UNIVERSIDADE FEDERAL DE JUIZ DE FORA
INSTITUTO DE CIÊNCIAS EXATAS
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Karen Braga Enes

Uma abordagem baseada em Perceptrons balanceados para geração de *ensembles* e redução do espaço de versões

Juiz de Fora

2016

UNIVERSIDADE FEDERAL DE JUIZ DE FORA
INSTITUTO DE CIÊNCIAS EXATAS
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Karen Braga Enes

Uma abordagem baseada em Perceptrons balanceados para geração de *ensembles* e redução do espaço de versões

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Ciências Exatas da Universidade Federal de Juiz de Fora como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Orientador: Raul Fonseca Neto
Coorientador: Saulo Moraes Villela

Juiz de Fora
2016

Karen Braga Enes

Uma abordagem baseada em Perceptrons balanceados para geração de *ensembles* e redução do espaço de versões

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação, do Instituto de Ciências Exatas da Universidade Federal de Juiz de Fora como requisito parcial para obtenção do título de Mestre em Ciência da Computação.

Aprovada em 8 de Janeiro de 2016.

BANCA EXAMINADORA

Prof. D.Sc. Raul Fonseca Neto - Orientador
Universidade Federal de Juiz de Fora

Prof. D.Sc. Saulo Moraes Villela - Coorientador
Universidade Federal de Juiz de Fora

Prof. D.Sc. Heder Soares Bernardino
Universidade Federal de Juiz de Fora

Prof. Ph.D. Antônio de Pádua Braga
Universidade Federal de Minas Gerais

Em memória dos meus avós

Dora, Lea e Hélio.

AGRADECIMENTOS

"A gratidão é a memória do coração."

Antístenes

"Não há no mundo exagero mais belo do que a gratidão."

Jean de la Bruyere

Eu sou extremamente grata a todos que passaram pela minha vida e mais ainda àqueles que contribuíram para que eu concluísse mais essa etapa da caminhada em direção a academia. Mas nenhum sentimento é maior do que a gratidão que eu sinto por ter o apoio incondicional da minha família em todos os momentos da minha vida. Eu nunca vou ser capaz de agradecer o suficiente!

Em primeiro lugar, agradeço aos meus pais, Katia e Marcos, por tudo que eu sou hoje. Por terem me ensinado a importância de estudar, buscar sempre aprender mais e desenvolver senso crítico. Mas não só por isso, agradeço também por me ensinarem a ouvir e respeitar toda e qualquer pessoa. Por serem muito mais do que presentes na minha vida, por serem verdadeiros amigos e parceiros. Agradeço ainda por cada palavra de incentivo, por cada risada e lágrima derramada. Agradeço por comemorarem comigo todas as minhas vitórias e me consolarem nos momentos difíceis. Agradeço por não hesitarem ao me apoiar quando decidi cursar o mestrado. Agradeço por me salvarem diversas vezes do R.U e por sacrificarem várias viagens e feriados quando eu precisei ficar em casa e estudar. Agradeço por cada momento em que vocês foram compreensivos quando eu estava estressada e solidários quando eu precisei de abraços. Eu NUNCA teria chegado até aqui se não tivesse vocês ao meu lado. Muito obrigada, vocês são os melhores!

À minha irmã, Karine, futura mestre em Química e minha maior incentivadora, agradeço por não me deixar fraquejar nos momentos de dúvida. Agradeço por me mostrar que não custa nada tentar e que se não der certo, a gente vai tentar de novo e vamos chegar lá. A sua amizade e a sua confiança em mim foram fundamentais para que eu chegasse até aqui. Agradeço por ser minha companheira de caminhada na área acadêmica e por me mostrar que eu não sou tão doida por fazer mestrado. Agradeço pelas noites mal dormidas assistindo maratonas infinitas de filmes. Pela paciência, pelas risadas, por me

ensinar a ser uma pessoa melhor, pela parceria, pela companhia e por me entender sempre. Estaremos sempre juntas, *brotha*! Obrigada!

À Gabrielle, minha prima-irmã, agradeço pelo companheirismo, por estar sempre ao meu lado, pelas risadas e pelos momentos de diversão! Obrigada por rir do meu medo de avião e permanecer me dando força! Ao meu priminho Matheus, pelo carinho, pelas brincadeiras, por ser um amigão e me ensinar as diferenças entre saguis.

Ao meu tio Ricardo, melhor biólogo do mundo, agradeço por ser referência e inspiração na minha vida, como pesquisador, educador e tio exemplar! Agora somos mestres! Partiu doutorado? À minha tia Valéria agradeço pelo apoio, por não medir esforços para me ajudar em todos os momentos. Agradeço pela torcida, por cuidar de mim e por estar sempre ao meu lado. À minha tia Mônica, agradeço pelo carinho, atenção e preocupação. Ao meu tio Ed, agradeço por dividir o amor pela leitura e por me incentivar sempre! Vocês são exemplos pra mim! Obrigada por tudo!

Aos melhores amigos que alguém pode ter, obrigada por vocês existirem! À minha irmã de coração, Taylla, obrigada pelo companheirismo e por estar sempre ao meu lado, desde o ensino fundamental e até a terceira idade. Marina, por me mostrar que a matemática é linda, pelo carinho e por cuidar sempre de mim. À Pink, por compartilhar as loucuras da vida acadêmica, pela calma, amizade em todas as horas! À Carol, por entender os momentos de ausência e não desistir de mim! Ao Marcelo, agradeço pela amizade pra vida toda, pela preocupação e cuidado, por dividir comigo a sala do laboratório e o desafio de fazer mestrado em Computação. Ao Fernando, pela companhia, pelos lanches, pelas discussões, por me ensinar um pouquinho de CG e aprender um pouquinho de IA. Obrigada!

Ao Roberto, obrigada por estar sempre ao meu lado, por lutar comigo, por torcer por mim, por cuidar de mim. Obrigada por me incentivar na matemática. Obrigada por entender os momentos de ausência, de estresse e dificuldades. Obrigada por me acompanhar nos momentos de diversão, pelo companheirismo incondicional, por me incentivar e me apoiar em todas as situações. Você foi fundamental nessa jornada. Obrigada por me ajudar a chegar até aqui.

Agradeço aos professores do Departamento de Matemática, por me incentivarem a cursar a terceira graduação, por torcerem por mim e entenderem os momentos nos quais eu priorizei o mestrado. Agradeço muito pela compreensão de vocês! Lucy, Julieta,

Beatriz, Fábio, Crocco, Nelson, Rogério e Adlai! Obrigada por tudo! Eu aprendi muito com vocês!

Agradeço ainda a todos os professores do PGCC/DCC pelos ensinamentos e em particular, ao Raul, por apostar em mim e pelo tema dessa dissertação. Ao Saulo, obrigada por me co-orientar, pelo apoio e atenção. Obrigada por ter acompanhado minha evolução, por ter me incentivado, por ter me ensinado tanto e por ter confiado no meu potencial desde o primeiro dia. Ao Barrére pelo carinho e atenção de sempre, desde a graduação. Ao Guilherme, por ser uma inspiração e pelas melhores aulas durante o mestrado. Agradeço ainda ao Alex, Cristiano, Heder, Itamar, Saul, Victor e Wagner, pelas disciplinas, discussões e por tudo que vocês me ensinaram. À Luciana e ao Edmar, agradeço pela companhia e apoio desde a graduação! Eu aprendi muito com todos vocês! E isso não se resume aos momentos dentro de sala de aula. Vou levar um pouquinho de cada um de vocês na minha carreira! Muito obrigada por tudo!

Gostaria ainda de agradecer aos funcionários do ICE e do DCC pelo suporte. Em especial a Núbia, pelo bom humor e por ser tão atenciosa e a Sarah, pela competência e por todo apoio com as demandas do mestrado.

Foram dois anos difíceis, de muita luta, muito aprendizado e muitas alegrias. Não passou rápido e não foi fácil, mas desistir nunca esteve nos meus planos. Se eu cheguei até aqui foi porque todos vocês torceram por mim, me deram força, me ajudaram e me encorajaram. Cada um da sua forma, vocês foram essenciais pra que eu seguisse na caminhada e para que hoje eu esteja aqui escrevendo os agradecimentos da minha Dissertação! Eu aprendi muito, cresci, amadureci e hoje posso dizer que sou Mestre em Ciência da Computação. Muito obrigada a cada um e a todos vocês! E que venha o doutorado!

"Conheça todas as teorias,
domine todas as técnicas,
mas ao tocar uma alma humana,
seja apenas outra alma humana."
Carl Gustav Jung

RESUMO

Recentemente, abordagens baseadas em *ensemble* de classificadores têm sido bastante exploradas por serem uma alternativa eficaz para a construção de classificadores mais acurados. A melhoria da capacidade de generalização de um *ensemble* está diretamente relacionada à acurácia individual e à diversidade de seus componentes. Este trabalho apresenta duas contribuições principais: um método *ensemble* gerado pela combinação de Perceptrons balanceados e um método para geração de uma hipótese equivalente ao voto majoritário de um *ensemble*. Para o método *ensemble*, os componentes são selecionados por medidas de diversidade, que inclui a introdução de uma medida de dissimilaridade, e avaliados segundo a média e o voto majoritário das soluções. No caso de voto majoritário, o teste de novas amostras deve ser realizado perante todas as hipóteses geradas. O método para geração da hipótese equivalente é utilizado para reduzir o custo desse teste. Essa hipótese é obtida a partir de uma estratégia iterativa de redução do espaço de versões. Um estudo experimental foi conduzido para avaliação dos métodos propostos. Os resultados mostram que os métodos propostos são capazes de superar, na maior parte dos casos, outros algoritmos testados como o SVM e o AdaBoost. Ao avaliar o método de redução do espaço de versões, os resultados obtidos mostram a equivalência da hipótese gerada com a votação de um *ensemble* de Perceptrons balanceados.

Palavras-chave: Perceptron. Classificação Binária. Métodos Ensemble. Espaço de Versões.

ABSTRACT

Recently, ensemble learning theory has received much attention in the machine learning community, since it has been demonstrated as a great alternative to generate more accurate predictors with higher generalization abilities. The improvement of generalization performance of an ensemble is directly related to the diversity and accuracy of the individual classifiers. In this work, we present two main contribuitions: we propose an ensemble method by combining Balanced Perceptrons and we also propose a method for generating a hypothesis equivalent to the majority voting of an ensemble. Considering the ensemble method, we select the components by using some diversity strategies, which include a dissimilarity measure. We also apply two strategies in view of combining the individual classifiers decisions: majority unweighted vote and the average of all components. Considering the majority vote strategy, the set of unseen samples must be evaluate towards the generated hypotheses. The method for generating a hypothesis equivalent to the majority voting of an ensemble is applied in order to reduce the costs of the test phase. The hypothesis is obtained by successive reductions of the version space. We conduct a experimental study to evaluate the proposed methods. Reported results show that our methods outperforms, on most cases, other classifiers such as SVM and AdaBoost. From the results of the reduction of the version space, we observe that the genareted hypothesis is, in fact, equivalent to the majority voting of an ensemble.

Keywords: Perceptron. Binary Classification. Ensemble Methods. Version Space.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AdaBoost – Adaptive Boosting

Bagging – Bootstrap Aggregating

EBP – Ensemble of Balanced Perceptrons

EBPd – Ensemble of Balanced Perceptrons with Dissimilarity

EBPK – Ensemble of Balanced Perceptrons Kernels

EBPKd – Ensemble of Balanced Perceptrons Kernels with Dissimilarity

GFMP – Geometric Fixed Margin Perceptron

GVMP – Geometric Variable Margin Perceptron

IA – Inteligência Artificial

PB – Perceptron Balanceado

PK – Perceptron Kernel

PKB – Perceptron Kernel Balanceado

RNA – Rede Neural Artificial

SVM – Support Vector Machine

VIPM – Variable-Increment Perceptron with Margin

VSRM – Version Space Reduction Machine

SUMÁRIO

1 INTRODUÇÃO

Todos os dias uma série de decisões são tomadas. Àquelas mais impactantes uma atenção maior é dispensada, seja essa relacionada à saúde ou ainda a questões políticas e sociais. Frequentemente, especialistas em um determinado assunto e opiniões distintas são procuradas, avaliadas e ponderadas visando obter um resultado mais preciso do que a resposta de cada especialista individualmente. Decisões baseadas na opinião de um comitê de especialistas são comuns em diversos níveis da sociedade há muito tempo. Por exemplo, sessões do congresso nacional, junta de médicos para perícias, bancas de avaliações, ou até mesmo reuniões para discussão de questões familiares.

No campo da Inteligência Artificial (IA), mais especificamente em Aprendizado de Máquinas, a ideia de formação de comitês de indivíduos que sejam capazes de dar respostas acuradas sobre um problema e ao mesmo tempo tenham opiniões, em certo grau, distintas dos demais indivíduos no comitê foi adotada nos chamados *ensembles* (HANSEN; SALAMON, 1990; DIETTERICH, 2000a; KUNCHEVA, 2004). Um *ensemble* consiste em um comitê, por exemplo de classificadores, cujas hipóteses individuais são induzidas separadamente e as decisões referentes a cada hipótese são combinadas através de um método de consenso para a classificação de novos dados. O objetivo principal é aumentar a capacidade de generalização do modelo individual.

Recentemente, abordagens baseadas em métodos *ensemble* têm sido amplamente difundidas, tornando-se uma área de pesquisa importante e ativa, por tratar-se de uma alternativa eficaz para a criação de classificadores mais acurados. De fato, resultados teóricos e empíricos demonstram que métodos *ensemble* apresentam acurácia superior em relação aos classificadores individuais que os compõem (TUMER; GHOSH, 1996; DIETTERICH, 2000a; KUNCHEVA, 2004). O possível aumento de acurácia está diretamente relacionado à capacidade que os métodos *ensemble* tem de agregar o conhecimento obtido por cada um de seus componentes, determinando assim, uma solução global potencialmente superior à melhor das soluções individuais. Para a construção de métodos *ensembles* eficazes, duas premissas básicas devem ser obedecidas: a acurácia individual do classificador de base e a diversidade dos componentes do *ensemble* (DIETTERICH, 2000a).

Muitas aplicações atuais motivam a utilização de métodos *ensemble* como, por exem-

plo: predição de séries temporais (MA et al., 2015); modelos de diagnósticos de falhas (XUE et al., 2015); aplicações referentes a geração de energia eólica (LIU et al., 2015); segurança cibernética (FOLINO; PISANI, 2015); e diagnósticos de doenças como diabetes (HAN et al., 2015).

1.1 TRABALHOS RELACIONADOS

A maior parte dos trabalhos desenvolvidos na área de Aprendizado de Máquinas para métodos de classificação consiste no desenvolvimento de novas estratégias com intuito de aumentar a capacidade de generalização dos modelos. Estudos demonstram que técnicas que envolvem a combinação de alguns classificadores de base em um único modelo tendem a prover ganhos de acurácia em relação a cada membro individual, evitando super ajuste (*overfitting*), aumentando a estabilidade e reduzindo o viés e a variância do modelo (BREIMAN, 1998; DIETTERICH, 2000a). É possível encontrar na literatura uma vasta gama de resultados empíricos e teóricos em relação a métodos que combinam hipóteses (DIETTERICH, 2000a; KUNCHEVA, 2004; ZHANG; MA, 2012).

Em geral, os métodos *ensemble* são classificados segundo três aspectos principais (ROLI et al., 2001): a escolha do classificador de base, a estratégia de combinação das saídas e o tratamento dos dados de entrada.

- Classificador de base: modelos de Árvore de Decisão e Redes Neurais Artificiais (RNAs) são os mais comumente empregados (DIETTERICH, 2000a; TIAN et al., 2012; PARVIN et al., 2015; ADAMU et al., 2015). Existem trabalhos que empregam o uso de algoritmos de Máquina de Vetores Suporte (*Support Vector Machines* – SVM) (LAI et al., 2015) ou ainda a mistura de diferentes tipos de classificadores, chamados *ensembles* heterogêneos ou mistura de especialistas (KUNCHEVA, 2004).

- Combinação de saídas: as estratégias mais comuns incluem a média das hipóteses e também votos, ponderados ou não (LAM; SUEN, 1997). Outros métodos de combinação de saídas incluem ainda *Naive Bayes Combination*, aproximações probabilísticas e métodos multinomiais (KUNCHEVA, 2004).

- Tratamento dos dados de entrada: geralmente, abordagens baseadas na manipulação dos dados de treinamento para geração de múltiplas hipóteses e a adoção de medidas de diversidade são as mais utilizadas (KUNCHEVA; WHITAKER, 2003).

Dois métodos para construção de *ensembles* amplamente utilizados são os chamados Bagging (BREIMAN, 1996) e Boosting, cujo principal representante é o algoritmo de Boosting Adaptativo (Adaptive Boosting – AdaBoost) (FREUND; SCHAPIRE, 1996). Uma característica interessante desses métodos, vale ressaltar, reside no fato de que, caso o classificador de base escolhido seja instável, pequenas alterações no conjunto de treinamento produzem grandes modificações na hipótese gerada, trazendo diversidade para o *ensemble* formado (BREIMAN, 1996; DIETTERICH, 2000b). Os métodos anteriormente citados utilizam um subconjunto diferente da base de treinamento para cada classificador individual com objetivo de produzir hipóteses diversas em relação aos dados de entrada.

Em relação ao Bagging, para cada classificador individual, uma permutação com repetição dos dados de entrada é gerado e o *ensemble* é construído paralelamente segundo a geração de seus componentes. Por outro lado, o AdaBoost emprega uma estratégia de penalização dos dados segundo uma distribuição atualizada a cada novo componente gerado, implicando em uma construção sequencial do modelo *ensemble*. Por fim, enquanto o método de Bagging emprega, como saída final da classificação, o voto majoritário não ponderado, o AdaBoost utiliza um esquema de voto ponderado por uma função da acurácia da fase de treinamento.

Estudos comparativos mostram que o AdaBoost é capaz de superar consistentemente algoritmos de Bagging na maior parte dos casos investigados (QUINLAN, 1996; BREIMAN, 1998; BAUER; KOHAVI, 1999; DIETTERICH, 2000a). Dessa forma, o consenso geral estabelecido é de que os métodos de Boosting, como o AdaBoost, imprimem taxas de erro inferiores ao Bagging e, portanto, são métodos mais precisos sobre uma ampla variedade de conjuntos de dados (BREIMAN, 1998). Assim, o AdaBoost é considerado o método do estado da arte na área de aprendizado por *ensembles* e é frequentemente usado em comparações e avaliações experimentais.

Em relação à construção de modelos *ensembles*, muito esforço tem sido concentrado no desenvolvimento de estratégias capazes de aumentar a diversidade dos componentes dos *ensembles* e, com isso, melhorar a acurácia do método global. Algumas estratégias focam em produzir diversidade ao aplicar métodos de seleção de características (CUNNINGHAM; CARNEY, 2000), ao passo que outras abordagens focam na aplicação de métodos de agrupamento (PARVIN et al., 2015) ou algoritmos bioinspirados (TIAN et al., 2012). Apesar disso, não existem estudos conclusivos definindo qual medida de diversidade é a

mais adequada. Assim, contribuições com intuito de prover um aumento na diversidade dos componentes do classificador são ainda relevantes. Vale ressaltar que, a melhoria na diversidade do *ensemble* deve ser observada em conjunto com a acurácia do classificador de base escolhido. De outra forma, a diversidade não é garantia de bons resultados.

1.2 MOTIVAÇÃO

Dois dos principais trabalhos na área de aprendizado de *ensembles* (DIETTERICH, 2000a; KUNCHEVA, 2004) são claros ao mencionar que não existe uma teoria unificada em relação aos métodos de construção, bem como não existe uma medida de diversidade mais adequada para cada classificador de base e cada problema avaliado. O consenso é de que não faz sentido a geração de componentes diversos se os classificadores de base que compõem o *ensemble* não são acurados. Os métodos para a combinação das saídas dos componentes são inúmeros e, da mesma forma, não existem resultados sobre a otimalidade de algum.

Em particular, métodos *ensemble* construídos a partir da combinação de RNAs, embora capazes de apresentar bons resultados, frequentemente tem um desempenho, em termos de capacidade de generalização, inferior a métodos como Bagging e AdaBoost. Uma justificativa para esse fato pode ser encontrada na acurácia das RNAs selecionadas ou ainda na diversidade dos componentes e forma de combinação das soluções. Trabalhos anteriores relacionados a *ensemble* de RNAs incluem os métodos propostos por Hansen e Salamon (1990); Opitz et al. (1996); Zhou et al. (2002).

Por essa razão, esse trabalho é motivado principalmente por contribuições relacionadas a investigação dessas questões. Em relação à acurácia do classificador de base, o intuito é fornecer uma alternativa para melhorar a solução de um classificador do tipo Perceptron, o modelo mais simples de RNAs. Considerando a geração de componentes, opta-se por investigar a combinação de algumas heurísticas de diversidade tratadas em conjunto a uma medida de dissimilaridade, ao invés de simplesmente aplicar uma única medida com intuito de aumentar a diversidade entre os componentes.

Além disso, duas abordagens distintas para a combinação das saídas dos componentes do *ensemble* são avaliadas com o objetivo de identificar possíveis melhorias de capacidade de generalização definidas a partir da estratégia utilizada. Uma das estratégias frequentemente utilizadas é o voto majoritário dos componentes (DIETTERICH, 2000b). Essa

abordagem, embora apresente bons resultados, demanda um alto custo computacional, uma vez que, durante a fase de teste, todos os componentes do *ensemble* devem ser testados. Esse problema não ocorre quando o método de combinação é, por exemplo, a média dos componentes. Nesse caso, o cálculo da média estabelece uma hipótese única a ser avaliada durante a fase de teste. Assim, esse trabalho apresenta uma alternativa de solução motivada por esse problema. Essa alternativa consiste no desenvolvimento de um algoritmo capaz de gerar uma hipótese única equivalente à votação majoritária de um *ensemble*.

1.3 OBJETIVOS

Os objetivos desse trabalho podem ser divididos em três pontos principais:

1. Apresentar de um método *ensemble* capaz de apresentar bons resultados em termos de capacidade de generalização, a partir da combinação de classificadores Perceptron;
2. Analisar a diferença na capacidade de generalização gerada por duas formas de combinação das saídas dos componentes: a média e o voto majoritário das soluções;
3. Apresentar um algoritmo capaz de determinar a hipótese equivalente à votação majoritária dos componentes de um *ensemble*.

1.4 CONTRIBUIÇÕES

As contribuições principais desse trabalho estão listadas a seguir:

- Introduzir o *Ensemble* de Perceptrons Balanceados (*Ensemble of Balanced Perceptrons* – EBP), derivado em variáveis primais (ENES et al., 2015a);
- Apresentar o *Ensemble* de Perceptrons *Kernels* Balanceados (*Ensemble of Balanced Perceptrons Kernels* – EBPK), extensão do modelo EBP, derivado em variáveis duais para introdução de funções *kernel* e solução de problemas não-linearmente separáveis (ENES et al., 2015b);
- Propor a Máquina de Redução do Espaço de Versões (*Version Space Reduction Machine* – VSRM) , derivado em variáveis primais, para geração da hipótese equivalente ao voto majoritário de um *ensemble* de Perceptrons Balanceados.

