Universidade Federal de Minas Gerais
Escola de Engenharia

Programa de Pós-Graduação em Engenharia Elétrica

Mapeamento Explícito como Kernel em Aprendizado de Máquinas de Vetores de Suporte

Carla Caldeira Takahashi

Dissertação de Mestrado

Belo Horizonte
12 de fevereiro de 2015

Universidade Federal de Minas Gerais
Escola de Engenharia

Carla Caldeira Takahashi

Mapeamento Explícito como Kernel em Aprendizado de Máquinas de Vetores de Suporte

Trabalho apresentado ao Programa de Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais como requisito para obtenção do grau de Mestre em Engenharia Elétrica.

Orientador: *Prof. Dr. Antônio de Pádua Braga*

Belo Horizonte
12 de fevereiro de 2015

Dedico este Mestrado à minha família, aos meus amigos e ao meu amado, pelo apoio e alegria durante esta etapa da minha vida.

Agradecimentos

Agradeço ao Professor Antônio de Pádua Braga pela sua orientação nos momentos necessários e aos meus colegas de laboratório por me ajudarem sempre que foi preciso. Agradeço aos meus pais, Ricardo e Cláudia, pelas sugestões, críticas construtivas e paciência. Agradeço a minha irmã, Fernanda, pela ajuda em diversos momentos da elaboração do projeto e do texto da dissertação, e, também, ao meu namorado, Fernando, por todo auxílio que pode me dar. Agradeço, por fim, aos funcionários e professores do Programa de Pós-Graduação em Engenharia Elétrica da UFMG pelo suporte, e à agência de fomento CAPES pelo apoio.

As I've tried to stress, at the heart of science is an essential balance between two seemingly contradictory attitudes - an openness to new ideas, no matter how bizarre or counterintuitive, and the most ruthlessly skeptical scrutiny of all ideas, old and new. This is how deep truths are winnowed from deep nonsense.

—CARL SAGAN (The Demon-Haunted World: Science as a Candle in the Dark, 1995)

Resumo

Os problemas que podem ser resolvidos por métodos de aprendizado de máquina também têm influência sobre os algoritmos implementados, eles são divididos em três grandes categorias: a regressão, a classificação e o agrupamento. O problema abordado ao longo desta dissertação é a classificação de padrões, que tem como objetivo criar superfícies de separações no espaço para o dividir em regiões de acordo com as classes dos padrões. A classificação é bastante parecida com o agrupamento, mas este não tem acesso à classe esperada para cada padrão, o que faz com que estes métodos estejam mais relacionados à distribuição dos dados no espaço. A maximização de margem é uma abordagem para problemas de aprendizado de máquina bastante apropriada uma vez que a capacidade de generalização de um método qualquer está relacionada a sua margem. Dessa forma é possível constatar que classificadores de margem larga são mais robustos quando devem determinar a classe de dados desconhecidos. Dentre os métodos de maximização de margem, as máquinas de vetores de suporte ou SVMs, utilizam o algoritmo de otimização baseados em Lagrangiano para determinar vetores de suporte, os quais constroem uma superfície de separação cuja distância, ou margem, seja o mais distante possível dos padrões de todas as classes. As SVMs utilizam kernels que tem como finalidade a projeção de dados de entrada em um espaço que permita a separação dos dados, possibilitando, assim, não só a sua classificação como também a regressão de funções. Atualmente as SVMs ainda são um dos métodos melhores e mais utilizados métodos na literatura. O Mapeamento Explícito, por sua vez, é uma abordagem mais recente que se tornou popular com o desenvolvimento das Máquinas de Aprendizado Extremo, as ELM. Estas máquinas possuem a implementação bem simples que permitem a criação de um classificador utilizando apenas cálculos analíticos e não iterativos. A ELM utiliza um mapeamento explícito aleatório para projetar o espaço de entrada em um espaço de características de maior dimensionalidade, o que possibilita a separação linear dos dados neste espaço projetado. Nas ELM o mapeamento é compreendido como a camada intermediária de uma rede neural cujos pesos foram atribuídos aleatoriamente, e o único parâmetro de ajuste é a quantidade de neurônios. A camada de saída da ELM pode ser ajustada de forma analítica, o que torna o método simples, rápido e elegante. No entanto, o mapeamento explícito pode, também, ser interpretado como um kernel com-

plexo, cujos parâmetros são, somente, a dimensão do mapeamento e a variância da distribuição que gerou os pesos. Como a quantidade de neurônios, ou seja, a dimensão do mapeamento, se torna insensível para a desempenho do classificador, quando ela é grande o suficiente, e a variância da função geradora dos pesos também não apresenta efeito algum, então este método pode ser considerado não paramétrico. É amplamente aceita a necessidade de se utilizar métodos que promovam a maximização de margem, uma forma possível de se aperfeiçoar a SVM é o emprego de kernels não paramétricos. Com isso o método se torna mais simples de ser implementado e empregado, já que dispensa uma longa metodologia para o ajuste apropriado dos parâmetros. Com esta motivação foi implementado um método que utiliza o mapeamento explícito como kernel, assim a grande dimensionalidade do espaço de características permite a separação linear dos dados ao mesmo tempo que é promovida a maximização da margem de classificação. Ao mesmo tempo, a utilização de uma máquina linear juntamente com o mapeamento que não requer o ajuste de qualquer parâmetro.

Palavras-chave: ELM, SVM, Kernel, Mapeamento Explícito

Abstract

The problems that can be solved through the machine learning approach also have influence on particularities of the implemented algorithms, they are divided in three large groups: regression, classification and clustering. This dissertation deals with pattern classification problems, which aim to create separating surfaces along the pattern space dividing it in regions according to the pattern classes. Classification problems are quite similar to clustering problems, however the latter does not have access to the expected class for each pattern, and therefore its methods use structural characteristics of the data distribution in the space. The margin maximization approach for machine learning problems is appropriated, since the capability of generalization of any classification method is related to its margin. Therefore, it is possible to assert that large margin classifiers are more robust when classifying unknown data. Among large margin classifiers methods, the support vectors machines, SVM, use a Lagrangian based algorithm to determine support vectors, which constructs a separating surface whose distance, or margin, to every class patterns is the largest as possible. The SVM use kernels with the purpose of mapping the input space into a feature space that allows the data separation, allowing not only the pattern classification but also the function regression. Nowadays the SVM are still one of the best and most used methods in the academia. Explicit mapping approach became popular recently with the proposal of the extreme learning machines, ELM. These machines have a rather simple implementation that allows the creation of a classifier that uses only analytical calculations, discarding any iterations. The ELM uses a random explicit mapping of the input space into a feature space of higher dimensionality, allowing the linear separability of the data in the mapped space. For the ELM, the mapping is construed as the hidden layer of a feedforward neural network whose weights are assigned randomly, and the single parameter to be tuned in it is the quantity of neurons. The output layer, in the ELM, has its weights tuned according to an analytical calculation, which makes this method simple, fast and very elegant. The explicit mapping can also be interpreted as a complex kernel, whose parameters are only the mapping dimension and the variance of the random distribution that generated the weights. Since the number of neurons, in other words the mapping dimension, is not sensible by the methods performance, when it is big enough, and the variance has no effect either, this method

can be considered non parametrical. The need of using large margin methods is widely accepted, hence it is possible to improve SVMs by using non parametric kernels. Thus the classifier becomes simpler to be implemented and used, since it is exempt of using a complicated methodology for a fine parameter tuning. With this motivation it was implemented a method that uses explicit mapping as kernel, therefore the great dimensionality of the feature space allows the linear separability of the data at the same time that the margin is maximized. Meanwhile, the use of the non-parametric explicit mapping and a linear support vectors machine allows a virtually non-parametric at all.

Keywords: ELM, SVM, Kernel, Explicit Mapping

Sumário

Lista de Figuras

Lista de Tabelas

CAPÍTULO 1

Introdução

O Aprendizado de Máquina é um ramo de Inteligência Computacional amplamente relacionado à estatística e probabilidade. Métodos de aprendizado devem ser capazes de abstrair informações estatísticas provenientes de um conjunto de dados e, a partir delas, definir o comportamento de uma máquina. Dessa forma os procedimentos implementados permitem que esta máquina extraia informações de dados empíricos disponíveis e, com este conhecimento adquirido, são definidos modelos capazes de resolver problemas específicos (Haykin, 2001).

Devido a grande variedade de máquinas presentes na literatura é fundamental o conhecimento das características do problema no qual a máquina será empregada e do contexto em que este problema está embutido para que seja escolhida a máquina apropriada. Da mesma forma, as máquinas são classificadas por diferentes critérios que veriam segundo diferentes características na sua construção e da sua aplicação.

1.1 Aprendizado Supervisionado, Não - Supervisionado e Semi - Supervisionado

Os problemas que podem empregar o aprendizado de máquinas ou redes neurais artificiais em sua solução podem ser classificados de diferentes formas. Umas destas classificações está relacionada a utilização de padrões com rótulos ou *targets* pela máquina para a solução do problema. Os rótulos são os valores esperados para cada conjunto características de entrada ou *features*. Por exemplo, em um problema de diagnóstico de câncer de mama, o rótulo poderia ser o resultado positivo ou negativo, para um problema binário que não trabalhe a possibilidade de resultado inconclusivo. Se um método de aprendizado utiliza estes rótulos ele é denominado Supervisionado, ou seja existem um professor que, durante o processo de aprendizado da máquina, avalia os resultados atingidos reforçando-os ou penalizando-os de acordo com a sua proximidade aos resultados esperados (Braga *et al.*, 2007; Chapelle *et al.*, 2010; Haykin, 2001; Zhu e Goldberg, 2009). Neste tipo de aprendizado é necessária a existência de um agente que

determina o rótulos de saída de acordo com as características de entrada, assim a máquina pode ajustar os seus parâmetros internos de forma que os resultados atingidos estejam o mais próximo possível dos esperados.

Quando estes rótulos não estão disponíveis é caracterizado o Aprendizado Não - Supervisionado. Neste caso, os dados não possuem valores esperados o que impossibilita a utilização de um oráculo ou de qualquer agente de rotulação. Dessa forma, a abordagem para se resolver o problema muda significativamente (Haykin, 2001). Os dados devem ser agrupados de acordo com as características estruturais intrínsecas à distribuição espacial ou estatística dos dados. Ou seja são utilizadas as informações de densidade de dados, distância entre padrões, a sua correlação, entre outros.

Há, ainda, um outro ramo do aprendizado de máquina, denominado Aprendizado Semi-Supervisionado (SSL, *Semi-Supervised Learning*), que foi apresentado como uma solução para problemas de classificação que possuem em sua base de dados tanto padrões rotulados quanto não rotulados, em particular para o caso em que há uma quantidade muito maior de dados não rotulados (Chapelle *et al.*, 2010; Zhu e Goldberg, 2009). Esta preocupação surgiu com constatação de problemas em que o conjunto de dados rotulados é uma parcela pequena dos dados disponíveis para treinamento. Em geral isto ocorre quando o esforço necessário para se gerar dados é muito pequeno em relação à dificuldade de se rotular cada um desses padrões gerados. Esse é o caso, por exemplo, da classificação de conteúdo na internet, enquanto a criação de novos textos, imagens e mídias em geral para abastecer a rede é fácil e intensa, o esforço necessário para classificá-los manualmente é inviável (Zhu e Goldberg, 2009).

1.2 Regressão, Classificação e Agrupamento

Ocorrem distinções entre os tipos de problemas que podem ser resolvidos com a utilização de aprendizado de máquinas. Existem, de forma geral, dois tipos principais de problemas de aprendizado de máquina supervisionado, regressão e classificação, e um não-supervisionado, o agrupamento ou *clustering* (Braga *et al.*, 2007; Haykin, 2001; Zhu e Goldberg, 2009). A regressão de funções consiste em um problema estatístico clássico, ela busca determinar uma função que descreva, com certa precisão, a o conjunto de dados de acordo com os pares de características dos padrões e seus respectivos rótulos. Ou seja ela identifica a relação entre as variáveis de entrada e as variáveis de saída de um determinado problema, por exemplo, na figura 1.1, um regressor foi desenvolvido utilizando uma Rede RBF (*Radial Basis Function*)

de forma que a sua saída correspondesse a uma função Sinc, de acordo com os padrões de entrada. A caracterização de funções é, ainda, um problema bastante complexo que possui diversos desdobramentos, como, por exemplo, problemas de previsão de séries temporais.

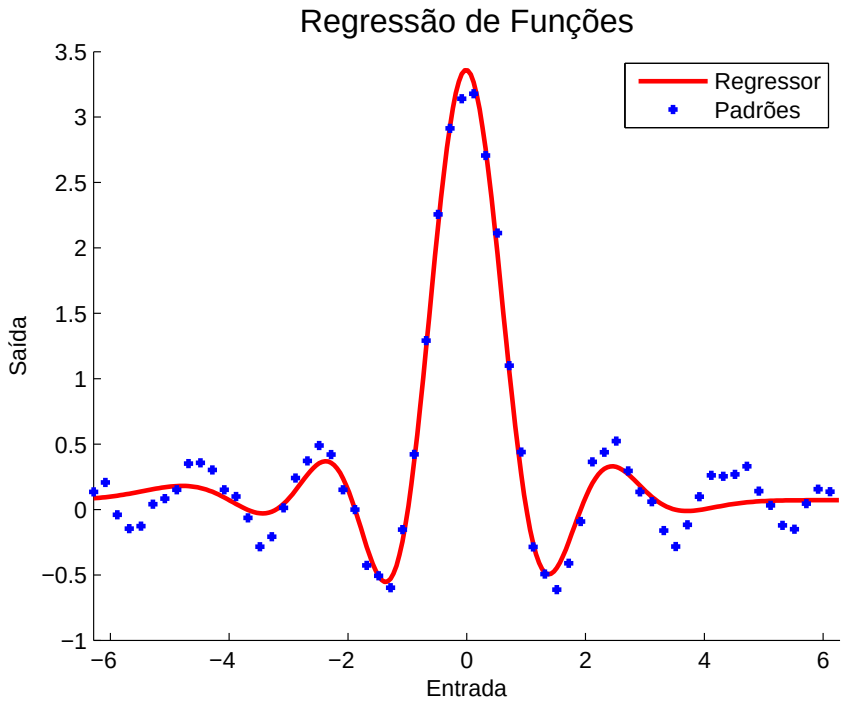


Figura 1.1: Regressão de uma função Sinc utilizando uma Rede Radial Basis Function.

A Classificação de Padrões está relacionada a criação de uma superfície de separação que divida o espaço entre duas ou mais classes de acordo com a distribuição dos padrões de cada classe(Braga *et al.*, 2007; Haykin, 2001). Com isso os métodos de aprendizado de máquinas para classificação devem ajustar os seus parâmetros livres de forma que a maior parte possível dos padrões de uma mesma classe estejam do mesmo lado da superfície de separação. Esta tarefa pode se tornar bastante complexa à medida que os dados apresentem um superposição ou que a geometria das classes no espaço se entrelace. Além disso um fenômeno indesejado, denominado *overtraining*, deve ser sempre evitado (Braga *et al.*, 2007). Quando há um treinamento inadequado da máquina ou quando ela é mais complexa do que o necessário, ela se adequa excessivamente aos dados de treinamento apresentado um ótimo desempenho fictício durante o aprendizado. No entanto este comportamento é bastante indesejado, já que ele implica em perdas na capacidade de generalização da máquina, uma vez que os dados de teste ou da aplicação real são diferentes dos dados de treinamento(Braga *et al.*, 2007; Haykin, 2001).

Por outro lado o Agrupamento está relacionada ao reconhecimento de padrões e não

possui rótulos esperados durante o treinamento. Desta forma o aprendizado é feito a partir da extração das informações intrínsecas da distribuição dos dados(Braga *et al.*, 2007). Os agrupamentos de dados são definidos a partir de relações que foram observadas em sua distribuição, como a média ou a mediana da distância euclidiana entre pares de dados, a correlação entre os dados, a existência de centróides nos dados, entre outros (Zhu e Goldberg, 2009).

A classificação e o agrupamento apresentam semelhanças, já que ambos consistem na obtenção de uma superfície de separação ou qualquer mecanismo semelhante que permita a separação do espaço em regiões que separem de forma bem definida os dados de dois ou mais grupos existentes neste espaço. O que diferencia a classificação do agrupamento é a existência de rótulos, ou seja, em um problema de classificação, existe para os dados de treinamento uma valor esperado que separa os dados em classes. Como estas classes são sumariamente arbitrárias, não existe necessariamente uma correlação entre a distribuição dos dados no espaço e valor dos seus rótulos. Desta forma é possível que os valores esperados em um problema de classificação seja extrínseca às informações estatísticas coletadas dos dados. Ou seja, é provável que um problema quando resolvido por métodos de classificação e clusterização, ignorando os rótulos, apresentem resultados bastante diferentes (Zhu e Goldberg, 2009), como é ilustrado na figura 1.2.

Na figura 1.2, distribuições Gaussinas bidimensionais estão distribuídas em duas classes, neste caso, o Classificador foi criado a partir de um Perceptron de Múltiplas Camadas (MLP ou *Multi Layer Perceptron*) que recebe a informação de rótulos de cada padrão para criar uma função de separação. A solução gerada a partir dos Agrupamentos, no entanto, são definidas com uma rede SOM, um Mapa Auto-Organizável (*Self-Organizing Map*), e definem os *clustes* de acordo com a distância entre os pontos no espaço.

1.3 Modelos Generativos e Discriminativos

Os modelos utilizados por métodos de Aprendizado Supervisionados e Semi-Supervisionados podem pertencer a dois tipos diferentes os modelos Generativos e os Discriminativos.

