UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

COMBINAÇÃO DE CLASSIFICADORES BASEADOS EM FLORESTA DE CAMINHOS ÓTIMOS

SILAS EVANDRO NACHIF FERNANDES

ORIENTADOR: PROF. DR. JOÃO PAULO PAPA

São Carlos – SP

Agosto/2017

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

COMBINAÇÃO DE CLASSIFICADORES BASEADOS EM FLORESTA DE CAMINHOS ÓTIMOS

SILAS EVANDRO NACHIF FERNANDES

Tese apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Doutor em Ciência da Computação, área de concentração: Ciência da Computação
Orientador: Prof. Dr. João Paulo Papa

São Carlos – SP

Agosto/2017

UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Ciência da Computação

Folha de Aprovação

Assinaturas dos membros da comissão examinadora que avaliou e aprovou a Defesa de Tese de Doutorado do candidato Silas Evandro Nachif Fernandes, realizada em 31/08/2017:

Prof. Dr. João Paulo Papa
UFSCar

Prof. Dr. Alexandre Luis Magalhães Levada
UFSCar

Profa. Dra. Heloisa de Arruda Camargo
UFSCar

Prof. Dr. Aparecido Nilceu Marana
UNESP

Prof. Dr. Moacir Antonelli Ponti
ICMC/USP

À minha querida esposa Elen pelo apoio em todos os momentos nessa caminhada,
aos meus pais José e Cecília e meus irmãos pelo apoio e incentivo.

AGRADECIMENTOS

Agradeço primeiramente a Deus, ao meu orientador Prof. Dr. João Paulo Papa, à seção administrativa, à Capes que financiou as minhas pesquisas, ao grupo RECOGNA - UNESP, aos professores do DC - UFSCar, aos meus amigos, à minha família e a minha esposa.

RESUMO

Técnicas de aprendizado de máquina têm sido amplamente estudadas nos últimos anos, principalmente devido ao grande número de aplicações que usam algum mecanismo de inteligência para tomar decisões. Nesse contexto, dentre os diversos estudos sobre técnicas de classificação e como melhorá-las, o campo de combinação de classificadores tem ganhado evidência na literatura. Nessa circunstância, um classificador com destaque crescente na literatura é a técnica denominada de Floresta de Caminhos Ótimos (*Optimum-Path Forest* - OPF), a qual, devido à sua facilidade de utilização, ausência de parâmetros em algumas versões e eficiência na etapa de treinamento de dados, tem se mostrado uma abordagem interessante para problemas de classificação. Por ser uma técnica relativamente recente na literatura e apresentar poucos estudos sobre estratégias de combinação de classificadores, a presente tese visa apresentar um estudo sobre combinação com foco no classificador OPF. A destacar, o estudo com aprendizado dos níveis de confiança baseados em pontuações para o conjunto de treinamento, o qual tem por finalidade aprender amostras mais confiáveis para a etapa de classificação, sendo estas utilizadas em um processo de combinação de classificadores OPF com votação por maioria. Além desse estudo, foi proposta também a combinação de classificadores utilizando a poda de conjunto guiada por otimização metaheurística baseada em quatérnions. Ademais, foi proposta uma extensão da poda de conjunto utilizando classificadores OPFs no contexto de imagens de sensoriamento remoto e, por fim, foi proposto o OPF probabilístico, visto que tradicionalmente o OPF apresenta saídas abstratas apenas. Testes empíricos sobre bases de dados reais e sintéticas evidenciaram que os estudos propostos neste trabalho alcançaram relevante eficácia e eficiência em diversos cenários.

Palavras-chave: Floresta de Caminhos Ótimos, Combinação de Classificadores, Reconhecimento de Padrões

ABSTRACT

Machine learning techniques have been actively pursued in the last years, mainly due to the great number of applications that make use of some sort of intelligent mechanism for decision-making processes. In this context, among the several studies on classification techniques and how to improve them, the ensemble of classifiers has achieved considerable evidence in the literature. In this circumstance, a classifier with significant growth is the technique called Optimum-Path Forest (OPF), which is considerable ease to manipulate, has no parameters in some versions, and it is efficient in the training phase. Since OPF is a relatively new technique in the literature, and we have few studies on ensemble of OPF classifiers only, this work aims to provide a more detailed study in ensemble techniques regarding the OPF classifier. This work has proposed an improved version of OPF, which learns a score-based confidence level for each training sample in order to turn the classification process "smarter" (i.e., more reliable), which is further used in a combination process with majority voting. Furthermore, we also proposed the combination of classifiers using an ensemble pruning strategy driven by meta-heuristics based on quaternions. In addition, we proposed an extension of the ensemble pruning using OPF classifiers in the context of remote sensing images. Finally, the probabilistic OPF was proposed, since the OPF presents only abstract outputs. Experimental results over synthetic and real datasets showed the effectiveness and efficiency of the proposed approaches for classification problems.

Keywords: Optimum-Path Forest, Ensemble Classifiers, Pattern Recognition

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

Capítulo 1

INTRODUÇÃO

Técnicas de reconhecimento de padrões visam o aprendizado de funções de decisão que separam um conjunto de dados em grupos de amostras que partilham propriedades semelhantes. Tal processo de aprendizagem de funções de decisão pode ser elencado por meio de três abordagens principais: (i) supervisionadas, onde se tem informação *a priori* sobre todo o conjunto de treinamento, (ii) semi-supervisionadas, onde é conhecida parte da informação do conjunto de treinamento, e (iii) abordagens não supervisionadas, nas quais não se têm informações sobre as amostras de treinamento.

Técnicas supervisionadas são conhecidas por serem mais eficazes, uma vez que a quantidade de informação disponível sobre as amostras de treinamento permite aprender propriedades específicas sobre cada classe. Um estudo aprofundado sobre o estado da arte de algumas das técnicas mais utilizadas na literatura, tais como Máquinas de Vetores de Suporte (*Support Vector Machines* - SVMs) (CORTES; VAPNIK, 1995), Redes Neurais Artificiais (*Artificial Neural Networks* - ANNs) amplamente discutido por Haykin (HAYKIN, 2007), classificadores Bayesianos, e *k*-vizinhos mais próximos (*k-nearest neighbours* - *k*-NN), dentre outros, são abordados por Duda et al. (DUDA; HART; STORK, 2000).