Entre as contribuições secundárias, destacam-se:

- O balanceamento da solução Perceptron e a introdução dos modelos Perceptron Balanceado (PB) e Perceptron *Kernel* Balanceado (PKB) no contexto de *ensembles* de Perceptrons;
- A introdução de medidas de dissimilaridade distintas para geração de diversidade nos componentes do *ensemble*, sendo distância Euclidiana na formulação primal e distância de Tanimoto na formulação dual.
- A apresentação do modelo Perceptron de Margem Geométrica Variável (*Geometric Variable Margin Perceptron* – GVMP), desenvolvido a partir de modificações no Perceptron de Margem Geométrica Fixa (*Geometric Fixed Margin Perceptron* – GFMP), proposto por Leite e Fonseca Neto (2008).

1.5 ORGANIZAÇÃO

Essa dissertação está organizada em 6 capítulos. Após o capítulo de introdução, a fundamentação teórica e os conceitos principais são apresentados no Capítulo 2. Discute-se o paradigma de aprendizado supervisionado e o problema de classificação binária. Os conceitos fundamentais referentes a teoria de aprendizado por *ensemble* são apresentados. Ao final, o modelo Perceptron é introduzido, bem como todas as formulações relacionadas ao Perceptron necessárias para o entendimento do trabalho e ainda o tópico relativo a classificadores *kernel*. O Capítulo 3 apresenta o método *ensemble* proposto derivado em suas formulações primal e dual. O principal objetivo desse capítulo é apresentar o classificador de base e as medidas de diversidade empregadas para a construção do método *ensemble*. Isso constitui a base do trabalho proposto. No Capítulo 4, o objetivo é apresentar o VSRM para geração de uma hipótese equivalente à um *ensemble* de Perceptrons balanceados combinados pelo voto majoritário. O estudo experimental realizado, as bases de dados avaliadas, os resultados obtidos, bem como as discussões relevantes em relação a eles são apresentadas no Capítulo 5. Por fim, o Capítulo 6 apresenta as conclusões do trabalho em questão, algumas considerações finais e alguns dos trabalhos futuros sugeridos.

2 FUNDAMENTAÇÃO TEÓRICA

As pesquisas na área de IA tiveram início no final da década de 40 (RUSSELL; NORVIG, 1995). Atualmente, trata-se de uma vasta área de pesquisa com diversas subáreas, entre elas, a área de Aprendizado de Máquinas, objeto de estudo desse trabalho. Esse subcampo da IA é dedicado ao desenvolvimento de algoritmos que buscam por padrões dentro de um conjunto de dados, os algoritmos de aprendizado. Essas técnicas são desenvolvidas com intuito de possibilitar que uma máquina possa "aprender", i.e, reconhecer padrões, aprimorando seu desempenho em determinada tarefa (MICHALSKI et al., 2013).

Na área de Aprendizado de Máquinas, três paradigmas de aprendizado destacam-se: supervisionado, não supervisionado e o aprendizado por reforço. Em particular, no caso do aprendizado supervisionado, se as classes possuirem valores discretos, o problema é categorizado como classificação. Caso as classes possuam valores contínuos, o problema é categorizado como regressão (MICHALSKI et al., 2013). O objetivo, no caso da classificação, é induzir conceitos a partir de exemplos que estão pré-classificados, ou seja, exemplos que estão rotulados com uma classe conhecida. Este trabalho está fundamentado no paradigma de aprendizado supervisionado, em particular, trata-se de problemas de classificação binária. Em problemas de classificação, várias técnicas podem ser empregadas. Este trabalho trata de umas das técnicas, a de aprendizado por *ensemble*.

Sendo assim, o problema de classificação binária é definido a seguir. Posteriormente, apesar de ser possível utilizar vários tipos de classificadores como componentes em *ensembles*, este trabalho está restrito ao uso do Perceptron, devido principalmente à sua simplicidade e eficácia, para implementação do método *ensemble* proposto. O modelo Perceptron é definido nesse capítulo em termos de variáveis primais e duais, apresentando o conceito de classificadores *kernel* e sua aplicação na solução de problemas não-linearmente separáveis. Por fim, define-se também o Perceptron de Margem Geométrica Fixa (Geometric Fixed Margin Perceptron – GFMP) (LEITE; FONSECA NETO, 2008) em variáveis primais, modelo base usado para implementação do VSRM. Por fim, os conceitos necessários referentes a teoria de aprendizado por *ensemble* são definidos.

2.1 CLASSIFICAÇÃO BINÁRIA

Considere o paradigma de aprendizado supervisionado, no qual são utilizados dados cujas classes são conhecidas a priori. Uma tarefa de classificação binária pode ser definida da seguinte forma: seja o conjunto Z dos dados de entrada de cardinalidade m, composto de um conjunto de vetores de entrada x_i e de um conjunto de escalares y_i. Defina $Z = \{(x_i, y_i) \ : \ i \in \{1, 2, ..., m\}\}$, como o conjunto de treinamento de m amostras de uma função desconhecida f. Cada componente dos vetores de entrada é um valor real e cada vetor de entrada está, portanto, inserido em um espaço de dimensão d, i.e., $x_i \in \mathbb{R}^d$. Esses componentes são chamados de atributos ou características e podem admitir valores reais ou discretos. Cada vetor de entrada é rotulado por um escalar y_i. No caso do problema de classificação binária, os valores de y_i são mapeados em um conjunto discreto de classes, i.e, $y_i \in \{-1, +1\}$, com $i \in \{1, 2, ..., m\}$. Dado um conjunto de treinamento, um algoritmo de aprendizado é capaz de gerar um classificador, dado por uma hipótese (representada por um hiperplano) em relação à função $f : \mathbb{R}^d \to \{-1, +1\}$.

O classificador gerado, componente do sistema responsável por classificar padrões de uma classe, prediz os valores correspondentes de y à medida que amostras adicionais x, não vistas anteriormente, são apresentadas. Essas amostras adicionais constituem o chamado conjunto de teste.

2.2 ALGORITMO PERCEPTRON

Baseado no modelo de neurônio artificial apresentado por McCulloch e Pitts (1943), Frank Rosenblatt propõe em 1958 um procedimento, simples e eficaz, para a atualização de um vetor de pesos, a partir de um elemento processador com múltiplas saídas. A medida de avaliação baseia-se na comparação do valor obtido pela saída do modelo com os valores de saída desejados. O modelo, chamado de Perceptron (ROSENBLATT, 1958), foi o primeiro modelo de aprendizado supervisionado e consiste na arquitetura mais simples de uma RNA.

Estruturalmente, o Perceptron realiza um mapeamento de um espaço de entrada de dimensão d para um espaço de saída de dimensão reduzida n, associando cada vetor de entrada, x_i, a um componente de um vetor de pesos w_i de dimensão d, com $i \in \{1, 2, ..., m\}$ e uma camada de saída y formada por n unidades. Quando empregado em problemas de

classificação binária, é suficiente a existência de somente um elemento processador na camada de saída. A estrutura do Perceptron é apresentada na Figura 2.1

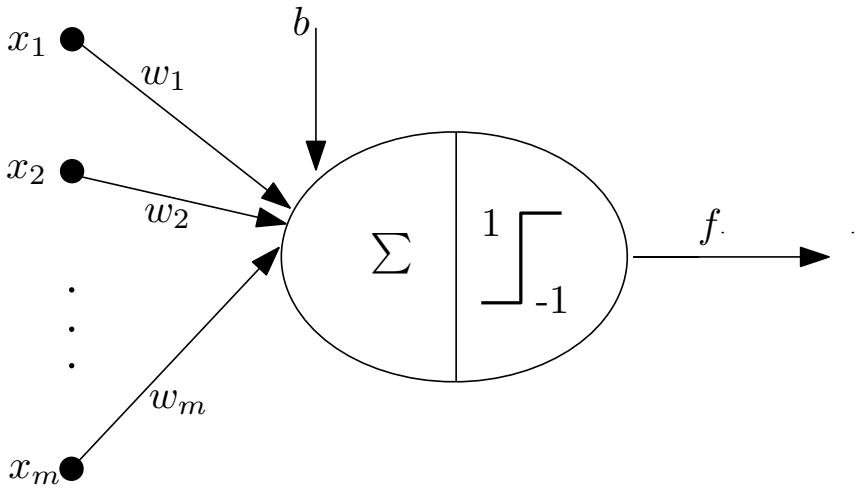


Figure 2.1: Modelo Perceptron

Novikoff (1963) provou que a convergência do algoritmo é garantida em um número finito de iterações, caso o conjunto de treinamento seja linearmente separável. O número de iterações está diretamente relacionado à quantidade de atualizações necessárias do vetor de pesos. Dessa forma, quanto mais erros o algoritmo comete, mais atualizações são necessárias, bem como mais iterações do algoritmo.

O vetor de pesos é, então, determinado com base em sucessivas correções e, portanto, o hiperplano separador das classes é construído de forma iterativa, caracterizando um processo de aprendizado *online*.

Em relação a aplicabilidade do algoritmo, trata-se de um classificador linear capaz de classificar apenas dados linearmente separáveis, em sua implementação original. Entretanto, como a maior parte dos problemas reais de predição são de natureza não-linearmente separável, a usabilidade do modelo passou a ser questionada. Por essa razão, em Aizerman et al. (1964), foi investigado e mostrou-se a possibilidade de utilização de funções *kernel* para a solução de problemas não-linearmente separáveis e, assim foi introduzida a formulação do algoritmo Perceptron em variáveis duais. A formulação matemática do modelo Perceptron em variáveis primais e duais é detalhado a seguir.

2.2.1 PERCEPTRON PRIMAL

Matematicamente, o modelo Perceptron proposto por Rosenblatt, em variáveis primais, consiste em encontrar um hiperplano separador obtido por meio da solução de um sistema

de inequações lineares, dado por

$$f(x_i) = \begin{cases} +1, & \langle w, x_i \rangle + b \geq 0 \\ -1, & \langle w, x_i \rangle + b < 0, \end{cases} \tag{2.1}$$

no qual w é o vetor de pesos, b o valor de *bias* e $\langle w, x_i \rangle$ é o produto interno de w por x_i.

Uma amostra de treinamento (x_i, y_i) é uma instância incorretamente classificada se $y_i(\langle w, x_i \rangle + b) < 0$. Enquanto uma amostra de treinamento for classificada incorretamente, a regra de correção deve ser aplicada até que não haja mais erros. A regra de correção é definida da seguinte forma

$$\begin{aligned} w^{t+1} &\leftarrow w^t + \eta x_i y_i \\ b^{t+1} &\leftarrow b^t + \eta y_i, \end{aligned} \tag{2.2}$$

na qual η é a taxa de aprendizado e t a variável de iteração.

A resposta final do algoritmo de aprendizado é obtida quando o hiperplano solução é capaz de classificar todas as amostras de treinamento corretamente. O Algoritmo 2.1 descreve o pseudocódigo do algoritmo primal relativo ao treinamento do Perceptron.

2.2.2 PERCEPTRON DUAL

A versão dual do modelo Perceptron, proposta por Aizerman et al. (1964), pode ser aplicada para solução tanto de problemas linearmente separáveis, quanto problemas não-linearmente separáveis. Entretanto, para os casos nos quais o conjunto de dados analisado é não-linearmente separável, sua utilização é imprescindível. Essa representação é também chamada de representação dependente dos dados e pode ser interpretada como um classificador *kernel*. Para modelar o Perceptron em termos de variáveis duais, torna-se necessário o mapeamento dos dados para um espaço de mais alta dimensão, chamado espaço de características, ou ϕ-*space*, comumente representado por F. Com o mapeamento $\phi : \mathbb{R}^d \times \mathbb{R}^d \rightarrow F$, é possível a representação do conjunto de amostras não-linearmente separável em um espaço de mais alta dimensão, $x \rightarrow \phi(x)$, no qual o problema se torna linearmente separável.

Essa modelagem permite a introdução de funções *kernel* e posterior solução do problema relacionado à construção de uma hipótese linear no espaço de variáveis duais. Por isso, é também chamada de Perceptron *Kernel* (PK). Para isso, algumas alterações no

Algorithm 2.1: Perceptron Primal

Entrada: conjunto de treinamento: $Z = \{(x_i, y_i)\}$ de cardinalidade m;
taxa de aprendizado: η;
limite superior no número de iterações: max;
Saída: vetor de pesos w e *bias* b;
início
  inicializar (w^0, b^0);
  $j \leftarrow 0$;
  $t \leftarrow 0$;
  $stop \leftarrow$ falso;
  enquanto $j \leq max$ ***e*** $\neg stop$ **faça**
    $erro \leftarrow$ falso;
    para i ***de*** 1 ***até*** m **faça**
      se $y_i \left(\langle w^t, x_i \rangle + b^t\right) < 0$ **então**
        $w^{t+1} \leftarrow w^t + \eta x_i y_i$;
        $b^{t+1} \leftarrow b^t + \eta y_i$;
        $t \leftarrow t + 1$;
        $erro \leftarrow$ verdadeiro;
      fim se
    fim para
    se $\neg erro$ **então**
      $stop \leftarrow$ verdadeiro;
    fim se
    $j \leftarrow j + 1$;
  fim enquanto
fim

modelo proposto por Rosenblatt (1958) devem ser consideradas. O vetor de pesos w é modificado para ser representado como um combinação linear dos vetores de entrada (x_i, y_i) da seguinte forma

$$w = \sum_{j=1}^{m} \alpha_j y_j \phi(x_j), \tag{2.3}$$

na qual $\alpha \in \mathbb{R}^m$, $\alpha \geq \mathbf{0}$, é o vetor de multiplicadores ou variáveis duais associado ao conjunto de entrada.

Substituindo a expansão do vetor w, dada pela Eq.(2.3), na equação original de variáveis primais, Eq. (2.1), a função f passa a ser definida da seguinte forma

$$f(x_i) = \begin{cases} +1, & \sum_{j=1}^{m} \alpha_j y_j y_i \left\langle \phi(x_i), \phi(x_j) \right\rangle + b \geq 0 \\ \\ -1, & \sum_{j=1}^{m} \alpha_j y_j y_i \left\langle \phi(x_i), \phi(x_j) \right\rangle + b < 0, \end{cases} \tag{2.4}$$

A grande vantagem de um classificador *kernel* é que não é necessário conhecer o tipo de mapeamento ou a função ϕ explicitamente. Para tanto, utiliza-se uma função *kernel*, simétrica e semi-definida positiva, definida por $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$. Os valores obtidos pela função K são correspondentes ao cálculo do produto interno dos vetores mapeados em um espaço de mais alta dimensão (KIVINEN et al., 2004).

A formulação dual do modelo Perceptron consiste em encontrar um hiperplano separador, dado pela solução do sistema de inequações lineares, definido da seguinte forma

$$f(x_i) = \begin{cases} +1, & \sum_{j=1}^{m} \alpha_j y_j K(x_j, x_i) + b \geq 0 \\ \\ -1, & \sum_{j=1}^{m} \alpha_j y_j K(x_j, x_i) + b < 0, \end{cases} \tag{2.5}$$

na qual b é o valor de *bias* e $K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$.

Uma amostra de treinamento (x_i, y_i) é classificada incorretamente se

$$y_i \left(\sum_{j=1}^{m} \alpha_j y_j K(x_j, x_i) + b \right) < 0. \tag{2.6}$$

Caso isso ocorra, a regra de correção do modelo deve ser aplicada até que não haja mais instâncias incorretamente classificadas. A regra de correção é então definida

$$\begin{aligned} \alpha_i^{t+1} &\leftarrow \alpha_i^t + \eta \cdot 1 \\ b^{t+1} &\leftarrow b^t + \Delta\alpha_i y_i, \end{aligned} \tag{2.7}$$

na qual $\Delta\alpha_i y_i$ refere-se ao somatório das correções nos valores dos multiplicadores considerando o respectivo sinal das classes, η é a taxa de aprendizagem e t a variável de iteração. O valor do *bias* pode ser computado separadamente em um esquema do tipo online conforme o vetor de pesos. Ao passo que todas as instâncias de treinamento são corretamente classificadas, está definido o hiperplano separador solução do problema no espaço de características. O Algoritmo 2.2 descreve o pseudocódigo do algoritmo dual relativo ao treinamento do PK.

Existem uma infinidade de funções *kernel* discutidas na literatura, como por exemplo, *kernel* exponencial, *kernel* Laplaciano e *kernel* sigmoidal (HOFMANN et al., 2008). Entretanto, três principais se destacam e são frequentemente utilizadas:

- *Kernel* linear é a função *kernel* mais simples e é frequentemente utilizado para

Algorithm 2.2: Perceptron *Kernel*

Entrada: conjunto de treinamento: $Z = \{(x_i, y_i)\}$ de cardinalidade m;
taxa de aprendizado: η;
limite superior no número de iterações: max;
Saída: vetor de multiplicadores α e *bias* b;
início
  inicializar (α^0, b^0);
  $j \leftarrow 0$;
  $t \leftarrow 0$;
  $stop \leftarrow$ falso;
  enquanto $j \leq max$ ***e*** $\neg stop$ **faça**
    $erro \leftarrow$ falso;
    para i ***de*** 1 ***até*** m **faça**
      se $y_i \left(\sum_{j=1}^{m} \alpha_j y_j K(x_j, x_i) + b \right) < 0$ **então**
        $\alpha_i^{t+1} \leftarrow \alpha_i^t + \eta \cdot 1$;
        $t \leftarrow t + 1$;
        $erro \leftarrow$ verdadeiro;
      fim se
    fim para
    $b^{t+1} \leftarrow b^t + \Delta\alpha_i y_i$;
    se $\neg erro$ **então**
      $stop \leftarrow$ verdadeiro;
    fim se
    $j \leftarrow j + 1$;
  fim enquanto
fim

solução de problemas linearmente separáveis nos modelos derivados em variáveis duais. A função *kernel* linear é dada por

$$K(x_i, x_j) = \langle x_i \cdot x_j \rangle + c; \tag{2.8}$$

- *Kernel* polinomial é normalmente utilizado em problemas normalizados, devido a sua característica não estacionária. Essa função é definida com base na variação do grau d do polinômio da seguinte forma

$$K(x_i, x_j) = (\langle x_i \cdot x_j \rangle + 1)^d; \tag{2.9}$$

- *Kernel* gaussiano é um exemplo de *Kernel* de Funções de Base Radial, cujo paramêtro a ser avaliado é a largura da gaussiana σ. Note na Eq. (2.10) de definição do *kernel*,

que $\|.\|_2^2$ define a norma Euclidiana ao quadrado.

$$K(x_j, x_i) = exp\left(-\frac{\|x_j - x_i\|_2^2}{2\sigma^2}\right). \tag{2.10}$$

2.2.3 PERCEPTRON DE MARGEM GEOMÉTRICA FIXA – GFMP

Uma característica da formulação original do Perceptron de Rosenblatt reside na definição do hiperplano solução. A primeira solução na qual todos os pontos do conjunto de treinamento são classificados corretamente determina o critério de parada do algoritmo. Em Duda et al. (2001), uma modificação do Perceptron original é proposta, o *Variable-Increment Perceptron with Margin* (VIPM), que inclui a utilização de uma regra de incremento variável e a utilização de um valor fixo de margem funcional, γ, no sentido de prover um critério de relaxação para o método.

Considerando a inclusão de um valor fixo de margem para geração do modelo, a função f é determinada pela solução do sistema de inequações lineares é dada por

$$f(x_i) = \begin{cases} +1, & \langle w, x_i \rangle + b \geq \gamma \\ -1, & \langle w, x_i \rangle + b < \gamma. \end{cases} \tag{2.11}$$

Assim, uma amostra de treinamento é classificada incorretamente se $y_i(\langle w, x_i \rangle + b) < \gamma$.

Essa formulação depende da limitação do valor da norma do vetor de pesos w, de forma que, $\|w\|_2 = 1$. Considerando que o problema em questão é linearmente separável, nos casos nos quais não é possível limitar o valor da norma em $\|w\|_2 = 1$, o sistema de inequações sempre apresentará uma solução viável para qualquer valor de margem funcional γ fixado. A existência da solução viável leva ao crescimento exagerado da norma de w e, consequentemente, do cálculo do produto interno do sistema de inequações. Dessa forma, é preciso estabelecer um critério de regularização para que seja possível controlar o crescimento do valor da norma de w.

Como alternativa para a solução do problema gerado pelo VIPM, Leite e Fonseca Neto (2008) propõem uma abordagem chamada de Perceptron de Margem Fixa Geométrica (*Geometric Fixed Margin Perceptron* – GFMP). Os autores sugerem uma modificação no Perceptron que determina que uma distância mínima seja estabelecida do conjunto de dados em relação ao hiperplano separador, ao passo que garante que todo o conjunto de

dados seja classificado corretamente. Essa restrição não limita diretamente o valor de norma do vetor w ao valor unitário, mas controla seu crescimento. Para tanto define-se o valor de margem fixa γ_f que corresponde ao valor de margem funcional γ da respectiva amostra dividido pelo valor da norma de w, $\|w\|_2$. Deve-se então adaptar o sistema de inequações lineares da seguinte forma

$$f(x_i) = \begin{cases} +1, & (\langle w, x_i \rangle + b)/\|w\|_2 \geq \gamma_f \\ -1, & (\langle w, x_i \rangle + b)/\|w\|_2 < \gamma_f. \end{cases} \tag{2.12}$$

De forma análoga, uma amostra de treinamento é classificada incorretamente se, só se, $y_i(\langle w, x_i \rangle + b) < \gamma_f \|w\|_2$. Em razão dessa modificação, é necessário também reescrever a regra de correção quando aplicada a uma determinada amostra (x_i, y_i), da seguinte forma

$$w^{t+1} \leftarrow w^t - \eta \left(\gamma_f w^t / \left\|w^t\right\|_2 - x_i y_i \right), \tag{2.13}$$

na qual $w^t/\|w^t\|_2$ representa o vetor unitário de direção w^t. Esse termo é equivalente a subtração do valor da margem fixa na parcela de correção de w^t. Geometricamente, esse procedimento é sintetizado pela Figura 2.2.