De forma geral os modelos Generativos têm a preocupação em determinar o mecanismo que gerou os dados e, dessa forma, tendem a incorporar informações estruturais dos dados no classificador (Chapelle *et al.*, 2010; Zhu e Goldberg, 2009). Dentre métodos baseados em modelos generativos, podem ser destacados os de mistura Gaussiana, como as máquinas de

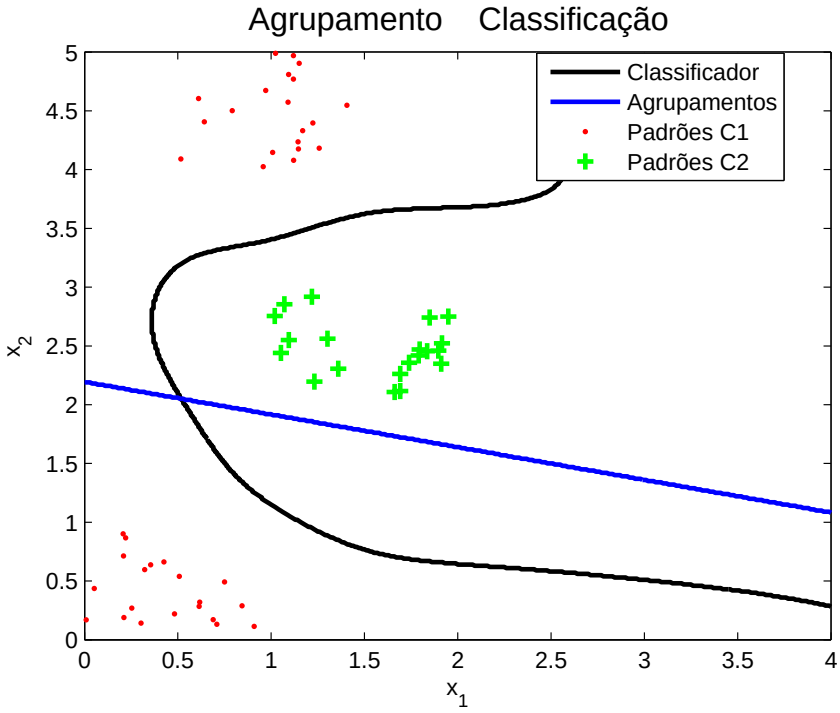


Figura 1.2: Três distribuições Gaussianas são classificadas em duas classes utilizando um MLP e agrupadas em dois *clusters* de acordo com uma rede SOM.

Bayes ou classificadores Bayesianos (*naive-Bayes*), métodos de maximização de esperança ou EM (*expectation maximization*) e métodos baseados em cadeias de Markov (Haykin, 2001).

Por outro lado os modelos Discriminativos não apresentam esta preocupação, focando apenas na geração de um modelo que consiga definir os resultados de acordo com os dados observados, como é o caso das Redes Neurais Artificiais e das Máquinas de Vetores de Suporte (*SVM*) (Chapelle *et al.*, 2010; Smola e Bartlett, 2000; Zhu e Goldberg, 2009).

1.4 Método Proposto com Mapeamento Explícito

Neste trabalho é abordada a classificação de dados e o reconhecimento de padrões a partir de modelos discriminativos. Problemas desta natureza são observados em diversas áreas do conhecimento, desde problemas na biologia, como a classificação de espécies de flores, até a literatura, na classificação do conteúdo de textos. Na literatura podem ser encontrados diversos artigos que resolvem problemas na área de saúde como o diagnóstico de doenças cardíacas, câncer de mama, diabetes, por exemplo. Existem ainda estudos na área de acústica, como

identificação de instrumentos ou mesmo de fonemas. Há também métodos de classificação automática de cadeias de proteínas, na área de genética na Biologia, da classificação de conteúdo web, na Informática, ou mesmo de falhas em motores na Engenharia de Elétrica de Potência. É inegável a importância dos métodos automáticos de classificação com o intuito de se agilizar diversos processos. Tornando estes mais acessíveis e dinâmicos já que reduzem a necessidade de uma mão de obra qualificada de alto custo, que poderia ser melhor empregada.

Existe, no entanto, uma dificuldade nos métodos de classificação de padrões relacionada a configuração otimizada de parâmetros. Neste contexto, foi proposto um método que que utiliza mapeamento explícito em máquinas de vetores de suporte (SVM, *Support Vector Machines*). Com base nas Máquinas de Aprendizado Extremo (ELM, *Extreme Learning Machines*), o Kernel é substituído por uma estrutura similar à camada escondida de uma Rede Neural Artificial, de grande dimensionalidade e pesos aleatórios. Dessa forma, é reduzida para um a quantidade de parâmetros a ser ajustado. No entanto, o método desenvolvido é também robusto a este parâmetro, que pode receber qualquer valor, desde que seja maior que a dimensão original do problemas, sem afetar de forma significativa o desempenho da máquina.

O método proposto se trata de um algoritmo de classificação discriminativo que consiste na integração de uma Máquina de Aprendizado Extremo a uma Máquina se Vetores de Suporte, o que o torna, assim, robusto a ruídos e incertezas na margem de separação entre classes. Ou seja, o algoritmo proposto é favorável se empregado em bases de dados em que há superposição entre os dados. Além disso este método não é muito sensível a nenhum dos seus parâmetros de entrada, o que o torna extremamente vantajoso para no treinamento, já que não requer nenhum ajuste de parâmetros.

1.4.1 Máquinas de Aprendizado Extremo

A proposta das Máquinas de Aprendizado Extremo é a implementação de uma rede neural artificial feedforward com uma camada escondida (SLFN *Single Layer Feedforward Network*) que possui um algoritmo rápido de aprendizado. Uma ELM é uma RNA cujos pesos e bias da camada escondida são atribuídos aleatoriamente e não são posteriormente ajustados, com isso o problema é projetado em um espaço de maior dimensionalidade o que facilita a separação dos dados. A camada de saída, por sua vez, tem seus pesos e bias ajustados analiticamente a partir da pseudo-inversa da camada escondida, esta solução foi baseada na minimização da norma dos mínimos quadrados. Este método permite que o erro de classificação da rede seja limitado desde que a função de ativação dos neurônios da camada de saída seja infinitamente

diferenciável. Diferentemente da maioria dos métodos de treinamento de SLFN que são baseados em gradiente, a estratégia da ELM reduz consideravelmente o tempo de treinamento de redes neurais artificiais do tipo SLFN (Huang *et al.*, 2006).

Apesar da simplicidade do método, as ELMs conseguem atingir altas taxas de sucesso equivalentes a máquinas de vetores de suporte. Neste caso, devido a redução do tempo de treinamento, apesar de não haver melhora no desempenho classificação da rede, é possível dizer que o mecanismo de aprendizagem apresentou uma evolução.

1.4.2 Máquinas de Vetores de Suporte

As máquinas de vetores de suporte são mecanismos baseados em Lagrangiano que tem como objetivo a definição de superfícies de separação, no caso da classificação de padrões, que maximize a margem de separação entre as duas classes. Elas definem os chamados vetores de suporte, que são os padrões mais significativos para de acordo com o Lagrangiano. A partir dos vetores de suporte uma superfície é construída com a maximização da distância entre os vetores de suporte e a superfície de separação, quanto maior esta distância maior a margem do classificador (Cortes e Vapnik, 1995; Smola e Bartlett, 2000).

Em termos de desempenho as máquinas de vetores de suporte ainda são umas das mais confiáveis em aplicações de classificação de padrões, e são, ainda, consideradas técnicas no estado da arte. Devido a sua característica de garantir a maximização da margem, inclusive com a utilização da variável de regularização, é um método que possui alta capacidade de generalização. Na tentativa de se contornar a questão do ajuste de parâmetros, foi implementado um método que combina as qualidades das SVMs com o mapeamento empregado em Máquinas de Aprendizado Extremo, dessa forma compondo uma SVM com Mapeamento Explícito.

1.4.3 Aprendizado Semi-Supervisionado

A partir do método proposto foi ainda implementado um método semi-supervisionado. O problema do Aprendizado Semi-Supervisionado pode ser elaborado de tal forma que ele seja constituído de dois problemas quase distintos, um supervisionado e um não supervisionado. Por exemplo, utilizando uma abordagem *cluster-then-label*, neste caso rótulos são atribuídos para os agrupamentos, gerados por um algoritmo não supervisionado, de acordo com um classificador supervisionado obtido com os dados rotulados. Outra possibilidade é a integração de

restrições para a superfície de separação obtida com um método supervisionado de acordo com a densidade dos padrões não rotulados. Dessa forma, os padrões não rotulados são utilizados para penalizar superfícies de separação que dividem possíveis agrupamentos, como é o caso das Máquinas Transdutivas e das Máquinas de Vetores de Suporte Semi-Supervisionadas (S3VM, *Semi-Supervised Support Vector Machines*) (Chapelle *et al.*, 2010; Zhu e Goldberg, 2009).

Foi proposto, neste contexto, uma variação das Máquinas de Vetores de Suporte Semi Supervisionadas que utiliza os objetivos de maximização de margem e da incorporação dos padrões não rotulados em dois objetivos distintos, caracterizando um problema multiobjetivo.

1.5 Estrutura da Dissertação

Esta dissertação está dividida em seis capítulos: Introdução, Referencial Teórico, Desenvolvimento do Método, Metodologia Experimental, Análise de Resultados e Conclusão.

Este primeiro capítulo contextualiza o problema abordado, no âmbito da inteligência computacional. É definido, segundo os conceitos da área, o tipo de problema que é abordado ao longo da dissertação e a sua importância prática.

Durante o Referencial Teórico, são revisados os artigos mais importantes na literatura relacionados aos temas abordados ao longo da dissertação, além de publicações recentes que apresentaram relevância acadêmica e avanços científicos relativos ao contexto da dissertação. Com base nas teorias, já consolidadas, apresentadas no segundo capítulo, foi elaborado o método proposto, cuja descrição e motivação estão apresentados no capítulo de Desenvolvimento do Método.

Afim de se validar a aplicabilidade do método, foi planejada uma série de experimentos que são capazes de aferir o comportamento do método de forma válida. Os experimentos realizados estão detalhados no quarto capítulo, nele são descritas, além da estrutura do testes, as bases de dados utilizadas, as medições coletadas e as decisões tomadas para o condicionamento dos dados. Segue a este capítulo, a apresentação e análise dos resultados experimentais obtidos e, por fim, o último capítulo contém uma breve discussão e as conclusões constatadas ao longo do trabalho.

CAPÍTULO 2

Referencial Teórico

2.1 Mapeamento

O mapeamento de padrões para um espaço de maior dimensionalidade é a base de diversos métodos propostos na literatura, o que ilustra a importância do artigo publicado por Cover (1965). Neste artigo foi publicado um teorema quanto a separabilidade geométrica e estatística de dicotomias quando é feita a projeção para um espaço não linear de maior dimensionalidade, que mais tarde viria a ser conhecido como o Teorema de Cover. No contexto da capacidade de separação de superfícies, Cover provou que, dado que o número de padrões seja fixo e a dimensionalidade do espaço em que eles estão situados possa aumentar, existe um tamanho de dimensão d^* a partir do qual é possível a separação linear dos padrões.

A classificação geral de um conjunto de padrões em duas classes de forma não ambígua somente pode ser atingida quando a capacidade de separação dos dados é conhecida. Ou seja, um dado conjunto de padrões $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$, cujas características são definidas $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$, pode ser separado se a sua capacidade separação permitir ou, também, se um mapeamento for feito de forma que esta capacidade aumente.

Neste caso, se um determinado mapeamento não linear tem como resultado uma matriz de posto completo no espaço de características, alguma classe de superfícies de de separação pode ser determinada. A quantidade de graus de liberdade de tais superfícies de separação é reduzida de acordo com o posto da matriz mapeada no espaço de características. Pode ser considerado então o seguinte mapeamento em um espaço de maior dimensionalidade:

$$\Phi : X \rightarrow E^d \tag{2.1}$$

Cuja projeção Φ é uma função não linear da forma $\Phi(x) = \{\phi_1(x), \phi_2(x), \ldots, \phi_d(x)\}, x \in X$ e E^d é um espaço Euclidiano d-dimensional.

Se a projeção dos N padrões X estiver em uma posição genérica e se existir uma ϕ-

superfície de separação $\{w_0 \cdot \phi(x) = 0\}$ restringida de forma a conter somente k dos N padrões, então existe uma quantidade determinada de pares de classes, denominadas dicotomias, ϕ-separáveis, dada por:

$$n_{\text{dicotomia}} = C(N, d-k) \tag{2.2}$$

Com isso, se o espaço E^d do mapeamento não linear for grande o suficiente, a quantidade de graus de liberdade da superfície de separação é reduzida e o problema pode se tornar linearmente separável. Isto pode ser observado dado que a quantidade possível de combinação dos padrões diminui quando a dimensão do espaço mapeado d aumenta, e as demais variáveis são constantes. Ou seja, se a quantidade de combinações dos N elementos em conjuntos de $(d-k)$ elementos tende a 1, então a quantidade de dicotomias e de graus de liberdade da superfície de separação é também 1 e, portanto, a separação dos padrões se torna linear.

Quando um dado conjunto de padrões é mapeado em um espaço não linear de dimensões suficientemente grandes é possível, então, diminuir a complexidade da superfície de separação do classificador desejado. Isso é possível uma vez que a quantidade de graus de liberdade da classe de superfícies de separação necessária para discriminar as classes do conjunto de padrões é reduzida de forma que uma superfície linear já seja capaz de separar os dados.

Dentre diversas técnicas, o mapeamento explicito, em especial, consiste na projeção dos dados de entrada em um espaço de dimensão maior a partir do produto escalar destes dados com uma matriz aleatória de dimensão arbitrária. O resultado deste produto é então submetido a uma não linearidade qualquer afim de se criar um espaço de características que permita a separabilidade linear dos dados. Uma Rede Neural Artificial pode ser utilizada como o mecanismo para a criação do mapeamento para o espaço de características (Hornik *et al.*, 1989). Neste caso, quando os dados de entrada passam pelos neurônios da RNA eles são submetidos a uma transformação linear com a soma de cada característica ponderada pelos pesos aleatórios, o que é equivalente ao produto escalar dos dados. E, em seguida, a não linearidade é imposta ao espaço pela função de ativação, que deve ser contínua, limitada e não constante. Para que o mapeamento promova a separação linear dos dados, é ainda necessário que a função de ativação seja não linear e infinitamente diferenciável(Cover, 1965; Huang *et al.*, 2006).

As publicações de Boser *et al.* (1992), Cortes e Vapnik (1995), Vapnik (1995, 1998) compreendem a contribuição de Vapnik quanto a elaboração e proposta das Máquinas de Vetores de Suporte, cujo aprendizado é baseado na minimização estrutural de risco, estudada em Vapnik (1998, 1992). Segundo Vapnik, conceitualmente a criação de um classificador pode

ser feita a partir da transformação do espaço inicial do padrão em um espaço de características de alta dimensionalidade. Então, um hiperplano de separação ótimo é construído a partir da distribuição dos padrões neste novo espaço.

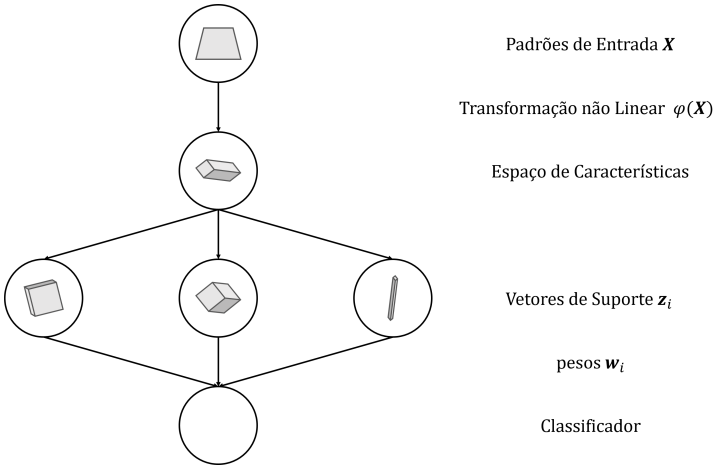
> *"The support-vector network implements the following idea: it maps the input vectors into some high dimensional feature space Z through some non-linear mapping chosen a priori. In this space a linear decision surface is constructed with special properties that ensure high generalization ability of the network."*
>
> Support Vector Networks, Cortes e Vapnik, 1995

No entanto, foi considerado um problema técnico a quantidade de operações necessárias para se tratar espaços de tão grande dimensionalidade, uma vez que a construção de hiperplanos ótimos com boa generalização para padrões com espaços de entrada de dimensões mais elevadas iria requerer espaços de características gigantescos (Cortes e Vapnik, 1995; Haykin, 2001).

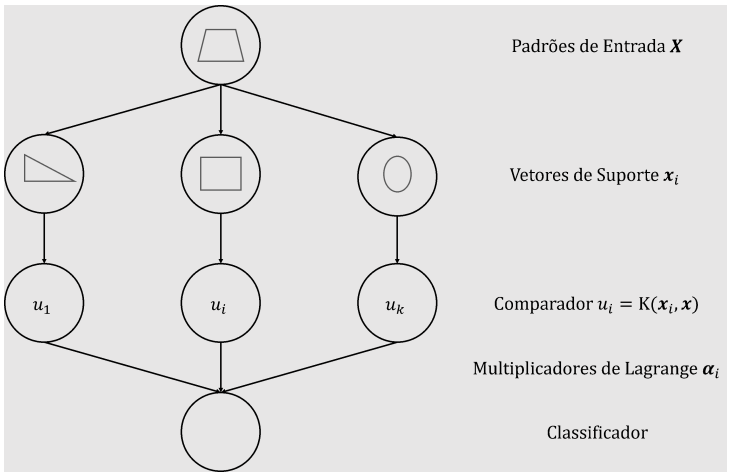
Em redes de vetores de suporte, é empregada a propriedade de que a ordem das operações na construção de uma função de decisão pode ser invertida. Neste caso dois vetores do espaço de entrada são comparados previamente e somente o resultado desta comparação sofre uma transformação não linear (Cortes e Vapnik, 1995; Haykin, 2001), conforme ilustrado na Figura 2.1.

As SVMs realizam um mapeamento do espaço de medidas de entrada para um espaço de características, sendo o Kernel o responsável por esta transformação. Quando é utilizada uma abordagem explícita do mapeamento, os padrões são projetados não linearmente em um espaço de características de alta dimensionalidade em relação ao espaço de entrada, e, somente depois do mapeamento, é medida a proximidade entre cada par de padrões, por meio de produtos escalares. Entretanto, como foi provado por Boser *et al.* (1992), a equação (2.3) permite que a ordem das operações possa ser invertida, de forma que as comparações com produto escalar sejam feitas no espaço de entrada e, neste caso, as medidas de proximidade formam o espaço de características que, posteriormente, sofre a transformação não linear. Essa inversão das operações possibilita o mapeamento implícito.

$$K(x_i \cdot x_j) = \Phi(x_i) \cdot \Phi(x_j) \tag{2.3}$$



(a) Mapeamento Explícito



(b) Emprego de um Kernel

Figura 2.1: Representação gráfica da realização do mapeamento em uma SVM.(2.1a) mostra um mapeamento realizado de forma explícita. (2.1b) utiliza uma abordagem implícita para gerar o classificador com o emprego de um Kernel.