Embora existam técnicas muito sofisticadas e complexas, é sempre importante ter em mente que novas abordagens podem alcançar melhores resultados. Ideias simples podem melhorar a eficácia de algumas técnicas bem conhecidas. Ahmadlou e Adeli (AHMADLOU; ADELI, 2010), por exemplo, propuseram um modelo melhorado das Redes Neurais Probabilísticas (*Enhanced Probabilistic Neural Networks*) com a ideia de evitar a influência de amostras ruidosas ao calcular a matriz de covariância de cada classe. Ao invés de considerar todas as amostras a partir de uma dada classe com a mesma importância (peso) para calcular a sua matriz de covariância, os autores propuseram considerar apenas a vizinhança de uma determinada amostra de treinamento. Portanto, cada amostra tem a sua própria variância, que será então utilizada para

calcular a matriz de covariância da sua classe. Valores atípicos (*outliers*) apresentam pouca ou nenhuma influência sobre esse processo. Mais tarde, Guo e Boukir (GUO; BOUKIR, 2015) apresentaram uma heurística simples para reduzir a carga computacional de SVMs mantendo uma boa generalização sobre os dados não rotulados (conjunto de teste). A abordagem consiste em extrair um conjunto relevante de candidatos a vetores de suporte mediante métodos de combinação de diferentes subconjuntos de treinamento.

O estudo de sistemas que fazem uso de múltiplos classificadores tem se tornado uma área de grande procura para pesquisadores em reconhecimento de padrões nos últimos anos. Essa busca tem sido motivada pelo interesse em agregar o campo de exploração que outros classificadores podem incluir durante o processo de decisão (DIETTERICH, 2000), dado que, na prática, não é trivial encontrar e, posteriormente, treinar um classificador que consegue generalizar a distribuição dos dados suficientemente bem (CHEN, 2005), uma vez que a maioria das implementações da literatura são tratadas como "caixas-pretas" (ROCHA; PAPA; MEIRA, 2012).

Nesse sentido, uma grande quantidade de métodos para combinação de classificadores tem sido propostos nos últimos anos, possibilitando combinar diferentes estratégias com o intuito de melhorar a eficácia obtida em uma tarefa de classificação qualquer. A literatura tem sugerido que a combinação das decisões dadas por vários classificadores pode levar a uma melhor taxa de reconhecimento do que empregar apenas um classificador, ou até mesmo do que empregar o melhor classificador de uma coleção deles (HANSEN; SALAMON, 1990; KITTLER et al., 1998; KUNCHEVA, 2004). Na verdade, é esperado que cada classificador dessa coleção aprenda diferentes aspectos dos dados. Assim, as deficiências de cada classificador podem ser compensadas pelos pontos positivos dos outros.

Dentre os classificadores mais procurados para compor sistemas de múltipla decisão, podemos citar as Redes Neurais, SVMs e classificadores Bayesianos (DUDA; HART; STORK, 2000), visto que são as técnicas mais conhecidas e utilizadas na literatura. Uma outra técnica que tem sido empregada em várias situações que envolvem classificação de dados é chamada de Floresta de Caminhos Ótimos (*Optimum-Path Forest* - OPF) (PAPA; FALCÃO; SUZUKI, 2009; PAPA et al., 2012) devido, principalmente, à sua facilidade de utilização, ausência de parâmetros em algumas versões e eficiência na etapa de treinamento dos dados. Entretanto, são muito poucos os estudos sobre combinação de classificadores OPF objetivando uma maior eficácia no processo de classificação.

Ponti e Papa (PONTI; PAPA, 2011), por exemplo, mostraram que o treinamento do classificador OPF pode ser mais eficiente e eficaz quando utilizados vários subconjuntos de treinamento disjuntos ao invés do conjunto original. No mesmo ano, Ponti et al. (PONTI; PAPA; LEVADA,

2011) propuseram a combinação de classificadores OPF utilizando Teoria dos Jogos e Campos Aleatórios Markovianos. Entretanto, nenhum estudo sobre técnicas de combinação mais complexas e atuais foi realizado no que diz respeito à esse classificador.

Nesse sentido, esta tese de doutorado objetiva preencher essa lacuna com estudos mais aprofundados sobre o impacto da combinação de classificadores OPF utilizando diferentes técnicas e abordagens, tais como otimização por meta-heurísticas, seleção de classificadores para composição de sistemas com múltiplas decisões, uso de um espaço quadridimensional (conhecido como quatérnions) como estratégia de otimização para seleção de classificadores, bem como a probabilidade de uma amostra pertencer a uma determinada classe por meio da versão probabilística do OPF. De maneira geral, as principais contribuições desse trabalho são:

- colaborar com a literatura específica da área de reconhecimento de padrões, em especial OPF;
- colaborar com o estudo de técnicas de combinação de classificadores OPF;
- modelar a problemática de seleção de classificadores como sendo uma tarefa de otimização baseada em quatérnions;
- investigar a eficácia e a eficiência da estratégia de combinação e seleção de classificadores OPF em imagens de sensoriamento remoto; e
- colaborar com o estudo de uma nova variação do classificador OPF que faz o uso de saídas probabilísticas.

O restante do trabalho está organizado da seguinte forma: o Capítulo 2 apresenta uma revisão bibliográfica sobre técnicas de combinação de classificadores, e no Capítulo 3 é apresentado um referencial teórico do classificador OPF. Os Capítulos 4, 5, 6 e 7 tratam de alguns estudos realizados sobre as técnicas de combinação utilizando o classificador OPF, sendo, respectivamente, um estudo da relevância de uma medida de confiança para aprimorar o processo de classificação final sobre o classificador OPF (Capítulo 4), a combinação de classificadores OPFs em conjunto com uma medida de confiança (Capítulo 5), a combinação de classificadores OPFs utilizando poda de conjunto (Capítulo 6), e uma extensão da combinação de classificadores utilizando poda de conjunto em sensoriamento remoto (Capítulo 7). Além disso, o Capítulo 8 aborda uma nova proposta de classificação utilizando o OPF com saída probabilística. Finalmente, o Capítulo 9 apresentada as conclusões e trabalhos futuros.