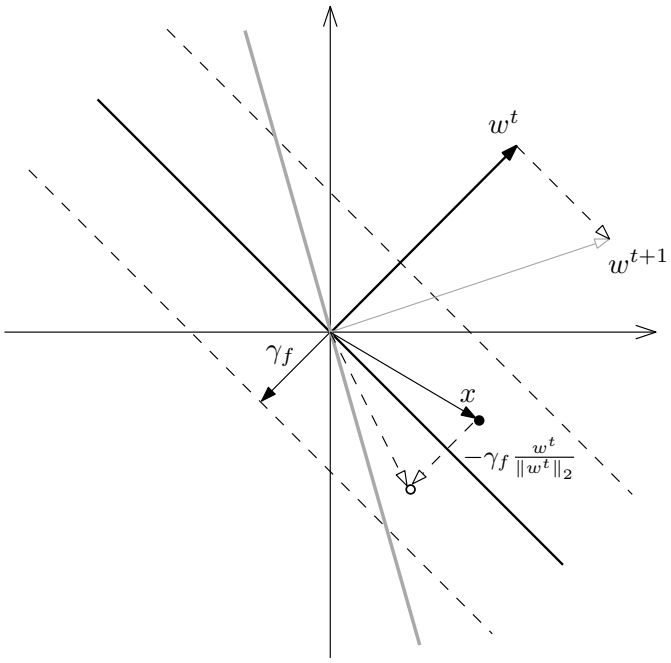


Figure 2.2: Correção geométrica do vetor w^t para o GFMP.

Novamente, a resposta final do algoritmo é obtida ao passo que o hiperplano separador é capaz de classificar todas as amostras de treinamento corretamente, seguindo a solução

do sistema de inequações dado pela Eq. (2.12).

O Algoritmo 2.3 descreve o pseudocódigo do algoritmo primal relativo ao treinamento do GFMP.

Algorithm 2.3: Perceptron de Margem Geométrica Fixa Primal

Entrada: conjunto de treinamento: $Z = \{(x_i, y_i)\}$ de cardinalidade m;
margem geométrica fixa: γ_f;
taxa de aprendizado: η;
limite superior no número de iterações: max;
Saída: solução: vetor de pesos w e *bias* b;

```
início
    inicializar (w^0, b^0);
    j ← 0;
    t ← 0;
    stop ← falso;
    enquanto j ≤ max e ¬stop faça
        erro ← falso;
        para i de 1 até m faça
            se y_i (⟨w^t, x_i⟩ + b^t) < γ_f ||w^t||_2 então
                w^{t+1} ← w^t − η (γ_f w^t / ‖w^t‖_2 − x_i y_i);
                b^{t+1} ← b^t + η y_i;
                t ← t + 1;
                erro ← verdadeiro;
            fim se
        fim para
        se ¬erro então
            stop ← verdadeiro;
        fim se
        j ← j + 1;
    fim enquanto
fim
```

2.3 MÁQUINA DE VETORES SUPORTE – SVM

SVMs são classificadores de máxima margem introduzidos por Boser et al. (1992). Essa técnica visa separar o conjunto de treinamento através do hiperplano que maximiza a distância entre as classes opostas. Para obter o hiperplano de máxima margem, capaz de classificar todas as amostras corretamente, é necessário que se resolva o seguinte problema de otimização

$$\begin{aligned} &\max_{(w,b)} \left(\min_i \frac{y_i \left(\langle w, x_i \rangle + b \right)}{||w||_2} \right) \\ &\text{sujeito a } y_i \left(\langle w, x_i \rangle + b \right) > 0. \end{aligned} \tag{2.14}$$

Esse problema é equivalente a seguinte solução

$$\begin{aligned} &\max \gamma_g \\ &\text{sujeito a } y_i \left(\langle w, x_i \rangle + b \right) \geq ||w||_2 \gamma_g, \end{aligned} \tag{2.15}$$

na qual γ_g é o valor de margem geométrica.

Definindo $\gamma_g ||w||_2 = 1$ como o valor mínimo de margem funcional, o trabalho de Vapnik (2013) apresenta a derivação da formulação clássica do modelo SVM, capaz de minimizar a norma Euclidiana do vetor, da seguinte forma

$$\begin{aligned} &\min \frac{1}{2} ||w||_2^2 \\ &\text{sujeito a } y_i \left(\langle w, x_i \rangle + b \right) \geq 1. \end{aligned} \tag{2.16}$$

Objetivando facilitar a solução desse problema, é conveniente relaxar as restrições das inequações por meio da introdução de um conjunto de multiplicadores Lagrangeanos não-negativos α_i, no qual $i \in \{1, 2, ..., m\}$. Ao incorporar as restrições relaxadas, a função Langrangeana é dada por

$$L \left(w, b, \alpha \right) = \frac{1}{2} ||w||_2^2 - \sum_i \alpha_i y_i \left(\langle w, x_i \rangle + b \right) + \sum_i \alpha_i. \tag{2.17}$$

Essa função deve ser minimizada em relação a w e b e maximizada em relação a α, sujeito a $\alpha_i \leq \mathbf{0}$ para todo $i \in \{1, 2, ..., m\}$. Essa solução pode ser obtida através da maximização da funções estritamente dual, na qual os parâmetros w e b são substituídos. Essa formulação, em particular, é chamada de *Wolfe's dual* (BURGES, 1998). Dessa forma, é possível obter a formulação dual do SVM estritamente em função dos valores dos

multiplicadores α, como segue

$$\max L(\alpha) = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j \langle x_i, x_j \rangle \tag{2.18}$$

$$\text{sujeito a} \begin{cases} \sum_i \alpha_i y_i = 0 \\ \alpha_i \geq 0. \end{cases}$$

Ao solucionar esse problema, os valores ótimos de α^* são obtidos. Assim, é possível reconstruir o vetor normal w para essa solução da seguinte forma

$$w^* = \sum_i \alpha_i^* y_i x_i. \tag{2.19}$$

2.4 MÉTODOS *ENSEMBLES*

O trabalho de Hansen e Salamon (1990) foi pioneiro em conduzir um estudo envolvendo a combinação de várias RNAs distintas para a solução de problemas de classificação binária. O argumento principal era que, em geral, esses problemas envolvem mais de um mínimo local. Por essa razão, a utilização de uma única rede, embora apresente bons resultados, poderia estar muitas das vezes restrita a esses pontos de mínimo local. Dessa forma, a solução poderia ser potencializada caso outras redes fossem avaliadas em conjunto, ao invés de descartadas após a escolha da melhor delas.

Os autores argumentam ainda que, caso as taxas de erro de cada modelo individual sejam distintas e menores do que 50%, então haverá melhoria de acurácia. Isso ocorre, já que a probabilidade da saída do modelo em conjunto estar errada seria inferior a menor das taxas de erro dos seus componentes isoladamente. Testes empíricos validaram o trabalho ao mostrar ganhos em termos da capacidade de generalização do modelo criado em relação a uma RNA individual. Nesse mesmo trabalho de Hansen e Salamon (1990), o termo *ensemble* já é empregado para designar a combinação dos classificadores. Entretanto, o termo "comitê" é frequentemente empregado com o mesmo significado.

A partir daí, estudos começaram a ser desenvolvidos com o intuito de investigar as vantagens de se combinar vários modelos no formato de um *ensemble*. Por definição, *ensemble* é um paradigma de aprendizado em que uma coleção finita de propostas alternativas para a solução de um mesmo problema, denominadas componentes, são empregadas em con-

junto na proposição de uma solução única (SOLLICH; KROGH, 1996). Nesse caso, cada componente representa, isoladamente, uma potencial solução para o problema.

Por se tratar de um área de pesquisa recente, ainda não existe uma teoria unificada de aprendizado por *ensembles*. Entretanto, existem muitas razões teóricas para combinar classificadores e ainda mais razões empíricas da eficácia dessa abordagem (BAUER; KOHAVI, 1999; DIETTERICH, 2000a).

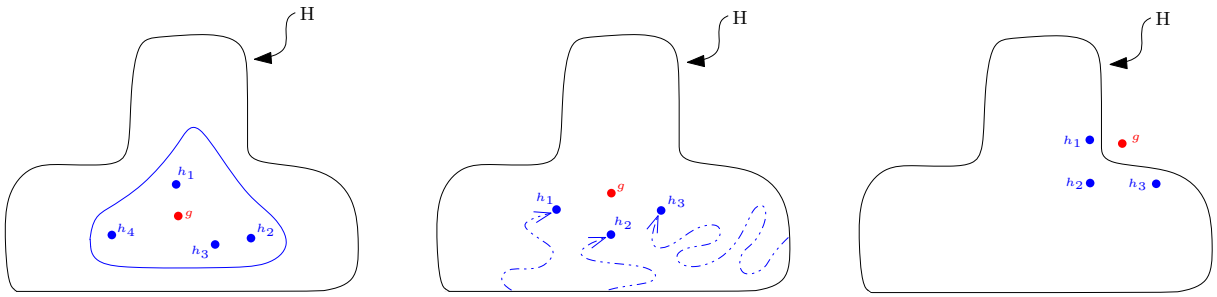
Tome H como o espaço de todas as hipóteses soluções para um problema qualquer, h_i uma hipótese solução viável e g a hipótese ótima hipotética para o problema. O trabalho de Dietterich (2000a) destaca três justificativas principais para a construção de *ensembles*, representadas na Figura 2.3 e descritas a seguir:

- Estatística: caso o conjunto de dados de treinamento no problema seja limitado, o espaço de soluções viáveis fica restrito às soluções que podem ser geradas pelo conjunto de treinamento limitado. O modelo então é treinado e está restrito ao subconjunto formado apenas pelas observações restritas à marcação linear interna da Figura 2.3-(a). O processo de geração de soluções pode determinar várias hipóteses com desempenhos semelhantes. Supondo que a seleção das hipóteses para a construção de um *ensemble* é efetiva, ou seja, bons modelos são selecionados, a solução combinada tende a prover uma boa aproximação em relação à solução ótima. Ao observar a Figura 2.3-(a), as 4 hipóteses sugeridas pelo mecanismo de seleção estão distribuídas em torno de g e o *ensemble* gerado pela combinação das 4 hipóteses tende a se aproximar mais de g do que qualquer uma das hipóteses avaliadas isoladamente.

- Computacional: algoritmos de aprendizado são empregados no refinamento de um processo de busca limitada e local, na qual modificações locais são propostas em relação à solução atual sempre que houver melhoria no desempenho. Dessa forma, suas soluções geradas podem estar restritas à convergência para um ponto de ótimo local. Particularmente, os problemas de aprendizado, em sua maioria, apresentam vários pontos de ótimo local e, mesmo que o conjunto de dados não seja restrito, não há garantias de que uma solução gerada seja a de ótimo global. De fato, em termos computacionais, pode ser muito difícil que uma solução seja guiada para o ponto de ótimo global. Supondo, por exemplo, que o ponto de partida de cada solução apresentada seja distinto, o *ensemble* formado por diferentes processos de

busca local podem proporcionar uma melhor aproximação para g do que um único componente qualquer gerado. Este fato pode ser observado na Figura 2.3-(b), na qual as linhas tracejadas representam as buscas locais.

- Representacional: em grande parte das aplicações, a solução ótima para o problema pode não ser capaz de ser representada por nenhuma das hipóteses pertencentes ao espaço de solução viável do modelo. Dessa forma, ao combinar vários modelos viáveis em H, o espaço de hipóteses é estendido de modo a se aproximar de g. De acordo com a teoria de *ensemble*, a combinação de classificadores pretende evitar que a capacidade de representação fique restrita ao conjunto finito de hipóteses, conforme ilustrado pela Figura 2.3-(c).



(a) Estatística (b) Computacional (c) Representacional

Figure 2.3: Três razões fundamentais para a construção de um *ensemble*.

Esses três fatores justificam o emprego de *ensembles* e ainda ilustram, a partir de conclusões computacionais, estatísticas e representacionais, motivos pelos quais os *ensembles* podem apresentar uma capacidade de generalização superior a um componente individual. Entretanto, embora uma vasta gama de resultados empíricos comprovem a eficácia, em termos de capacidade de generalização da construção de um *ensemble*, não existe uma garantia de que a precisão do *ensemble* será sempre superior ao melhor de seus componentes. Sendo assim, o emprego dessa abordagem é motivado principalmente pela possibilidade de aumento da capacidade de generalização do modelo quando combinado em um *ensemble*. Para que essa possibilidade tenha chances de se concretizar, duas premissas básicas devem ser obedecidas:

- Acurácia individual: os classificadores que irão compor o *ensemble* devem ter um bom desempenho isoladamente. Quanto melhor o desempenho do classificador in-

dividual, melhor tende a ser o resultado da composição dos classificadores. Além disso, um classificador é um candidato viável se apresentar um desempenho superior àquele produzido por um classificador aleatório, ou seja, um classificador que rotula uma amostra qualquer de entrada com um índice aleatório que apresenta uma distribuição uniforme entre as classes candidatas.

- Diversidade dos componentes: claramente não haverá ganho na capacidade de generalização do *ensemble*, em comparação ao classificador individual, caso todos os seus componentes sejam idênticos. Dessa forma, outra premissa fundamental para a construção de um *ensemble* reside na diversidade de seus componentes. Dois classificadores são ditos diversos entre si se seus erros de classificação são distintos e não correlacionados, frente a um mesmo problema. Essa diversidade pode ser gerada a partir da construção de comitês heterogêneos, de configurações distintas de parâmetros e ainda da inclusão de medidas de diversidade (KUNCHEVA; WHITAKER, 2003).

Sendo determinados o modelo de classificador individual e as estratégias para geração de diversidade no modelo, o processo de construção de um modelo *ensemble* compreende o cumprimento de três etapas (KUNCHEVA, 2004):

- Geração de um conjunto de candidatos a componentes do *ensemble*: quanto a geração do conjunto de componentes viáveis, deve-se atentar para a acurácia de cada componente, que deve ser superior a 50% para problemas de classificação binária, a não correlação entre os erros e a diversidade dos componentes do conjunto em questão.

- Seleção dos candidatos viáveis: após a geração do conjunto de candidatos, o processo de seleção funciona separando aqueles componentes que contribuem mais fortemente para a criação do modelo. Essa contribuição pode ser definida de diversas formas, como por exemplo, pela introdução de medidas de dissimilaridade de componentes. O processo de seleção é ainda responsável por determinar a quantidade de componentes que formarão o *ensemble*, com base no critério de seleção adotado.

- Combinação das saídas geradas: o processo de combinação corresponde à forma com que as soluções dos componentes serão avaliadas como a solução única de um

ensemble. Em geral, métodos baseados na média dos componentes e esquemas de votações, ponderadas ou não, são comumente empregadas.

Note que, para diferenciar dois modelos *ensemble*, deve-se levar em conta a escolha do classificador de base, a presença ou não de tratamento nos dados de entrada dos modelos e ainda o método de combinação adotado, como mencionado em (ROLI et al., 2001), e não o processo de construção do mesmo. Isso se dá, uma vez que diferenças relativas à geração de um conjunto de componentes e seleção dos candidatos viáveis não são suficientes para garantir que os modelos gerados são distintos. Processos diferentes podem levar a construção de um mesmo modelo *ensemble*.

2.4.1 ADAPTIVE BOOSTING – ADABOOST

A tarefa de construir bons classificadores, funcionais e com boa capacidade de generalização, não é trivial. Por outro lado, existem diversas formas para desenvolver estratégias eficazes para a construção de *ensemble* que obedeçam as duas premissas básicas de combinação de classificadores e que sejam capazes de prover uma capacidade de generalização igual ou superior à de classificadores complexos. Em particular, uma técnica de construção de *ensembles* têm se destacado na literatura, o *Boosting* (FREUND; SCHAPIRE, 1995), cujo principal representante é o algoritmo de Boosting Adaptativo (*Adaptive Boosting* – AdaBoost), apresentado por Freund e Schapire (1996).

O AdaBoost é considerado, na literatura, o estado da arte em métodos *ensemble*. O método combina um conjunto de classificadores fracos, cuja influência é ponderada por uma importância definida para cada solução, para a construção de um único classificador forte. A injeção de diversidade na geração dos componentes é determinada por uma reamostragem dos dados de treinamento. Essa reamostragem é dada com base em um vetor de distribuição de probabilidades, definido para cada amostra do conjunto de entrada. O vetor de probabilidades é atualizado conforme a dificuldade de classificação de cada amostra isoladamente, no qual um peso maior é dado para aquelas amostras que foram classificadas incorretamente. Por essa razão, os componentes do comitê são gerados de forma sequencial, após a atualização do vetor de probabilidades.

Esse método pode ser usado em conjunto com qualquer classificador de base, desde que o erro de cada componente seja inferior a 50% e superior a 0, nos problemas de

classificação binária. O modelo pode ser utilizado também para solução de problemas multiclasses.

A primeira etapa do algoritmo é a definição do vetor de probabilidades de ocorrência das amostras, D, com base no conjunto de dados entrada Z, atualizado a cada iteração do algoritmo. Nessa primeira iteração, todas as amostras tem o mesmo peso, i.e, a mesma chance ocorrência, determinada da seguinte forma

$$D_i^1 = \frac{1}{m}, \tag{2.20}$$

para todo $i \in \{1, 2, ..., m\}$, no qual m é o número de amostras de Z.

Segue então que, para o treinamento de cada classificador de base h^k, um subconjunto dos dados de entrada é selecionado (Z^k), formando o conjunto de treinamento do componente individual. Esse subconjunto de dados é composto por m amostras selecionadas de Z com base no vetor de probabilidades, de forma que a inclusão de amostras repetidas é permitida. Note que, podem ocorrer inclusões de amostras repetidas em Z^k. A hipótese h_k é treinada sobre o subconjunto de treinamento Z^k e testada em relação ao conjunto de treinamento original Z. O erro ϵ^k da hipótese h^k é dado por

$$\epsilon^k = Pr_{i \sim D^k} \left(h^k(x_i) \neq y_i \right). \tag{2.21}$$

Com base nas taxas de erro de teste em Z, importância β^k do componente h^k é calculada

$$\beta^k = \frac{1}{2} \ln \left(\frac{1 - \epsilon^k}{\epsilon^k} \right). \tag{2.22}$$

Posteriormente, o vetor de probabilidades D^k é atualizado da seguinte forma

$$D_i^{k+1} \leftarrow \frac{D_i^k exp\left(-\beta^k y_i h^k(x_i)\right)}{\psi^k}, \tag{2.23}$$

para todo $i \in \{1, 2, ..., m\}$, com $(x_i, y_i) \in Z^k$, no qual ψ^k é um fator de normalização definido para que D^{k+1} seja uma distribuição.

O processo iterativo ocorre até que o número total T de componentes definido, ou total de iterações, seja alcançado. A hipótese final, ou hipótese forte h^f, é obtida com base na função sinal da votação ponderada de todos os componentes do *ensemble* pela importância dada a cada classificador de base diante de uma nova amostra x, definida por

$$h^f(x) = sinal\left(\sum_{k=1}^{T} \beta^k h^k(x)\right). \tag{2.24}$$

O Algoritmo 2.4 descreve o pseudocódigo do AdaBoost.

Algorithm 2.4: Adaptive Boosting

Entrada: conjunto de treinamento: $Z = \{(x_i, y_i)\}$ de cardinalidade m;
classificador de base: CB;
número de iterações: T;

Saída: hipótese forte h^f;

início
  para i ***de*** 1 ***até*** m **faça**
    $D_i^1 = 1/m$;
  fim para
  para k ***de*** 1 ***até*** T **faça**
    obter Z^k a partir de Z e D^k;
    $h^k \leftarrow CB(Z^k)$;
    calcular ϵ^k;
    $aceite \leftarrow$ verdadeiro;
    se $\epsilon^k > 1/2$ ***ou*** $\epsilon^k = 0$ **então**
      $k \leftarrow k - 1$;
      $aceite \leftarrow$ falso;
    fim se
    se $aceite$ **então**
      $\beta^k = 1/2 \ln\left(\left(1 - \epsilon^k\right)/\epsilon^k\right)$;
      para i ***de*** 1 ***até*** m **faça**
        $D_i^{k+1} \leftarrow D_i^k exp\left(-\beta^k y_i h^k(x_i)\right)/\psi^k$;
      fim para
      $h^f \leftarrow h^f + \beta^k h^k$;
    fim se
  fim para
fim

3 ENSEMBLE DE PERCEPTRONS BALANCEADOS – EBP

Esse capítulo é destinado à descrição de um dos métodos propostos nesse trabalho, o Ensemble de Perceptrons Balanceados (*Ensemble of Balanced Perceptrons* – EBP). O modelo Perceptron Balanceado (PB) é apresentado. O *ensemble* proposto é uma combinação de classificadores do tipo PB, selecionados conforme as medidas de diversidade adotadas. As seguintes seções discutem a estratégia empregada para melhoria da acurácia do classificador de base, bem como as estratégias usadas para introduzir diversidade na formação do *ensemble*. O método é também derivado em relação às variáveis duais. A extensão, para a formulação dual, é chamada *Ensemble* de Perceptrons *Kernel* Balanceados (*Ensemble of Balanced Perceptrons Kernel* – EBPK) e os ajustes necessários para a construção do modelo derivado em variáveis duais são destacadas.

3.1 FORMULAÇÃO PRIMAL

3.1.1 PERCEPTRON BALANCEADO – PB

O modelo original do algoritmo Perceptron, devido à sua formulação, tende a manter uma distância pequena entre o hiperplano separador e as últimas amostras corrigidas. Esse fato pode culminar em um hiperplano solução desbalanceado, no qual o hiperplano solução pode se manter a uma distância muito grande em relação a uma classe e muito pequena em relação a outra. Em alguns casos, o valor de margem pode ser significativamente melhorado através de um reposicionamento do hiperplano, mantendo a mesma direção da solução. Esse reposicionamento pode ser efetuado através de um deslocamento no valor do *bias*, movimentando o hiperplano para uma posição equidistante das duas classes, no caso da classificação binária. Essa nova posição é a solução de máxima margem da hipótese obtida originalmente. Em geral, essa solução de máxima margem implica em uma capacidade de generalização mais elevada para o modelo. Uma vez que, ao modificar o modelo Perceptron original e permitir o deslocamento do hiperplano solução, ocorre um balanceamento desse hiperplano. Ao algoritmo resultante dessa modificação, deu-se o nome de Perceptron Balanceado (PB).

Para determinar o deslocamento a ser realizado, seja $Z^+ = \{(x_i, y_i) \in Z : y_i = +1\}$, $Z^- = \{(x_i, y_i) \in Z : y_i = -1\}$. Considerando a distância entre o hiperplano separador e as duas classes do problema, as semi-margens são definidas da seguinte forma

$$\begin{aligned} \gamma^+ &= \min \{y_i(\langle w, x_i \rangle + b),\ \forall\ x_i \in Z^+\} \\ \gamma^- &= \min \{y_i(\langle w, x_i \rangle + b),\ \forall\ x_i \in Z^-\}\,. \end{aligned} \tag{3.1}$$

O deslocamento é determinado com base na média dos valores das margens. Assim, define-se o coeficiente de deslocamento como $\Gamma = (\gamma^+ + \gamma^-)/2$. Dado o valor de Γ, a atualização do valor do *bias* é dado por

$$b \leftarrow b - \gamma^- + \Gamma. \tag{3.2}$$

A Figura 3.1 ilustra esse procedimento. Note que, uma vez que o vetor de pesos não é alterado, a solução final do Perceptron Balanceado é equivalente a solução do Perceptron, capaz de classificar os dados de treinamento da mesma forma. Entretanto, ao introduzir o deslocamento do *bias*, o novo hiperplano obtido mantém a margem mais larga possível, considerando a mesma solução.