2.1.1 Máquinas de Aprendizado Extremo

No contexto do mapeamento explícito, as Máquinas de Aprendizado Extremo foram propostas por Huang *et al.* (2006). Este método é uma implementação específica de Redes Neurais com uma única camada escondida alimentadas adiante que propõe uma atribuição aleatória dos pesos da camada escondida e uma solução algébrica para a camada escondida, com as condições de que a quantidade de neurônios da camada escondida seja suficientemente grande e as suas funções de ativação infinitamente diferenciáveis. Por esses motivos, as ELMs apresentam uma velocidade de aprendizado bastante rápida, e ainda assim o seu desempenho

de classificação é elevado, equiparados aos das SVMs.

Em *Extreme Learning Machines* (Huang *et al.*, 2006), foi provado que um mapeamento aleatório, para qualquer problema de probabilidade contínua, resulta em uma matriz de posto completo. Isso ocorre desde que a função de ativação de cada neurônio na camada escondida da ELM seja não linear, não regular e infinitamente diferenciável em qualquer intervalo.

As Máquinas de Aprendizado extremo estabeleceram uma nova abordagem do aprendizado de máquina, ao transpor a limitação técnica apontada por Cortes e Vapnik (1995). O intervalo de aproximadamente dez anos entre as duas publicações são muito significativos quanto à evolução da capacidade de processamento dos computadores, por este motivo algumas das limitações antes existentes podem ser desconsideradas ocasionalmente. No caso da ELM(Huang *et al.*, 2006), ainda com uma quantidade enorme de dimensões, a dificuldade computacional é reduzida já que o algoritmo de aprendizado é baseado na pseudo inversa da matriz de pesos da camada intermediária. Dessa forma esta abordagem é semelhante ao caso do aprendizado Hebbiano quando os padrões são ortonormais ou mesmo à solução OLAM (Optimal Linear Associative Memory)(Braga *et al.*, 2007). Portanto, este aprendizado não é iterativo e não requer, a princípio, uma grande quantidade de operações. Contudo, o número de neurônios da camada escondida influencia diretamente na quantidade de elementos da matriz pseudo-inversa, isto é no seu tamanho, e de produtos a serem calculados, assim o fato do método trabalhar com o aumento do espaço de características implica no aumento inevitável do custo computacional.

2.1.1.1 Aprendizado

As Máquinas de Aprendizado Extremo possuem uma motivação matemática relativamente simples, cujo algoritmo é baseado em uma abordagem na minimização da norma de mínimos quadrados.

Dado um conjunto de treinamento dado por:

$$(\mathbf{X}_l, \mathbf{Y}) = \{(\mathbf{x}_{l1}, y_1), (\mathbf{x}_{l2}, \mathbf{y}_2), \ldots, (\mathbf{x}_{lN}, y_N)\} \tag{2.4}$$

Em que Y são os rótulos de a cada padrão e X_l é o conjunto de características de cada padrão, dadas por:

$$\mathbf{x}_{li} = \{x_{l1}, x_{l2}, \ldots, x_{ln}\} \tag{2.5}$$

O método é proposto para uma rede com uma única camada escondida sem realimentação, uma SLFN (*Single hidden Layer Feedforward Network*), segundo modelo matemático em (2.6):

$$\sum_{i=1}^{d} \beta_i g(\mathbf{x}_{lj}) = \sum_{i=1}^{d} \beta_i \varphi(\mathbf{w}_i \cdot \mathbf{x}_{lj} + b_i) = t_j, \quad j = 1, \ldots, N \tag{2.6}$$

Em que d é a quantidade de neurônios na camada escondida, isto é equivalente a dimensão do espaço de mapeamento. E $\mathbf{w}_i = [w_{i1}, w_{i2}, \ldots, w_{in}]$ são os pesos das conexões entre as entradas e os neurônios da camada escondida, b_i são os seus limiares ou *bias* e β são os pesos das conexões dos neurônios da camada de saída da rede. o símbolo "$\cdot$" denota o produto interno dos vetores A função de ativação φ deve ser uma função não linear, não regular e infinitamente diferenciável em qualquer intervalo.

Uma matriz intermediária $\mathbf{H}$, denominada matriz de saída da camada escondida, é construída no espaço de características, de forma que:

$$\mathbf{H} = \begin{bmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,d} \\ h_{2,1} & h_{2,2} & \cdots & h_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ h_{N,1} & h_{N,2} & \cdots & h_{N,d} \end{bmatrix} \tag{2.7}$$

E os elementos de $\mathbf{H}$ sejam dados por:

$$h_{i,j} = \varphi(\mathbf{w}_i \cdot \mathbf{x}_{lj} + b_i), \quad i = 1, \ldots, d \; j = 1, \ldots, N \tag{2.8}$$

A equação em 2.6 pode, então, ser escrita na forma matricial como:

$$\mathbf{H}\beta = \mathbf{Y} \tag{2.9}$$

Com isso o treinamento da ELM é obtido a partir da minimização da norma quadrática da equação (2.9):

$$\|\mathbf{H}\hat{\beta} - \mathbf{Y}\| = \min_{\beta} \|\mathbf{H}\hat{\beta} - \mathbf{Y}\| \tag{2.10}$$

Assim os pesos estimados da camada de saída, $\hat{\beta}$ podem ser obtidos a partir da inversa generalizada de Moore-Penrose:

$$\hat{\beta} = \mathbf{H}^{\dagger}\mathbf{Y} \tag{2.11}$$

Com a equação (2.11), é possível definir, de forma simplificada, o algoritmo 2.1 de aprendizado do método.

Entradas: $\mathbf{X_1}$, $\mathbf{Y}$, d, φ
Saídas : $\mathbf{W}$, $\mathbf{b}$, $\hat{\beta}$
for $i \leftarrow 1$ **to** d **do**
  for $j \leftarrow 1$ **to** n **do**
    $w_{i,j} \leftarrow$ rand ; `// Atribui valores aleatórios para os pesos`
  end
  $b_i \leftarrow$ rand ; `// Atribui valores aleatórios para o bias`
end
for $i \leftarrow 1$ **to** d **do**
  for $j \leftarrow 1$ **to** N **do**
    $h_{i,j} \leftarrow \varphi(w_i \cdot x_j + b_i)$; `//` φ `é infinitamente diferenciável`
  end
end
$\hat{\beta} \longleftarrow \mathbf{H}^{\dagger}\mathbf{Y}$; `//` $\mathbf{H}^{\dagger}$ `é a inversa de Moore-Penrose`

Algoritmo 2.1: Aprendizado da ELM

2.1.1.2 Factibilidade

Os artigos de Liu *et al.* (2015) e Lin *et al.* (2015), do mesmo grupo de autores, compõe uma análise de factibilidade das ELMs dividida em duas partes.

O primeiro artigo (Liu *et al.*, 2015) mostra que a capacidade limite teórica de generalização atingida pelas ELMs não degrada mesmo que os pesos internos da rede não são ajustados, para certos tipos de função de ativação, como a polinomial, a sigmóide e a Nadaraya-Watson. Existem ainda uma quantidade mínima necessária de neurônios para atingir este limite teórico.

Outro aspecto importante avaliado no artigo é a utilização da técnica da generalização da inversa de Moore-Penrose, quando as funções de ativação são polinomiais, e da regularização de Tikhonov, quando elas não são. Foi constatado, ainda, que a capacidade de generalização dos métodos do tipo ELM não sofrem deterioração com o cálculo da inversa, mesmo que ela seja regularizada. Por outro lado, a capacidade de generalização sofre uma degradação quando a função de ativação não é apropriada, como, por exemplo, uma função Gaussiana que, quando utilizada em uma ELM, possui desempenho pior do que em uma Rede Neural Artificial comum, como a segunda parte do estudo (Lin *et al.*, 2015) evidencia.

Em (Lin *et al.*, 2015), outros os efeitos indesejados causados pela aleatoriedade do método também foram constatados. Neste caso, foi verificado que, para certos tipos de função de ativação para o Kernel Gaussiano, existem incertezas relacionadas à aproximação e ao aprendizado.

2.1.1.3 Mapeamento e Kernel

Foi observado por Frénay e Verleysen (2011) que o mapeamento explícito utilizado pela ELM com uma dada função de ativação é equivalente a um kernel mais complexo específico. A equação 2.12 é equivalente ao Kernel de ELM com função de ativação sigmoidal quando a quantidade de dimensões da camada escondida tende a infinito.

$$K(x_i, x_j) = \frac{2}{\pi} \arcsin \frac{1 + x_i \cdot x_j}{\sqrt{\left(\frac{1}{2\sigma^2} + 1 + x_i \cdot x_i\right)\left(\frac{1}{2\sigma^2} + 1 + x_j \cdot x_j\right)}} \tag{2.12}$$

O σ representa a variância da distribuição Gaussiana isotrópica que gerou a camada escondida da rede neural da ELM.

Uma vez que essa relação entre o mapeamento explícito e um Kernel complexo foi comprovada, ficou clara a caracterização deste mapeamento como um mecanismo possível de ser empregado em diversos métodos de aprendizado de máquina, tal como as Máquinas de Vetores de Suporte.

2.2 Vetores de Suporte

O classificador SVM utiliza, a partir de um kernel, um mapeamento implícito não linear afim de se atingir a separabilidade dos padrões. Este mapeamento permite a computação do produto escalar para a criação do espaço de características utilizando apenas alguns padrões no espaço de entrada. Os padrões escolhidos, denominados vetores de suporte, são capazes de definir um hiperplano que maximiza a margem de separação no espaço de características. A computação destes vetores de suporte é feita a partir de um problema de otimização quadrático e, com isso, um classificador com maximização de margem pode ser implementado (Cortes e Vapnik, 1995).

Estritamente, a abordagem utilizada pelas SVMs com a utilização de Kernels para promover a separabilidade dos dados é equivalente ao mapeamento explicito. Já é provado que a ordem das operações necessárias para determinação da superfície de separação pode ser invertida (Boser *et al.*, 1992), ou seja, a realização de uma transformação não linear seguida da avaliação dos vetores de suporte no espaço de características é equivalente a alguma comparação entre os padrões e o vetores de suporte no espaço de entrada, somente então, seguida da transformação não linear. Segundo Cortes e Vapnik (1995), esta propriedade foi fundamental para a superar as limitações computacionais presentes na ocasião da proposta das Máquinas de Vetores de Suporte.

De forma geral, a estratégia utilizada pelas Máquinas de Vetores de Suporte (Boser *et al.*, 1992; Cortes e Vapnik, 1995; Vapnik, 1995, 1998) consiste em determinar o hiperplano ótimo, o qual representa a superfície de separação com a margem maximizada. Para isso, é utilizado o princípio indutivo do método de minimização estrutural de risco. A implementação deste método de aprendizagem para problemas de classificação permite que a máquina atinja um bom desempenho de generalização ainda que o domínio do problema seja desconhecido. Os vetores de suporte, sobre os quais foi concebida esta categoria de máquina, são padrões extraídos dos dados de entrada que se encontram mais próximos da superfície de decisão. O que torna a sua classificação mais complexa e faz com eles que sejam mais significativos para a determinação do hiperplano ótimo de separação. Haykin (2001).

2.2.1 Hiperplanos Ótimos

O cálculo dos vetores de suporte e da superfície de separação com margem maximizada são obtidos com a resolução de problemas quadráticos, como foi detalhadamente descrito por (Cortes e Vapnik, 1995).

Dado os conjuntos de padrões rotulados:

$$(\mathbf{X}_l, \mathbf{Y}) = \{(\mathbf{x}_{l1}, y_1), (\mathbf{x}_{l2}, \mathbf{y}_2), \ldots, (\mathbf{x}_{lN}, y_N)\} \tag{2.13}$$

em que X_l é o conjunto de características de cada padrão, dadas por:

$$\mathbf{x}_{li} = \{x_{l1}, x_{l2}, \ldots, x_{ln}\} \tag{2.14}$$

e Y são os rótulos relativos a cada padrão, que podem assumir os seguintes valores:

$$y_i \in (-1, 1) \tag{2.15}$$

os padrões são ditos linearmente separáveis caso as condições estabelecidas pelas seguintes inequações sejam atendidas:

$$\begin{array}{lll} \mathbf{w_{SVM}} \cdot \mathbf{x}_{li} + b & \geq 1 \quad \text{se} & y_i = 1 \\ \mathbf{w_{SVM}} \cdot \mathbf{x}_{li} + b & \leq -1 \quad \text{se} & y_i = -1 \end{array} \tag{2.16}$$

Assim, deve haver um vetor de pesos $\mathbf{w_{SVM}}$ e um escalar bias b que sejam válidos para todo o conjunto de treinamento definido em (2.13).

As inequações em (2.16) podem ser reescritas na seguinte forma:

$$y_i(\mathbf{w_{SVM}} \cdot \mathbf{x}_{li} + b) \geq 1, \quad i = 1, 2, \ldots, N \tag{2.17}$$

O hiperplano ótimo é, então, definido como o uma superfície única que separa as duas classes com a margem máxima. Dessa forma ele pode ser descrito pela equação:

$$\mathbf{w_{SVM0}} \cdot \mathbf{x}_{li} + b_0 = 0 \tag{2.18}$$

Com isso a distância entre as projeções dos padrões do vetor de treinamento das diferentes classes atingem seu valor máximo na direção $\mathbf{w_{SVM}}/\|\mathbf{w_{SVM}}\|$. Dada esta a direção, as distâncias entre as projeções podem ser calculadas por:

$$\rho(\mathbf{w_{SVM}}, b) = \min_{\{x_l : y=1\}} \frac{\mathbf{x_l} \cdot \mathbf{w_{SVM}}}{|\mathbf{w_{SVM}}|} - \max_{\{x_l : y=-1\}} \frac{\mathbf{x_l} \cdot \mathbf{w_{SVM}}}{|\mathbf{w_{SVM}}|} \tag{2.19}$$

Substituindo a inequação (2.17) no cálculo da distância da equação (2.19), obtemos a seguinte relação:

$$\rho(\mathbf{w_{SVM}}, b) = \frac{2}{|\mathbf{w_{SVM}}|} \tag{2.20}$$

Assim é obtido o hiperplano ótimo, ao se maximizar o valor da distância na equação (2.20)resultando em uma valor de distância $\rho_{\max}$, determinado através da equação (2.21). Portanto é necessário que os valores de w_{SVM0} resultem da minimização de $\mathbf{w_{SVM}} \cdot \mathbf{w_{SVM}}$ segundo as restrições em (2.16).

$$\rho(\mathbf{w_{SVM0}}, b_0) = \frac{2}{\sqrt{\mathbf{w_{SVM0}} \cdot \mathbf{w_{SVM0}}}} \tag{2.21}$$

O problema de otimização pode, então, ser escrito na forma:

$$\begin{aligned} \min_{\mathbf{w_{SVM}}, b} \quad & \Phi(\mathbf{w_{SVM}}, b) = \mathbf{w_{SVM}} \cdot \mathbf{w_{SVM}} \\ \text{suj. a} \quad & y_i(\mathbf{w_{SVM}} \cdot \mathbf{x}_{li} + b) \geq 1, \qquad i = 1, \ldots, N. \end{aligned} \tag{2.22}$$

Utilizando o método de Langrange para a resolução do problema, são aplicados os multiplicadores de Lagrange $\Lambda = \alpha_1, \alpha_,2, \ldots, \alpha_N$ às penalidades impostas ao Lagrangiano do problema:

$$L(\mathbf{w_{SVM}}, b, \Lambda) = \frac{1}{2}\mathbf{w_{SVM}} \cdot \mathbf{w_{SVM}} - \sum_{i=1}^{N} \alpha_i \left[y_i \left(\mathbf{x}_{li} \cdot \mathbf{w_{SVM}} + b \right) - 1 \right] \tag{2.23}$$

Assim o ponto de mínimo é obtido com as derivações:

$$\frac{\partial L(\mathbf{w_{SVM}} = \mathbf{w_{SVM0}}, b, \Lambda)}{\partial \mathbf{w_{SVM}}} = \mathbf{w_{SVM0}} - \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_{li} \tag{2.24}$$

$$\frac{\partial L(\mathbf{w_{SVM}}, b = b_0, \Lambda)}{\partial \mathbf{w_{SVM}}} = \sum_{a_i=a_1}^{a_N} y_i a_i = 0 \tag{2.25}$$

Com essa determinação do mínimo da função de Lagrange, o valor dos pesos do hiperplano ótimo são definidos por:

$$\mathbf{w_{SVM0}} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_{li} \tag{2.26}$$

Neste momento também podem ser definidos os vetores de suporte, os quais são os padrões de que atendem a seguinte igualdade:

$$y_i(\mathbf{w_{SVM0}} \cdot \mathbf{x}_{li} + b_0) = 1 \tag{2.27}$$

Esta relação é observada quando as condições de Kuhn-Tucker são atendidas, com elas a igualdade na equação (2.28) deve ser verdadeira.

$$\alpha_i \left[y_i \left(\mathbf{w_{SVM}} \cdot \mathbf{x}_{li} + b \right) - 1 \right] = 0 \tag{2.28}$$

Dessa forma, os valores de α_i somente serão diferentes de zero quando existir a igualdade em 2.27. Por sua vez, os multiplicadores de Lagrange, a partir da maximização do problema quadrático obtidos com a substituição das equações (2.24) e (2.25) na equação de Lagrange (2.32). Assim a seguinte equação para determinar o vetor de pesos é encontrada:

$$W_{SVM}(\Lambda) = \Lambda^T \mathbf{1} - \frac{1}{2} \Lambda^T \mathbf{D} \Lambda \tag{2.29}$$

Na qual, $\mathbf{1}$ é um vetor unitário $N-$dimensional e $\mathbf{D}$ é uma matriz simétrica $N \times N$, cujos elementos são $d_{ij} = y_i y_j \mathbf{x}_i \mathbf{x}_j$.

2.2.1.1 Problema Regularizado

Quando há algum ruido nos dados e os padrões não podem ser separados linearmente utilizando este método, é possível introduzir uma variável de regularização no método para flexibilizar as restrições dos problemas e, assim, as margens.