Capítulo 2

REVISÃO BIBLIOGRÁFICA

2.1 Combinação de Classificadores

É conhecido que uma das propriedades mais relevantes de um classificador está na sua capacidade de responder ao tentar reconhecer novos padrões, em outras palavras, sua qualidade de generalizar. Implementar uma série de classificadores com diferentes limiares de decisão e, por conseguinte, escolher aqueles com maior capacidade de generalização para futuras tarefas de classificação nem sempre é trivial e, além disso, essa estratégia pode desperdiçar informações ao ignorar determinados classificadores. Uma forma de evitar essa situação seria por meio da combinação das saídas do grupo de classificadores, possibilitando que a decisão final inclua diferentes tipos de conhecimento sobre os dados, melhorando, assim, a capacidade de generalização. Este capítulo introduz algumas conceitos e técnicas de combinação de classificadores que serão abordadas no trabalho.

2.1.1 Contextualização

Para problemas linearmente separáveis ou, que de modo geral apresentem um "bom" comportamento da distribuição dos dados, combinar múltiplos classificadores pode não apresentar ganhos significativos. Pelo contrário, pode ser até mais dispendioso do que apenas um único classificador com boa capacidade de generalização. A combinação de múltiplos classificadores torna-se interessante em situações onde os problemas apresentam certo grau de complexidade, como amostras sobrepostas, grande quantidade de classes, conjuntos com alta dimensionalidade ou dados ruidosos. Nesses casos, o uso de múltiplos classificadores poderia fornecer uma visão do problema utilizando como base diferentes classificadores especializados em diferentes contextos treinados para o mesmo conjunto de dados, ou ainda L-partes distintas do conjunto

de treinamento fornecendo diferentes representações/descrições de um mesmo problema para *L*-classificadores, quer sejam iguais ou diferentes (JAIN; DUIN; MAO, 2000; DUDA; HART; STORK, 2000; KUNCHEVA, 2004).

Atualmente, a literatura contempla, ao menos, duas principais estratégias de combinação de classificadores: fusão e seleção. Na primeira, cada classificador detém um conhecimento completo de todo o espaço de características, sendo que ao final são aplicadas abordagens para combinar essas decisões, como votação por maioria, por exemplo. Para o segundo caso, a seleção fornece a ideia de que cada classificador possui competência acerca de uma sub-região, sendo esse responsável apenas por essa área. Ao final, é selecionado o classificador mais adequado para rotular um determinado padrão de entrada. Outras estratégias utilizam medidas de confidência que, dependendo do padrão de entrada, atribuem pesos aos possíveis classificadores responsáveis por rotular o dado de entrada. Ao final, são utilizadas estratégias de fusão para as saídas (KUNCHEVA, 2004).

Com base nisso, é possível categorizar os sistemas de classificação em três principais arquiteturas (LU, 1996; BIANCHINI; MAGGINI; JAIN, 2013):

- **Cascata**: compreende uma estrutura sequencial onde o resultado de um classificador corresponde à entrada do próximo até que o processo de decisão seja aplicado. Uma desvantagem desse modelo está na dificuldade dos classificadores posteriores identificarem/corrigirem erros cometidos pelos classificadores anteriores.

- **Paralela**: são integrados os resultados de todos os classificadores em um processo de decisão final. Considerada uma das arquiteturas mais utilizadas pela sua simplicidade, o foco dessa abordagem está na escolha mais adequada da metodologia de combinação.

- **Hierárquica**: fornece uma combinação utilizando as duas configurações anteriores. Tem como proposta introduzir a verificação de erros no modelo em cascata.

Um estudo mais detalhado sobre a categorização de múltiplos classificadores pode ser encontrado em Lam (LAM, 2000), o qual divide as técnicas em Condicionais, Hierárquicas (sequenciais), Múltiplas (paralelas) e Híbridas. Resumidamente, de acordo com Lam (LAM, 2000), pode-se defini-las como:

- **Condicional**: um primeiro classificador é selecionado para a tarefa de classificação e, caso falhe, um próximo da lista é selecionado. Essa metodologia depende da escolha

adequada do processo pelo qual os classificadores são rejeitados e, como principal desvantagem, um número elevado de classificadores pode tornar essa metodologia demasiadamente custosa.

- **Hierárquica** (sequencial): nesse modelo, cada classificador empregado tem a finalidade de reduzir o número de possíveis classes que poderiam ser atribuídas ao conjunto de dados de entrada. A estratégia dessa topologia consiste em inserir na fila ordenada os classificadores de acordo com o seu erro, isto é, o primeiro da fila compreende o classificador com maior erro e o último da fila com menor erro.

- **Múltipla** (paralela): como já descrito anteriormente por Lu (LU, 1996), essa topologia executa todos os classificadores em paralelo e suas saídas são então combinadas de acordo com algum critério.

- **Híbrida**: fornece um equilíbrio entre a topologia sequencial e paralela, em que, dentre um conjunto inicial de classificadores, o melhor deles ou um subconjunto ótimo de classificadores são selecionados de acordo com algum critério para compor o processo final. Usualmente conhecido pela literatura como seleção de classificadores (*classifier selection*) ou, para o caso onde temos mais de um conjunto de classificadores selecionados, dá se o nome de poda de conjunto (*ensemble pruning* ou *ensemble selection*) (TSOUMAKAS; PARTALAS; VLAHAVAS, 2009; MARKATOPOULOU; TSOUMAKAS; VLAHAVAS, 2010).

Em geral, a saída dos algoritmos de classificação pode ser categorizada em três níveis (ALANI; DERICHE, 2002): abstrata, *ranking* e confidência. No primeiro nível, os classificadores associam um único rótulo à cada amostra do conjunto de dados, enquanto que no modo *ranking* os possíveis rótulos para uma dada amostra são armazenados em uma fila de prioridades de acordo com algum critério. Já no modo confidência, o classificador calcula alguma métrica que irá refletir na probabilidade de cada um dos rótulos ser atribuído à uma dada amostra.