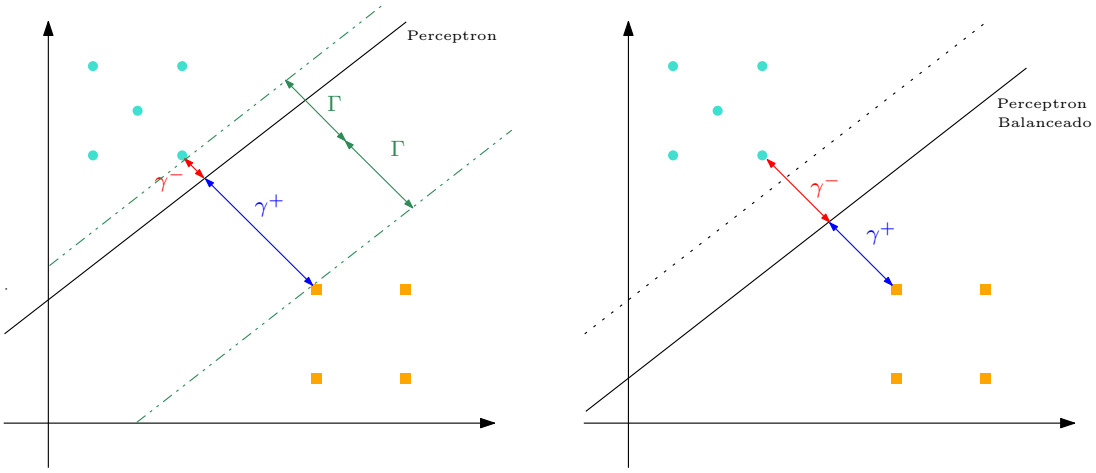


Figure 3.1: Estratégia para o balanceamento da solução Perceptron.

3.1.2 INTRODUZINDO DIVERSIDADE NO *ENSEMBLE*

Uma das premissas básicas para a construção de um *ensemble* é a diversidade dos componentes do modelo. Como mencionado anteriormente, não existem vantagens ou ganho

de acurácia em um modelo que combina classificadores idênticos. Por outro lado, a combinação de componentes diversos é vantajosa, uma vez que existe a possibilidade de recuperar dados classificados incorretamente por um determinado classificador. Entretanto, não existe um consenso sobre qual forma de introduzir medidas de dissimilaridade é a mais adequada para a construção de *ensembles*. Por essa razão, o método proposto combina 3 técnicas de geração de diversidade. Cada componente do *ensemble* é gerado a partir de uma nova permutação aleatória dos dados de entrada, e o vetor de pesos iniciais é preenchido com valores aleatórios. Essas estratégias são comuns na geração de *ensembles* de Perceptrons (KUNCHEVA, 2004). Além disso, é combinada aqui uma medida de dissimilaridade capaz de aumentar significativamente a distância de cada componente do *ensemble* em relação aos demais, como estratégia para aumentar a diversidade do modelo. As técnicas empregas são descritas adiante.

3.1.2.1 Pesos iniciais aleatórios

O modo mais comum de gerar *ensembles* de Perceptrons é a partir da injeção de fatores aleatórios no algoritmo de aprendizado. Nos casos em que o classificador de base é um Perceptron, o vetor inicial de pesos do modelo pode ser definido com valores aleatórios. No método proposto, o vetor de pesos é inicializado com valores aleatórios gerados no intervalo entre -1 e 1.

Modificar o vetor de pesos iniciais leva a uma inicialização diferente do modelo e, assim, uma sequência diferente de atualizações. Essa sequência distinta de atualizações pode propiciar a convergência para um conjunto de pesos distintos, implicando em um ótimo local diferente, dependente da condição inicial. Assim, uma solução diferente pode ser determinada. Com isso, espera-se que os modelos gerados resultem em capacidades de generalização distintas. Essa estratégia foi discutida por Dietterich (2000a) e Opitz e Maclin (1999).

3.1.2.2 Permutação aleatória dos dados de entrada

Uma outra forma de gerar diversidade no modelo consiste na manipulação dos dados de treinamento para a geração de múltiplas hipóteses. Para cada componente do *ensemble*, o algoritmo de aprendizado é executado com uma permutação diferente do conjunto de treinamento. Essa estratégia eleva a diversidade do *ensemble*, em relação ao caso anterior,

já que alterar a ordem de correção das amostras pode culminar em uma solução final diferente. Além disso, todos os componentes do *ensemble* são especialistas no mesmo conjunto de dados, embora distintos em relação a visualização dos mesmos, portanto diversos em relação as soluções geradas. Essa estratégia foi abordada por Parmanto et al. (1996).

3.1.2.3 Medida de Dissimilaridade

Para algoritmos estáveis, como o Perceptron, simplesmente aleatorizar a ordem das amostras de entrada e inicializar o vetor de pesos com valores aleatório não é suficiente para produzir uma diversidade nos componentes de forma adequada, permitindo a geração de componentes similares. Nesse caso, ainda assim, o *ensemble* gerado não é muito diverso.

Com o intuito de gerar um conjunto de hipóteses o mais diverso possível, uma outra estratégia é adotada. Em adição às estratégias descritas anteriormente, define-se um valor, $\varepsilon > 0$, que representa a distância mínima que todos os componentes pertencentes ao comitê devem manter entre si. Em outras palavras, dado o conjunto de hipóteses $\{h_1, h_2, \cdots, h_j\}$ já aceitas no comitê, uma nova hipótese h_k é tomada como um componente válido se, e somente se

$$\min_{i \in \{1,\cdots,j\}} \{\|h_i, h_k\|_2\} > \varepsilon. \tag{3.3}$$

Essa estratégia é chamada de *medida de dissimilaridade*.

No EBP, essa distância é calculada com base na distância Euclidiana entre as hipóteses aceitas no comitê. Vale ressaltar ainda que, comparar distâncias só faz sentido caso o vetor de pesos estendido, i.e., o vetor de pesos w acrescido do valor do *bias*, esteja normalizado. Caso contrário, seria impossível definir valores de ε adequados, considerando, por exemplo, problemas de alta dimensão e o intervalo de valores possíveis para o preenchimento do vetor de pesos. O cálculo da norma, adotada aqui, é definido pela Eq. (3.4) e a normalização do vetor de pesos estendido é dada pela Eq. (3.5)

$$\|w\|_2 = \left(\sum_{i=1}^{d} w_i^2\right)^{1/2}, \tag{3.4}$$

$$(w, b) = (w, b) / \|w\|_2 . \tag{3.5}$$

3.1.3 PSEUDOCÓDIGO

O Algoritmo 3.1 descreve o pseudocódigo do algoritmo primal relativo ao treinamento do EBP.

Algorithm 3.1: Ensemble de Perceptrons Balanceados

Entrada: conjunto de treinamento: $Z = \{(x_i, y_i)\}$ de cardinalidade m;
taxa de aprendizado: η;
tamanho do comitê: tam;
valor de dissimilaridade: ε;
limite superior no número de iterações: max;
Saída: componentes do comitê;
início
  para k ***de*** 1 ***até*** tam **faça**
    permutar Z;
    $(w_k, b_k) \leftarrow \text{PerceptronPrimal}(Z, \eta, max)$;
    $\gamma^+ = \min\{y_i(\langle w_k, x_i\rangle + b_k),\ \forall\ x_i \in Z^+\}$;
    $\gamma^- = \min\{y_i(\langle w_k, x_i\rangle + b_k),\ \forall\ x_i \in Z^-\}$;
    $\Gamma = (\gamma^+ + \gamma^-)/2$;
    $b_k = b_k - \gamma^- + \Gamma$;
    normalizar (w_k, b_k);
    $aceite \leftarrow$ verdadeiro;
    para j ***de*** 1 ***até*** $k-1$ **faça**
      se $\|(w_k, b_k), (w_j, b_j)\|_2 < \varepsilon$ **então**
        $k \leftarrow k - 1$;
        $aceite \leftarrow$ falso;
      fim se
    fim para
    se $aceite$ **então**
      $comite_k \leftarrow (w_k, b_k)$;
    fim se
  fim para
fim

3.2 FORMULAÇÃO DUAL

Essa seção apresenta o modelo *Ensemble* de Perceptrons *Kernel* Balanceados (Ensemble of Balanced Perceptrons Kernel – EBPK), uma extensão da formulação do EBP, que permite a utilização de funções *kernel* e a solução de problemas não-linearmente separáveis. O EBPK emprega uma estratégia análoga para o balanceamento da solução do Perceptron *Kernel* (PK) e estratégias para geração de componentes diversos são aplicadas. O mod-

elo Perceptron *Kernel* Balanceado (PKB) apresentado mais adiante nesse capítulo é o classificador de base adotado para construção do *ensemble* em questão. A seleção dos componentes do *ensemble* é feita a partir das duas medidas de diversidade.

3.2.1 PERCEPTRON *KERNEL* BALANCEADO – PKB

Assim como no PB, o Perceptron *Kernel* Balanceado (PKB) é uma modificação do PK que desloca o hiperplano solução em direção à solução de máxima margem daquela hipótese.

Para determinar o deslocamento no PKB, novamente, seja $Z^+ = \{(x_i, y_i) \in Z : y_i = +1\}$, $Z^- = \{(x_i, y_i) \in Z : y_i = -1\}$ Considere a distância entre o hiperplano separador da solução do PK e as duas classes do problema. Analogamente, as semi-margens relativas ao PKB são definidas da seguinte forma

$$\begin{aligned} \gamma^+ &= \min \left\{ y_i \left(\sum_{j=1}^{m} \alpha_j y_j K(x_j, x_i) + b \right), \forall x_i \in Z^+ \right\} \\ \gamma^- &= \min \left\{ y_i \left(\sum_{j=1}^{m} \alpha_j y_j K(x_j, x_i) + b \right), \forall x_i \in Z^- \right\}, \end{aligned} \tag{3.6}$$

na qual α é o vetor de multiplicadores e K é a função *kernel* definida na Seção 2.2.2

O deslocamento é determinado de maneira análoga ao PB. Defina $\Gamma = (\gamma^+ + \gamma^-)/2$. Dado o valor de Γ, o novo valor do *bias* é dado pela Eq. (3.2).

3.2.2 INTRODUZINDO DIVERSIDADE NO *ENSEMBLE*

Diferentemente do EBP, apenas duas medidas de diversidade são empregadas no EBPK. No caso do *ensemble* em variáveis duais, não é possível introduzir diversidade a partir da injeção de aleatoriedade no vetor de multiplicadores, uma vez que o Perceptron não é capaz de convergir nesses casos. Por outro lado, a permutação dos dados de entrada é mantida de maneira análoga a apresentada na Seção 3.1.2.3 para cada componente do *ensemble*. Esse tratamento nos dados de entrada, para algoritmos em variáveis duais foi utilizada no contexto de classificadores *ensemble* no trabalho de Herbrich et al. (2001).

Além disso, a medida de dissimilaridade em relação à distância Euclidiana não é efetiva no espaço de características, uma vez que a distância entre os componentes do *ensemble* não é preservada ao retornar ao espaço de entrada. Assim, não é possível precisar sobre a real diversidade das soluções. Isso acontece, uma vez que, a dissimilaridade no espaço de características deve ser calculada com base nos vetores de multiplicadores α, ao passo que,

quando a dissimilaridade é calculada no espaço de entrada deve ser definida para o vetor de pesos w. Dessa forma, uma alteração na medida de dissimilaridade deve ser considerada, visando obter uma medida de distância que seja efetiva para soluções definidas no espaço de características.

A medida de dissimilaridade é empregada de maneira análoga ao EBP, definida pela Eq. (3.3), usando a distância de Tanimoto, ou distância de Jaccard (LIPKUS, 1999; UMBAUGH, 2005). Trata-se de uma medida de distância específica para aplicações no espaço de características. Os vetores normais ao espaço de características devem ser normalizados. Sejam dois vetores $\alpha_i,\ \alpha_k \in F$, definidos na Seção 2.2.2, referentes a duas hipóteses $h_i,\ h_k$. Define-se a distância, ou a dissimilaridade, entre os mesmos da seguinte forma

$$dist(h_i, h_k) = dist(\alpha_i, \alpha_k) = 1 - \frac{\alpha_i \alpha_k}{\alpha_i^2 + \alpha_k^2 - \alpha_i \alpha_k}. \tag{3.7}$$

Note que, essa medida gera um coeficiente de dissimilaridade entre 0 e 1, no qual $dist(h_i, h_k) = 0$ representa vetores idênticos, cuja distância entre os mesmos é nula. Por outro lado, $dist(h_i, h_k) = 1$ representa vetores completamente distintos.

3.2.3 ALGORITMO DUAL

O Algoritmo 3.2 descreve o pseudocódigo do algoritmo dual relativo ao treinamento do EBPK.

Algorithm 3.2: Ensemble de Perceptrons *Kernel* Balanceados

Entrada: conjunto de treinamento: $Z = \{(x_i, y_i)\}$ de cardinalidade m;
taxa de aprendizado: η;
tamanho do comitê: tam;
valor de dissimilaridade: ε;
limite superior no número de iterações: max;
Saída: componentes do comitê;
início
  para k ***de*** 1 ***até*** tam **faça**
    permutar Z;
    $(\alpha_k, b_k) \leftarrow \text{PerceptronDual}(Z, \eta, max)$;
    $\gamma^+ = \min\left\{y_i\left(\sum_{j=1}^{m} \alpha_j y_j K(x_j, x_i) + b\right), \forall x_i \in Z^+\right\}$;
    $\gamma^- = \min\left\{y_i\left(\sum_{j=1}^{m} \alpha_j y_j K(x_j, x_i) + b\right), \forall x_i \in Z^-\right\}$;
    $\Gamma = (\gamma^+ + \gamma^-)/2$;
    $b_k = b_k - \gamma^- + \Gamma$;
    normalizar (α_k, b_k);
    $aceite \leftarrow$ verdadeiro;
    para j ***de*** 1 ***até*** $k-1$ **faça**
      se $1 - \left(\frac{\alpha_j \alpha_k}{\alpha_j^2 + \alpha_k^2 - \alpha_j \alpha_k}\right) < \varepsilon$ **então**
        $k \leftarrow k - 1$;
        $aceite \leftarrow$ falso;
      fim se
    fim para
    se $aceite$ **então**
      $comiteDual_k \leftarrow (\alpha_k, b_k)$;
    fim se
  fim para
fim

4 MÁQUINA DE REDUÇÃO DO ESPAÇO DE VERSÕES – VSRM

Este capítulo tem o objetivo de descrever a Máquina de Redução do Espaço de Versões (*Version Space Reduction Machine* – VSRM). Esse método é desenvolvido com intuito de apresentar um algoritmo que seja capaz de fornecer uma única hipótese concordante com um EBP combinado pelo voto majoritário dos seus componentes. Frequentemente, esquemas de votação, como estratégia para combinação de saídas dos componentes, apresentam resultados bastante satisfatórios em termos de capacidade de generalização. Entretanto, ao avaliar um novo conjunto de amostras para a fase de testes, a votação de um *ensemble* leva em consideração a solução de cada uma das hipóteses para cada novo dado, ao passo que outras estratégias são capazes de combinar as hipóteses geradas em uma única hipótese equivalente. Dessa forma, a construção de um método capaz de gerar a hipótese equivalente a um *ensemble* combinado pelo voto majoritário é de grande valia.

Este algoritmo pode ser construído a partir da geração de sucessivos planos de cortes no espaço de versões. A adição de um plano de corte divide o espaço de versões em dois semi-espaços. Um *ensemble*-guia é gerado para definição da concordância. Esse *ensemble*-guia é obtido a partir de um EBPd combinado pelo voto majoritário de seus componentes. O semi-espaço concordante com o comitê é preservado, ao passo que o semi-espaço restante é descartado. Dessa forma, após um número finito de iterações e a constante redução do espaço de versões, o algoritmo converge para a hipótese aproximada equivalente e concordante com um EBP combinado pelo voto majoritário.

O algoritmo foi desenvolvido a partir de uma modifição introduzida no GFMP, chamada de Perceptron de Margem Geométrica Variável (*Geometric Variable Margin Perceptron – GVMP*). O GVMP considera a utilização de duas margens para cada ponto. A redução do espaço de versões é feita com base nos valores de margem obtidos e nas restrições do problema avaliado.

As seções a seguir definem os conceitos fundamentais em relação ao VSRM. Inicialmente, define-se formalmente o conceito de espaço de versões. Posteriormente, é discutido o GVMP e a modificação introduzida ao GFMP, para o emprego da técnica de redução do espaço. Discute-se ainda a questão da diversidade na geração dos componentes do *en-*

semble. Finalmente, é apresentada a formulação matemática e o pseudocódigo do método proposto em variáveis primais.

4.1 ESPAÇO DE VERSÕES

Segundo Herbrich (2001), dado o conjunto de amostras de treinamento Z e o conjunto de hipóteses soluções H de um problema qualquer, define-se $V(Z)$ em função de H tal que

$$V_H(Z) \stackrel{def}{=} \{h \in H : i \in \{1, 2, ..., m\} : h(x_i) = y_i\} \subseteq H, \tag{4.1}$$

como o *espaço de versões*, i.e, o conjunto de todos os classificadores consistentes com o conjunto de treinamento. Em particular, para classificadores lineares, como o Perceptron, o espaço de versões também pode ser definido como o conjunto de vetores de pesos consistentes, dado por

$$V_W(Z) \stackrel{def}{=} \{(w, b) \in W : i \in \{1, 2, ..., m\} : y_i(\langle w, x_i \rangle + b) > 0\} \subseteq W, \tag{4.2}$$

na qual $W = \{(w, b) : \|(w, b)\|_2 = 1\}$ é a esfera unitária isomorfa ao espaço de hipóteses do problema em questão. Para facilitar o emprego da notação, o vetor de pesos (w, b), equivalente a um ponto no espaço de versões, será representado simplesmente por w, partir de agora.

4.2 PERCEPTRON DE MARGEM GEOMÉTRICA VARIÁVEL – GVMP

O processo de redução do espaço de versões é baseado em uma modificação do GFMP, na qual associa-se, para cada restrição do problema, dois tipos de margens, uma inferior γ^{inf} e outra superior γ^{sup}. A inclusão de dois valores de margem distintos define o Perceptron de Margem Geométrica Variável (*Geomtric Variable Margin Perceptron – GVMP*).

Para a construção do GVMP, a regra de correção do GFMP deve ser modificada para permitir que cada ponto possa ser corrigido em relação a cada uma das margens. Consequentemente, o critério de viabilidade do ponto também deve ser modificado. Um ponto w é viável no espaço de versões se obedece as seguintes inequações

$$\begin{aligned} y_i(\langle w, x_i \rangle + b) &\geq \gamma_i^{inf} \\ y_i(\langle w, x_i \rangle + b) &\leq \gamma_i^{sup} \end{aligned} \tag{4.3}$$

Caso o critério de viabilidade em relação ao γ_i^{inf} não seja satisfeito para um ponto w, ou seja, caso $y_i(\langle w, x_i\rangle + b) < \gamma_i^{inf}$, a regra de correção a ser aplicada é dada por

$$w^{t+1} \leftarrow w^t \left(1 - \left(\eta\gamma^{inf}\right) \frac{1}{\|w^t\|_2}\right) + \eta y_i x_i. \tag{4.4}$$

Por outro lado, se o critério de viabilidade não é satisfeito para um ponto w em relação ao γ_i^{sup}, ou seja, caso $y_i(\langle w, x_i\rangle + b) > \gamma_i^{sup}$ ou $-y_i(\langle w, x_i\rangle + b) < -\gamma_i^{sup}$, a regra de correção a ser aplicada é a seguinte

$$w^{t+1} \leftarrow w^t \left(1 + (\eta\gamma_i^{sup}) \frac{1}{\|w^t\|_2}\right) + \eta y_i x_i. \tag{4.5}$$

Note que, embora haja a inclusão de uma nova restrição e o um novo valor de margem, o custo do método se mantém o mesmo do GFMP.

A Figura 4.1 ilustra os dois casos descritos. Considere o espaço de versões $V(Z)$ em relação a W, e a introdução de dois novos pontos viáveis w' e w'' que apresentam dois valores distintos de margem, uma inferior e uma superior, caracterizando dois casos do GVMP em relação à restrição R_1.

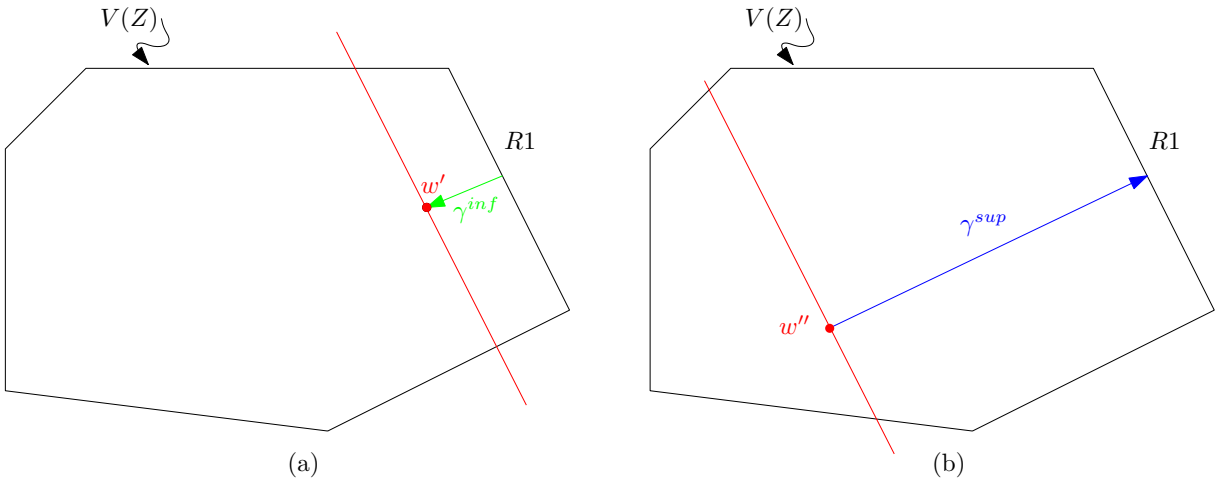


Figure 4.1: Introdução de dois pontos viáveis, w' e w'', dados pela convergência do GVMP

O Algoritmo 4.1 descreve o pseudocódigo do GVMP em termos das variáveis primais.