Assim, devem ser incluídas variáveis não negativas que representam erros de classificação do grupo de treinamento, os quais são representados na forma:

$$\Xi = \{\xi_1, \xi_2, \ldots, \xi_N\} \tag{2.30}$$

O objetivo se torna a minimização dos valores do erro, ou seja de Ξ, com isso, problema de que passa a ser minimizado é:

$$\begin{array}{lll} \min\limits_{\mathbf{w_{SVM}},b,\xi} & \Phi(\mathbf{w_{SVM}},b,\xi) = \mathbf{w_{SVM}}\cdot\mathbf{w_{SVM}} + C\left(\sum\limits_{i=1}^{N}\xi_i\right)^2 & \\ \text{suj. a} & y_i(\mathbf{w_{SVM}}\cdot\mathbf{x}_{li}+b) \geq 1-\xi_i, & i=1,\ldots,N. \\ & \xi_i \geq 0, & i=1,\ldots,N. \end{array} \tag{2.31}$$

O segundo termo da função Φ foi elevado ao quadrado, mantendo a função monotônica e convexa, e foi multiplicado por uma constante C suficientemente grande. A partir dessa abordagem é encontrado um subconjunto de padrões que minimize os erros de treinamento.

Dessa forma o Lagrangiano do problema de otimização é dado por:

$$L(\mathbf{w_{SVM}},b,\xi,\Lambda,\mathbf{R}) = \frac{1}{2}\mathbf{w_{SVM}}\cdot\mathbf{w_{SVM}} + C\left(\sum_{i=1}^{N}\xi_i\right)^2 - \sum_{i=1}^{N}\alpha_i\left[y_i\left(\mathbf{x}_{li}\cdot\mathbf{w_{SVM}}+b\right)-1+\xi_i\right] - \sum_{i=1}^{N} r_i\xi_i \tag{2.32}$$

Resolvendo este Lagrangiano, obtém-se as seguintes igualdades:

$$\mathbf{w}_{\mathbf{SVM0}} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_{l\,i} \tag{2.33}$$

$$\sum_{i=1}^{N} \alpha_i y_i = 0 \tag{2.34}$$

$$\delta = \alpha_i + r_i \tag{2.35}$$

Em que $\sum_{i=1}^{N} \xi_i^0 = \left(\frac{\delta}{2C}\right)$.

Com isso o problema de otimização final para se obter os multiplicadores de Lagrange é dado por:

$$\begin{aligned} \max_{\Lambda,\delta} \quad & W_{SVM}(\Lambda,\delta) = \Lambda^T \mathbf{1} - \frac{1}{2}\left[\Lambda^T \mathbf{D} \Lambda + \frac{\delta^2}{C}\right] \\ \text{suj. a} \quad & \Lambda^T \mathbf{Y} = 0, \\ & \delta \geq 0, \\ & 0 \leq \Lambda \leq \delta \mathbf{1}. \end{aligned} \tag{2.36}$$

2.2.2 Máquinas de Vetores de Suporte Extremas

Liu *et al.* (2008) e He *et al.* (2011) propuseram um método denominado *Extreme Support Vector Machine*, no qual é determinada a orientação e a posição do plano hiperplano de separação em relação à origem. Neste método, ao invés de se calcular os vetores de suporte, as variáveis que definem o plano de separação são iterativamente atualizadas.

A seguinte função de decisão foi utilizada, com base no princípio das máquinas de vetores de suporte:

$$I(x) = \text{sign}\left(\sum_i \alpha_i \phi_i(x)\right) \tag{2.37}$$

A qual pode ser reescrita, em termos de pesos para os hiperplanos de separação, da seguinte forma:

$$\begin{aligned} \mathbf{w_{ESVM}} \cdot \Phi(\mathbf{x}_i) + b \geq 1, \quad & \text{se} \quad y_i = 1 \\ \mathbf{w_{ESVM}} \cdot \Phi(\mathbf{x}_i) + b \leq -1, \quad & \text{se} \quad y_i = -1 \end{aligned} \tag{2.38}$$

A função de mapeamento Φ na equação 2.1 é o produto escalar dos pesos que são aleatoriamente atribuídos, semelhantemente a ELM. A não-linearidade é imposta na transformação pode ser exponencial ou senoidal, entre outras, desde que satisfaçam as condições da ELM mencionadas na seção anterior. A função de ativação foi arbitrariamente escolhida:

$$\varphi(\mathbf{x'}_i) = \exp(-\mathbf{x'}_i) \tag{2.39}$$

A determinação do hiperplano ótimo $\{\mathbf{w_{ESVM0}} \cdot \Phi(\mathbf{x}) + b_0 = 0\}$ no espaço de características á atingida a partir da resolução do seguinte problema de otimização:

$$\begin{aligned} \min \quad & \mathbf{w_{ESVM}} \cdot \mathbf{w_{ESVM}}, \\ \text{sujeito a} \quad & y_i(\mathbf{w_{ESVM}} \cdot \Phi(\mathbf{x}_i) + b) \geq 1, i = 1, \ldots, N \end{aligned} \tag{2.40}$$

Os padrões cujo lagrangiano forem diferentes de 0 são os vetores de suporte. Uma vez que eles são obtidos, a função de decisão é reduzida a:

$$I(\phi) = \text{sign}\left(\sum_i \alpha_i \phi_i \cdot \phi + b_0\right), i \in \text{vet. suporte} \tag{2.41}$$

Neste caso , os pesos são dados por:

$$\mathbf{w_{ESVM0}} = \sum_{i=1}^{N} y_i \alpha_i^0 \Phi(\mathbf{x}_i) \tag{2.42}$$

Implicitamente, o método proposto por Liu *et al.* (2008), e mais tarde aperfeiçoado por He *et al.* (2011), calcula os hiperplanos de separação para um espaço de características com mapeamento aleatório para uma dimensão maior, a partir da criação de hiperesferas de camada de saída em que seus centros e raios são ajustados de forma a satisfazer a maximização de margem. Essa abordagem é bastante criativa e ilustra uma concepção diferente do problema de maximização de margem com a utilização do mapeamento explícito.

2.3 Aprendizado Semi-Supervisionado

As Máquinas de Vetores de Suporte Semi-Supervisionadas ou S3VM *Semi-Supervised Support Vector Machines*, são métodos de aprendizados Semi-Supervisionados baseados em máquinas transdutivas. Elas incorporam ao problema de otimização das máquinas de vetores de suporte um termo relacionado aos padrões não rotulados, segundo a seguinte equação:

$$\begin{aligned} \min_{\mathbf{w_{SVM}},b} \quad & \Phi(\mathbf{w_{SVM}},b) = \mathbf{w_{SVM}} \cdot \mathbf{w_{SVM}} \\ \text{suj. a} \quad & y_i(\mathbf{w_{SVM}} \cdot \mathbf{x}_{li} + b) \geq 1, \qquad i = 1,\ldots,N_l. \\ & (|\mathbf{w_{SVM}} \cdot \mathbf{x}_{ui} + b|) \geq 1, \qquad i = 1,\ldots,N_u. \end{aligned} \tag{2.43}$$

Em que o indicador l está relacionado aos dados rotulados (*labeled*) e o u aos dados não rotulados (*unlabeled*).

De forma geral a S3VM desenvolve um problema quadrático com penalizações atribuídas a partir das restrições. No entanto esta solução apresenta um problema empírico de desbalanceamento das predições classes para os dados não rotulados (Zhu e Goldberg, 2009). Com isso é, geralmente, introduzida uma heurística na forma de uma restrição que faz com que a proporção das classes atribuídas aos dados não rotulados seja igual a proporção das classes dos dados rotulados:

$$\frac{1}{N_u}\sum_{i=1}^{N_u} \mathbf{w_{SVM}} \cdot \mathbf{x}_{ui} + b = \frac{1}{N_l}\sum_{i=1}^{N_l} y_i \tag{2.44}$$

Este artifício, no entanto, faz com que o problema deixe de ser convexo, o que dificulta a solução do problema de otimização, a ponto de se impossibilitar a sua solução ótima.

CAPÍTULO 3

Desenvolvimento do Método

No contexto atual da classificação de padrões, as Máquinas de Vetores de Suporte ainda são consideradas técnicas no estado da arte. Elas apresentam altos índices de desempenho para diversos tipos de problemas reais e sintéticos. Apesar da grande variedade de novos métodos propostos na área nos últimos anos as SVMs se mantém como métodos de referência. Embora exista esta dificuldade em se aprimorar o desempenho das técnicas de classificação de padrões, a área possui outros aspectos que devem ser melhor estudados. Um dos quais é a minimização da quantidade de parâmetros livres a serem ajustados.

De forma geral os métodos utilizados em classificação de padrões possuem um ou mais parâmetros que requerem um ajuste preciso. Com isto, é necessário o emprego de técnicas bem elaborados de busca exaustiva para definir o valor ótimo para cada um destes parâmetros. O que se torna fundamental uma vez que, comumente, variações mínimas nestes parâmetros causam uma diminuição significativa no desempenho do método. A diferença entre duas superfícies de separação geradas pela mesma Máquina de Vetores de Suporte com o mesmo Kernel, cujos parâmetros, no entanto, deferem

Visto que existe tal limitação, são estudadas diversas abordagens afim de se melhorar os métodos utilizados atualmente. Com isso, foi elaborado um método capaz de atingir um desempenho de classificação semelhante ao da SVM que, no entanto, não possua parâmetros que precisem de ajuste.

Em resumo, este método consiste em uma Máquina de Vetores de Suporte que, ao invés de utilizar um kernel, faz explicitamente a projeção dos dados de entrada em um espaço de maior dimensionalidade com a utilização de uma rede neural. Esta projeção dos dados é baseada na solução proposta nas Máquinas de Aprendizado Extremo (ELM), em que uma rede neural com uma grande quantidade de neurônios e a função de ativação não-linear é a camada intermediária de uma *Single Layer Feedforward Network*. Este mapeamento em um espaço de maior dimensionalidade promove a separabilidade de tal forma que a camada de saída da ELM é calculada analiticamente a partir da pseudo-inversa da camada intermediária, análogo à OLAM. Por outra perspectiva a utilização da pseudo-inversa é equivalente a utilização da

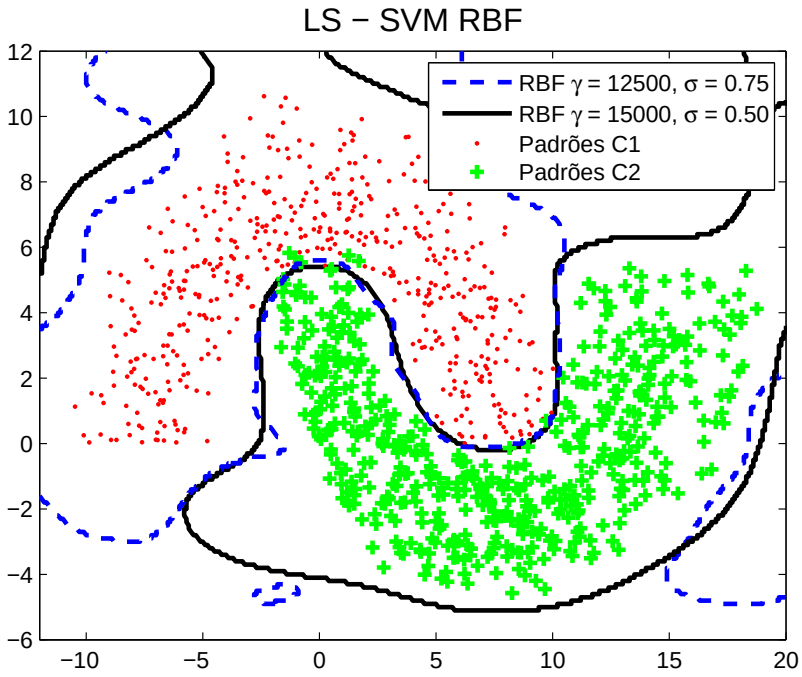


Figura 3.1: Problema de classificação duas meia-luas resolvido por duas LS-SVM de kernel Gaussinao com parâmetros γ e σ^2 diferentes.

matriz transposta da Regra de Hebb no caso em que os padrões de entrada formam uma base ortonormal. É importante observar que, para que seja possível a utilização da pseudo-inversa, é necessário que a matriz de entrada tenha posto completo, do contrário, não é possível sua determinação. Como a quantidade de neurônios da camada escondida é muito grande e a função de ativação é não linear é probabilidade da obtenção de uma matriz de posto completo é alta. Com isso, é possível obter uma solução ótima e única para a minimização do erro de classificação (Braga *et al.*, 2007). Contudo, a maximização da margem ainda não está garantida, já que, para tanto, são necessárias outras premissas.

As ELMs possuem um bom desempenho e, devido a sua simplicidade, tem se difundido bem. No entanto, a abordagem de mapeamento explícito complementaria a maximização de margem das SVM de forma interessante, uma vez que integraria um método não paramétrico com um método com um grande poder de generalização. A capacidade de generalização de classificadores de padrões está relacionada a maximização da margem de classificação (Smola e Bartlett, 2000). Existe na literatura diversas técnicas com esta finalidade, dentre elas a SVM é uma das mais utilizadas.

Tendo em vista as características promissoras do mapeamento explícito empregado na

Máquinas de Aprendizado Extremo, é possível combinar a esta abordagem às Máquinas de Vetores de Suporte, em conformidade com a proposta original de Cortes e Vapnik (1995). A proposta é substituir o mapeamento das SVM, feitos com a utilização de um kernel, pelo mapeamento explícito a partir de uma RNA com a função de ativação não linear. Neste caso, os padrões de entrada são projetados em um espaço de características de maior dimensionalidade onde são determinados os vetores de suporte.

No método proposto de máquinas de vetores de suporte com mapeamento explícito, o próprio espaço de características é considerado de forma explícita na construção do hiperplano de separação. Com isso, o espaço de características é criado previamente à extração dos vetores de suporte. Assim é construído um kernel linear simples para substituir um núcleo do produto interno arbitrário mais complexo. A equivalência do mapeamento explícito a um kernel de maior complexidade foi exemplificado por Frénay e Verleysen (2011).

Com esta abordagem, é possível atingir resultados similares a SVMs com kernels mais complexos, além de apresentar também a vantagem de não ajustar nenhum parâmetro. É possível fazer essa consideração, uma vez que o aumento da quantidade de neurônios da RNA reflete em um efeito assimptótico sobre a exatidão da classificação. Ou seja, se a quantidade de neurônios for grande o suficiente, a exatidão da rede tenderá ao melhor resultado possível, independentemente do valor exato de neurônios, desta forma o método não é sensível a este parâmetro.

A partir do método de classificação proposto, foi também implementada uma solução para o problema de aprendizado semi supervisionado. Neste contexto, são apresentados ao algoritmo de treinamento dados rotulados, ou seja que possuem uma saída esperada, e dados não rotulados, para os quais as informações intrínsecas da sua distribuição são utilizadas para o agrupamento dos dados. Em particular, esta variedade de problemas apresenta, de forma geral, uma quantidade muito maior de dados sem rótulos.

A abordagem escolhida para a solução do problema semi supervisionado é semelhante a *cluster-then-label*, e se resume em quatro etapas principais. Primeiramente é implementado um classificador utilizando os dados rotulados a partir do método de mapeamento explícito para SVMs apresentado neste trabalho. Em seguida é realizado o agrupamento dos dados rotulados e dos não rotulados utilizado um método genérico do algoritmo *k-means*. A partir da avaliação dos rótulos presentes em cada cluster é possível determinar um rótulo provável para cada cluster ou se é necessário dividir algum agrupamento em conjuntos menores afim de se atribuir um rótulo mais apropriado. Na etapa seguinte, a partir da função de decisão obtida pelo

primeiro método e dos clusters rotulados obtido pelo segundo, são definidas regiões em que ambos os métodos concordam na classificação, as quais são consideradas regiões de certeza de classificação. São também definidas as regiões em que não há consenso entre os dois métodos, neste caso foram observadas possíveis estratégias a serem tomadas que dependem dos motivos que acarretaram os resultados discrepantes.

3.1 Métodos Propostos

3.1.1 SVM com Mapeamento Explícito

O método proposto combina uma máquina de vetores de suporte com mapeamento explícito, na forma de um Kernel virtualmente aparamétrico. Como o método foi desenvolvido para lidar com classes de padrões que possuam uma superposição intrínseca, foi admitida a utilização das SVM regularizadas, com uma constante C, que pode, também, ser ajustada.

O método apresenta alguns elementos que devem ser parâmetros de entrada ou escolhidos durante o projeto, que são: A base de dados de treinamento, com características e rótulos para os padrões; uma função aleatória para gerar o matriz $\mathbf{W}$ de pesos; a função de ativa não linear não regular e infinitamente diferenciável; a constante C para a regularização da SVM; e a dimensão d da matriz $\mathbf{W}$. Dentre estes parâmetros, o primeiro é a variável de entrada do método e os outros podem ser obtidos como escolha de projeto, de forma que o método se adeque ao problema a ser resolvido. Assim, não há a necessidade de realizar o ajuste de nenhum parâmetro.

Dados os conjuntos de padrões de treinamento:

$$(\mathbf{X}_l, \mathbf{Y}) = \{(\mathbf{x}_{l1}, y_1), (\mathbf{x}_{l2}, \mathbf{y}_2), \ldots, (\mathbf{x}_{lN}, y_N)\} \tag{3.1}$$

Cujos rótulos estão no vetor Y e X_l contém as características de cada padrão, com a seguinte forma:

$$\mathbf{x}_{li} = \{x_{l1}, x_{l2}, \ldots, x_{ln}\} \tag{3.2}$$

O mapeamento é feito segundo o mapeamento da ELM (Huang *et al.*, 2006) na forma:

$$\Phi : X \to E^d \tag{3.3}$$

Utilizando um mecanismo semelhante a uma rede neural com uma camada, ou seja a matriz $\mathbf{X}$ os padrões de entrada são multiplicados por uma matriz de pesos $[\mathbf{W}_{n\times d}$, com isso a matriz intermediária $\mathbf{H}$ pode ser criada da seguinte forma:

$$\mathbf{H}(x) = \begin{bmatrix} \varphi(\mathbf{w}_1 \cdot \mathbf{x}_{l1} + b_1) & \varphi(\mathbf{w}_2 \cdot \mathbf{x}_{l1} + b_2) & \cdots & \varphi(\mathbf{w}_d \cdot \mathbf{x}_{l1} + b_d) \\ \varphi(\mathbf{w}_1 \cdot \mathbf{x}_{l2} + b_1) & \varphi(\mathbf{w}_2 \cdot \mathbf{x}_{l2} + b_2) & \cdots & \varphi(\mathbf{w}_d \cdot \mathbf{x}_{l2} + b_d) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi(\mathbf{w}_1 \cdot \mathbf{x}_{lN} + b_1) & \varphi(\mathbf{w}_2 \cdot \mathbf{x}_{lN} + b_2) & \cdots & \varphi(\mathbf{w}_d \cdot \mathbf{x}_{lN} + b_d) \end{bmatrix} \tag{3.4}$$

É possível considerar a matriz $\mathbf{H}$ como o espaço de características de treinamento do método e, a partir da teoria de SVMs (Cortes e Vapnik, 1995), calcular, nesse novo espaço, os vetores de suporte e definir o classificador linear.