Assim sendo, baseadas no tipo de saída do classificador, diferentes estratégias de combinação de classificadores têm sido propostas, tais como votação (LAM; SUEN, 1997), votação ponderada (TOMAN et al., 2012), métodos baseados na teoria de evidência de Dempster-Shafer (BI et al., 2007), dentre outras. Alguns trabalhos modelam a combinação de classificadores como sendo um problema de otimização (JIA; WANG; FAN, 2014), sendo que os pesos dos níveis de confidência de cada classificador são determinados por técnicas de otimização meta-heurísticas (JOLY; VERSTRAETE; PANIAGUA, 2014; REYES; MORELL; VENTURA, 2014; BOLOURCHI; MASRI; ALDRAIHEM, 2015). Nesse contexto, Nabavi-Kerizi et al. (NABAVI-KERIZI; ABADI; KABIR, 2010) propuseram a combinação linear de Redes Neurais Artificiais (ADELI; HUNG,

1994; HAYKIN, 1998; ADELI; PARK, 1998; SIDDIQUE; ADELI, 2013) utilizando Otimização por Enxame de Partículas (*Particle Swarm Optimization* - PSO) (KENNEDY; EBERHART, 2001), enquanto que Sheen et al. (SHEEN et al., 2012) utilizaram a Busca Harmônica (*Harmony Search* - HS) (GEEM, 2009; ZENG et al., 2014; SHABBIR; OMENZETTER, 2015; SIDDIQUE; ADELI, 2015b, 2015c, 2015a) para otimizar a composição de um sistema de múltiplos classificadores.

Dentre os diferentes métodos de combinação apresentados na literatura (KUNCHEVA, 2004), podemos citar o *Bagging* (BREIMAN, 1996), *Boosting* (BREIMAN, 1998; KUNCHEVA; SKURICHINA; DUIN, 2002; SCHAPIRE et al., 1998) e *Random Subspaces* (HO, 1998) como os mais utilizados. Lee et al. (LEE et al., 2013) propõem um método de aprendizagem de combinação que utiliza um conjunto de classificadores com base em votação ponderada. Também pode-se destacar a Programação Genética (*Genetic Programming* - GP) (LUNA et al., 2014; PARIS; PEDRINO; NICOLETTI, 2015; RASHIDI; RANJITKAR, 2015), que tem sido utilizada para aprender estratégias de fusão de classificadores, possibilitando uma combinação não-linear de mecanismos que poderiam melhor explorar as saídas de múltiplos classificadores (ACOSTA-MENDOZA et al., 2014).

Com respeito a seleção de classificadores, o trabalho proposto por Diao e Shen (DIAO; SHEN, 2011) fornece uma abordagem de seleção com base na escolha de características utilizando a teoria dos conjuntos difusos em conjunto com a teoria dos conjuntos aproximativos (*fuzzy-rough*) (MENDOZA; VELLASCO; FIGUEIREDO, 2014) e Busca Harmônica. Essa última técnica é utilizada para selecionar um subconjunto mínimo de características visando maximizar a regra *fuzzy-rough*. Mais tarde, Coletta et al. (COLETTA et al., 2015) utilizaram meta-heurísticas para ajustar os parâmetros de um estrutura de combinação para construção de conjuntos de classificadores e agrupamentos.

Em geral, pode-se separar o projeto de Sistemas de Múltiplos Classificadores (*Multiple Classifier Systems* - MCS) de acordo com o estágio onde são aplicados. Caso o estágio ocorra no processo inicial, procura-se definir um conjunto ótimo de classificadores e, ao final, utilizar uma estratégia simples para combinar as saídas. Por conseguinte, há o projeto dos métodos de decisão, em que os classificadores de entrada são imutáveis e, portanto, o objetivo concentra-se em otimizar o processo de decisão das saídas (HO, 2001; PONTI, 2011).

Além disso, algumas observações devem ser consideradas sobre os classificadores a serem combinados. Geralmente, a decomposição de *bias-variância* (GEMAN; BIENENSTOCK; DOURSAT, 1992) é utilizada para entender o comportamento de MCS em relação aos seus erros. Tumer e Ghosh (TUMER; GHOSH, 1996), por exemplo, mostraram que para classificadores correlacionados, o erro do conjunto será igual ao erro médio para os classificadores individuais. Caso

os classificadores sejam estatisticamente independentes, o erro será menor do que o erro para os classificadores individuais. Além disso, a combinação de classificadores com alta variância (comumente associada aos sobre-ajustes das amostras - *overfitting*) pode reduzir a variância final do conjunto, e a combinação de classificadores com baixa variância poderá reduzir as *bias* (geralmente associado aos sob-ajustes das amostras - *underfitting*) (PONTI, 2011).

2.2 Regras de combinação

Algoritmos como *Bagging*, *Boosting* e outros especializados em métodos de combinação (que serão discutidos na Seção 2.3) fornecem as suas próprias regras de combinação, tal como votação por maioria ou votação por maioria utilizando pesos. Contudo, os classificadores podem ser combinados utilizando diferentes regras de combinação. Algumas dessas regras operam somente nos rótulos (nível abstrato), como a votação por maioria, votação por maioria utilizando pesos, espaço de conhecimento comportamental (*Behavior-Knowledge Space* - BKS) e combinação bayesiana (*bayesian combination*). Por outro lado, existem regras que atuam em saídas contínuas (nível de confidência) com suporte à classes fornecidas pelos classificadores, como na abordagem Dempster-Shafer (BI et al., 2007), na regra do produto, soma, média, máximo, mínimo e da mediana, dentre outras que podem ser utilizadas (XU; KRZYZAK; SUEN, 1992; KUNCHEVA, 2004).