Algorithm 4.1: Perceptron de Margem Geométrica Variável Primal

Entrada: conjunto de treinamento: $Z = \{(x_i, y_i)\}$ de cardinalidade m;
vetor de margem geométrica inferior: γ^{inf};
vetor de margem geométrica superior: γ^{sup};
taxa de aprendizado: η;
limite superior no número de iterações: max;
Saída: vetor de pesos w e *bias* b;
início
  inicializar (w^0, b^0);
  $j \leftarrow 0$;
  $t \leftarrow 0$;
  $stop \leftarrow$ falso;
  enquanto $j \leq max$ ***e*** $\neg stop$ **faça**
    $erro \leftarrow$ falso;
    para i ***de*** 1 ***até*** m **faça**
      se $y_i \left(\langle w^t, x_i \rangle + b^t\right) < \gamma^{inf}$ **então**
        $w^{t+1} \leftarrow w^t \left(1 - \left(\eta\gamma^{inf}\right) 1/ \left\|w\right\|_2\right) + \eta y_i x_i$;
        $b^{t+1} \leftarrow b^t + \eta y_i$;
        $t \leftarrow t + 1$;
        $erro \leftarrow$ verdadeiro;
      fim se
      se $y_i \left(\langle w^t, x_i \rangle + b^t\right) > \gamma^{sup}$ **então**
        $w^{t+1} \leftarrow w^t \left(1 + \left(\eta\gamma^{sup}\right) 1/ \left\|w\right\|_2\right) + \eta y_i x_i$;
        $b^{t+1} \leftarrow b^t + \eta y_i$;
        $t \leftarrow t + 1$;
        $erro \leftarrow$ verdadeiro;
      fim se
    fim para
    se $\neg erro$ **então**
      $stop \leftarrow$ verdadeiro;
    fim se
    $j \leftarrow j + 1$;
  fim enquanto
fim

4.3 A DIVERSIDADE NO VSRM

O processo de redução do espaço de versões leva em consideração, para a escolha do semi-espaço a ser preservado, a concordância da hipótese avaliada em relação ao *ensemble*-guia gerado. A concordância da hipótese avaliada reflete o maior semi-espaço resultante da divisão definida pelo GVMP, ou seja, o semi-espaço que contém o maior número de componentes do *ensemble*-guia.

Um *ensemble* bem distribuído é aquele cujos componentes são os mais diversos pos-

síveis (HERBRICH, 2001). Assim, para que a aproximação da hipótese equivalente a um EBP combinado pelo voto majoritária seja efetiva, é imprescindível que o *ensemble*-guia seja bem distribuído no espaço de versões. Do contrário, a decisão de descartar um dos semi-espaços pode ser determinada de maneira incorreta.

As Figuras 4.2 e 4.3 ilustram os dois casos em discussão. No primeiro caso, a Figura 4.2 apresenta um *ensemble*-guia mal distribuído, gerado sem levar em consideração a diversidade dos componentes. Observa-se que a hipótese concordante com o *ensemble* gerado determina que o menor semi-espaço deve ser preservado, o que contraria a teoria de decisão do método. Por outro lado, o caso da Figura 4.3 apresenta um comitê bem distribuído no espaço de versões. Dessa forma, a hipótese concordante com o *ensemble*-guia determina o corte que preserva o maior semi-espaço dentre os dois avaliados. Note que, o EBPd é capaz de gerar componentes bastante diversificados e, por essa razão, o método gera um conjunto de hipóteses bem distribuídas no espaço de versões.

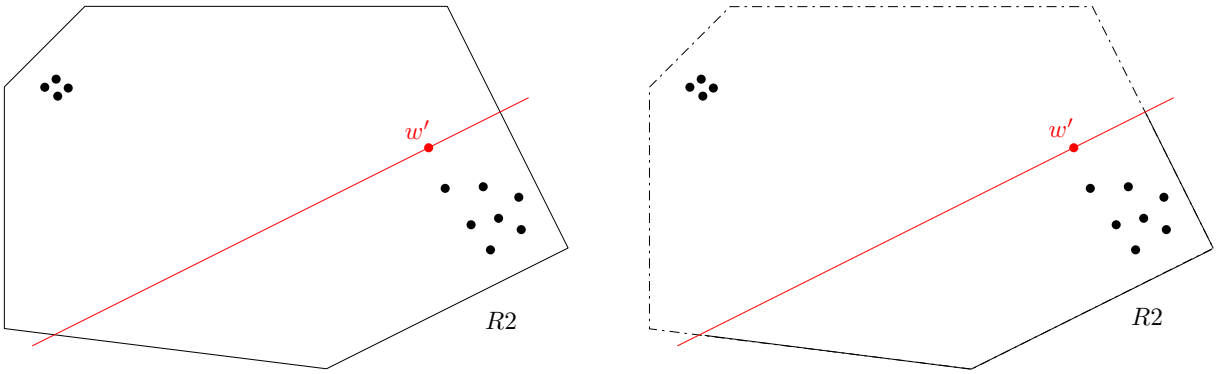


Figure 4.2: *Ensemble* mal distribuído.

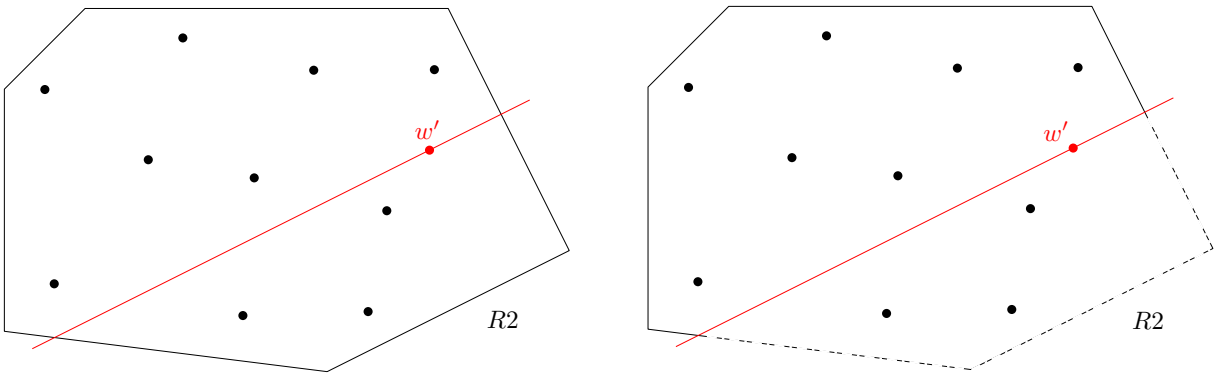


Figure 4.3: *Ensemble* diverso com componentes bem distribuídos.

4.4 FORMULAÇÃO PRIMAL

O primeiro passo do VSRM consiste na geração do *ensemble*-guia a partir do EBPd, cujos componentes são viáveis em relação às restrições do problema. A Figura 4.4 apresenta as iterações até a convergência do VSRM caracterizando o passo a passo do método, para um problema hipotético contendo 6 restrições no espaço de versões.

Ao aplicar a medida de dissimilaridade na geração do comitê, os componentes obtidos tendem a ser bem distribuídos no espaço de versões, como visto na Figura 4.4-(a). Assim, $comite = \{h_1, h_2, ..., h_{11}\}$ representa o conjunto de componentes do *ensemble*-guia.

O procedimento iterativo do VSRM deve ser repetido enquanto existir um ponto w viável no espaço de versões $V(Z)$. Um novo ponto w é viável em $V(Z)$ se e somente se

$$w \in V(Z) \leftrightarrow y_i \left(\langle w, x_i \rangle + b \right) \geq 0, \tag{4.6}$$

para todo $i \in \{1, 2, ..., m\}$, com $(x_i, y_i) \in Z$.

Iniciando as iterações do algoritmo, a partir do GVMP, um novo ponto w^1 é gerado e um hiperplano separador é definido paralelamente à restrição R_1 do problema, representado pela Figura 4.4-(b). Essa direção d_1 é dada pela seguinte equação

$$d_1 = y_1 x_1. \tag{4.7}$$

Ao calcular o hiperplano separador, o valor da margem do hiperplano em relação a restrição R_1 é dado por

$$\gamma^{new} = \left(\left\langle w^1, x_1 \right\rangle + b^1 \right) / ||w^1||_2, \tag{4.8}$$

na qual $||w^1||_2$ é a norma de w^1. O hiperplano definido por w^1 divide o $V(Z)$ em dois semi-espaços. O semi-espaço a ser preservado é aquele que concorda com o voto majoritário dos componentes do *ensemble*-guia. A votação é determinada com base na avaliação de cada componente do conjunto $comite = \{h_1, h_2, ..., h_{11}\}$, com $h_k = (w_k, b_k)$, para todo $k \in \{1, 2, ..., 11\}$. O processo de votação para cada amostra do conjunto de treinamento Z é dado por

$$\begin{aligned} &\text{se } y_1(\langle (w_k - w^1), x_1 \rangle + (b_k - b^1)) \geq 0 \Rightarrow \quad inf \leftarrow inf + 1, \\ &\text{se } y_1(\langle (w_k - w^1), x_1 \rangle + (b_k - b^1)) < 0 \Rightarrow \quad sup \leftarrow sup + 1, \end{aligned} \tag{4.9}$$

na qual inf e sup são as duas possibilidades de voto de cada componente do $comite = \{h_1, h_2, ..., h_{11}\}$. As margens γ^{inf} e γ^{sup} estão associadas aos votos inf e sup, respectivamente. Dessa forma, a atualização das margens para o descarte de um semi-espaço depende do resultado da votação. Para a iteração representada na Figura 4.4-(b), a votação determinou 9 votos para alteração da margem γ_{inf} contra 2 votos para alteração da margem γ_{sup}. Dessa forma, o semi-espaço a ser descartado é o semi-espaço com menos componentes, dado pela margem γ^{inf}, como discutido na Figura 4.1. O espaço é então reduzido e a restrição R_1 é atualizada para o novo valor de margem da seguinte forma

$$\gamma_1^{inf} \leftarrow \gamma^{new} = \left(\left\langle w^1, x_1 \right\rangle + b^1\right) / ||w^1||_2 \tag{4.10}$$

A linha tracejada representa o semi-espaço descartado.

A próxima iteração tem início na correção de w^1 que deixa de ser um ponto viável para o $V(Z)$ reduzido. A regra de correção utilizada é a do GVMP. O ponto w^2 é obtido a partir dessa correção. Novamente, um hiperplano separador é traçado na direção da restrição R_2 e o semi-espaço concordante com a votação entre inf e sup é preservado, ao passo que o outro semi-espaço é descartado. Para a iteração representada pela Figura 4.4-(c), a votação dos componentes do *ensemble*-guia determinou 1 voto para alteração da margem γ^{inf} contra 10 votos para alteração da margem γ^{sup}. Assim, descarta-se o semi-espaço com menos componentes votantes, dado pela margem γ^{sup}. Ao final da iteração, o espaço é reduzido novamente e a restrição R_2 é atualizada para o valor de margem definido pelo hiperplano separador gerado para o w^2. O processo continua, como ilustrado pela Figura 4.4.

Note que o $V(Z)$ reduzido é considerado apenas para a atualização do ponto w e consequente geração de uma nova solução viável. Entretanto, para a definição do semi-espaço concordante com o *ensemble*-guia, o espaço de versões original é considerado e, assim, todos os componentes do comitê são utilizados para a votação. No caso do problema hipotético representado pela Figura 4.4, em todas as iterações, todas as 11 componentes geradas a partir do *ensemble*-guia são avaliadas para definição dos semi-espaços a serem preservados ou descartados. Note ainda que a redução do espaço de versões $V(Z)$ é realizada sem a introdução de novas restrições para o problema.

O critério de parada do algoritmo é definido quando não há mais a possibilidade de geração de novos pontos viáveis no espaço de versões reduzido. A geração de um novo

ponto viável depende da convergência do GVMP. Caso o GVMP não atinja a convergência no número máximo de iterações estabelecidas, o critério de parada é atingido. Note que, ao final, todas as restrições serão atualizadas em relação às duas margens γ^{inf} e γ^{sup}, atendendo a seguinte condição

$$\gamma_i^{inf} \leq (\langle w, x_i \rangle + b)/||w||_2 \leq \gamma_i^{sup}. \tag{4.11}$$

O último ponto viável encontrado define a aproximação da hipótese equivalente a um EBP combinado pelo voto majoritário. De posse dessa hipótese, durante a fase de teste, essa é a única hipótese a ser verificada perante a introdução de uma nova amostra para classificação.

A estratégia para construção do VSRM foi definida para permitir a obtenção da equação do hiperplano equivalente à votação majoritária de um EBP de forma linear em relação à dimensão do problema. Essa formulação possibilita a extensão do método para introdução de funções *kernel* e solução de problemas não-linearmente separáveis.

O Algoritmo 4.2 descreve o pseudocódigo do VSRM em termos das variáveis primais.

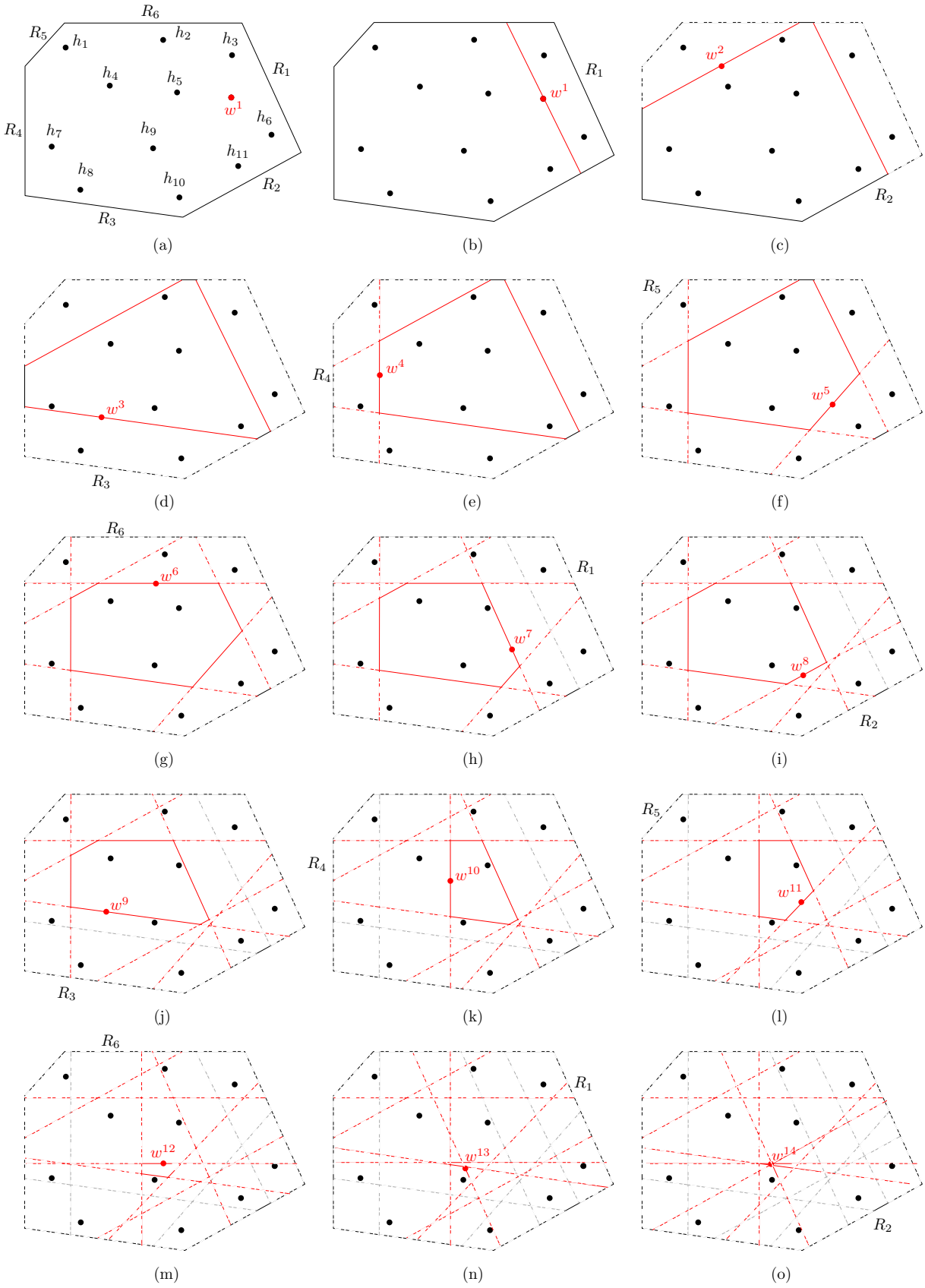


Figure 4.4: Ilustração da estratégia empregada pelo VSRM.

Algorithm 4.2: Máquina de Redução do Espaço de Versões

Entrada: conjunto de treinamento: $Z = \{(x_i, y_i)\}$ de cardinalidade m;
taxa de aprendizado: η;
limite superior no número de iterações: max;
Saída: vetor de pesos w e *bias* b;
início
  $comite \leftarrow \text{EBP}(Z, \eta, tam, \varepsilon, max)$;
  inicializar (w^0, b^0);
  para i ***de*** 1 ***até*** m **faça**
    $\gamma_i^{inf} \leftarrow 0$;
    $\gamma_i^{sup} \leftarrow \infty$;
  fim para
  $i \leftarrow 0$;
  repita
    $(w, b) \leftarrow \text{GVMP}(Z, \gamma^{inf}, \gamma^{sup}, max)$;
    normalizar (w, b);
    $inf \leftarrow 0$;
    $sup \leftarrow 0$;
    para k ***de*** 1 ***até*** tam **faça**
      se $y_i(\langle (w_k - w), x_i \rangle + (b_k - b)) \geq 0$ **então**
        $inf \leftarrow inf + 1$;
      senão
        $sup \leftarrow sup + 1$;
      fim se
    fim para
    se $inf > sup$ **então**
      $\gamma_i^{inf} \leftarrow (\langle w, x_i \rangle + b) / ||w||_2$;
    senão
      $\gamma_i^{sup} \leftarrow (\langle w, x_i \rangle + b) / ||w||_2$;
    fim se
    $i \leftarrow i + 1 \bmod m$;
  até *que a convergência do GVMP em max iterações não seja atingida*;
fim

5 ANÁLISE EXPERIMENTAL E RESULTADOS

Esse capítulo apresenta o estudo experimental realizado para avaliar os métodos propostos. Em relação ao método primal, o EBP, o estudo foi conduzido em 6 bases de dados linearmente separáveis de *microarray*. O método dual, o EBPK, foi avaliado em 8 bases de dados não-linearmente separáveis e, ainda, a base Sonar que, embora seja linearmente separável, dificilmente é resolvida por modelos lineares. Em seguida, considerando o VSRM, as mesmas bases usadas para o EBP foram empregadas.

A avaliação foi baseada na comparação dos métodos propostos (primal e dual) considerando duas estratégias de combinação de saídas: a média das hipóteses e o voto não ponderado. Com o intuito de avaliar a efetividade da medida de dissimilaridade, foram considerados duas variações de cada um dos métodos: a primeira, EBP e EBPK, na qual a medida de dissimilaridade não é empregada, e a segunda, EBPd e EBPKd, a qual emprega a medida de dissimilaridade.

As próximas seções descrevem as bases de dados utilizadas, as avaliações em relação a escolha do tamanho do comitê e o parâmetro da medida de dissimilaridade de cada método proposto em cada base avaliada. Também são descritos os parâmetros envolvidos no estudo experimental e como foram calibrados. Para observar a funcionalidade da medida de dissimilaridade, dois problemas simples foram construídos e os modelos EBP e EBPd avaliados. Finalmente, um estudo experimental foi conduzido para cada um dos três métodos propostos: EBP, EBPK e VSRM. Para os dois primeiros, a avaliação foi dividida em duas partes discutidas mais a diante nesse capítulo.

Em todos os conjuntos de experimentos, são apresentados resultados referentes ao modelo individual do Perceptron ou PK e ainda ao PB ou PKB. Ao fazê-lo, o objetivo é validar a premissa básica da construção de *ensembles*, que diz que um modelo *ensemble* deve obter uma desempenho superior ao classificador individual em questão. Além disso, comparou-se o Perceptron com a proposta de solução balanceada com intuito de mostrar que o balanceamento da solução gera melhorias na capacidade de generalização. As implementações dos algoritmos SVM e AdaBoost consideradas nesse trabalho são o Otimização Mínima Sequencial (*Sequential Minimal Optimization* – SMO) (PLATT, 1998) sem flexibilização de margem e o AdaBoost.M1 (FREUND; SCHAPIRE, 1996). Para o AdaBoost

utilizou-se um Perceptron como classificador de base para a comparação com o EBP e um Perceptron *kernel* como classificador de base para a comparação com o EBPK, respectivamente. Todos os métodos utilizados nesse trabalho foram reimplementados na linguagem C.

5.1 BASES DE DADOS

Para análise dos resultados, foram utilizadas 15 bases de dados. Sete bases de dados são linearmente separáveis. Dessas, seis são de *microarray* disponíveis nos trabalhos Glaab et al. (2012); Zhu et al. (2007); Golub et al. (1999). Essas bases foram empregadas para avaliação do EBP e do VSRM. A sétima base linearmente separável é a Sonar que, devido ao seu nível de dificuldade, raramente é resolvida por modelos lineares e, por isso, consta apenas como parte da avaliação do modelo dual, EBPK, bem como as outras 8 bases de dados utilizadas, por serem não-linearmente separáveis. Todas as bases utilizadas para avaliação do EBPK estão disponíveis em UCI Machine Learning Repository (BACHE; LICHMAN, 2013). As principais informações referentes as bases de dados utilizadas nesse trabalho encontram-se sumarizadas na Tabela 5.1.

Table 5.1: Informações sobre as bases de dados consideradas.

Base	Atributos	Amostras		
		Pos.	Neg.	Total
Breast	12625	10	14	24
Colon	2000	22	40	62
DLBCL	5468	58	19	77
Leukemia	7129	47	25	72
Prostate	12600	50	52	102
CNS	7129	21	39	60
Sonar	60	97	111	208
Ionosphere	34	225	126	351
Tic Tac	9	626	332	958
Bupa	6	145	200	345
Pima	8	268	500	768
Wine	13	107	71	178
BreastII	10	444	239	683
Heart	13	120	150	270
Live	6	145	200	345

Vale ressaltar ainda que não houve qualquer tipo de pré-processamento em qualquer uma das bases. Todas as bases, mesmo as de alta dimensão, foram utilizadas sem redução de dimensionalidade, exatamente como obtidas a partir das fontes anteriormente citadas. As bases selecionadas também não contém dados faltantes. Não houve ainda normalização

dos dados das bases. Os dados são compostos por valores reais e discretos.

5.2 AVALIAÇÃO DO BALANCEAMENTO E DA MEDIDA DE DISSIMILARIDADE

Com o objetivo de investigar a influência da medida de dissimilaridade e do balanceamento dos componentes do *ensemble*, realizou-se testes em dois problemas simples. Ambos os problemas são linearmente separáveis e bidimensionais com 8 amostras de treinamento para construção do comitê. Esses problemas foram escolhidos para facilitar a visualização dos resultados. Além disso, essa avaliação foi realizada somente para o modelo primal, já que o cálculo da medida de dissimilaridade, no modelo dual, deve ser feito no espaço de características, no qual a visualização é comprometida pelo aumento da dimensionalidade do problema.