Assim a otimização estudada na seção 2.2 pode ser aplicada. O hiperplano ótimo de separação é desejado então as inequações em (3.5) devem ser atendidas.

$$\begin{aligned} \mathbf{w} \cdot \mathbf{h}_{li} + b &\geq 1 && \text{if} \quad y_i = 1 \\ \mathbf{w} \cdot \mathbf{h}_{li} + b &\leq -1 && \text{if} \quad y_i = -1 \end{aligned} \tag{3.5}$$

O método desenvolvido enfatiza problemas que possuem uma certa superposição dos dados, de forma que a robustez de seu Kernel seja melhor aproveitada. Dessa forma foi empregado o o problema de otimização regularizado, em que as variáveis de erro $\Xi = \{\xi_1, \xi_2, \ldots, \xi_N\}$ são introduzidas e o problema de otimização é dado por:

$$\begin{aligned} \min_{\mathbf{w},b,\xi} \quad & \Phi(\mathbf{w}, b, \xi) = \mathbf{w} \cdot \mathbf{w} + C \left(\sum_{i=1}^{N} \xi_i \right)^2 \\ \text{suj. a} \quad & y_i(\mathbf{w} \cdot \mathbf{h}_{li} + b) \geq 1 - \xi_i, && i = 1, \ldots, N. \\ & \xi_i \geq 0, && i = 1, \ldots, N. \end{aligned} \tag{3.6}$$

Ao se resolver este problema de otimização utilizando um método tradicional Lagrangiano, a equação de Lagrange encontrada é:

$$L(\mathbf{w},b,\xi,\Lambda,\mathbf{R}) = \frac{1}{2}\mathbf{w}\cdot\mathbf{w} + C\left(\sum_{i=1}^{N}\xi_i\right)^2 - \sum_{i=1}^{N}\alpha_i\left[y_i\left(\mathbf{h}_{li}\cdot\mathbf{w}+b\right)-1+\xi_i\right] - \sum_{i=1}^{N}r_i\xi_i \quad (3.7)$$

Cuja solução é:

$$\mathbf{w}_0 = \sum_{i=1}^{N}\alpha_i y_i \mathbf{h}_{li} \quad (3.8)$$

$$\sum_{i=1}^{N}\alpha_i y_i = 0 \quad (3.9)$$

$$2C\left(\sum_{i=1}^{N}\Xi_i^0\right) - \alpha_i - r_i = 0 \quad (3.10)$$

A igualdade (3.10) pode ser reescrita da seguinte forma:

$$\delta = \alpha_i + r_i \quad (3.11)$$

Se for considerado que $\sum_{i=1}^{N}\xi_i^0 = \left(\frac{\delta}{2C}\right)$.

Os multiplicadores de Lagrange $\Lambda = \alpha_1, \alpha,2, \ldots, \alpha_N$ podem, então, ser encontrados com a resolução do problema 3.12.

$$\begin{aligned} \max_{\Lambda,\delta} \quad & \mathbf{W}(\Lambda,\delta) = \Lambda^T\mathbf{1} - \frac{1}{2}\left[\Lambda^T\mathbf{D}\Lambda + \frac{\delta^2}{C}\right] \\ \text{suj. a} \quad & \Lambda^T\mathbf{Y} = 0, \\ & \delta \geq 0, \\ & \mathbf{0} \leq \Lambda \leq \delta\mathbf{1}. \end{aligned} \quad (3.12)$$

Em que $\mathbf{D}$ é a matriz simétrica $N \times N$, cujos elementos são $d_{ij} = y_i y_j \mathbf{h}_i \mathbf{h}_j$.

Por fim, o classificador é formado a partir da seguinte equação:

$$\hat{y} = \text{sign} f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{N} y_i \alpha_i \mathbf{H}_i(\mathbf{x}_l)\right) \cdot \mathbf{H}(\mathbf{x}) \tag{3.13}$$

A parir desse desenvolvimento matemático, é possível definir o algoritmo 3.1 de treinamento do método proposto, que pode ser considerada uma abordagem de Mapeamento Explícito para Máquinas de Vetores de Suporte.

Entradas: $\mathbf{X_l}$, $\mathbf{Y}$, d, φ, C
Saídas : $\mathbf{W}$, $\mathbf{b}$, Λ
for $i \leftarrow 1$ **to** d **do**
  for $j \leftarrow 1$ **to** n **do**
    $w_{i,j} \leftarrow$ rand ; `// Atribui valores aleatórios para os pesos`
  end
  $b_i \leftarrow$ rand ; `// Atribui valores aleatórios para o bias`
end
for $i \leftarrow 1$ **to** d **do**
  for $j \leftarrow 1$ **to** N **do**
    $h_{i,j} \leftarrow \varphi(w_i \cdot x_j + b_i)$
  end
end
`// Cria a Matriz` $\mathbf{D}$ `para o problema de Otimização 3.12`
for $i \leftarrow 1$ **to** N **do**
  for $j \leftarrow 1$ **to** N **do**
    $d_{ij} = y_i y_j \mathbf{h}_i \mathbf{h}_j$.
  end
end
`// O Problema de Otimização na equação 3.12 é resolvido`
while $\mathbf{W} \neq \max \mathbf{W}$ **do**
  $\delta \leftarrow \alpha_{\max}$; `// A terceira condição de contorno implica nesta igualdade`
  $\max \Lambda^T \mathbf{1} - \frac{1}{2}\left[\Lambda^T \mathbf{D} \Lambda + \frac{\delta^2}{C}\right]$; `// Resolve o problema quadrático`
end

Algoritmo 3.1: Aprendizado Proposto para SVMs com Mapeamento Explícito

3.1.2 Aprendizado Semi-Supervisionado

O problema de aprendizado semi-supervisionado pode ser desenvolvido a partir da equação:

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \Phi(\mathbf{w},b) = \mathbf{w}\cdot\mathbf{w} \\ \text{suj. a} \quad & y_i(\mathbf{w}\cdot\mathbf{x}_{li}+b) \geq 1, \quad i = 1,\ldots,N_l. \\ & (|\mathbf{w}\cdot\mathbf{x}_{ui}+b|) \geq 1, \quad i = 1,\ldots,N_u. \end{aligned} \tag{3.14}$$

Neste caso é possível criar dois problemas convexos, aplicando uma penalizações a partir da primeira restrição e, separadamente, definindo uma segunda função objetivo, resultando, assim, em duas funções objetivo:

$$L(\mathbf{w},b,\Lambda) = \frac{1}{2}\mathbf{w}\cdot\mathbf{w} - \sum_{i=1}^{N}\alpha_i\left[y_i\left(\mathbf{x}_{li}\cdot\mathbf{w}+b\right)-1\right] \tag{3.15}$$

$$L(\mathbf{w},b,\Lambda) = -\sum_{i=1}^{N_u}\left[|\mathbf{x}_{li}\cdot\mathbf{w}+b|-1\right] \tag{3.16}$$

Como não existe uma necessidade de se regularizar a superfície de separação, já que os rótulos não são conhecidos, então o método é baseado na formulação não regularizada.

A partir das equações 3.15 e 3.16, é possível criar um problemas multiobjetivo em que ambos os objetivos são convexos e, portando, possuem uma solução relativamente simples.

Esta solução multiobjetivo, foi estabelecida uma vez que o problema empírico de desbalanceamento observado na literatura (Zhu e Goldberg, 2009), é, provavelmente, causado pela própria implementação baseada em penalizações que, para diversos casos, são muito dificilmente controláveis.

Da mesma forma, a heurística utilizada para resolver este erro sistemático, além de dificultar desnecessariamente o problema de otimização tornado-o não convexo, não possui uma fundamentação válida para a maioria das bases de dados reais. Por exemplo, em um problema de diagnóstico pode existir uma preocupação maior em se diagnosticar os casos mais característicos de serem positivos, com isso os dados rotulados podem apresentar uma proporção maior de diagnósticos positivos do que um grupo heterogêneo de dados não rotulados.

Observando a equação 3.16, é possível perceber que também pode existir uma importante limitação relacionada ao balanceamento das classes. De forma geral, a região de menor densidade de pontos está ao redor dos agrupamentos, isso pode causar um desbalanceamento se for considerado que todos os pontos pertencem ao mesmo agrupamento. Para evitar este problema podem ser incluídas duas restrições da seguinte forma:

$$\begin{aligned} &\max\left(\mathbf{w}\cdot\mathbf{x}_{ui}+b\right)\geq 1 \\ &\min\left(\mathbf{w}\cdot\mathbf{x}_{ui}+b\right)\leq -1 \end{aligned} \tag{3.17}$$

CAPÍTULO 4

Metodologia

4.1 Desempenho da SVM com Mapeamento Explícito

A fim de se avaliar os métodos propostos foram necessários uma série de testes para se identificar as suas principais características de desempenho, no caso a exatidão da classificação e o tempo de treinamento foram verificados. Tendo como princípio essas duas características, foram elaboradas metodologias para os experimentos que minimizassem ou, ao menos, identificassem efeitos extrínsecos ao método.

Primeiramente a performance do método precisa ser avaliada de forma comparativa com outros métodos existentes na literatura, por isso foram escolhidos a SVM, na forma da a LS-SVM (*least square support vector machines*), e a ELM. A LS-SVM foi implementada utilizando um Kernel RBF, enquanto o método proposto, a partir deste momento denominado EMS (*Explicit Mapping* SVM), e a ELM foram implementados utilizando como função de ativação uma simples sigmoidal:

$$\varphi(\mathbf{x'}_i) = \frac{1}{1+\exp(-\mathbf{x'}_i)} \tag{4.1}$$

Tendo em vista os três métodos, o proposto e os dois escolhidos, foi, então, elaborado um experimento para aferir o desempenho do método proposto em relação aos demais. A medida de desempenho escolhida foi a exatidão na forma da taxa de acertos de classificação do conjunto de testes:

$$\text{taxa de acerto} = \frac{\text{quantidade de acertos de classificação}}{\text{tamanho do conjunto de teste}} \tag{4.2}$$

O desempenho do método de classificação proposto possui um parâmetro que merece um certo cuidado em sua análise, a quantidade de neurônios na camada escondida. É aceito que, se a dimensão do espaço de características gerado por estes neurônios for grande o suficiente, a

função de separação dos dados será linear. Neste sentido existe a questão de se determinar qual tamanho mínimo necessário da camada escondida para que este comportamento seja observado e o desempenho da rede atinja o seu máximo. O comportamento da rede com camadas escondidas com menos neurônios é igualmente relevante, uma vez que a partir da curva empírica do desempenho em relação a quantidade de neurônios é possível caracterizar sensibilidade do método a esse parâmetro.

Neste contexto, a exatidão do método, a partir do percentual de acertos, e a dificuldade em se gerar uma superfície de separação, por meio da quantidade de vetores de suporte encontrados, foram caracterizadas para uma quantidade crescente de neurônios na camada escondida, isto é, de dimensões do espaço de características. A quantidade de vetores de suporte deve ser observada já que a separabilidade dos problemas tendem a aumentar a medida que a dimensão da camada escondida aumenta, com isso é admissível que a quantidade de vetores de suporte necessários para garantir a separação e a maximização de margem tende a diminuir.

Tabela 4.1: Blocos e Níveis do Fator do Experimento para a SVM com Mapeamento Explícito

Fator	Níveis								
d	10	20	40	80	160	320	640	1280	2560
N[1]	1000			1500			2000		
Blocos	Grupos de Experimento								
Bases	Gauss	Spiral	Circle	Ion	PID	BLD	Iris	Wine	

O método proposto possui, neste caso, um fator e um conjunto de blocos que devem ser avaliados, a quantidade d de dimensões do mapeamento, ou neurônios na camada escondida, e as bases de dados para classificação, respectivamente. Os níveis atribuídos a ambos estão relacionados na Tabela 4.2 bases de dados dados são detalhadas na Seção 4.4. Com isso as medições a serem realizadas devem compreender os 72 pares de combinações possíveis entre os níveis do fator e os blocos observados.

São ajustados também, no caso das bases de dados sintéticas a quantidade de padrões de entrada N e, para a base *Circle* da Tabela 5.1, a quantidade de dimensões do problema n. Para esses casos entrarem na análise dos resultados é necessário que a análise dos blocos seja feita em duas partes, uma com os dados sintéticos e outra com os dados reais. No caso das bases de dados Sintéticas, existe um fator de três níveis a mais, o que resulta em 216 testes, para cada base de dados.

[1] Somente Para Dados Sintéticos

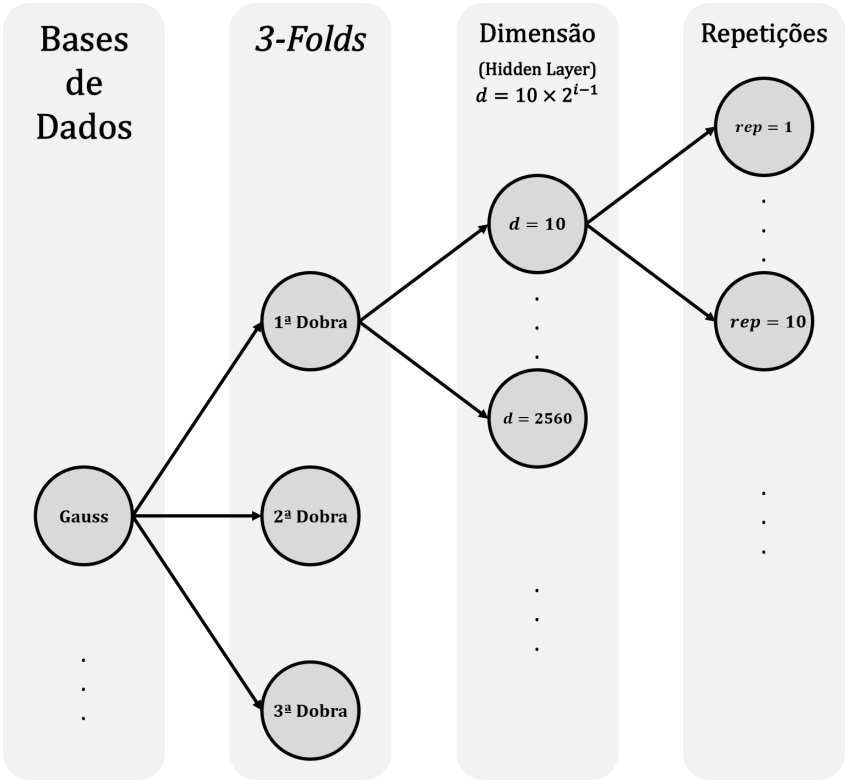


Figura 4.1: Diagrama do experimento realizado com bases de dados reais.

Devido a natureza aleatória do método, é necessário não só a aleatorização dos dados de entrada, mas também a repetição da medição de forma a garantir a estimação da sua variabilidade. Cada uma das bases de dados foram aleatorizadas utilizando o método ***k**-fold* para determinar os grupos de treinamento e teste, com 3 *folds*. E, uma vez que os valores dos pesos dos neurônios e os bias são atribuídos aleatoriamente, para cada uma das dobras obtidas o treinamento é realizado 10 vezes. A estrutura do experimento está exemplificada em resumo na Figura 4.2.

Para os outros dois métodos serão realizadas testes com as mesmas dobras definidas anteriormente e no caso da ELM, também serão realizadas as 10 repetições. Os Kernels utilizados pelas máquinas de vetores de suporte e os valores dos seus parâmetros foram escolhidos automaticamente segundo heurísticas provenientes dos dados de treinamento. Os resultados foram ainda conferidos em concordância com o benchmark desenvolvido por Van Gestel *et al.* (2004). A utilização destes recurso permite uma comparação mais justa em relação aos demais métodos na literatura.

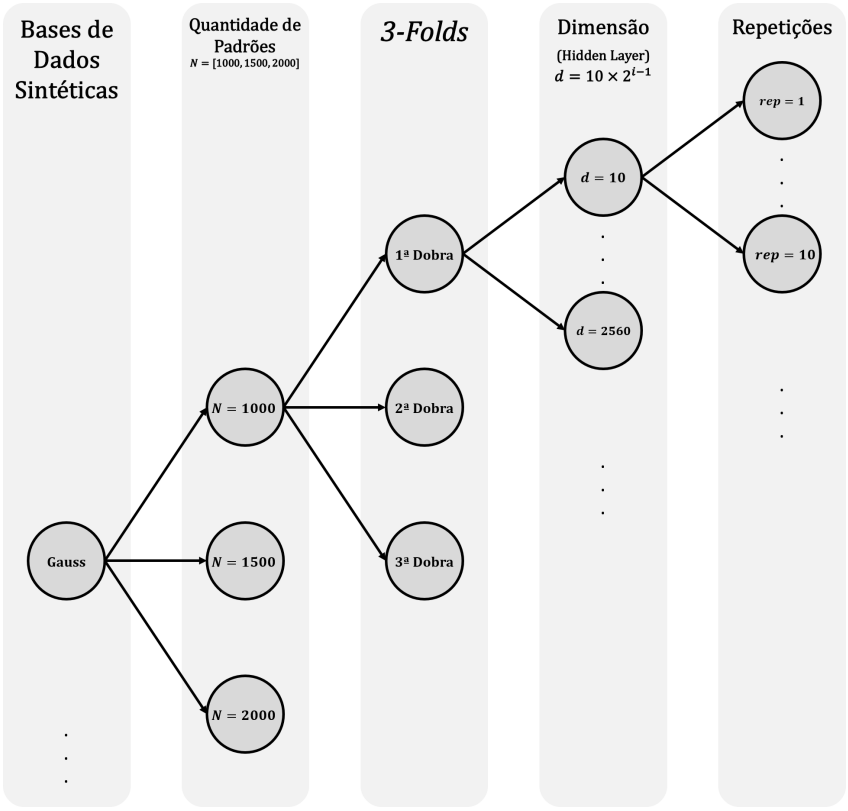


Figura 4.2: Diagrama do experimento realizado com bases de dados sintéticos.