Considerando classificadores com saídas abstratas, seja $\mathscr{D} = \{D_1, D_2, \ldots, D_L\}$ um conjunto de L classificadores, e $d_{i,j} \in \{0,1\}$ a decisão do i-ésimo classificador considerando a classe j, para $i = 1, \ldots, L$ e $j = 1, \ldots, K$, onde K denota o número de classes. Cada classificador utiliza como entrada um vetor de características $\mathbf{z} \in \Re^n$, e associa ao mesmo uma classe pertencente ao conjunto $\Omega = \{\omega_1, \omega_2, \ldots, \omega_K\}$, isto é, temos que $D_i : \Re^n \rightarrow \Omega$. Caso o i-ésimo classificador escolha a classe ω_j, então $d_{i,j} = 1$, ou $d_{i,j} = 0$ para o caso contrário. Desta forma, podemos definir $d_{i,j}(\mathbf{z})$ como sendo a decisão do i-ésimo classificador com relação à classe j para a amostra $\mathbf{z}$.

2.2.1 Votação por Maioria

A estratégia de votação das decisões dos classificadores que são considerados complementares pode melhorar a eficácia da classificação de um conjunto em comparação aos aprendizados individuais (MARTÍNEZ-MUÑOZ; HERNÁNDEZ-LOBATO; SUÁREZ, 2009). Dois classificadores são considerados complementares quando os seus erros não são correlacionados. Quando classificadores complementares são combinados, as decisões corretas são amplificadas pelo pro-

cesso de agregação (HANSEN; SALAMON, 1990; KROGH; VEDELSBY, 1995; MARTÍNEZ-MUÑOZ; HERNÁNDEZ-LOBATO; SUÁREZ, 2009). Segundo Kuncheva (KUNCHEVA, 2004), podem ser mencionadas três principais abordagens de votação: unanimidade, simples e pluralidade. A unanimidade ocorre quando todas as saídas de cada classificador são do mesmo rótulo, isto é, todos estão de acordo. O segundo caso estabelece $50\% + 1$ das saídas como sendo o rótulo final para o dado de entrada. Por último, a pluralidade, que também é conhecida como votação por maioria, fornece um rótulo com base na maioria das saídas. Para problemas de duas classes apenas, a votação por maioria coincide com a votação simples.

Assumindo que a saída dos rótulos do classificador D_i pode ser dada como um vetor binário de K dimensões $[d_{i,1},\ldots,d_{i,K}], i = 1,2,\ldots,L$, a votação por maioria computará a decisão da classe ω_j como segue:

$$\omega_j = \arg_j \max \sum_{i=1}^{L} d_{i,j}(\mathbf{z}). \tag{2.1}$$

No nível abstrato, $d_{i,j}(\mathbf{z})$ será somente um elemento não zero, o qual corresponde à classe decidida. Para casos em que os classificadores não forneçam um mesmo padrão de cálculo de acurácia, pesos podem ser atribuídos com a finalidade de favorecer classificadores mais competentes para aquele determinado problema. Reescrevendo a Equação 2.1, a votação por maioria utilizando pesos para a decisão da classe ω_j é dada por:

$$\omega_j = \arg_j \max \sum_{i=1}^{L} w_i d_{i,j}(\mathbf{z}), \tag{2.2}$$

onde w_i é o peso do classificador D_i. É válido notar que, caso w_i seja igual para todos os classificadores, a votação por maioria utilizando pesos apresentará o mesmo comportamento da tradicional votação por maioria (Equação 2.1).

Kuncheva (KUNCHEVA, 2004) destaca que uma provável maneira de obter os pesos w_i para a Equação 2.2 pode ser pelo seguinte método:

$$w_i \propto \log \frac{p_i}{1-p_i}, \tag{2.3}$$

em que p_i representa a acurácia de D_i.

2.2.2 Combinação Bayesiana

A regra de combinação Bayesiana baseia-se na probabilidade *a posteriori* em que os termos são representados pelas saídas de cada classificador individual. No geral, uma amostra $\mathbf{z}$ de entrada será rotulada com uma classe ω_j que maximiza a probabilidade dessa última (BIANCHINI; MAGGINI; JAIN, 2013).

Considere a saída $d_{i,j}(\mathbf{z})$ de L classificadores para uma determinada amostra $\mathbf{z}$ e classe j. Atribui-se a classe $\omega_j, j = 1,2,\ldots,K$, conforme segue:

$$\omega_j = \max_j P(\omega_j|d_{1,j}, d_{2,j}, \ldots, d_{L,j}), \tag{2.4}$$

em que a probabilidade *a posteriori* $P(\omega_j|d_{1,j}, d_{2,j}, \ldots, d_{L,j})$ é definida por:

$$P(\omega_j|d_{1,j}, d_{2,j}, \ldots, d_{L,j}) = \frac{P(d_{1,j}, d_{2,j}, \ldots, d_{L,j}|\omega_j)P(\omega_j)}{P(d_{1,j}, d_{2,j}, \ldots, d_{L,j})}, \tag{2.5}$$

onde $P(\omega_j)$ é a probabilidade *a priori* de ocorrência da classe ω_j, $P(d_{1,j}, d_{2,j}, \ldots, d_{L,j}|\omega_j)$ é a função de densidade de probabilidade conjunta, e o denominador representa a densidade de probabilidade conjunta incondicional, a qual é definida por:

$$P(d_{1,j}, d_{2,j}, \ldots, d_{L,j}) = \sum_{j=1}^{K} P(d_{1,j}, d_{2,j}, \ldots, d_{L,j}|\omega_j)P(\omega_j). \tag{2.6}$$

Para determinar $P(d_{1,j}, d_{2,j}, \ldots, d_{L,j}|\omega_j)$, algumas regras podem ser utilizadas como a da soma, produto e suas derivações como a regra do máximo, mínimo, mediana, dentre outras.

2.2.3 Espaço de Conhecimento Comportamental

Assim como na votação por maioria (Seção 2.2.1) e na combinação bayesiana (Seção 2.2.2) que também utilizam o nível abstrato, o espaço de conhecimento comportamental (*Behavior-Knowledge Space* - BKS) faz uso dessas saídas para construir um espaço de conhecimento acerca das decisões de todos os classificadores sobre cada amostra.

