte obtidos com o modelo SVM. Também observa-se que na maioria das bases de dados o método GMC-RVM gera modelos com quantidade média de vetores relevantes menor que o modelo RVM.

Analisando em conjunto os resultados apresentados nas tabelas 8 e 9, nota-se que na base de dados HAB, o modelo GMC-RVM com $K = 10$ obteve resultados de

Tabela 9 – Quantidade de vetores relevantes obtidos com os modelos GMC-RVM, RVM e SVM. Cada valor foi calculado sobre a média de 30 realizações independentes.

<i>base</i>	$K = 10$ GMC-RVM	$K = 50$ GMC-RVM	$K = 100$ GMC-RVM	RVM	SVM
HAB	1.07 ± 0.37	4.87 ± 6.83	5.43 ± 8.76	21.47 ± 4.87	187.80 ± 5.38
VCP	6.10 ± 3.69	6.13 ± 3.69	5.47 ± 3.16	11.33 ± 3.70	122.17 ± 52.15
PID	18.63 ± 4.82	21.57 ± 4.05	20.43 ± 6.37	27.97 ± 2.06	369.57 ± 24.19
AUS	19.87 ± 3.88	18.00 ± 6.80	13.57 ± 8.73	20.50 ± 1.50	274.40 ± 52.16
RIP	18.00 ± 8.64	16.63 ± 6.63	19.60 ± 7.83	17.53 ± 7.25	245.50 ± 25.02
BNA	23.60 ± 2.63	16.17 ± 2.17	15.50 ± 0.73	23.17 ± 3.78	1135.27 ± 86.13

acurácia superiores aos modelos RVM e SVM utilizando um número muito menor de vetores relevantes. O mesmo comportamento se repete na base de dados RIP para o modelo GMC-RVM com $K = 50$, mas com uma discrepância menor em termos de quantidade de vetores relevantes. Um comportamento interessante ocorre nas bases de dados PID e AUS, pois os resultados de acurácia do modelo GMC-RVM são inferiores, mas isso foi obtido utilizando uma quantidade aproximada de vetores relevantes do RVM. Esse comportamento ocorre devido a perda de padrões de treinamento importantes durante a aplicação do processo de mistura de Gaussianas.

D.3 Conclusão

O método GMC-RVM é um modelo que constrói um conjunto reduzido de padrões de treinamento para iniciação do classificador RVM. A abordagem proposta, chamada de GMC-RVM, consiste na aplicação de um processo de mistura de Gaussianas para identificação dos centros das componentes que são utilizadas como padrões de treinamento. Os resultados demonstram que o modelo GMC-RVM atingiu resultados significativos em relação aos modelos RVM e SVM em termos de acurácia, gerando modelos com poucos vetores relevantes ao final do processo de treinamento.