4.2 Tempo de Treinamento

O tempo de treinamento observado é, de fato, o custo computacional para se resolver o problema em cada uma das configurações possíveis. Para a sua análise, é preciso identificar os fatores que podem exercer qualquer influenciar nele, que podem ser, além do tamanho da camada escondida, a quantidade de padrões de treinamento, a dimensão do problema e a sua própria complexidade inata. Com isso é possível verificar como a dimensão dos padrões de entrada e a sua quantidade, além da dimensão da camada escondida afetam o custo computacional do método proposto.

É possível considerar que um mesmo problema formulado com diferentes dimensões permite a análise do efeito da dimensionalidade da entrada desconsiderando a sua complexidade, que é mantida constante. Neste sentido a complexidade é isolada utilizando um problema básico específico que, no caso, é um círculo em dentro de um quadrado em duas dimensões. Este problema pode ser escalonado para uma esfera dentro de um cubo em três dimensões e em seguida para a hiperesfera dentro do hipercubo para as demais dimensões, como apresentado

nos dados sintéticos da Seção 4.4.1.

O experimento para a análise do tempo assume uma configuração diferente. Somente uma base de dados é utilizada, por isto não há a necessidade da sua separação em blocos, no entanto é feita a blocagem quanto a quantidade de padrões de entradas uma vez que não há a necessidade de se verificar o efeito da quantidade de padrões de entrada no tempo de treinamento neste momento. Isto foi determinado uma vez que é desejável, em treinamento de máquinas, que se utilize o máximo possível de informações disponíveis para a caracterização do problema. A dimensão n dos padrões entrada e a dimensão d da camada escondida são, dessa forma, os fatores do experimento.

Tabela 4.2: Blocos e Níveis do Fator do Experimento para o Avaliação de Tempo, a Base de Dados é a *Circle*

Fatores	Níveis								
d	10	20	40	80	160	320	640	1280	2560
n	2		5		10		20		40
Blocos	Grupos de Experimento								
N	1000			1500			2000		

Uma representação gráfica do experimento está presenta na figura 4.3.

4.3 Padronização

A função não-linear utilizada neste estudo requer argumentos de valor positivo. Além disso, existe a necessidade de se minimizar os efeitos de valores muito discrepantes dos atributos. Assim, melhor solução é a padronizados de cada atributos de todos os dados para o intervalo $[0, 1]$.

Existe, no entanto, uma discussão sobre a necessidade de se normalizar os dados de entrada, por um lado há a necessidade de se regularizar as informações de forma que uma característica não seja mais significativa que a outra. Como é o caso de um problema de classificação de pessoas em que o peso de cada indivíduo em quilogramas não pode ser mais significativo do que a sua altura em metros por que tal medida é uma ordem de grandeza maior, além disso este panorama mudaria totalmente se a altura fosse atribuída em centímetros. Essas irregularidades devem, no contexto do aprendizado de máquina, ser devidamente ajustadas.

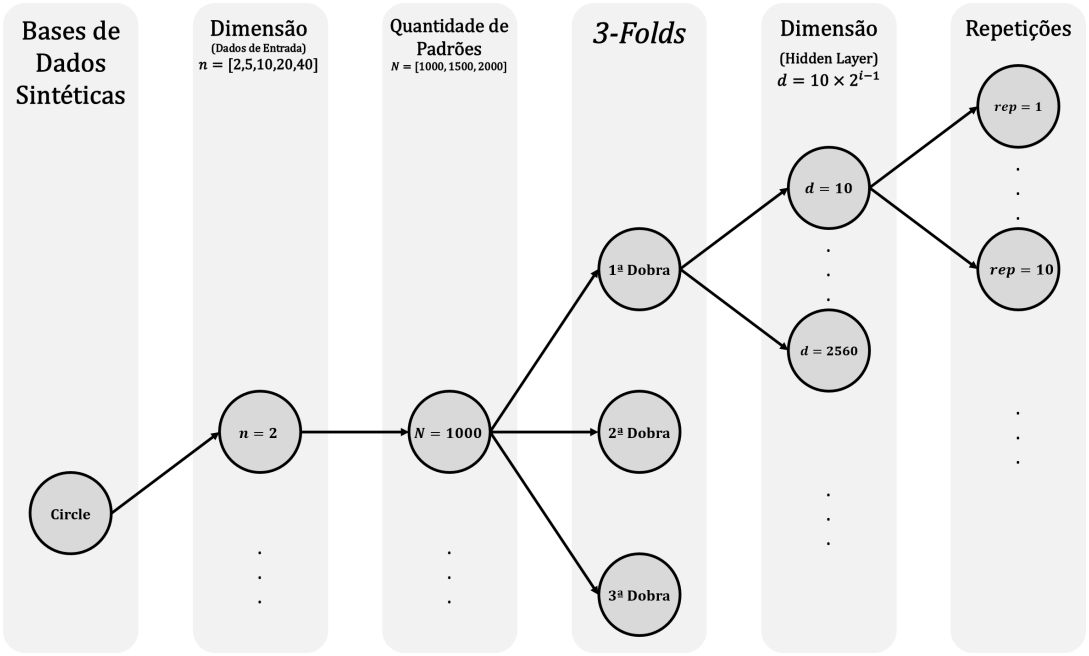


Figura 4.3: Diagrama do experimento de análise do tempo realizado com a base de dados *Circle*.

Entretanto, há a questão da máquina, durante o treinamento, ter acesso somente a um conjunto limitado de dados, e que a normalização não poderia ter acesso aos dados de validação ou teste. Esta restrição pode ser contornada por uma estratégia apropriada a cada problema. Se for considerado que o conjunto de amostras de treinamento representam de forma significativa o universo dos dados é possível criar uma função de normalização que aplique uma correção, determinada estatística ou empiricamente, aos os valores que caracterizam a distribuição, como a média ou os valores máximo e mínimo.

Por exemplo, um valor máximo pode ser estimado para a padronização como $\widehat{\max} = \max + 0.1(\max - \min)$. De outra forma pode-se admitir que o programador tenha conhecimento suficiente do universo do problema para criar arbitrariamente os critérios de normalização. Por exemplo, no caso da altura de seres humanos, é admissível supor que não há seres humanos adultos com mais de 2,5 metros de altura nem com menos de 50 cm. Com isso, é possível considerar que estes valores estão suficientemente perto dos limites de altura de seres humanos para que a normalização seja satisfatória. Inclusive podem ser utilizados dados estatísticos da população alvo para estimar melhor tais valores.

4.4 Bases de Dados

Para os testes foram utilizadas Bases de Dados cujas características são todas numéricas, principalmente, pelo fato de não ter sido definido o comportamento dos algoritmos para variáveis categóricas e, dessa forma, as implementações não foram feitas de forma a lidar com as características não mensuráveis, como forma, cor ou gênero.

Afim de se realizar uma análise dos métodos mais expressiva, foram realizados experimentos tanto com bases sintéticas quanto com bases reais, sendo suas quantidades respectivamente 3 e 4 e elas estão descritas nas tabelas 5.1 e 4.4.

4.4.1 Dados Sintéticos

As três bases de dados sintéticas estão especificadas na seguinte tabela:

Tabela 4.3: Descrição das Bases de Dados Sintéticas

Nome	Atributos	Quantidade de Padrões	Descrição
Gaussian	2	500 1000 2000	Duas distribuições Gaussianas levemente superpostas
Spiral	2	500 1000 2000	Duas espirais entrelaçadas com ruído (sd=0.05).
Circle	2	500 1000 2000	Um círculo ou uma hiperesfera dentro de um quadrado ou hipercubo.
	5	500 1000 2000	
	10	500 1000 2000	
	20	500 1000 2000	
	40	500 1000 2000	

4.4.2 Dados Reais

As bases de dados reais foram obtidas do Repositório de Aprendizado de Máquina da UCI Bache e Lichman (2013), e estão relacionados na tabela a seguir.

Tabela 4.4: Descrição das Bases de Dados Reais

Nome	Atrib.	Padr.	Class.	Descrição
Iono-sphere (Ion)	33	351	2	Elétrons livres observados na ionosfera, foi avaliada a existência de estrutura.
Pima Indian Diabetes (PID)	8	392	2	Dados clínicos de diagnóstico de diabetes.
Bupa Liver Disease (BLD)	6	345	2	Exames de sangue e quantidade de bebida para diagnóstico de doenças hepáticas em homens.
Iris (Iri)	4	150	3	Catálogo de três variedades da flor Íris.
Wine (Win)	13	178	3	Chemical analysis of green wines for discriminating three different types.

CAPÍTULO 5

Resultados

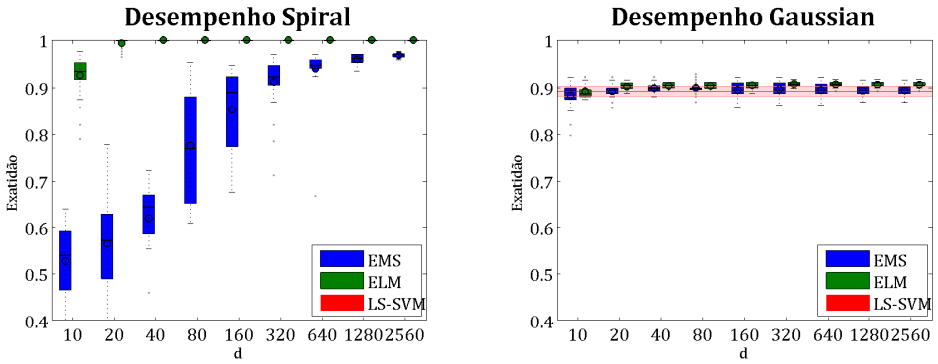
5.1 Dados Sintéticos

A Tabela 5.1 apresenta os resultados para todos os testes realizados com dados sintéticos. Nela é possível verificar a melhora do desempenho do método proposto, aqui denominado EMS (*Explicit Mappinf SVM*), à medida que a quantidade de neurônios da camada de mapeamento aumento. O seu desempenho pode, também, ser comparado ao da ELM padrão e com uma LS-SVM RBF.

Como pode ser observado na Tabela 5.1, as quantidades de padrões de entrada escolhidos tem pouco efeito sobre o desempenho da rede, por este motivo somente são apresentados gráficos para $N = 500$. Neste contexto não é possível afirmar que a quantidade de padrões de treinamento não tem efeito sobre o desempenho. É mais provável que a faixa de valores escolhidos já estivesse atingido um resultado limite, e que conjuntos de treinamentos menores degradariam o desempenho das máquinas. Entretanto a escolha de tais valores para a quantidade de padrões foi motivada pela análise do tempo computacional do método.

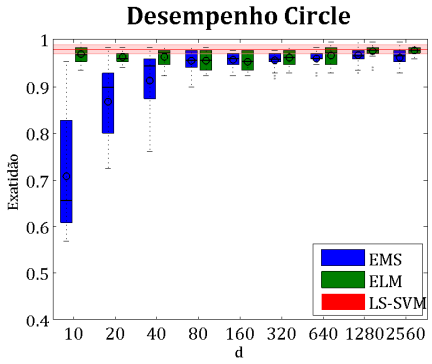
Os gráficos apresentados na Figura 5.1 ilustram o desempenho do método proposto, o EMS, em comparação á ELM padrão caom diferentes tamanhos da camada escondida e à LS-SVM com kernel RBF. Nesta figura é possível perceber que, com o aumento de d, o desempenho do método proposto é estatisticamente semelhante aos desempenhos da ELM e da LS-SVM RBF para os casos das bases de dados Gaussian e Circle. No entanto isto não é observado no teste realizado com a base de dados Spiral, no qual o desempenho do método proposto é claramente inferior. Este comportamento ocorre por causa da superposição entre as classes classificadas, tanto a base de dados Gaussian quanto a Circle apresentam uma superposição entre as classes de forma que uma fronteira entre elas não é suave, enquanto a base de dados Spiral, mesmo com as classes entrelaçadas apresenta uma fronteira clara entre as classes. O método proposto tem um aproveitamento melhor no primeiro caso.

A Figura 5.2 indica que o desempenho do método proposto é superior ao da ELM à



(a) Spiral

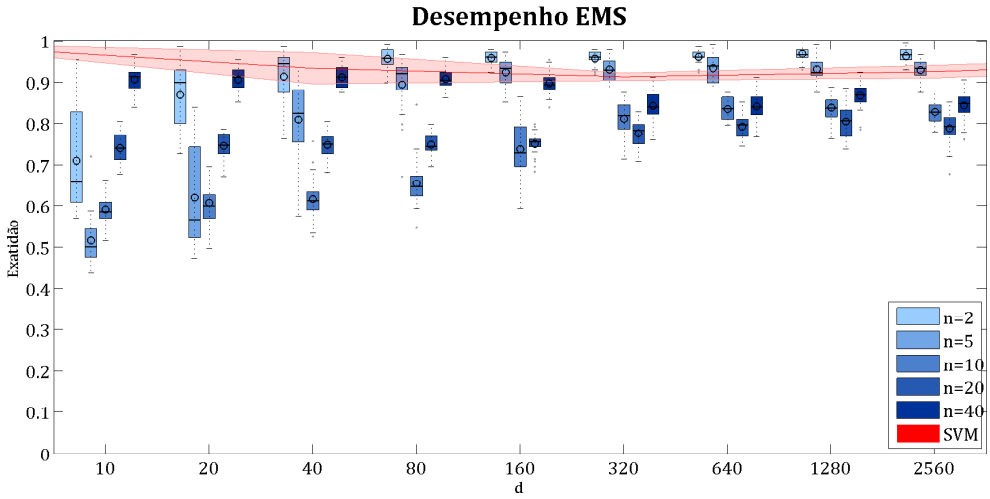
(b) Gaussian



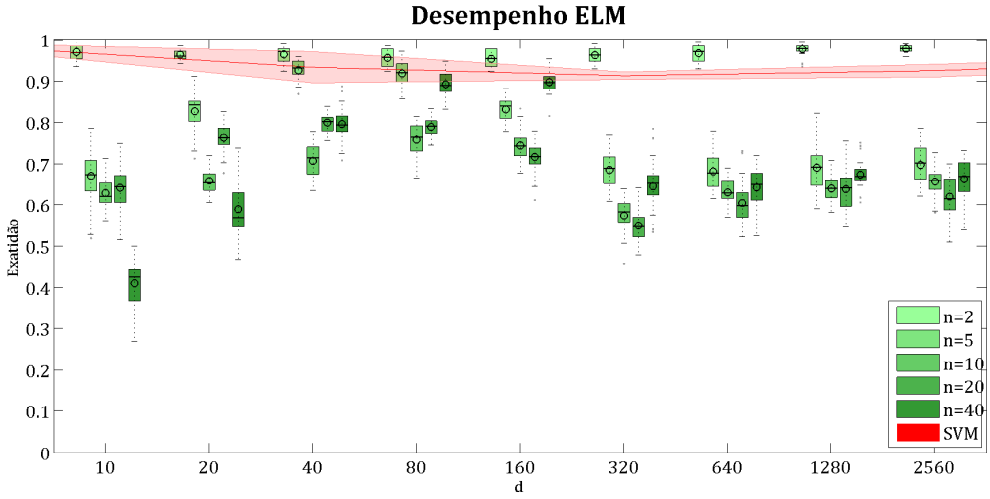
(c) Circle

Figura 5.1: Resultado para o teste de desempenho para os problemas sintéticos bidimensionais, para as EMS, ELM e LS-SVM RBF.(5.1a) Exatidão para a base de dados com duas espirais entrelaçdas (Spiral). (5.1b) Exatidão para a base de dados com duas distribuições gaussianas (Gaussian).(5.1c) Exatidão para a base de dados do círculo dentro do quadrado(Circle bidimensional).

medida que a dimensão do espaço de entrada aumenta. Ainda assim, o desempenho de ambas é reduzido enquanto o da SVM permanece bastante alto.



(a) Spiral



(b) Gaussian

Figura 5.2: Resultado para o teste de desempenho para o problema do círculo dentro do quadrado com diferentes dimensões de entrada, para as EMS, ELM e LS-SVM RBF.(5.2a) Exatidão para o método proposto EMS e para LS-SVM RBF. (5.2b) Exatidão para a ELM e para a LS-SVM RBF.

A quantidade de vetores de suporte no método proposto apresenta um comportamento bastante curioso. A princípio, ela aparenta diminuir assimptoticamente á medida que a dimensão de mapeamento aumenta, ao mesmo tempo esta quantidade aumenta com o crescimento do número de padrões de treinamento, no entanto isto não é verificado quando a quantidade de dimensões dos padrões de entrada aumenta. Existem casos em que, com o aumento da quantidade de neurônios na camada escondida aumento, a quantidade de vetores de suporte diminui e depois volta a aumentar ou ela pode, ainda, somente aumentar.