O BKS, proposto inicialmente por Huang e Suen (HUANG; SUEN, 1993), procura estimar a probabilidade *a posteriori* por meio da frequência de cada classe para todo possível conjunto de decisões dos classificadores com base em um conjunto de dados de treinamento. Como resultado, é gerada uma tabela com todas as possíveis combinações das saídas dos classificadores para amostras de um conjunto de validação, e cada célula representa a frequência em que

ocorre uma determinada saída para uma dada classe ω_j em relação a um conjunto de validação. Ao final, o maior valor correspondente, isto é, a classe mais representativa, é selecionada e atribuída (BUNKE; WANG, 1997). Empates são resolvidos arbitrariamente e saídas com todas células vazias são rotuladas por voto majoritário.

Considere como exemplo a Tabela 2.1 para duas classes e três classificadores $D_1(\mathbf{z})$, $D_2(\mathbf{z})$, $D_3(\mathbf{z})$ para um dado padrão de entrada $\mathbf{z} \in \Re^n$, conforme descrito por Ponti (PONTI, 2011). Nesse caso, assuma que $\mathbf{z}$ é classificado pelo conjunto de classificadores D_1, D_2 e D_3 respectivamente com os seguintes rótulos "$1,2,1$". Em seguida, os dados da Tabela 2.1 são, então, consultados e a classe mais representativa é escolhida, isto é, a classe ω_1. Ponti (PONTI, 2011) ainda destaca que esse método não faz suposições sobre a independência, porém apresenta algumas desvantagens em pequenos conjuntos de dados e pode apresentar-se demasiadamente custoso para altas dimensionalidades.

Tabela 2.1: Espaço de conhecimento do método BKS. A coluna com os valores em destaque representa uma consulta para as saídas dos classificadores nos seguintes rótulos: "1,2,1".

$\Omega/\mathscr{D}$	1,1,1	1,1,2	**1,2,1**	1,2,2	2,1,1	2,1,2	2,2,1	2,2,2
ω_1	98	48	**74**	87	52	76	85	3
ω_2	6	86	**15**	93	18	88	93	98

2.2.4 *Decision Templates*

Para classificadores com saídas contínuas (nível de confidência), tais como funções de base radial (*Radial Basis Function* - RBF) e naïve Bayes, dentre outros, a saída dos classificadores são definidas como $d_{i,j} \in [0,1]$, as quais podem ser interpretadas como graus de suporte à classe ω_j ou, em determinadas condições, estimados pela probabilidade *a posteriori* definida por $P(\omega_j|\mathbf{z})$.

Uma forma de se obter os graus de suporte das classes é mediante *Decision Templates* - *DT*. A literatura categoriza esse modelo como Combinadores Treináveis, isto é, são estratégias que necessitam de uma etapa de treinamento para determinados ajustes dos parâmetros ou para a construção de algum template de decisão (KUNCHEVA, 2004). Para isso, faz-se necessário um perfil de decisão (*Decision Profile* - *DP*), o qual é definido como uma matriz $DP(\mathbf{z})_{L\times K}$ organizada por meio de L classificadores e K classes. Cada elemento $d_{i,j} \in [0,1]$ representa o suporte dado pelo i-ésimo classificador para a classe ω_j. A matriz $DP(\mathbf{z})_{L\times K}$ pode ser definida

como:

$$DP(\mathbf{z}) = \begin{bmatrix} d_{1,1}(\mathbf{z}) & d_{1,2}(\mathbf{z}) & \dots & d_{1,K}(\mathbf{z}) \\ \dots & \dots & \dots & \dots \\ d_{i,1}(\mathbf{z}) & d_{i,2}(\mathbf{z}) & \dots & d_{i,K}(\mathbf{z}) \\ \dots & \dots & \dots & \dots \\ d_{L,1}(\mathbf{z}) & d_{L,2}(\mathbf{z}) & \dots & d_{L,K}(\mathbf{z}) \end{bmatrix}. \tag{2.7}$$

Nessa matriz, os valores na coluna j correspondem aos suportes individuais para a classe ω_j considerando todos os classificadores, sendo que os valores da linha i denotam as saídas do classificador D_i para a amostra $\mathbf{z}$.

Os métodos de combinação utilizam essa matriz $DP(z)_{L\times K}$ com o intuito de encontrar um valor de suporte $\mu_j(z)$ que possua o maior poder discriminativo para cada classe. A ideia é comparar o perfil de decisão de um novo padrão de entrada com a sua *decision template* para cada classe usando alguma medida de similaridade como a distância Euclidiana, por exemplo (KUNCHEVA, 2004).

Para treinar esse método de combinação, isto é, observar qual é o perfil de decisão mais comum para cada classe ω_j, o qual é denominado de *decision template* (DT_j), calcula-se para cada classe a média dos perfis de decisão para todas as amostras de ω_j do conjunto de treinamento, como segue:

$$DT_j = \frac{1}{N_j} \sum_{\mathbf{s}\in\omega_j} DP(\mathbf{s}), \tag{2.8}$$

em que N_j representa o número de amostras pertencentes à classe ω_j.

Considere um novo padrão de entrada $\mathbf{x} \in \Re^n$: calcula-se, então, o perfil de decisão $DP(\mathbf{x})$ utilizando as saídas dos conjuntos de classificadores e, por conseguinte, computa-se a similaridade S entre $DP(\mathbf{x})$ e cada DT_j, gerando o suporte μ_j para cada classe:

$$\mu_j(\mathbf{x}) = S(DP(\mathbf{x}), DT_j), \quad j = 1,\dots,K. \tag{2.9}$$

A média de similaridade S, como exemplo a distância Euclidiana quadrada, pode ser definida por:

$$S(DP(\mathbf{x}), DT_j) = 1 - \frac{1}{L\times K} \sum_{i=1}^{L}\sum_{t=1}^{K} [DT_j(i,t) - d_{i,t}(\mathbf{x})]^2. \tag{2.10}$$

Ao final, será atribuído à $\mathbf{x}$ a classe com maior suporte $\mu_j(\mathbf{x})$.