A Figura 5.1 apresenta os resultados. Um *ensemble* de Perceptrons sem balanceamento e sem a medida de dissimilaridade foi gerado (Fig. 5.1-(a),(d)), para cada problema, para comparações relativas aos métodos EBP e EBPd. O objetivo foi investigar se a solução de balanceamento foi eficaz e se a medida de dissimilaridade produziu o efeito desejado na geração de componentes.

Em relação ao primeiro problema, Fig. 5.1-(a),(b),(c), os comitês gerados têm 5 componentes. Observa-se que o *ensemble* gerado sem balanceamento e sem a medida de dissimilaridade, Fig. 5.1-(a), apresenta componentes muito similares, i.e., pouco diversificados, mesmo com a permutação aleatória dos dados de entrada e a inicialização aleatória do vetor de pesos. Ao balancear as soluções, Fig. 5.1-(b), ocorre um deslocamento em todos os componentes do *ensemble*, sem modificar as soluções. Ao aplicar a medida de dissimilaridade, o modelo EBPd, Fig. 5.1-(c), conta com componentes mais bem distribuídos no espaço disponível tornando o *ensemble* gerado bastante diversificado.

Considerando o segundo problema, Fig. 5.1-(d),(e),(f), foram gerados comitês com 10 componentes. De maneira análoga, é possível observar que os componentes do comitê sem balanceamento e sem dissimilaridade, Fig. 5.1-(d), são pouco diversificados. Aplicar o balanceamento das soluções para o EBP, Fig. 5.1-(e), a diversidade dos componentes é aumentada, mas ainda existem muitos componentes similares. Assim, ao introduzir a medida de dissimilaridade no modelo, Fig. 5.1-(f), o comitê gerado ficou diverso e

bem distribuído em relação ao espaço. Dessa forma, tem-se indícios de que a medida de dissimilaridade é efetiva e capaz de maximizar a diversidade no *ensemble*.

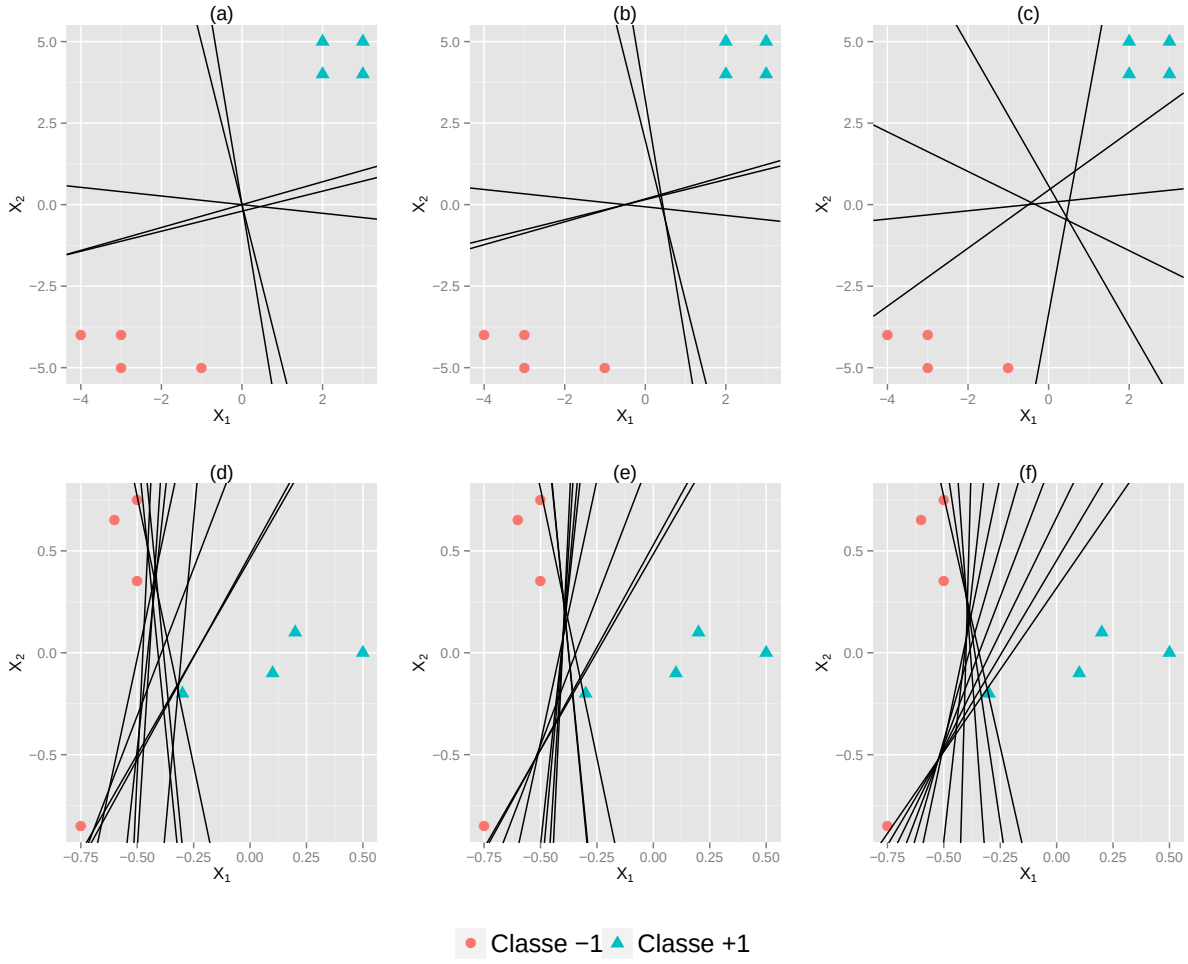


Figure 5.1: Balanceamento e medida de dissimilaridade:(a)(d) *Ensemble* de Perceptrons sem balanceamento e sem dissimilaridade (b)(e) *Ensemble* obtido a partir do método EBP; (c)(f) *Ensemble* obtido a partir do método EBPd.

5.3 DEFINIÇÃO DE PARÂMETROS

Essa seção apresenta os dados referentes à parametrização dos modelos avaliados no estudo experimental. Os métodos de avaliação empregados na análise dos resultados numéricos são apresentados, bem como os testes estatísticos realizados para verificar a significância dos resultados. Posteriormente, as condições para escolha do *kernel* são discutidas. Finalmente, a forma com que o parâmetro da medida de dissimilaridade foi empregado e os dados referentes à quantidade de componentes definido para a formação do *ensemble*

são apresentados. Vale ressaltar que, para todos os modelos que utilizam o algoritmo Perceptron, seja em variáveis primais ou duais, o parâmetro referente à taxa de aprendizado foi fixado em $\eta = 0,05$.

5.3.1 *K-FOLD CROSS-VALIDATION*

A técnica de *cross-validation* é amplamente empregada em problemas de predição e tem como objetivo avaliar a capacidade de generalização de um modelo dado um conjunto de dados a ser testado (KOHAVI, 1995). Essa técnica foi proposta em Mosteller e Tukey (1988) e revisada em Geisser (1975); Wahba e Wold (1975). O método funciona a partir da geração de uma permutação aleatória do conjunto de treinamento Z, de cardinalidade m, e posterior subdivisão em k subconjuntos, portanto *k-fold cross-validation*. Assim, para todo $i \in \{1, 2, ..., k\}$, Z_i tem tamanho m/k. A partir daí, k modelos são gerados da seguinte forma: para cada i, o i-ésimo modelo é gerado tomando $Z - Z_i$, como conjunto de treinamento. O modelo obtido é então testado no conjunto Z_i. O *cross-validation* é então estabelecido. Por fim, calcula-se a média dos erros obtidos para cada um dos conjuntos de teste Z_i, $i \in \{1, 2, ..., k\}$. Dessa forma, não é necessário dividir previamente o conjunto de dados de entrada em amostras de treino e teste e todos os dados disponíveis são usados nas fases de treinamento e teste.

O método de *k-fold* estratificado é uma derivação da técnica original que emprega uma estratificação do conjunto de dados de entrada visando obter modelos mais justos e com melhores resultados (KOHAVI, 1995). Estratificar os dados implica em dividi-los de forma que em cada conjunto seja mantida a proporcionalidade entre as classes do problema. Essa divisão é feita verificando a porcentagem de dados disponíveis para cada classe no conjunto de amostras. O processo de treinamento e teste dos dados permanece o mesmo, o que muda é a forma de se dividir o conjunto de dados em k partes de tamanho m/k (KOHAVI, 1995; EFRON; TIBSHIRANI, 1997).

Apesar de tratar-se de uma técnica bem difundida, não existe um consenso sobre um número adequado para k. A maior parte dos trabalhos utiliza ou sugere a utilização de $k = 10$ (GEISSER, 1993; KOHAVI, 1995). Considerando a natureza aleatória dos modelos abordados, em todos os experimentos foi empregada uma estratégia de *cross-validation* em 10x10-10-*fold*, i.e, 10 execuções independentes de 10 execuções de um 10-*fold*, exceto para o caso do SVM, em que foi empregado a estratégia em 1x10-10-*fold*. Esse esquemas foram

adotados com intuito de reduzir o viés dos métodos analisados. Em relação à estratégia estratificada de *cross-validation*, em cada conjunto sempre foi mantida a porcentagem de dados referentes a cada classe (KOHAVI, 1995). Visando comparações mais precisas, para cada base de dados, sempre foram selecionados os mesmos 10 subconjuntos do *cross-validation*, preservando a geração da semente associada ao processo aleatório.

5.3.2 MÉTODO DE AVALIAÇÃO DE ERRO

Como método de avaliação dos algoritmos testados, optou-se por medir a acurácia de classificação dos modelos por meio da taxa de erro convencional, i.e., a porcentagem de amostras de teste classificadas incorretamente, considerando a média e o desvio padrão dos erros obtidos em cada execução independente.

Com intuito de investigar a significância estatística dos modelos avaliados, foram aplicados os testes *Friedman Rank Sum* (FRIEDMAN, 1937) e o *post-hoc Nemenyi* (NEMENYI, 1962; DEMSAR, 2006), responsáveis por identificar a não equivalência dos resultados apresentados e comparar dois a dois os métodos avaliados, respectivamente. Os testes estatísticos foram realizados a partir do pacote R PMCMR (POHLERT, 2015).

5.3.3 MEDIDA DE DISSIMILARIDADE

A medida de dissimilaridade é o único parâmetro não fixado no modelo tratado, i.e., depende de cada problema especificamente. Esse parâmetro é responsável por imprimir diversidade na construção do *ensemble*. A escolha de um valor adequado depende tanto da complexidade do problema quanto do tamanho pretendido de comitê. Para a construção de comitês com muitos componentes, valores pequenos devem ser estabelecidos para garantir admissão dos componentes requisitados para construção do *ensemble*, independentemente da medida de distância empregada por cada modelo.

No decorrer do estudo experimental, a medida de dissimilaridade foi definida com base em testes empíricos para cada algoritmo analisado. O parâmetro foi estabelecido conforme o maior valor (aproximado) possível capaz de construir o *ensemble* com a quantidade de componentes desejado.

As tabelas a seguir apresentam os valores do parâmetro da medida de dissimilaridade empregados para cada base de dados e tamanho de comitês. A Tabela 5.2 apresenta os valores obtidos para todas as bases testadas pelo EBPd considerando três tamanhos

diferentes de *ensemble*, 10, 100 e 1000. Já a Tabela 5.3 apresenta os resultados análogos à tabela anterior para as bases testadas pelo EBPKd, com *kernel* Gaussiano e a largura $\sigma = 1$, considerando as mesmas quantidades de componentes descritas acima.

Table 5.2: Valores de Dissimilaridade aplicados para cada uma das bases para o EBPd.

Base	Tamanho do comitê		
	10	100	1000
Prostate	10,0	1,00	0,55
Breast	70,0	8,00	1,30
Colon	11,0	1,50	0,70
Leukemia	97,0	17,0	2,50
DLBCL	150,0	15,0	3,00
CNS	125,0	10,0	0,70

Table 5.3: Valores de Dissimilaridade aplicados para cada uma das bases para o EBPKd, com *kernel* gaussiano e largura $\sigma = 1$.

Base	Tamanho do comitê		
	10	100	1000
Sonar	0,570	0,500	0,450
Ionosphere	0,500	0,400	0,350
Tic Tac	0,880	0,850	0,820
Bupa	0,180	0,160	0,140
Pima	0,060	0,056	0,050
Wine	0,195	0,150	0,120
BreastII	0,100	0,093	0,090
Heart	0,175	0,160	0,140
Live	0,190	0,165	0,153

A partir das tabelas 5.2 e 5.3 é possível observar que, de fato, à medida que o número de componentes no *ensemble* aumenta, o valor da medida de dissimilaridade diminui. Esse fato é esperado, uma vez que, a diminuição do valor do parâmetro de dissimilaridade reflete na relaxação da restrição, possibilitando a aceitação de mais componentes.

5.3.4 TAMANHO DO *ENSEMBLE*

Com o intuito de investigar o efeito causado pela variação do tamanho do *ensemble* na capacidade de generalização do modelo obtido, foram realizados testes nas referidas bases de dados. Executou-se um esquema de *cross-validation* segundo a configuração 10x10-10-*fold* considerando as duas formas de combinação de saída empregadas nesse estudo e aplicada aos métodos EBPd e EBPKd. Os valores usados no parâmetro da medida de dissimilaridade são os mencionados na Seção 5.3.3.

O tamanho do *ensemble* foi variado segundo $s = \{1, ..., 10, 20, ..., 100, 200, ..., 1000\}$. Entretanto, optou-se por condensar os resultados apresentando apenas três valores principais: 10, 100 e 1000. Vale ressaltar que não houve diferença significativa nos valores intermediários do número de componentes.

Table 5.4: Resultados da comparação entre diferentes tamanhos de comitê considerando o erro médio de classificação e desvio padrão para o EBPd.

Base	Tamanho	Média	Votação Majoritária
Prostate	10	10,10 ± 1,51	**9,47 ± 1,64**
	100	**9,81 ± 1,49**	9,81 ± 1,49
	1000	9,81 ± 1,49	9,81 ± 1,49
Breast	10	**19,95 ± 3,06**	**19,90 ± 3,17**
	100	20,85 ± 2,62	20,83 ± 2,69
	1000	20,97 ± 2,39	21,00 ± 2,38
Colon	10	15,07 ± 2,26	15,60 ± 2,19
	100	**14,98 ± 2,38**	**15,15 ± 2,43**
	1000	14,98 ± 2,12	15,20 ± 2,34
Leukemia	10	3,43 ± 1,22	3,62 ± 1,27
	100	3,35 ± 0,95	3,38 ± 0,96
	1000	**3,30 ± 0,93**	**3,33 ± 0,94**
DLBCL	10	**3,68 ± 0,73**	**3,59 ± 0,79**
	100	3,81 ± 0,63	3,79 ± 0,66
	1000	3,81 ± 0,63	3,81 ± 0,68
CNS	10	**33,00 ± 2,40**	**31,80 ± 2,75**
	100	33,24 ± 2,34	32,07 ± 2,78
	1000	33,46 ± 2,19	32,15 ± 2,43

Considerando todos os resultados obtidos, observou-se que, na maioria dos casos, o aumento do número de componentes do comitê não necessariamente resulta em redução da taxa de erro. Além disso, em alguns casos, observa-se que, quanto maior a quantidade de componentes no comitê, maior o erro de generalização obtido. Esse efeito pode ser observado claramente nas tabelas 5.4 e 5.5. Tais resultados sugerem que o aumento no tamanho do *ensemble* pode culminar em *overfitting* do modelo, como mencionado em (QUINLAN, 1996). Como consequência, a geração de um número de componentes pequeno tende a evitar *overfitting* no comitê.

Observou-se ainda que na grande maioria dos casos nos quais o método de combinação é a média, os *ensembles* com 10 componentes apresentam a menor taxa de erro em relação aos demais valores. De forma análoga, quando o método de combinação é o voto majoritário, o mesmo padrão foi predominante. Por essa razão, sem eventuais perdas

Table 5.5: Resultados da comparação entre diferentes tamanhos de comitê considerando o erro médio de classificação e desvio padrão para o EBPKd.

Base	Tamanho	Média	Votação Majoritária
Sonar	10	14,26 ± 1,05	14,56 ± 1,36
	100	14,11 ± 0,94	14,31 ± 0,89
	1000	**14,03 ± 0,96**	**14,19 ± 0,83**
Ionosphere	10	**5,92 ± 0,51**	**6,08 ± 0,43**
	100	5,96 ± 0,53	6,10 ± 0,42
	1000	5,99 ± 0,47	6,11 ± 0,31
Tic Tac	10	0,02 ± 0,04	0,11 ± 0,10
	100	0,01 ± 0,03	0,01 ± 0,04
	1000	**0,00 ± 0,00**	**0,01 ± 0,03**
Bupa	10	**37,73 ± 0,68**	37,55 ± 0,74
	100	39,79 ± 0,67	37,51 ± 0,64
	1000	40,45 ± 0,27	**37,36 ± 0,70**
Pima	10	**34,89 ± 0,02**	**34,68 ± 0,62**
	100	34,89 ± 0,00	34,68 ± 0,13
	1000	34,89 ± 0,00	34,68 ± 0,11
Wine	10	**36,30 ± 1,18**	32,93 ± 1,38
	100	39,74 ± 0,35	**32,84 ± 1,18**
	1000	39,90 ± 0,00	33,27 ± 0,97
BreastII	10	**34,23 ± 0,19**	**34,23 ± 0,19**
	100	34,23 ± 0,19	34,23 ± 0,19
	1000	34,23 ± 0,19	34,23 ± 0,19
Heart	10	**45,39 ± 1,30**	46,82 ± 1,04
	100	45,76 ± 0,69	**45,19 ± 0,71**
	1000	45,93 ± 0,70	45,24 ± 1,04
Live	10	**37,70 ± 0,73**	**37,87 ± 1,09**
	100	38,72 ± 0,78	41,92 ± 1,58
	1000	40,56 ± 0,35	43,55 ± 1,96

significativas da capacidade de generalização, optou-se por fixar o tamanho do modelo *ensemble* em $10 + 1$ componentes, com intuito de evitar possíveis empates ao utilizar a estratégia de votação.

5.3.5 *KERNEL*

Para a escolha do *kernel* que foi utilizado para as bases não-linearmente separáveis, foram realizadas variações de parâmetro dos *kernels* polinomial e gaussiano e o erro médio e desvio padrão de 10 execuções de 10-10-*fold* foram verificados. Para o *kernel* polinomial, foram utilizados os graus $d = 2$ e $d = 3$. Entretanto, na maior parte dos problemas avaliados, não foi possível obter a convergência dos algoritmos avaliados e, por essa razão,

esse método foi desprezado.

Em relação ao *kernel* gaussiano, 5 larguras de σ foram testadas com valores iguais a $\{0,01,\ 0,1,\ 1,\ 10,\ 100\}$. Fixado o tamanho do *ensemble* em 10+1 componentes, a Tabela 5.6 apresenta todos os valores obtidos de medida de dissimilaridade para todas as largura de *kernel* gaussiano testadas.

Table 5.6: Valores de dissimilaridade aplicados para cada uma das bases para o EBPKd, com *kernel* gaussiano e 5 larguras testadas.

Base	σ				
	0,01	0,1	1	10	100
Sonar	0,380	0,450	0,570	0,600	0,300
Ionosphere	0,180	0,330	0,500	0,410	0,260
Tic Tac	0,600	0,750	0,880	0,825	0,880
Bupa	0,180	0,350	0,180	0,150	0,180
Pima	0,330	0,310	0,060	0,042	0,060
Wine	0,500	0,420	0,195	0,120	0,120
BreastII	0,135	0,115	0,100	0,100	0,100
Heart	0,175	0,330	0,175	0,175	0,175
Live	0,200	0,350	0,190	0,150	0,190

Para comparação e apresentação dos resultados numéricos, ao invés de fixar uma largura para o *kernel* gaussiano, optou-se por utilizar o melhor resultados obtido para cada método e para cada base de dados considerando todas as larguras avaliadas. Essa decisão foi tomada com intuito de promover uma avaliação justa, permitindo que a melhor solução de cada método seja comparada adequadamente para a avaliação dos algoritmos. Os resultados completos obtidos para todos os algoritmos e todas as bases de dados avaliadas em cada uma das larguras em questão encontram-se no Apêndice A, para fins de reprodutibilidade.

5.4 RESULTADOS NUMÉRICOS

Essa seção tem como objetivo discutir e apresentar os resultados numéricos obtidos da comparação dos métodos propostos. Em todos os casos, os métodos propostos foram comparados ao Perceptron ou ao PK, ao respectivo classificador de base (PB ou PKB) e aos algoritmos SVM e AdaBoost. As condições nas quais os experimentos foram realizados foram anteriormente descritas, na Seção 5.3, bem como os dados necessários para reprodução dos experimentos.

Para facilitar a compreensão, os resultados numéricos serão discutidos separadamente em relação aos métodos propostos. Inicialmente, discute-se os resultados obtidos para o EBP para as 6 bases de dados linearmente separáveis analisadas. Em relação ao EBPK, testado em 8 bases de dados não-linearmente separáveis, além da base Sonar, optou-se por discutir os resultados obtidos para as melhores configurações de parâmetro de *kernel* de cada um dos métodos. Entretanto, os resultados completos estão disponíveis no Apêndice A, para fins de reprodutibilidade. Os resultados para o VSRM são apresentados em seguida.

Dois grupos distintos foram considerados para avaliação dos resultados do EBP e EBPK. Os grupos foram divididos considerando as duas estratégias para combinação dos componentes adotadas: a média e o voto majoritário não ponderado. O primeiro grupo considera apenas métodos que podem ser representados por uma única hipótese ou pela média das hipóteses. Nesse grupo, são apresentados os resultados para os métodos propostos indicados com *m-*, significando que a hipótese final é obtida a partir da média das hipóteses do comitê. Esses resultados são comparados a um classificador SVM. O segundo grupo trata dos métodos cuja solução é obtida por esquemas de votação. Os resultados são apresentados para os métodos indicados *v-* e comparados ao AdaBoost. Essa separação é aplicada para que métodos que avaliam diversas hipóteses, como nos casos de votação, não sejam comparados a métodos que definem uma única hipótese, como nos casos de média das soluções. Para o VSRM essa divisão não foi aplicada, uma vez que o método gera uma hipótese equivalente a votação majoritária de um EBP.

5.4.1 RESULTADOS EBP

5.4.1.1 *Ensemble* representado pela média das hipóteses

A primeira parte da análise do EBP compara métodos que geram uma única hipótese, como o SVM, o Perceptron e o EBP combinado pela média dos componentes. A Tabela 5.7 apresenta os resultados obtidos em relação ao percentual de amostras classificadas incorretamente dado pelo erro médio e desvio padrão para cada método comparado e cada base de dados avaliada durante a fase de teste.

Os valores em destaque apresentam os melhores resultados para cada base de dados. A partir dos resultados obtidos, observou-se que o m-EBP e o m-EBPd apresentaram

Table 5.7: Comparação entre o m-EBP e o m-EBPd com o SVM.