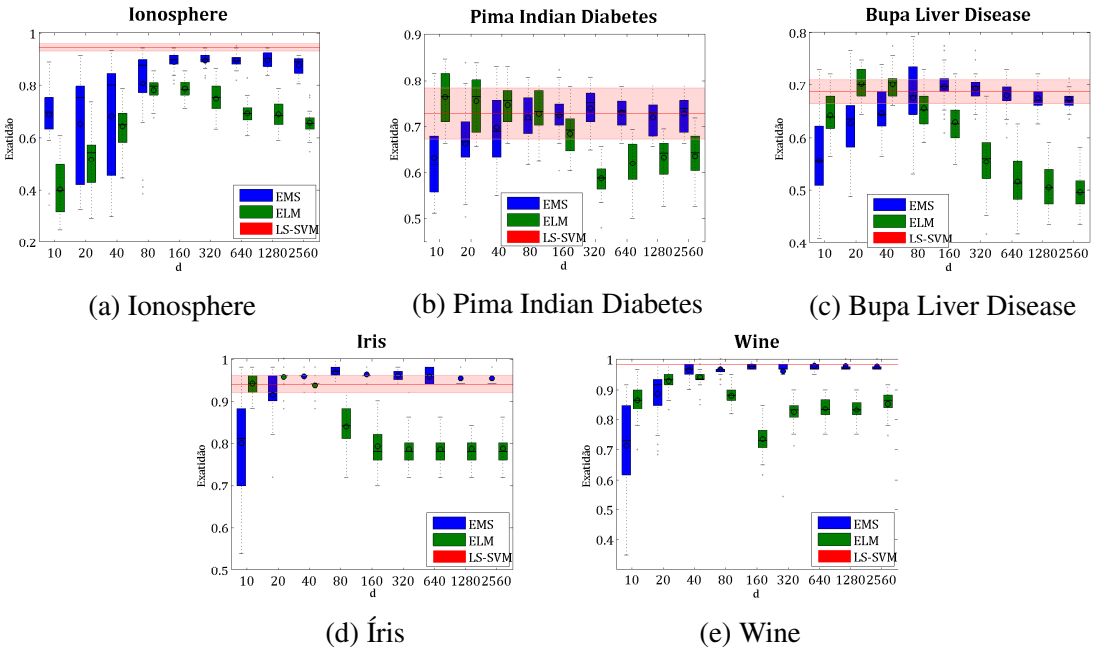
Tabela 5.1: Resultados dos testes com as bases de dados sintéticas

Nome	n	N	d																		SVM
			10		20		40		80		160		320		640		1280		2560		
			EMS	ELM	EMS	ELM	EMS	ELM	EMS	ELM	EMS	ELM	EMS	ELM	EMS	ELM	EMS	ELM	EMS	ELM	
Spiral	2	500	0.53 ± 0.083	0.93 ± 0.044	0.57 ± 0.1	0.99 ± 0.011	0.62 ± 0.085	1 ± 0	0.78 ± 0.12	1 ± 0	0.85 ± 0.088	1 ± 0	0.91 ± 0.056	1 ± 0	0.94 ± 0.053	1 ± 0	0.96 ± 0.01	1 ± 0	0.97 ± 0.0062	1 ± 0	1 ± 0
		1000	0.53 ± 0.096	0.93 ± 0.052	0.59 ± 0.091	0.99 ± 0.016	0.63 ± 0.12	1 ± 0.001	0.74 ± 0.13	1 ± 0.00092	0.82 ± 0.12	1 ± 0.00076	0.91 ± 0.06	1 ± 0.001	0.96 ± 0.016	1 ± 0.00092	0.96 ± 0.0081	1 ± 0.00055	0.97 ± 0.0059	1 ± 0	1 ± 0
		2000	0.55 ± 0.094	0.92 ± 0.072	0.58 ± 0.13	0.99 ± 0.011	0.61 ± 0.11	1 ± 0.00027	0.73 ± 0.14	1 ± 0.00027	0.84 ± 0.11	1 ± 0	0.92 ± 0.072	1 ± 0.00038	0.95 ± 0.043	1 ± 0	0.97 ± 0.012	1 ± 0	0.98 ± 0.0062	1 ± 0	1 ± 0
Gaussian	2	500	0.88 ± 0.029	0.89 ± 0.013	0.89 ± 0.014	0.9 ± 0.0086	0.9 ± 0.011	0.9 ± 0.0069	0.9 ± 0.013	0.9 ± 0.0065	0.89 ± 0.016	0.9 ± 0.0065	0.9 ± 0.015	0.91 ± 0.0053	0.89 ± 0.016	0.91 ± 0.0062	0.89 ± 0.014	0.91 ± 0.008	0.89 ± 0.012	0.91 ± 0.0061	0.89 ± 0.011
		1000	0.9 ± 0.035	0.92 ± 0.018	0.91 ± 0.025	0.92 ± 0.018	0.91 ± 0.019	0.92 ± 0.018	0.92 ± 0.018	0.92 ± 0.017	0.92 ± 0.02	0.92 ± 0.018	0.91 ± 0.017	0.92 ± 0.018	0.91 ± 0.018	0.92 ± 0.018	0.91 ± 0.016	0.92 ± 0.017	0.91 ± 0.019	0.92 ± 0.017	0.91 ± 0.022
		2000	0.89 ± 0.059	0.92 ± 0.006	0.91 ± 0.032	0.92 ± 0.0079	0.92 ± 0.011	0.92 ± 0.008	0.91 ± 0.026	0.92 ± 0.0076	0.91 ± 0.014	0.92 ± 0.007	0.91 ± 0.013	0.92 ± 0.0059	0.91 ± 0.017	0.92 ± 0.0051	0.91 ± 0.017	0.92 ± 0.0054	0.91 ± 0.015	0.92 ± 0.0048	0.92 ± 0.0092
Circle	2	500	0.71 ± 0.12	0.97 ± 0.017	0.87 ± 0.081	0.96 ± 0.014	0.91 ± 0.067	0.96 ± 0.021	0.95 ± 0.023	0.95 ± 0.02	0.96 ± 0.015	0.95 ± 0.02	0.95 ± 0.018	0.96 ± 0.019	0.96 ± 0.017	0.97 ± 0.02	0.97 ± 0.014	0.97 ± 0.013	0.96 ± 0.021	0.98 ± 0.0069	0.98 ± 0.0091
		1000	0.67 ± 0.12	0.95 ± 0.023	0.8 ± 0.1	0.96 ± 0.0071	0.89 ± 0.079	0.96 ± 0.0089	0.95 ± 0.025	0.95 ± 0.0047	0.97 ± 0.012	0.95 ± 0.0051	0.96 ± 0.018	0.96 ± 0.0089	0.96 ± 0.014	0.97 ± 0.011	0.96 ± 0.016	0.98 ± 0.0085	0.96 ± 0.015	0.99 ± 0.0071	0.98 ± 0.0017
		2000	0.68 ± 0.12	0.97 ± 0.012	0.82 ± 0.11	0.97 ± 0.0067	0.93 ± 0.036	0.97 ± 0.0085	0.97 ± 0.016	0.96 ± 0.0074	0.98 ± 0.006	0.96 ± 0.0062	0.98 ± 0.006	0.97 ± 0.011	0.98 ± 0.006	0.98 ± 0.0077	0.97 ± 0.0055	0.98 ± 0.0082	0.97 ± 0.006	0.99 ± 0.0043	0.99 ± 0.0046
	5	500	0.52 ± 0.054	0.67 ± 0.061	0.62 ± 0.12	0.83 ± 0.05	0.81 ± 0.099	0.92 ± 0.026	0.89 ± 0.063	0.92 ± 0.027	0.92 ± 0.031	0.83 ± 0.028	0.93 ± 0.029	0.68 ± 0.044	0.93 ± 0.032	0.68 ± 0.04	0.93 ± 0.027	0.69 ± 0.053	0.93 ± 0.023	0.7 ± 0.046	0.93 ± 0.037
		1000	0.52 ± 0.046	0.67 ± 0.061	0.57 ± 0.099	0.81 ± 0.035	0.81 ± 0.12	0.92 ± 0.026	0.9 ± 0.06	0.93 ± 0.017	0.92 ± 0.032	0.9 ± 0.01	0.92 ± 0.015	0.82 ± 0.013	0.93 ± 0.014	0.76 ± 0.023	0.93 ± 0.015	0.7 ± 0.026	0.94 ± 0.011	0.68 ± 0.027	0.94 ± 0.0045
		2000	0.52 ± 0.025	0.68 ± 0.039	0.55 ± 0.049	0.83 ± 0.051	0.76 ± 0.12	0.93 ± 0.015	0.92 ± 0.042	0.94 ± 0.0095	0.94 ± 0.022	0.95 ± 0.0064	0.94 ± 0.013	0.91 ± 0.0095	0.94 ± 0.015	0.89 ± 0.012	0.95 ± 0.011	0.86 ± 0.012	0.95 ± 0.01	0.83 ± 0.016	0.96 ± 0.0074
	10	500	0.59 ± 0.034	0.63 ± 0.04	0.61 ± 0.048	0.66 ± 0.029	0.62 ± 0.048	0.71 ± 0.041	0.65 ± 0.056	0.76 ± 0.039	0.74 ± 0.07	0.74 ± 0.036	0.81 ± 0.042	0.57 ± 0.042	0.83 ± 0.027	0.63 ± 0.032	0.84 ± 0.031	0.64 ± 0.029	0.83 ± 0.027	0.66 ± 0.034	0.91 ± 0.0089
		1000	0.61 ± 0.045	0.66 ± 0.032	0.6 ± 0.063	0.69 ± 0.022	0.62 ± 0.032	0.74 ± 0.03	0.65 ± 0.041	0.83 ± 0.025	0.71 ± 0.043	0.87 ± 0.02	0.77 ± 0.043	0.81 ± 0.019	0.83 ± 0.024	0.59 ± 0.031	0.86 ± 0.025	0.64 ± 0.036	0.86 ± 0.015	0.65 ± 0.035	0.95 ± 0.015
		2000	0.6 ± 0.016	0.65 ± 0.022	0.59 ± 0.054	0.68 ± 0.012	0.61 ± 0.016	0.73 ± 0.023	0.63 ± 0.019	0.84 ± 0.018	0.67 ± 0.04	0.89 ± 0.015	0.73 ± 0.039	0.88 ± 0.0099	0.81 ± 0.033	0.82 ± 0.015	0.85 ± 0.018	0.59 ± 0.027	0.87 ± 0.011	0.64 ± 0.022	0.97 ± 0.0068
	20	500	0.74 ± 0.038	0.64 ± 0.054	0.74 ± 0.032	0.76 ± 0.037	0.75 ± 0.027	0.8 ± 0.024	0.75 ± 0.03	0.79 ± 0.023	0.75 ± 0.025	0.72 ± 0.037	0.77 ± 0.034	0.55 ± 0.041	0.79 ± 0.029	0.6 ± 0.053	0.8 ± 0.038	0.64 ± 0.056	0.79 ± 0.039	0.62 ± 0.05	0.92 ± 0.015
		1000	0.76 ± 0.024	0.65 ± 0.053	0.75 ± 0.026	0.78 ± 0.035	0.76 ± 0.027	0.82 ± 0.024	0.76 ± 0.031	0.82 ± 0.02	0.76 ± 0.027	0.81 ± 0.022	0.76 ± 0.028	0.75 ± 0.026	0.79 ± 0.031	0.56 ± 0.028	0.8 ± 0.029	0.62 ± 0.031	0.81 ± 0.025	0.64 ± 0.025	0.93 ± 0.014
		2000	0.76 ± 0.026	0.62 ± 0.051	0.77 ± 0.019	0.77 ± 0.029	0.76 ± 0.018	0.83 ± 0.016	0.76 ± 0.018	0.83 ± 0.016	0.77 ± 0.019	0.83 ± 0.017	0.77 ± 0.021	0.84 ± 0.016	0.78 ± 0.03	0.79 ± 0.017	0.8 ± 0.026	0.58 ± 0.024	0.81 ± 0.021	0.63 ± 0.022	0.94 ± 0.018
	40	500	0.91 ± 0.032	0.41 ± 0.058	0.91 ± 0.028	0.59 ± 0.066	0.91 ± 0.027	0.79 ± 0.039	0.91 ± 0.025	0.89 ± 0.029	0.9 ± 0.028	0.9 ± 0.027	0.84 ± 0.038	0.64 ± 0.059	0.84 ± 0.035	0.64 ± 0.047	0.87 ± 0.032	0.67 ± 0.032	0.84 ± 0.035	0.66 ± 0.048	0.95 ± 0.013
		1000	0.9 ± 0.021	0.35 ± 0.062	0.9 ± 0.026	0.56 ± 0.051	0.91 ± 0.02	0.77 ± 0.044	0.91 ± 0.023	0.89 ± 0.024	0.91 ± 0.022	0.93 ± 0.02	0.89 ± 0.024	0.91 ± 0.019	0.86 ± 0.025	0.67 ± 0.065	0.84 ± 0.015	0.64 ± 0.041	0.86 ± 0.032	0.66 ± 0.028	0.95 ± 0.021
		2000	0.91 ± 0.013	0.32 ± 0.044	0.92 ± 0.014	0.53 ± 0.054	0.91 ± 0.019	0.75 ± 0.045	0.92 ± 0.018	0.9 ± 0.018	0.92 ± 0.017	0.95 ± 0.0059	0.92 ± 0.014	0.95 ± 0.0032	0.9 ± 0.012	0.93 ± 0.0068	0.88 ± 0.016	0.7 ± 0.049	0.86 ± 0.017	0.66 ± 0.055	0.95 ± 0.0023

Tabela 5.2: Quantidade de Vetores de Suporte obtidos para as bases de dados sintéticos

Nome	n	N	d								
			10	20	40	80	160	320	640	1280	2560
Spiral	2	500	285 ± 5	279 ± 6	262 ± 22	203 ± 21	147 ± 12	112 ± 7	87 ± 4	69 ± 4	56 ± 3
		1000	564 ± 15	536 ± 39	424 ± 54	325 ± 34	230 ± 15	178 ± 12	138 ± 8	109 ± 6	87 ± 4
		2000	1108 ± 38	1000 ± 109	761 ± 105	487 ± 42	364 ± 31	273 ± 20	213 ± 12	168 ± 12	135 ± 9
Gaussian	2	500	82 ± 4	81 ± 4	81 ± 4	81 ± 4	81 ± 4	81 ± 4	82 ± 4	83 ± 4	83 ± 3
		1000	125 ± 10	123 ± 10	123 ± 10	123 ± 10	122 ± 10	122 ± 10	122 ± 10	122 ± 10	122 ± 11
		2000	265 ± 14	264 ± 14	263 ± 14	263 ± 14	263 ± 14	263 ± 14	263 ± 14	262 ± 14	263 ± 13
Circle	2	500	112 ± 24	72 ± 6	54 ± 3	42 ± 2	33 ± 2	27 ± 2	22 ± 1	19 ± 1	16 ± 1
		1000	236 ± 53	137 ± 12	104 ± 6	80 ± 3	63 ± 2	48 ± 1	38 ± 1	30 ± 1	26 ± 1
		2000	326 ± 89	203 ± 16	144 ± 8	113 ± 6	87 ± 3	70 ± 2	55 ± 2	45 ± 1	38 ± 1
	5	500	213 ± 31	141 ± 26	83 ± 13	57 ± 7	45 ± 5	39 ± 4	38 ± 3	38 ± 2	39 ± 4
		1000	446 ± 60	276 ± 48	157 ± 17	107 ± 7	83 ± 4	69 ± 4	62 ± 4	57 ± 4	54 ± 4
		2000	868 ± 99	473 ± 91	237 ± 38	144 ± 10	108 ± 6	85 ± 3	73 ± 2	65 ± 3	63 ± 2
	10	500	259 ± 13	212 ± 16	156 ± 14	106 ± 8	89 ± 4	98 ± 4	111 ± 4	123 ± 7	137 ± 9
		1000	484 ± 22	398 ± 38	260 ± 33	162 ± 12	120 ± 6	122 ± 5	138 ± 5	159 ± 5	187 ± 4
		2000	996 ± 43	819 ± 64	556 ± 47	310 ± 29	206 ± 10	176 ± 7	193 ± 6	221 ± 6	252 ± 5
	20	500	178 ± 16	176 ± 14	163 ± 13	133 ± 10	129 ± 7	169 ± 7	217 ± 6	265 ± 6	289 ± 9
		1000	327 ± 39	327 ± 36	301 ± 20	249 ± 25	199 ± 13	228 ± 12	305 ± 11	389 ± 11	483 ± 13
		2000	625 ± 60	591 ± 47	571 ± 39	484 ± 37	344 ± 25	300 ± 18	407 ± 20	546 ± 26	701 ± 17
	40	500	49 ± 8	58 ± 8	69 ± 6	92 ± 8	125 ± 11	200 ± 8	295 ± 6	327 ± 3	322 ± 2
		1000	95 ± 16	104 ± 17	115 ± 15	139 ± 14	170 ± 18	248 ± 17	390 ± 15	569 ± 10	655 ± 4
		2000	149 ± 9	160 ± 10	176 ± 11	194 ± 9	235 ± 16	292 ± 16	481 ± 36	753 ± 34	1107 ± 10

5.2 Dados Reais



(a) Ionosphere (b) Pima Indian Diabetes (c) Bupa Liver Disease

(d) Íris (e) Wine

Figura 5.3: Resultado para o teste de desempenho para os problemas reais.(5.3a) Base de dados *Ionosphere* do repositório de Bache e Lichman (2013). (5.3b) Base de dados *Pima Indian Diabetes* do repositório de Bache e Lichman (2013), modificada sem os valores NaN.(**??**) Base de dados *Bupa Liver Disease* do repositório de Bache e Lichman (2013). (5.3d) Base de dados Íris. (5.3e) Base de dados *Wine*.

O resultado da exatidão do método verificado para as bases de dados reais estão ilustrados na Figura 5.3, estão presentes, também, na Tabela 5.3 para avaliar os seus valores médios e o seu desvio padrão. O desempenho do classificador proposto, assim como o da ELM padrão aumenta quando a quantidade de neurônios da camada escondida, e portanto do espaço de características, aumenta. No entanto, o desempenho da ELM padrão sofre uma certa degeneração dependendo da sua camada escondida o que o torna relativamente inferior ao método proposto e a LS-SVM. Neste contexto, de solução de problemas reais, em que há superposição, ruídos e uma quantidade elevada de características, o método proposto é comparável a um método no estado-da-arte, a LS-SVM RBF.