2.2.5 Método Dempster-Shafer

Uma outra forma de investigar a relação das saídas contínuas é mediante a combinação Dempster-Shafer, a qual fornece graus de crença para decidir a classe de um novo padrão de entrada. Dempster formulou a teoria da crença (DEMPSTER, 1967) que, em seguida, foi melhorada por Shafer (SHAFER, 1976), resultando na teoria Dempster-Shafer (D-S) (KUNCHEVA, 2004). Frequentemente aplicada para lidar com incerteza e raciocínio incompleto (GORDON; SHORTLIFFE, 1990), a teoria D-S difere da teoria Bayesiana clássica, pois a primeira pode modelar explicitamente a ausência de informações, enquanto que para a abordagem Bayesiana a ausência de informações implica na mesma probabilidade para todos os eventos (BIANCHINI; MAGGINI; JAIN, 2013).

Considerando a mesma *decision template DT* e *decision profile DP* da Seção 2.2.4, ao invés de usarmos uma medida de similaridade como na Equação 2.10, podemos construir a decisão com base na medida de crença, conforme descreve Kuncheva (KUNCHEVA, 2004):

1. Considere DT_{ij} a i-ésima linha da *decision template* DT_j e a saída do classificador $D_i(\mathbf{x}) = [d_{i,1}(\mathbf{x}), d_{i,2}(\mathbf{x}), \ldots, d_{i,K}(\mathbf{x})]$ (i-ésima linha da $DP(\mathbf{x})$) para o cálculo da proximidade Φ entre DT_{ij} e $D_i(\mathbf{x})$:

$$\Phi_{i,j}(\mathbf{x}) = \frac{(1 + ||DT_{i,j} - D_i(\mathbf{x})||^2)^{-1}}{\sum_{t=1}^{K}(1 + ||DT_{i,t} - D_i(\mathbf{x})||^2)^{-1}}, \tag{2.11}$$

   em que $||\cdot||$ denota a norma da matriz.

2. Com base na Equação 2.11, calcula-se a medida de crença b_j como segue:

$$b_j(D_i(\mathbf{x})) = \frac{\Phi_{i,j}(\mathbf{x}) \prod_{t \neq j}(1 - \Phi_{i,t}(\mathbf{x}))}{1 - \Phi_{i,j}(\mathbf{x})[1 - \prod_{t \neq j}(1 - \Phi_{i,t}(\mathbf{x}))]}, \quad \forall i = 1, \ldots, L, \quad \forall j = 1, \ldots, K. \tag{2.12}$$

3. Por fim, é atribuída a classe com maior grau de suporte $\mu_j(\mathbf{x})$:

$$\omega_j = \arg_j \max \mu_j(\mathbf{x}) = \alpha \prod_{i=1}^{L} b_j(D_i(\mathbf{x})), \quad \forall j = 1, \ldots, K, \tag{2.13}$$

   sendo α uma constante de normalização.

2.2.6 Combinadores Algébricos

Utilizando a matriz de suporte $DP(\mathbf{z})$ (Seção 2.2.4), o suporte geral para cada classe pode ser obtido por meio de $\mu_j(\mathbf{z}) = F[d_{i,j}(\mathbf{z}), \ldots, d_{L,K}(\mathbf{z})]$, onde F representa algumas das funções de combinação como, por exemplo, a regra da média, regra do mínimo, máximo e da mediana, regra do produto e da soma.

Regra da média

Uma primeira abordagem seria calcular a média dos suportes para cada classe e associar o rótulo com maior média. Nesse caso, o suporte μ_j da classe ω_j é dado por:

$$\mu_j(\mathbf{z}) = \frac{1}{L}\sum_{i=1}^{L} d_{i,j}(\mathbf{z}). \tag{2.14}$$

Uma outra forma consiste em adicionar pesos na Equação 2.14 combinados à média com a regra da votação por maioria utilizando pesos. Nesse caso, o suporte μ_j da classe ω_j utilizando L pesos, $w_1, \ldots, w_L$, é dado por:

$$\mu_j(\mathbf{z}) = \frac{1}{L}\sum_{i=1}^{L} w_i d_{i,j}(\mathbf{z}). \tag{2.15}$$

Os pesos para cada classificador podem ser obtidos durante a construção do próprio conjunto de classificadores ou utilizando alguma outra técnica. Uma segunda abordagem considera agora também uma matriz $W'_{L\times K}$ de pesos, ou seja, um peso para cada par classificador-classe. O suporte μ_j é dado por:

$$\mu_j(z) = \frac{1}{L}\sum_{i=1}^{L} W'_{i,j} d_{i,j}(z), \tag{2.16}$$

onde $W'_{i,j}$ denota o peso associado ao i-ésimo classificador para classificar instâncias da classe ω_j.

Ainda nesse contexto, uma terceira abordagem considera uma matriz $W'_{L\times K}$ de pesos calculada para cada classe. Nesse caso, o suporte μ_j para a classe ω_j é obtido por meio de uma combinação linear entre todos os elementos da matriz conforme segue:

$$\mu_j(\mathbf{z}) = \frac{1}{L}\sum_{k=1}^{K}\sum_{i=1}^{L} W'_{i,k,j} d_{i,k}(\mathbf{z}), \tag{2.17}$$

onde $W'_{i,k,j}$ corresponde ao peso (i,k) para a classe ω_j.

Alguns trabalhos têm proposto a utilização de técnicas de otimização por meta-heurísticas para encontrar os pesos que maximizam a taxa de acerto para as abordagens descritas pelas Equações 2.3, 2.15, 2.16, e 2.17. Como dito anteriormente, Nabavi-Kerizi et al. (NABAVI-KERIZI; ABADI; KABIR, 2010) utilizaram a técnica PSO para encontrar tais valores, sendo que Günter and Bunke (GÜNTER; BUNKE, 2004) empregaram Algoritmos Genéticos (GOLDBERG, 1989) para o mesmo fim.