Base	Perceptron	PB	m-EBP	m-EBPd	SVM
Prostate	11,86 ± 1,14	10,42 ± 1,73	10,02 ± 1,48	10,10 ± 1,51	**9,58 ± 1,35**
Breast	23,07 ± 4,71	20,85 ± 4,52	20,00 ± 2,78	**19,95 ± 3,06**	20,67 ± 2,49
Colon	19,97 ± 2,44	19,03 ± 3,00	15,11 ± 2,27	**15,07 ± 2,26**	18,69 ± 2,32
Leukemia	6,55 ± 0,93	4,00 ± 1,45	3,48 ± 1,05	3,43 ± 1,22	**2,75 ± 0,88**
DLBCL	5,97 ± 1,14	4,87 ± 1,67	3,71 ± 0,64	**3,68 ± 0,73**	3,81 ± 0,68
CNS	36,57 ± 3,50	36,70 ± 4,26	33,17 ± 2,67	**32,99 ± 2,40**	33,50 ± 1,99

taxas de erro inferiores ao SVM em 4 das 6 bases de dados avaliadas. Como esperado, o m-EBPd superou, em poder de generalização o m-EBP, em todos os casos, já que a introdução da medida de dissimilaridade contribui para o aumento da diversidade do *ensemble* resultando em uma média dos componentes mais representativa do espaço. É possível observar ainda que o PB apresentou taxa de erro inferior ao Perceptron na maioria dos casos, mostrando que houve aumento de acurácia a partir do balanceamento.

Ao checar a significância estatística dos resultado, aplicou-se o teste de Friedman Rank Sum que assume, como hipótese nula, a equivalência da capacidade de generalização dos cinco algoritmos comparados. O teste de Friedman indicou significância ($\chi^2 = 19,0667,\ df = 4,\ p\text{-valor} = 0,0007626$), o que permite que seja rejeitada a hipótese nula. Uma vez que a hipótese nula foi rejeitada, deve-se prosseguir com o teste post-hoc Nemenyi, que promove uma comparação par-a-par da significância dos resultados dos algoritmos. A partir do teste de Nemenyi, foi possível concluir que os métodos propostos m-EBP e m-EBPd diferem significativamente do algoritmo do Perceptron original, com p-valor $= 0,0052$ e p-valor $= 0,0048$, respectivamente. Além disso, foi possível concluir que o m-EBPd difere significativamente do PB, p-valor $= 0,0395$. Apesar das outras comparações não indicarem significância estatística, ou seja, os outros métodos comparados dois a dois têm resultados equivalentes, vale ressaltar que o método proposto nesse trabalho supera, em capacidade de generalização, o SVM na maior parte dos casos apresentados.

5.4.1.2 *Ensemble* representado pelo voto das hipóteses

A segunda parte da análise do EBP compara métodos combinados por esquemas de votação. Dessa forma, os resultados obtidos para o v-EBP e v-EBPd, que são combinados pelo voto majoritário não ponderado das soluções são comparados com os do AdaBoost, que emprega um esquema de voto ponderado. Adicionalmente, os resultados para o Perceptron e o PB são apresentados novamente, para comparação da efetividade

do modelo proposto e para verificar a satisfabilidade das premissas de construção do *ensemble*. Os resultados estão dispostos na Tabela 5.8 considerando a média do erro e o desvio padrão obtidos durante a fase de teste.

Table 5.8: Comparação entre o v-EBP e o v-EBPd com o AdaBoost.

Base	Perceptron	PB	v-EBP	v-EBPd	AdaBoost
Prostate	11,86 ± 1,14	10,42 ± 1,73	**9,03 ± 1,43**	9,47 ± 1,64	11,17 ± 1,65
Breast	23,07 ± 4,71	20,85 ± 4,52	**19,70 ± 2,53**	19,90 ± 3,17	22,57 ± 4,81
Colon	19,97 ± 2,44	19,03 ± 3,00	**15,22 ± 2,12**	15,60 ± 2,19	18,30 ± 2,72
Leukemia	6,55 ± 0,93	4,00 ± 1,45	**3,57 ± 1,30**	3,62 ± 1,27	6,66 ± 1,62
DLBCL	5,97 ± 1,14	4,87 ± 1,67	3,67 ± 0,64	**3,59 ± 0,79**	4,07 ± 1,38
CNS	36,57 ± 3,50	36,70 ± 4,26	**30,72 ± 2,65**	31,80 ± 2,75	35,45 ± 3,32

Os resultados obtidos mostram que ambos os métodos v-EBP e v-EBPd superam a capacidade de generalização do AdaBoost, apresentando erro de teste inferior em todos os casos. Como esperado, o método v-EBP apresenta erro no generalização inferior ao v-EBPd na maior parte dos testes. Isso ocorre pois já que o v-EBPd é combinado por voto majoritário não ponderado e a boa distribuição das hipóteses no espaço de solução, obtidos através da aplicação da medida de dissimilaridade, não contribuiu para o aumento da acurácia do modelo, uma vez que permite a consideração de classificadores pouco acurados, em termos de capacidade de generalização, para construção do *ensemble*.

Analisando a relevância estatística dos resultados obtidos, segundo o teste de Friedman a hipótese nula é rejeitada ($\chi^2 = 20,13,\ df = 4,\ p\text{-valor} = 0,00047$), mostrando que existem diferenças significativas em relação aos métodos comparados. Segue então o teste post-hoc Nemenyi. Os resultados do teste mostram que o v-EBP e o v-EBPd diferem significativamente do Perceptron com p-valor $= 0,00044$ e p-valor $= 0,0012$, respectivamente. Além disso, o teste de Nemenyi mostrou que o v-EBP apresenta diferenças significativas em relação ao PB e ao AdaBoost, com p-valor $= 0,03951$ e p-valor $= 0,0485$, respectivamente.

5.4.2 RESULTADOS EBPK

5.4.2.1 *Ensemble* representado pela média das hipóteses

Essa seção tem por objetivo discutir os resultados obtidos pelo método proposto, combinado pela média dos componentes do *ensemble* (m-EBPK e m-EBPKd), com os modelos SVM, PKB e PK. Todos os métodos foram aplicados às bases de dados descritas na Seção

5.1 levando em consideração os parâmetros definidos na Seção 5.3. A Tabela 5.9 apresenta os melhores resultados de erro médio e desvio padrão obtidos para os problemas avaliados, considerando a melhor execução em relação aos 5 valores de largura σ testados.

Table 5.9: Comparação entre o m-EBPK e o m-EBPKd com o SVM.

Base	PK	PKB	m-EBPK	m-EBPKd	SVM
Sonar	15,72 ± 1,50	15,67 ± 1,32	13,78 ± 1,09	14,26 ± 1,05	**13,40 ± 0,77**
Ionos	6,30 ± 0,78	5,83 ± 0,78	4,96 ± 0,47	**4,94 ± 0,47**	7,15 ± 0,40
Tictac	1,38 ± 0,31	0,22 ± 0,19	**0,01 ± 0,02**	0,02 ± 0,04	0,66 ± 0,03
Bupa	38,56 ± 1,28	38,03 ± 1,63	36,62 ± 1,22	**36,61 ± 0,93**	36,93 ± 1,09
Pima	32,90 ± 1,06	32,87 ± 1,00	32,76 ± 0,88	**32,22 ± 1,02**	34,89 ± 0,00
Wine	21,56 ± 1,16	20,55 ± 1,43	20,50 ± 1,60	19,08 ± 1,05	**18,46 ± 0,96**
BreastII	65,06 ± 0,67	34,23 ± 0,19	33,89 ± 0,24	33,83 ± 0,25	**33,57 ± 0,21**
Heart	39,86 ± 1,32	40,33 ± 1,73	39,47 ± 1,07	**39,32 ± 1,12**	39,93 ± 1,08
Live	38,97 ± 1,70	37,95 ± 1,62	36,82 ± 1,16	**36,66 ± 0,97**	37,22 ± 1,10

Através dos resultados apresentados, observa-se que as abordagens m-EBPK e m-EBPKd superam o classificador individual PKB e o PK em todos os casos avaliados. O método proposto apresenta também uma taxa de erro de generalização inferior ao modelo SVM em 6 das 9 bases de dados testadas. Observa-se ainda que a inclusão da medida de dissimilaridade, como esperado, produz efeitos positivos na redução da taxa de erro do modelo. Assim, a média das hipóteses obtida através do m-EBPKd é mais representativa do que a média obtida pelo m-EBPK. O PKB apresentou taxas de erro inferiores ao PK na maior parte das bases de dados, o que indica que o balanceamento da solução pode implicar em melhoria de acurácia do modelo.

Em relação à relevância estatística dos resultados encontrados, segundo o teste de Friedman, a hipótese nula de equivalência dos modelos analisados é rejeitada ($\chi^2 = 21,2444$, $df = 4$, p-valor $= 0,0002832$), mostrando que existem diferenças significativas entre os métodos comparados. Prosseguindo com o teste post-hoc Nemenyi par a par, os resultados mostram que o m-EBPK e o m-EBPKd apresentam diferenças estatisticamente significativas em relação ao modelo Perceptron, com p-valor $= 0,00920$ e p-valor $= 0,00055$, respectivamente. Além disso, o método m-EBPKd difere significativamente do PB, com p-valor $= 0,02398$. Embora as demais comparações não apresentem significância estatística em relação a diferença dos métodos para os casos avaliados, vale ressaltar que o método proposto apresenta resultados de acurácia superiores ao SVM na maior parte dos casos.

5.4.2.2 *Ensemble* representado pelo voto das hipóteses

A segunda parte da análise experimental do EBPK compara-o combinado pelo voto majoritário não ponderado dos componentes (v-EBPK e v-EBPKd) aos modelos PKB, PK e AdaBoost. O AdaBoost utiliza um Perceptron *kernel* simples como classificador de base e considera o voto ponderado dos membros do comitê. A Tabela 5.10 apresenta os melhores resultados de erro médio e desvio padrão obtidos para os problemas em questão são destacados em negrito, considerando a melhor execução em relação aos 5 valores de σ testados.

Table 5.10: Comparação entre o v-EBPK e o v-EBPKd com o AdaBoost.

Base	PK	PKB	v-EBPK	v-EBPKd	AdaBoost
Sonar	15,72 ± 1,50	15,67 ± 1,32	**13,87 ± 1,00**	14,56 ± 1,36	14,99 ± 1,58
Ionos	6,30 ± 0,78	5,83 ± 0,78	**5,04 ± 0,53**	5,05 ± 0,58	5,91 ± 0,82
Tictac	1,38 ± 0,31	0,22 ± 0,19	**0,01 ± 0,03**	0,11 ± 0,10	0,63 ± 0,24
Bupa	38,56 ± 1,28	38,03 ± 1,63	**37,10 ± 1,15**	37,18 ± 1,43	39,11 ± 1,77
Pima	32,90 ± 1,06	32,87 ± 1,00	**32,40 ± 0,93**	32,47 ± 0,95	33,82 ± 0,96
Wine	21,56 ± 1,16	20,55 ± 1,43	**19,35 ± 0,97**	19,69 ± 1,19	20,12 ± 1,49
BreastII	65,06 ± 0,67	34,23 ± 0,19	**34,20 ± 0,37**	34,23 ± 0,19	34,48 ± 0,35
Heart	39,86 ± 1,32	40,33 ± 1,73	**39,04 ± 1,04**	39,37 ± 1,16	40,38 ± 1,37
Live	38,97 ± 1,70	37,95 ± 1,62	**37,03 ± 1,52**	37,12 ± 1,12	38,92 ± 1,70

Observa-se, a partir dos resultados reportados, que as formulações v-EBPK e v-EBPKd apresentam resultados bastante satisfatórios. Ambas as abordagens superam o classificador individual PKB e o modelo Perceptron *kernel*. Além disso, o método proposto apresenta uma taxa de erro inferior ao AdaBoost em todas as 9 bases de dados consideradas nesse estudo. Em relação a eficácia da medida de dissimilaridade é possível observar que o emprego da medida não gerou redução na taxa de erro do modelo. Assim como no EBP, isso aconteceu devido ao esquema de combinação das saídas, no qual o voto não ponderado garante o mesmo peso na votação a todos os componentes do comitê. Entretanto, vale destacar que mesmo ao aplicar a medida de dissimilaridade, os resultados ainda sim superam em todos os casos o AdaBoost.

Observando os resultados de relevância estatística dos resultados obtidos, de acordo com o teste de Friedman, a hipótese nula pode ser rejeitada ($\chi^2 = 31,2179,\ df = 4,\ p$-valor $= 0,000002764$), evidenciando que os métodos comparados não são equivalentes. Segue o teste de post-hoc Nemenyi. Os resultados do teste mostram que os métodos v-EBPK e v-EBPKd apresentam diferenças relevantes em relação ao Perceptron, com p-valor $= 0,000018$ e p-valor $= 0,00712$, respectivamente. Observa-se ainda que o v-

EBPK apresenta diferenças estatisticamente significativas em relação aos métodos PB e AdaBoost, com p-valor $= 0,01905$ e p-valor $= 0,00029$, respectivamente. Finalmente, o teste de Nemenyi mostrou ainda que o método v-EBPKd apresenta resultados estatísticos relevantes em relação ao AdaBoost, mostrando diferença significativa entre os métodos com p-valor $= 0,04602$.

5.4.3 RESULTADOS VSRM

Inicialmente, para identificar a corretude do método proposto, foram gerados *ensembles*-guia com 1 componente e observou-se que o algoritmo é capaz de convergir para o ponto representado pelo componente gerado. Dessa forma, conclui-se que o método é capaz de escolher o maior semi-espaço adequadamente, a partir da votação majoritária dos componentes. Para validar o algoritmo, um m-EBP foi gerado e testado segundo um esquema de k-*fold* para uma base de dados. Esse mesmo *ensemble* foi utilizado como entrada de *ensemble*-guia para o VSRM. Após a convergência do algoritmo, a solução gerada foi testada segundo o mesmo esquema de k-*fold*, a partir da semente preservada, para a mesma base de dados e os resultados obtidos foram equivalentes.

O estudo experimental subsequente para o VSRM foi gerado a partir dos dados linearmente separáveis apresentados anteriormente. A comparação foi realizada entre o v-EBPd e o VSRM, com intuito de identificar a equivalência das soluções dadas pelo v-EBPd e pela aproximação provida pelo VSRM. A hipótese gerada pelo VSRM foi ainda comparada aos classificadores de base, Perceptron e PB, e com os métodos AdaBoost e SVM, ambos considerados estado da arte. Como a solução do VSRM é uma única hipótese equivalente a um v-EBPd a comparação com o SVM é, a partir de agora, viável. A Tabela 5.11 apresenta os resultados de erro médio de classificação na fase de teste e desvio padrão. Os melhores resultados para cada base de dados estão destacados em negrito.

Table 5.11: Comparação entre o v-EBPd e o VSRM com o AdaBoost e o SVM.

Base	Perceptron	PB	v-EBPd	VSRM	AdaBoost	SVM
Prostate	11,86 ± 1,14	10,42 ± 1,73	9,47 ± 1,64	**9,46 ± 1,77**	11,17 ± 1,65	9,58 ± 1,35
Breast	23,07 ± 4,71	20,85 ± 4,52	19,90 ± 3,17	**19,83 ± 2,39**	22,57 ± 4,81	20,67 ± 2,49
Colon	19,97 ± 2,44	19,03 ± 3,00	15,60 ± 2,19	**15,44 ± 2,29**	18,30 ± 2,72	18,69 ± 2,32
Leukemia	6,55 ± 0,93	4,00 ± 1,45	3,62 ± 1,27	3,39 ± 1,30	6,66 ± 1,62	**2,75 ± 0,88**
DLBCL	5,97 ± 1,14	4,87 ± 1,67	3,59 ± 0,79	**3,49 ± 0,87**	4,07 ± 1,38	3,81 ± 0,68
CNS	36,57 ± 3,50	36,70 ± 4,26	**31,80 ± 2,75**	32,87 ± 2,15	35,45 ± 3,32	33,50 ± 1,99

A partir dos resultados apresentados é possível observar que os valores obtidos para o

v-EBPd e o VSRM são bastante próximos, o que é um possível indicativo da equivalência das soluções. Além disso, assim como o EBP, o VSRM também satisfaz a premissa de construção de *ensembles* ao apresentar resultados superiores ao PB, em termos de capacidade de generalização. Observa-se ainda que o método avaliado é capaz de superar o AdaBoost e o SVM na maior parte das bases de dados consideradas.

Em relação a relevância estatística dos resultados, segundo o teste de Friedman, a hipótese nula relativa a equivalência de resultados de todos os métodos avaliados pode ser rejeitada $\chi^2 = 24,7619,\ df = 5, p-\text{valor} = 0,0001549$. Ao prosseguir com o teste post-hoc de Nemenyi, os resultados obtidos indicaram que os métodos v-EBPd e VSRM são estatisticamente equivalentes ($p-\text{valor} = 0,98982$). Observou-se ainda que os resultados do VSRM apresentam diferença significativas em relação ao Perceptron, PB e AdaBoost, $p-\text{valor} = 0,00085$, $p-\text{valor} = 0,02481$ e $p-\text{valor} = 0.03951$ respectivamente.

6 CONCLUSÕES E TRABALHOS FUTUROS

Esse trabalho apresentou um nova abordagem *ensemble* baseada em Perceptrons Balanceados e um método para geração de uma hipótese equivalente a um EBP combinado por voto majoritário.

Em relação à primeira contribuição, o EBP foi proposto em duas versões. A versão primal, implementada exclusivamente para a solução de problemas linearmente separáveis combina três heurísticas para geração de componentes, entre elas uma medida de dissimilaridade. Além disso, o hiperplano solução do Perceptron é balanceado como uma estratégia para aumentar a acurácia do modelo. A segunda versão, a dual, é uma extensão da versão primal que permite a utilização de funções *kernel* e a solução de problemas não-linearmente separáveis. Para essa versão, além do balanceamento da solução, apenas duas das três medidas de geração de diversidade foram empregadas. Um estudo experimental foi realizado para as duas versões do EBP e os resultados obtidos foram estatisticamente avaliados. Mostrou-se que o balanceamento da solução foi responsável por um aumento de acurácia do PB em relação ao Perceptron. Além disso, a premissa para desenvolvimento de *ensembles* foi confirmada, ao mostrar que o modelo proposto foi superior, em acurácia, ao classificador de base PB. Em relação à medida de dissimilaridade aplicada, para ambas as versões, observou-se que os resultados foram bastante satisfatórios quando o método de combinação de saídas adotado foi a média dos classificadores. Esse resultado era esperado, uma vez que o *ensemble* gerado é diverso e bem distribuído, culminando em uma média bem representativa no espaço de solução do problema. Por outro lado, quando a estratégia de combinação de saídas foi o voto majoritário, na maior parte dos casos, os resultados do EBPd e EBPKd foram inferiores ao EBP e EBPK, respectivamente. Esse resultado também é esperado, já que a votação majoritária não é ponderada e, assim, classificadores ruins têm o mesmo poder de voto que os classificadores bons. Em relação aos resultados comparativos com os métodos SVM e AdaBoost, o método proposto foi capaz de superar ambos os classificadores na maior parte dos casos. Em particular, testes estatísticos mostraram que o algoritmo proposto difere significativamente do Perceptron, PB e AdaBoost.

Em relação a segunda contribuição, o VSRM foi proposto em variáveis primais. Essa

versão permite a solução de problemas linearmente separáveis. O algoritmo converge para uma hipótese equivalente a votação majoritária de um EBP através de sucessivas reduções do espaço de versões. O estudo experimental foi realizado com o objetivo de validar a geração da hipótese equivalente a partir da comparação das taxas de erro de classificação do VSRM e do v-EBPd. A equivalência das soluções foi comprovada por testes estatísticos. O VSRM foi ainda comparado ao Perceptron, PB, AdaBoost e SVM. O método proposto superou o Perceptron, PB e o AdaBoost em todos os casos analisados. A relevância dos resultados foi também confirmada por testes estatísticos. Em relação ao SVM, a comparação não apresentou resultados estatisticamente diferentes. Entretanto, o método apresentado superou, em termos de capacidade de generalização, o SVM em 5 das 6 bases testadas.

Como algumas possibilidades de trabalhos futuros, quanto ao VSRM cuja formulação é extensível tanto à introdução de funções *kernel* quanto à flexibilização da margem, o que permite também a solução de problemas não-linearmente separáveis, sugere-se a implementação do método segundo essas abordagens. Quanto ao processo de decisão do lado concordante com o comitê, atualmente, não existe garantia de que a decisão é sempre tomada de maneira correta, i.e, o lado concordante é sempre o maior lado do espaço de versões. Por essa razão, uma solução que possa garantir que o semi-espaço a ser desprezado é o menor dos dois semi-espaços avaliados pode ser implementada.

Ainda em relação ao VSRM, outra possibilidade de trabalho futuro reside na comparação da solução proposta com a Máquina de Ponto de Bayes (*Bayes Point Machine* – BPM), apresentada por Herbrich et al. (2001), um modelo para aproximação do centro de massa do espaço de versões. De acordo com o trabalho de Herbrich et al. (2001), o centro de massa do espaço de versões, equivalente ao Ponto de Bayes, consiste na solução ótima do problema de classificação. Em teoria, o algoritmo que alcança o ponto referente ao centro de massa, ou é capaz de obter a melhor aproximação para ele, é chamado de classificador ótimo de Bayes. O Ponto de Bayes pode ser determinado pela média de um *ensemble* infinito, cujos componentes são uniformemente distribuídos no espaço de versões. Atualmente, o BPM é a melhor aproximação para o centro de massa (HERBRICH, 2001). Entretanto, ao aplicar a medida de dissimilaridade proposta no EBP, foi possível observar que os componentes gerados são muito bem distribuídos no espaço de entrada, o que pode indicar boa distribuição no espaço de versões. Dessa forma, a convergência do

VSRM levaria também a uma aproximação do centro massa. Por essa razão, sugere-se um estudo comparativo entre os métodos BPM e VSRM com intuito de confirmar o método proposto como uma aproximação para o centro de massa.

As perspectivas de trabalhos futuros relacionados ao método *ensemble* proposto envolvem a implementação do algoritmo permitindo a flexibilização de margem em relação aos Perceptrons Balanceados. Outro possível trabalho envolve a implementação de um comitê cujas saídas são combinadas através do voto ponderado pelo valor de margem de cada solução.

Por fim, em relação aos dados usados para comparação dos modelos, observou-se que os dados analisados não apresentavam grandes desbalanceamentos em relação as classes. Resultados anteriores (SUN et al., 2015; DÍEZ-PASTOR et al., 2015) mostram que métodos *ensembles* tendem a apresentar bons resultados quando aplicados a bases desbalanceadas. Por essa razão, um estudo experimental para a avaliação do método *ensemble* proposto em bases de dados desbalanceados pode ser realizado.