Tabela 5.3: Resultados dos testes com as bases de dados reais

Nome	d																		SVM
	10		20		40		80		160		320		640		1280		2560		
	EMS	ELM	EMS	ELM	EMS	ELM	EMS	ELM	EMS	ELM	EMS	ELM	EMS	ELM	EMS	ELM	EMS	ELM	
Ion	0.69± 0.12	0.4 ± 0.11	0.65± 0.2	0.52± 0.11	0.68± 0.22	0.64± 0.088	0.81± 0.15	0.78± 0.049	0.89± 0.038	0.79± 0.037	0.89± 0.025	0.75± 0.059	0.9 ± 0.029	0.69± 0.048	0.9 ± 0.029	0.69± 0.053	0.88± 0.036	0.66± 0.041	0.94± 0.013
PID	0.63± 0.086	0.76± 0.058	0.66± 0.078	0.75± 0.059	0.7 ± 0.081	0.75± 0.045	0.72± 0.051	0.73± 0.048	0.72± 0.047	0.68± 0.051	0.74± 0.045	0.59± 0.036	0.73± 0.034	0.62± 0.049	0.72± 0.038	0.63± 0.04	0.73± 0.039	0.64± 0.054	0.73± 0.055
BLD	0.56± 0.084	0.64± 0.039	0.63± 0.068	0.7 ± 0.03	0.65± 0.067	0.7 ± 0.031	0.67± 0.07	0.66± 0.037	0.7 ± 0.042	0.63± 0.042	0.69± 0.028	0.55± 0.054	0.68± 0.03	0.52± 0.055	0.67± 0.025	0.5 ± 0.042	0.67± 0.019	0.5 ± 0.037	0.69± 0.023
Iri	0.8 ± 0.11	0.94± 0.024	0.91± 0.062	0.96± 0.024	0.96± 0.018	0.94± 0.025	0.97± 0.016	0.84± 0.054	0.96± 0.016	0.79± 0.054	0.96± 0.017	0.79± 0.043	0.96± 0.018	0.79± 0.037	0.95± 0.019	0.79± 0.031	0.95± 0.019	0.79± 0.034	0.94± 0.02
Win	0.71± 0.16	0.86± 0.053	0.89± 0.084	0.93± 0.035	0.96± 0.023	0.94± 0.03	0.97± 0.021	0.88± 0.034	0.98± 0.0096	0.73± 0.051	0.96± 0.08	0.82± 0.046	0.98± 0.015	0.83± 0.039	0.98± 0.015	0.83± 0.039	0.98± 0.014	0.85± 0.038	0.98± 0.00016

A Tabela 5.4 relaciona as quantidades de vetores de suporte observadas para diferentes tamanhos do espaço de características, nela é possível perceber um decrescimento assimptótico deste valor. A redução da quantidade de vetores de suporte encontrados está relacionada ao aumento da distância entre os padrões provocado pelo aumento da dimensionalidade, ou seja, com o aumento do espaço em que quantidade fixa de dados está espalhado a quantidade de padrões significativos o suficiente para se tornarem vetores de suporte diminui.

Tabela 5.4: Quantidade de Vetores de Suporte obtidos para as bases de dados reais

Nome	d								
	10	20	40	80	160	320	640	1280	2560
Ion	124 ± 18	94 ± 15	74 ± 10	61 ± 8	56 ± 5	51 ± 5	50 ± 4	50 ± 6	52 ± 6
PID	135 ± 18	127 ± 13	122 ± 12	120 ± 12	118 ± 12	117 ± 12	118 ± 13	115 ± 11	115 ± 11
BLD	177 ± 12	162 ± 8	153 ± 6	146 ± 6	142 ± 6	141 ± 5	140 ± 4	139 ± 5	139 ± 5
Iri	26 ± 5	21 ± 2	19 ± 2	17 ± 1	16 ± 1	15 ± 2	14 ± 2	14 ± 2	14 ± 2
Win	34 ± 6	27 ± 3	27 ± 3	28 ± 3	30 ± 2	29 ± 3	30 ± 2	30 ± 2	30 ± 2

5.3 Efeitos dos Parâmetros de Teste

Se o desempenho dos testes forem linearizados a partir de uma função exponencial pode ser traçado um modelo linear de acordo com a Tabela ANOVA 5.5.

Tabela 5.5: Tabela ANOVA de modelos Lineares para análise do desempenho do método

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
DB	7	117.84	16.83	277.89	0.0000
N	1	0.27	0.27	4.51	0.0337
d	1	49.03	49.03	809.32	0.0000
n	1	1.01	1.01	16.70	0.0000
N:d	1	1.21	1.21	20.02	0.0000
N:n	1	0.19	0.19	3.14	0.0767
d:n	1	14.67	14.67	242.17	0.0000
N:d:n	1	1.71	1.71	28.15	0.0000
Residuals	7005	424.37	0.06		

A partir deste modelo é possível verificar o efeito dos parâmetros no desempenho do método proposto. As relevâncias estatísticas para o modelo da Base de Dados (*BD*), para a dimensão de entrada n e para a dimensão de mapeamento d, e para os efeitos combinados de

$N:d$ e $d:n$ e $N:d:n$ foram verificadas pelo seu p valor, para uma significância $\alpha = 0.1\%$. Considerando $\alpha = 5\%$, a quantidade de padrões de treinamento pode apresentar relevância para o modelo e o efeito combinado de $N:n$ somente seria considerado para um $\alpha = 10\%$.

5.4 Escalonamento do Tempo Computacional

Com a finalidade de se avaliar o custo computacional do método proposto foram realizados experimentos com a medição do tempo computacional para se treinar a uma máquina EMS. Assim, foram testados os tempos de treinamento para com o aumento da dimensão de entrada, da quantidade de padrões de treinamento e da dimensão do mapeamento.

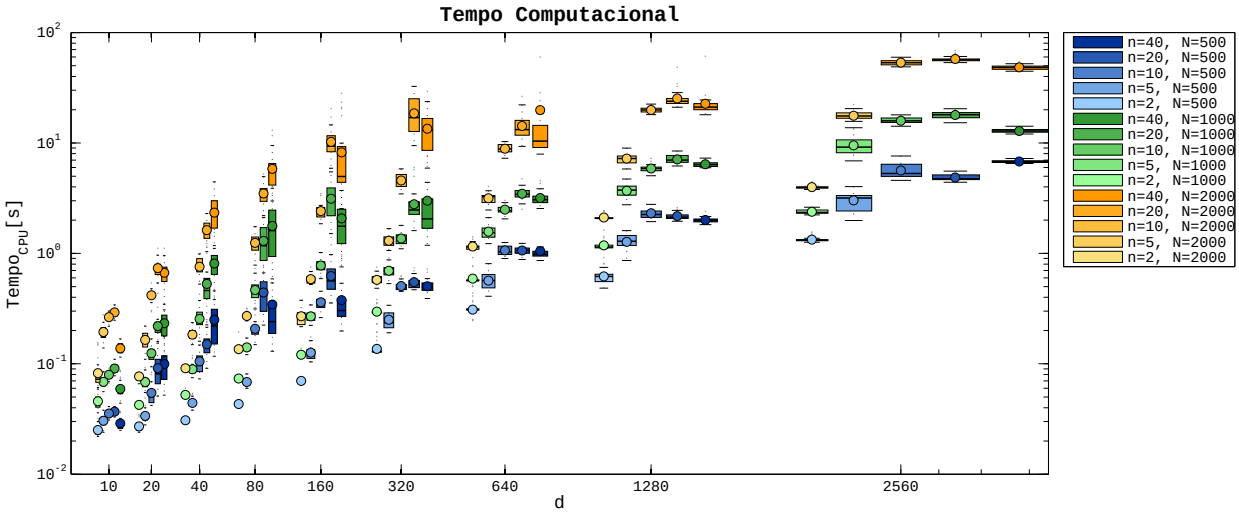


Figura 5.4: Tempo de treinamento por dimensão da camada escondida d para diferentes dimensões do espaço de entrada n, representadas pela gradação de cada cor, e para diferentes quantidades de padrões de treinamento N, representadas pelas diferentes cores.

É possível observar na Figura 5.4 um crescimento linear do tempo com o aumento da dimensão de mapeamento. Da mesma forma, a quantidade de padrões de treinamento apresentou uma influencia significativa sobre o tempo computacional, com um aumento do custo computacional aproximadamente proporcional ao crescimento do conjunto de treinamento.

Da mesma forma, o escalonamento do espaço de entrada promoveu um aumento no tempo computacional, como pode ser observado em cada conjunto de cores da Figura 5.4. Para cada tamanho da camada escondida, é possível observar no bloco de mesma cor que o crescimento é linear, considerando que há uma distorção dos últimos blocos em relação aos

primeiros devido a escala.

Tabela 5.6: Tabela ANOVA de modelos Lineares para análise de tempo

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
N	1	2526.02	2526.02	2380.84	0.0000
d	1	7160.36	7160.36	6748.84	0.0000
n	1	1466.56	1466.56	1382.27	0.0000
d:n	1	14.76	14.76	13.91	0.0002
N:d	1	0.14	0.14	0.13	0.7175
N:n	1	102.34	102.34	96.46	0.0000
N:d:n	1	1.03	1.03	0.97	0.3257
Residuals	4042	4288.47	1.06		

Estas observações são corroboradas pelos resultados obtidos na análise estatística da tabela ANOVA 5.6. Considerando uma significância $\alpha = 0.1\%$, os p valores encontrados para os fatores quantidade de padrões de treinamento (N), quantidade dimensões de entrada (n) e quantidade de dimensões de mapeamento (d), além dos efeitos combinados de $d : n$ e $N : n$, apresentam relevância estatística no modelo de tempo do método.

CAPÍTULO 6

Conclusões

O método proposto foi avaliado segundo critérios de desempenho e tempo computacional, em ambos os casos o seu comportamento se manteve dentro do esperado segundo os premissas da construção do método.

Existem características específicas de problemas de classificação que podem tornar o método desenvolvido mais eficiente em relação aos outros dois métodos avaliados. A principal delas está relacionada a existência de alguma superposição entre as classes, quando os grupos não estão bem definidos a estrutura do método promove alguns benefícios bastante propícios. O mapeamento com aumento da dimensionalidade permite uma maior probabilidade de separação, dessa forma a dimensionalidade não é apresentada como uma desvantagem já que ela acarreta uma maior dispersão entre os padrões. O hiperplano ótimo de separação regularizado a partir de vetores de suporte, por sua vez, implica em um classificador robusto que promove uma grande capacidade de generalização. Esta vantagem se destaca, sobretudo quando a quantidade de dimensões de entrada aumenta. A combinação do aumento da separabilidade dos padrões com um classificador de margem larga com alta capacidade de generalização é apropriado, principalmente, para problemas reais, já que nestes estão presentes ruídos, que podem tornar a fronteira entre as classes bastante difusa, além de uma superposição intrínseca de diversos problemas reais, por exemplo no caso de diagnóstico de câncer, ao se observar dois pacientes com resultados de exames, características genéticas e estilos de vida quase idênticos é, ainda assim, possível que um apresente o quadro da doença e o outro não.

Quando, no entanto, o problema de classificação apresentar as classes bem definidas, de forma que não exista qualquer superposição entre elas mesmo que a sua geometria esteja entrelaçada, existem métodos mais eficientes na literatura, como a própria ELM.

Comparativamente, os métodos com Vetores de Suporte avaliados, o proposto e a LS-SVM, apresentam resultados estatisticamente semelhantes para vários casos. Esta análise apresenta uma certa relevância, uma vez que ela constata que o método proposto atinge resultados compatíveis com métodos no estado-da-arte, mesmo que não tenha ocorrido o ajuste minucioso dos parâmetros. Existe uma dificuldade em se ajustar os diversos parâmetros de um método e,

mesmo utilizando heurísticas, é necessário um procedimento complexo e custoso para se definir os valores ideais de cada parâmetro. Esta desvantagem dos métodos de vetores de suporte tradicionais pode tornar o processo de aprendizado e validação das SVMs tão computacionalmente caros quanto o método proposto implementado com um espaço de características de enormes proporções.

Por outro lado, o desempenho da ELM tradicional é bastante peculiar, como já foi mencionado por Liu *et al.* (2015) e Lin *et al.* (2015), existe uma degeneração associada ao aumento da camada escondida da ELM. Dessa forma a sua quantidade de neurônios deve ser escolhida de forma que seja grande o suficiente para permitir a generalização mas, ao mesmo tempo, deve ter o seu tamanho limitado de forma a evitar tal degradação. Outra abordagem possível seria utilizar a regularização de Tikhonov, mas isto deveria ser implementado para ambos os métodos.

Certamente o método proposto pode apresentar um tempo computacional maior para resolver um mesmo problema que uma ELM clássica, principalmente com mapeamentos maiores (Cortes e Vapnik, 1995), isso é intuitivo uma vez que o método de treinamento da ELM é analítico e não requer nenhuma iteração para definir os parâmetros do classificador. O método proposto, em oposição, é mais lento já que o método de Lagrange para resolver o problema quadrático para a obtenção de pesos é, geralmente, baseado em gradiente e, portanto, requer várias iterações para se atingir os resultados desejados. Esta dificuldade também está presente nos métodos de Máquinas de Vetores de Suporte mais tradicionais, como a LS-SVM.

Neste contexto foram avaliados os efeitos da variação da dimensão de mapeamento, da quantidade de padrões de entrada e da sua dimensão. De forma geral o aumento de cada um destes parâmetros causou um aumento linear no tempo de treinamento do método. Isto está em conformidade com o método, já que o aumento de qualquer um destes parâmetros tem efeitos semelhantes sobre o espaço de características gerado. Como a etapa mais demorada do método está relacionada ao cálculo dos vetores de suporte sobre tal espaço de características, é compreensível que a variação no tempo seja semelhante nos três casos.

A partir dos experimentos realizados, foi considerado que, para as bases de dados testadas, o método de Mapeamento Explícito para Máquinas de Vetores de Suporte atinge os resultados esperados e são adequados para resolver os problemas no escopo proposto.

CAPÍTULO 7

Trabalhos Futuros

O Estudo do método Semi-Supervisionado se encontre, ainda, em um estágio bastante inicial. Com implementação do método multiobjetivo completo será possível verificar o desempenho do método. Primeiramente é preciso verificar a viabilidade da função de otimização relacionada aos padrões não rotulados, sem a sua integração ao método já consolidado das Máquinas de Vetores de Suporte.

Com otimização dos dois objetivos em conjunto será possível criar um fronteira pareto-ótima que permite soluções que apresentem tanto um um bom desempenho para o objetivo de maximização de margem quanto da criação de uma fronteira em uma região de baixa densidade. Dentre as soluções na fronteira pareto-ótima a escolha da solução ideal depende do problema estudado, por exemplo, em um problema de diagnóstico médico, a solução com uma menor ocorrência de falsos negativos seria a ideal.

Referências Bibliográficas

Bache e Lichman(2013) K. Bache e M. Lichman. UCI machine learning repository, 2013. URL http://archive.ics.uci.edu/ml. Citado na pág. xiii, 41, 48

Boser *et al.***(1992)** Bernhard E. Boser, Isabelle M. Guyon, e Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. Em *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT '92, páginas 144–152, New York, NY, USA. ACM. Citado na pág. 10, 11, 17

Braga *et al.***(2007)** Antônio de Pádua Braga, André Ponce de Leon F. Carvalho, e Teresa Bernarda Ludermir. *Redes neurais artificiais: teoria e aplicações*. LTC, 2a edição. Citado na pág. 1, 2, 3, 4, 13, 26

Chapelle *et al.***(2010)** Olivier Chapelle, Bernhard Schlkopf, e Alexander Zien. *Semi-Supervised Learning*. The MIT Press, 1a edição. Citado na pág. 1, 2, 4, 5, 8

Cortes e Vapnik(1995) Corinna Cortes e Vladimir N. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297. Citado na pág. 7, 10, 11, 13, 17, 18, 27, 29, 54

Cover(1965) Thomas M. Cover. Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition. *Electronic Computers, IEEE Transactions on*, 14(3):326–334. Citado na pág. 9, 10

Frénay e Verleysen(2011) Benoît Frénay e Michel Verleysen. Parameter-insensitive kernel in extreme learning for non-linear support vector regression. *Neurocomputing*, 74(16):2526 – 2531. Citado na pág. 16, 27

Haykin(2001) Simon Haykin. *Redes Neurais Princípios e Práticas*. BOOKMAN, 2a edição. Citado na pág. 1, 2, 3, 5, 11, 17

He *et al.***(2011)** Qing He, Changying Du, Qun Wang, Fuzhen Zhuang, e Zhongzhi Shi. A parallel incremental extreme {SVM} classifier. *Neurocomputing*, 74(16):2532 – 2540. Citado na pág. 22, 23

Hornik *et al.***(1989)** Kur Hornik, Maxwell Stinchcombe, e Halber White. Multilayer feed-forward networks are universal approximators. *Neural Networks*, 2(5):359–366. Citado na pág. 10

Huang *et al.***(2006)** Guang-Bin Huang, Qin-Yu Zhu, e Chee-Kheong Siew. Extreme learning machine: theory and applications. *Neurocomputing*, 70(1-3):489–501. Citado na pág. 7, 10, 12, 13, 28

Lin *et al.***(2015)** Shaobo Lin, Xia Liu, Jian Fang, e Zongben Xu. Is Extreme Learning Machine Feasible? A Theoretical Assessment (Part II). *Neural Networks and Learning Systems, IEEE Transactions on*, 26(1):21–34. Citado na pág. 15, 16, 54

Liu *et al.***(2008)** Qiuge Liu, Qing He, e Zhongzhi Shi. Extreme support vector machine classifier. Em *Advances in Knowledge Discovery and Data Mining*, volume 5012 of *Lecture Notes in Computer Science*, páginas 222–233. Springer Berlin Heidelberg. Citado na pág. 22, 23

Liu *et al.***(2015)** Xia Liu, Shaobo Lin, Jian Fang, e Zongben Xu. Is Extreme Learning Machine Feasible? A Theoretical Assessment (Part I). *Neural Networks and Learning Systems, IEEE Transactions on*, 26(1):7–20. Citado na pág. 15, 54

Smola e Bartlett(2000) Alexander J. Smola e Peter J. Bartlett, editors. *Advances in Large Margin Classifiers*. MIT Press. Citado na pág. 5, 7, 26

Van Gestel *et al.***(2004)** Tony Van Gestel, Johan A. K. Suykens, Bart Baesens, Stijn Viaene, Jan Vanthienen, Guido Dedene, Bart De Moor, e Joos Vandewalle. Benchmarking least squares support vector machine classifiers. *Machine Learnig*, 54(1):5–32. Citado na pág. 36

Vapnik(1995) Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., New York, NY, USA. Citado na pág. 10, 17

Vapnik(1998) Vladimir N. Vapnik. *Statistical learning theory*. Wiley, 1a edição. Citado na pág. 10, 17

Vapnik(1992) Vladmir N. Vapnik. Principles of risk minimization for learning theory. *Advances in Neural Information Processing Systems*, 4:831–838. Citado na pág. 10

Zhu e Goldberg(2009) Xiaojin Zhu e Andrew B. Goldberg. *Introduction to Semi-Supervised Learning*. Morgan and Claypool Publishers. Citado na pág. 1, 2, 4, 5, 8, 24, 32