REFERÊNCIAS

ADAMU, A.; MAUL, T.; BARGIELA, A.; ROADKNIGHT, C. Preliminary experiments with ensembles of neurally diverse artificial neural networks for pattern recognition. In: **Recent Advances in Information and Communication Technology 2015**, 2015.

AIZERMAN, A.; BRAVERMAN, E. M.; ROZONER, L. Theoretical foundations of the potential function method in pattern recognition learning. **Automation and remote control**, 1964.

BACHE, K.; LICHMAN, M. **UCI Machine Learning Repository**. 2013. Disponível em: <http://archive.ics.uci.edu/ml>.

BAUER, E.; KOHAVI, R. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. **Machine learning**, Springer, 1999.

BOSER, B. E.; GUYON, I. M.; VAPNIK, V. N. A training algorithm for optimal margin classifiers. In: ACM. **Proceedings of the fifth annual workshop on Computational learning theory**, 1992. p. 144–152.

BREIMAN, L. Bagging predictors. **Machine learning**, Springer, 1996.

BREIMAN, L. Arcing classifier. **The annals of statistics**, Institute of Mathematical Statistics, 1998.

BURGES, C. J. A tutorial on support vector machines for pattern recognition. **Data mining and knowledge discovery**, Springer, v. 2, n. 2, p. 121–167, 1998.

CUNNINGHAM, P.; CARNEY, J. Diversity versus quality in classification ensembles based on feature selection. In: **Machine Learning: ECML 2000**, 2000.

DEMSAR, J. Statistical comparisons of classifiers over multiple data sets. **The Journal of Machine Learning Research**, JMLR. org, 2006.

DIETTERICH, T. G. Ensemble methods in machine learning. In: **Multiple classifier systems**, 2000. p. 1–15.

DIETTERICH, T. G. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. **Machine learning**, Springer, 2000.

DÍEZ-PASTOR, J. F.; RODRÍGUEZ, J. J.; GARCÍA-OSORIO, C.; KUNCHEVA, L. I. Random balance: Ensembles of variable priors classifiers for imbalanced data. **Knowledge-Based Systems**, Elsevier, 2015.

DUDA, R. O.; HART, P. E.; STORK, D. G. et al. Pattern classification. **International Journal of Computational Intelligence and Applications**, Imperial College Press, v. 1, p. 335–339, 2001.

EFRON, B.; TIBSHIRANI, R. Improvements on cross-validation: the 632+ bootstrap method. **Journal of the American Statistical Association**, Taylor & Francis, 1997.

ENES, K. B.; VILLELA, S. M.; FONSECA NETO, R. A novel ensemble approach based on balanced perceptrons applied to microarray datasets. In: **Proceedings of the 4th Brazilian Conference on Intelligent Systems**, 2015.

ENES, K. B.; VILLELA, S. M.; FONSECA NETO, R. Um classificador kernel composto por um comitê de perceptrons balanceados. In: **Anais do 12 Congresso Brasileiro de Inteligência Computacional**, 2015.

FOLINO, G.; PISANI, F. S. Combining ensemble of classifiers by using genetic programming for cyber security applications. In: **Applications of Evolutionary Computation**, 2015.

FREUND, Y.; SCHAPIRE, R. E. A desicion-theoretic generalization of on-line learning and an application to boosting. In: **Computational learning theory**, 1995.

FREUND, Y.; SCHAPIRE, R. E. Experiments with a new boosting algorithm. In: **Proceedings of the 13th International Conference on Machine Learning**, 1996.

FRIEDMAN, M. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. **Journal of the American Statistical Association**, Taylor & Francis, 1937.

GEISSER, S. The predictive sample reuse method with applications. **Journal of the American Statistical Association**, Taylor & Francis Group, 1975.

GEISSER, S. **Predictive inference**, 1993.

GLAAB, E.; BACARDIT, J.; GARIBALDI, J. M.; KRASNOGOR, N. Using rule-based machine learning for candidate disease gene prioritization and sample classification of cancer gene expression data. **PloS one**, Public Library of Science, 2012.

GOLUB, T. R.; SLONIM, D. K.; TAMAYO, P.; HUARD, C.; GAASENBEEK, M.; MESIROV, J. P.; COLLER, H.; LOH, M. L.; DOWNING, J. R.; CALIGIURI, M. A. et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. **Science**, American Association for the Advancement of Science, 1999.

HAN, L.; LUO, S.; YU, J.; PAN, L.; CHEN, S. Rule extraction from support vector machines using ensemble learning approach: An application for diagnosis of diabetes. **Biomedical and Health Informatics, IEEE Journal of**, IEEE, 2015.

HANSEN, L. K.; SALAMON, P. Neural network ensembles. **IEEE transactions on pattern analysis and machine intelligence**, IEEE Computer Society, 1990.

HERBRICH, R. **Learning Kernel classifiers: theory and algorithms**, 2001.

HERBRICH, R.; GRAEPEL, T.; CAMPBELL, C. Bayes point machines. **JMLR**, 2001.

HOFMANN, T.; SCHÖLKOPF, B.; SMOLA, A. J. Kernel methods in machine learning. **The annals of statistics**, JSTOR, 2008.

KIVINEN, J.; SMOLA, A. J.; WILLIAMSON, R. C. Online learning with kernels. **IEEE Transactions on Signal Processing**, 2004.

KOHAVI, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: **Proceedings of the 14th IJCAI**, 1995.

KUNCHEVA, L. I. **Combining pattern classifiers: methods and algorithms**, 2004.

KUNCHEVA, L. I.; WHITAKER, C. J. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. **Machine learning**, Springer, 2003.

LAI, W.-C.; HUANG, P.-H.; LEE, Y.-J.; CHIANG, A. A distributed ensemble scheme for nonlinear support vector machine. In: **IEEE 10th International Conference on ISSNIP**, 2015.

LAM, L.; SUEN, C. Y. Application of majority voting to pattern recognition: an analysis of its behavior and performance. **Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on**, IEEE, 1997.

LEITE, S. C.; FONSECA NETO, R. Incremental margin algorithm for large margin classifiers. **Neurocomputing**, Elsevier, 2008.

LIPKUS, A. H. A proof of the triangle inequality for the tanimoto distance. **Journal of Mathematical Chemistry**, Springer, 1999.

LIU, H.; TIAN, H.; LIANG, X.; LI, Y. New wind speed forecasting approaches using fast ensemble empirical model decomposition, genetic algorithm, mind evolutionary algorithm and artificial neural networks. **Renewable Energy**, Elsevier, 2015.

MA, Z.; DAI, Q.; LIU, N. Several novel evaluation measures for rank-based ensemble pruning with applications to time series prediction. **Expert Systems with Applications**, Elsevier, 2015.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **The bulletin of mathematical biophysics**, Springer, 1943.

MICHALSKI, R. S.; CARBONELL, J. G.; MITCHELL, T. M. **Machine learning: An artificial intelligence approach**, 2013.

MOSTELLER, F.; TUKEY, J. W. Data analysis, including statistics. **The Collected Works of John W. Tukey: Graphics 1965-1985**, CRC Press, 1988.

NEMENYI, P. Distribution-free multiple comparisons. In: **Biometrics**, 1962.

NOVIKOFF, A. B. **On convergence proofs for perceptrons**, 1963.

OPITZ, D.; MACLIN, R. Popular ensemble methods: An empirical study. **Journal of Artificial Intelligence Research**, 1999.

OPITZ, D. W.; SHAVLIK, J. W. et al. Generating accurate and diverse members of a neural-network ensemble. **Advances in neural information processing systems**, Citeseer, 1996.

PARMANTO, B.; MUNRO, P. W.; DOYLE, H. R. Improving committee diagnosis with resampling techniques. In: **Advances in neural information processing systems**, 1996.

PARVIN, H.; MIRNABIBABOLI, M.; ALINEJAD-ROKNY, H. Proposing a classifier ensemble framework based on classifier selection and decision tree. **Engineering Applications of Artificial Intelligence**, Elsevier, 2015.

PLATT, J. Sequential minimal optimization: A fast algorithm for training support vector machines. technical report msr-tr-98-14, Microsoft Research, 1998.

POHLERT, T. The pairwise multiple comparison of mean ranks package (pmcmr). 2015.

QUINLAN, J. R. Bagging, boosting, and c4.5. In: **AAAI/IAAI**, 1996.

ROLI, F.; GIACINTO, G.; VERNAZZA, G. Methods for designing multiple classifier systems. In: **Multiple Classifier Systems**, 2001.

ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. **Psychological Review**, 1958.

RUSSELL, S.; NORVIG, P. **Artificial intelligence: a modern approach**, 1995.

SOLLICH, P.; KROGH, A. Learning with ensembles: How overfitting can be useful. In: **Advances in Neural Information Processing Systems**, 1996.

SUN, Z.; SONG, Q.; ZHU, X.; SUN, H.; XU, B.; ZHOU, Y. A novel ensemble method for classifying imbalanced data. **Pattern Recognition**, Elsevier, 2015.

TIAN, J.; LI, M.; CHEN, F.; KOU, J. Coevolutionary learning of neural network ensemble for complex classification tasks. **Pattern Recognition**, Elsevier, 2012.

TUMER, K.; GHOSH, J. Analysis of decision boundaries in linearly combined neural classifiers. **Pattern Recognition**, Elsevier, 1996.

UMBAUGH, S. E. **Computer imaging: digital image analysis**, 2005.

VAPNIK, V. **The nature of statistical learning theory**, 2013.

WAHBA, G.; WOLD, S. A completely automatic french curve: Fitting spline functions by cross validation. **Communications in Statistics-Theory and Methods**, Taylor & Francis Group, 1975.

XUE, X.; ZHOU, J.; XU, Y.; ZHU, W.; LI, C. An adaptively fast ensemble empirical mode decomposition method and its applications to rolling element bearing fault diagnosis. **Mechanical Systems and Signal Processing**, Elsevier, 2015.

ZHANG, C.; MA, Y. **Ensemble machine learning**, 2012.

ZHOU, Z.-H.; WU, J.; TANG, W. Ensembling neural networks: many could be better than all. **Artificial intelligence**, Elsevier, 2002.

ZHU, Z.; ONG, Y.-S.; DASH, M. Markov blanket-embedded genetic algorithm for gene selection. **Pattern Recognition**, Elsevier, 2007.

APÊNDICE A - RESULTADOS COMPLETOS

As tabelas a seguir apresentam os resultados completos em relação à variação da largura do *kernel* gaussiano para os algoritmos avaliados. Os melhores resultados estão destacados em cada tabela. Os valores do parâmetro de dissimilaridade podem ser encontrados na Tabela 5.6.

Observou-se, a partir dos resultados, que ao passo que a largura da gaussiana é aumentada, a taxa de erro aumenta significativamente nos métodos propostos independente do método de combinação adotado e nos modelos Perceptron e PB. Isso ocorre devido ao *overfitting* gerado pelo excesso de ajuste da curva gaussiana aos dados de treinamento. As Tabelas A.1, A.2, A.3, A.4, A.5 e A.6 mostram os resultados.

Em relação ao AdaBoost, observou-se que com o aumento da largura da curva gaussiana, o método não conseguiu concluir a convergência para várias bases de dados. Os resultados encontram-se na Tabela A.8. Isso acontece, uma vez que o algoritmo AdaBoost não consegue completar a formação do *ensemble*, devido à restrição implementada no algoritmo de que cada componente aceito não pode apresentar erro superior a 50% (DIETTERICH, 2000a). Por sua vez, o SVM consegue evitar *overfitting* total do modelo, como mostrado na Tabela A.7.

Table A.1: Resultados Completos para o Perceptron *kernel*.

Base	σ				
	0,01	0,1	1	10	100
Sonar	19,32 ± 1,54	16,37 ± 1,27	**15,72 ± 1,50**	18,59 ± 1,43	17,71 ± 1,36
Ionos	9,53 ± 0,84	**6,30 ± 0,78**	14,46 ± 0,99	13,25 ± 0,82	27,4 ± 0,43
Tic Tac	2,00 ± 0,38	**1,38 ± 0,31**	2,24 ± 0,42	1,53 ± 0,40	**0,00 ± 0,00**
Bupa	**38,56 ± 1,28**	39,51 ± 1,50	39,86 ± 1,17	65,07 ± 0,67	97,91 ± 0,38
Pima	**32,90 ± 1,06**	33,80 ± 1,14	36,48 ± 0,74	83,58 ± 0,38	100 ± 0,00
Wine	**21,56 ± 1,16**	21,61 ± 1,12	24,67 ± 0,84	63,16 ± 1,46	98,1 ± 0,50
BreastII	**65,06 ± 0,67**	78,66 ± 0,62	83,06 ± 0,53	88,58 ± 0,44	94,28 ± 0,28
Heart	**39,86 ± 1,32**	42,59 ± 1,44	44,27 ± 1,36	90,41 ± 0,67	100 ± 0,00
Live	**38,97 ± 1,70**	39,75 ± 1,66	39,95 ± 1,18	65,08 ± 0,66	97,91 ± 0,38

Table A.2: Resultados Completos para o Perceptron *kernel* Balanceado.

Base	σ				
	0,01	0,1	1	10	100
Sonar	18,81 ± 1,73	16,54 ± 1,46	**15,67 ± 1,32**	17,78 ± 1,63	31,97 ± 1,88
Ionos	9,34 ± 0,90	**5,83 ± 0,78**	6,50 ± 0,80	13,98 ± 4,87	34,68 ± 0,49
Tic Tac	**0,22 ± 0,19**	0,65 ± 0,21	2,18 ± 0,46	2,82 ± 0,40	**0,00 ± 0,00**
Bupa	**38,03 ± 1,63**	38,94 ± 1,68	40,59 ± 2,19	65,08 ± 0,68	97,91 ± 0,38
Pima	**32,87 ± 1,00**	36,26 ± 1,07	35,41 ± 1,48	83,58 ± 0,38	100 ± 0,00
Wine	21,71 ± 1,58	**20,55 ± 1,43**	36,38 ± 1,34	63,16 ± 1,46	98,1 ± 0,50
BreastII	36,58 ± 2,77	37,31 ± 3,06	**34,23 ± 0,19**	**34,23 ± 0,19**	94,28 ± 0,28
Heart	**40,33 ± 1,73**	42,36 ± 1,58	46,49 ± 2,29	90,41 ± 0,67	100 ± 0,00
Live	**37,95 ± 1,62**	38,47 ± 1,63	40,71 ± 2,23	65,05 ± 0,71	97,91 ± 0,38

Table A.3: Resultados Completos para o m-EBPK.

Base	σ				
	0,01	0,1	1	10	100
Sonar	18,38 ± 1,03	15,03 ± 1,19	**13,78 ± 1,09**	16,67 ± 1,14	33,28 ± 1,19
Ionos	8,80 ± 0,96	**4,96 ± 0,47**	5,71 ± 0,66	6,42 ± 1,58	35,44 ± 0,22
Tic Tac	**0,01 ± 0,02**	0,33 ± 0,16	0,05 ± 0,07	1,67 ± 0,00	**0,00 ± 0,00**
Bupa	36,89 ± 1,35	**36,62 ± 1,22**	38,77 ± 0,76	65,00 ± 0,64	97,91 ± 0,38
Pima	**32,76 ± 0,88**	35,5 ± 0,69	34,89 ± 0,02	83,58 ± 0,38	100 ± 0,00
Wine	22,14 ± 1,04	**20,5 ± 1,60**	39,12 ± 0,95	63,16 ± 1,46	98,1 ± 0,50
BreastII	**33,89 ± 0,24**	33,94 ± 0,20	34,23 ± 0,19	34,23 ± 0,19	94,28 ± 0,28
Heart	**39,47 ± 1,07**	42,3 ± 1,60	45,76 ± 1,10	90,41 ± 0,67	100 ± 0,00
Live	36,95 ± 1,43	**36,82 ± 1,16**	39,34 ± 0,80	65,01 ± 0,62	97,91 ± 0,38

Table A.4: Resultados Completos para o m-EBPKd.

Base	σ				
	0,01	0,1	1	10	100
Sonar	18,27 ± 0,97	14,87 ± 1,07	**14,26 ± 1,05**	16,67 ± 1,08	30,8 ± 1,20
Ionos	8,76 ± 0,91	**4,94 ± 0,47**	5,92 ± 0,51	6,11 ± 1,36	35,39 ± 0,23
Tic Tac	**0,02 ± 0,04**	0,49 ± 0,17	**0,02 ± 0,04**	1,67 ± 0,00	**0,00 ± 0,00**
Bupa	36,86 ± 1,23	**36,61 ± 0,93**	37,73 ± 0,68	64,99 ± 0,64	97,91 ± 0,38
Pima	**32,22 ± 1,02**	36,16 ± 0,64	34,89 ± 0,02	83,58 ± 0,38	100 ± 0,00
Wine	22,00 ± 0,84	**19,08 ± 1,05**	36,3 ± 1,18	63,16 ± 1,46	98,1 ± 0,50
BreastII	**33,83 ± 0,25**	34,23 ± 0,19	34,23 ± 0,19	34,23 ± 0,19	94,28 ± 0,28
Heart	**39,32 ± 1,12**	41,69 ± 1,62	45,39 ± 1,30	90,41 ± 0,67	100 ± 0,00
Live	36,88 ± 1,14	**36,66 ± 0,97**	37,7 ± 0,73	65,00 ± 0,64	97,91 ± 0,38

Table A.5: Resultados Completos para o v-EBPK.

Base	σ				
	0,01	0,1	1	10	100
Sonar	18,74 ± 0,99	15,15 ± 1,25	**13,87 ± 1,00**	17,16 ± 1,24	31,54 ± 0,84
Ionos	8,86 ± 1,00	**5,04 ± 0,53**	5,86 ± 0,58	8,96 ± 3,32	34,74 ± 0,41
Tic Tac	**0,01 ± 0,03**	0,35 ± 0,17	0,10 ± 0,09	1,67 ± 0,06	**0,00 ± 0,00**
Bupa	**37,10 ± 1,15**	37,19 ± 1,20	37,29 ± 0,64	65,1 ± 0,67	97,91 ± 0,38
Pima	**32,40 ± 0,93**	36,2 ± 0,68	34,70 ± 0,49	83,58 ± 0,38	100 ± 0,00
Wine	22,35 ± 1,07	**19,35 ± 0,97**	33,82 ± 1,18	63,16 ± 1,46	98,1 ± 0,50
BreastII	**34,20 ± 0,37**	34,92 ± 1,34	34,23 ± 0,19	34,23 ± 0,19	94,28 ± 0,28
Heart	**39,04 ± 1,04**	41,79 ± 1,33	45,18 ± 1,18	90,41 ± 0,67	100 ± 0,00
Live	**37,03 ± 1,52**	37,25 ± 1,32	37,45 ± 0,66	65,08 ± 0,64	97,91 ± 0,38

Table A.6: Resultados Completos para o v-EBPKd.

Base	σ				
	0,01	0,1	1	10	100
Sonar	18,74 ± 0,96	15,46 ± 1,20	**14,56 ± 1,36**	17,02 ± 1,28	32,46 ± 1,01
Ionos	8,82 ± 0,91	**5,05 ± 0,58**	6,08 ± 0,43	10,17 ± 2,90	34,48 ± 0,39
Tic Tac	0,12 ± 0,11	0,49 ± 0,14	**0,11 ± 0,10**	1,67 ± 0,00	**0,00 ± 0,00**
Bupa	**37,18 ± 1,43**	37,90 ± 1,27	37,55 ± 0,74	65,07 ± 0,65	97,91 ± 0,38
Pima	**32,47 ± 0,95**	36,25 ± 0,81	34,68 ± 0,62	83,58 ± 0,38	100 ± 0,00
Wine	22,38 ± 1,06	**19,69 ± 1,19**	32,93 ± 1,38	63,16 ± 1,46	98,10 ± 0,50
BreastII	34,60 ± 0,55	**34,23 ± 0,19**	**34,23 ± 0,19**	**34,23 ± 0,19**	94,28 ± 0,28
Heart	**39,37 ± 1,16**	41,52 ± 1,49	46,82 ± 1,04	90,41 ± 0,67	100 ± 0,00
Live	**37,12 ± 1,12**	37,54 ± 1,26	37,87 ± 1,09	65,09 ± 0,67	97,91 ± 0,38

Table A.7: Resultados Completos para o SVM.

Base	σ				
	0,01	0,1	1	10	100
Sonar	17,98 ± 2,05	14,45 ± 1,35	**13,40 ± 0,77**	30,39 ± 0,85	46,62 ± 0,00
Ionos	10,11 ± 0,74	8,38 ± 0,74	**7,15 ± 0,40**	34,19 ± 0,01	35,33 ± 0,00
Tic Tac	**0,66 ± 0,03**	0,73 ± 0,10	1,68 ± 0,03	34,66 ± 0,00	34,66 ± 0,00
Bupa	**36,93 ± 1,09**	38,02 ± 0,62	40,45 ± 0,27	40,45 ± 0,27	40,45 ± 0,27
Pima	35,87 ± 0,64	**34,89 ± 0,00**	**34,89 ± 0,00**	**34,89 ± 0,00**	**34,89 ± 0,00**
Wine	**18,46 ± 0,96**	35,41 ± 0,90	39,90 ± 0,00	39,90 ± 0,00	39,90 ± 0,00
BreastII	**33,57 ± 0,21**	33,94 ± 0,20	34,23 ± 0,19	34,23 ± 0,19	34,23 ± 0,19
Heart	**39,93 ± 1,08**	44,14 ± 0,15	44,44 ± 0,00	44,44 ± 0,00	44,44 ± 0,00
Live	**37,22 ± 1,10**	38,02 ± 0,62	40,45 ± 0,27	40,45 ± 0,27	40,45 ± 0,27

Table A.8: Resultados Completos para o AdaBoost.

Base	σ				
	0,01	0,1	1	10	100
Sonar	19,16 ± 1,73	15,25 ± 1,63	**14,99 ± 1,58**	18,65 ± 1,45	19,12 ± 1,48
Ionos	9,15 ± 0,87	**5,91 ± 0,82**	12,1 ± 0,92	12,68 ± 0,98	29,40 ± 0,64
Tic Tac	0,66 ± 0,25	0,79 ± 0,23	**0,63 ± 0,24**	1,03 ± 0,29	0,98 ± 0,23
Bupa	**39,11 ± 1,77**	40,61 ± 2,00	41,62 ± 1,98	49,45 ± 1,06	- ± -
Pima	**33,82 ± 0,96**	34,54 ± 1,05	36,51 ± 0,97	62,87 ± 0,53	- ± -
Wine	**20,12 ± 1,49**	21,74 ± 1,41	28,24 ± 1,53	35,88 ± 1,55	- ± -
BreastII	35,97 ± 0,43	34,53 ± 0,46	**34,48 ± 0,35**	- ± -	- ± -
Heart	**40,38 ± 1,37**	41,20 ± 1,66	41,10 ± 1,55	55,00 ± 0,70	- ± -
Live	**38,92 ± 1,70**	40,88 ± 1,44	41,53 ± 1,61	49,44 ± 1,02	- ± -