Regra do mínimo, máximo e da mediana

As regras de mínimo, máximo e mediana encontram o suporte de acordo com regra estabelecida para cada classe, e atribuem a decisão final para a classe ω_j de acordo com o suporte μ_j dado por:

$$\mu_j(\mathbf{z}) = \min_{i=1,\ldots,L}\{d_{i,j}(\mathbf{z})\}, \tag{2.18}$$

$$\mu_j(\mathbf{z}) = \max_{i=1,\ldots,L}\{d_{i,j}(\mathbf{z})\}, \tag{2.19}$$

$$\mu_j(\mathbf{z}) = \underset{i=1,\ldots,L}{\text{mediana}}\{d_{i,j}(\mathbf{z})\}, \tag{2.20}$$

respectivamente para mínimo, máximo e mediana, onde a decisão é dada pela escolha do maior suporte.

Regra do produto

A regra do produto escolhe a classe em que o produto dos suportes μ_j para cada classe for maior. A equação pode ser representada por:

$$\omega_j = \arg_j \max \mu_j(\mathbf{z}) = \prod_{i=1}^{L} d_{i,j}(\mathbf{z}). \tag{2.21}$$

Regra da soma

Por fim, a regra da soma é dada por:

$$\omega_j = \arg_j \max \mu_j(\mathbf{z}) = \sum_{i=1}^{L} d_{i,j}(\mathbf{z}). \tag{2.22}$$

2.3 Algoritmos de combinação

Como já mencionado, há diferentes algoritmos que propõem combinar os classificadores. Os principais são *Bagging* (BREIMAN, 1996), *Boosting* (BREIMAN, 1998; KUNCHEVA; SKURICHINA; DUIN, 2002; SCHAPIRE et al., 1998), *AdaBoost* (FREUND; SCHAPIRE, 1999), Empilhamento (*Stacking*) (WOLPERT, 1992), Mistura de Especialistas (*Mixture of Experts*) (JACOBS et al., 1991) e Subespaços Aleatórios (*Random Subspaces*) (HO, 1998). As próximas seções tratam de descrever alguns deles de maneira mais detalhada.

2.3.1 *Bagging*

A técnica *Bagging*, acrônimo para *Bootstrap AGGregatING*, gera diferentes classificadores selecionando aleatoriamente subconjuntos de amostras para treinar os classificadores-base. As decisões dos classificadores são combinadas pela votação por maioria ou por meio da média (KUNCHEVA, 2004). O método é descrito no Algoritmo 1.

Algoritmo 1 – BAGGING

ENTRADA: O tamanho L do conjunto de classificadores, e conjunto de treinamento $\mathscr{Z}$ de tamanho N.

1. **Para** $i = 1, \ldots, L$, **Faça**
2. $\quad$ $\mathscr{Z}_i \leftarrow N$ *amostragens de* $\mathscr{Z}$ *com reposição (bootstrap).*
3. $\quad$ *Treinar classificador* D_i *utilizando* $\mathscr{Z}_i$.
4. **Para** *cada novo padrão de entrada* $\boldsymbol{x}$ **Faça**
5. $\quad$ *Classificar* $\boldsymbol{x}$ *utilizando* $D_i, i = 1, \ldots, D_L$
6. $\quad$ **Se** *as saídas forem contínuas,* **Então**
7. $\quad\quad$ *Calcular a média das decisões de* $D_i, i = 1, \ldots, D_L$
8. $\quad$ **Caso contrário, Se** *as saídas forem rótulos de classes,* **Então**
9. $\quad\quad$ *Computar o voto da maioria utilizando* $D_i, i = 1, \ldots, D_L$

É válido ressaltar que a eficiência desse modelo depende do classificador ser considerado "instável", isto é, pequenas alterações no conjunto de entrada para treinamento provocam diferenças significativas nas saídas durante uma etapa de classificação, como ocorre nos classificadores por Árvores de Decisão ou Redes Neurais (BREIMAN, 1996). Se essa condição não for satisfeita, ou seja, se o classificador for considerado estável como ocorre para o algoritmo dos *k*-vizinhos mais próximos, as saídas dos classificadores poderão apresentar um comportamento muito próximo, reduzindo qualquer possível ganho adicional em construir e treinar classificadores-base (KUNCHEVA, 2004; PONTI, 2011). Nesse caso, quando classificadores apresentam baixo *bias* (erros) e alta variância, o método *bagging* ajuda a reduzir a variância final (FUMERA; FABIO; ALESSANDRA, 2008).

2.3.2 *AdaBoost*

A técnica *AdaBoost* (*Adaptive Boosting*) (FREUND; SCHAPIRE, 1996) é fundamentada no modelo *Boosting* (SCHAPIRE, 1990), sendo que esse último também utiliza parte das amostras para treinar os classificadores como no modelo *Bagging*, porém com a diferença de que a técnica *AdaBoost* seleciona as amostras consideradas mais difíceis de serem classificadas para compor o conjunto de treinamento. O *AdaBoost*, considerado o primeiro algoritmo *Boosting* prático, apresenta como ideia combinar classificadores considerados "fracos" para formar um classificador-base "robusto".

Diferente do método *Bagging* que trabalha em paralelo, o *AdaBoost* representa um método interativo sequencial, ou seja, classificadores são treinados sequencialmente em cada rodada. Ao final de cada uma delas, os padrões que não foram classificadores corretamente recebem pesos mais significativos para que o seu erro, na próxima roda, possa ser melhorado pelo classificador na sequência (PONTI, 2011). A abordagem tradicional, utilizando apenas duas classes, é apresentada no Algoritmo 2.

Para cada iteração i, é criado um conjunto de treinamento $\mathscr{Z}_i$ amostrado de $\mathscr{Z}$ de acordo com o peso w_i inicial. Em seguida, treina o classificador-base D_i sobre $\mathscr{Z}_i$ e classifica com o conjunto original $\mathscr{Z}$. Após isso, calcula-se a taxa de erro para esse classificador-base ε_i e, caso esse erro seja maior ou igual que 50%, um outro classificador-base é treinado ou o algoritmo é encerrado. Caso $\varepsilon_i < 0.5$, a iteração continua e a "importância" do classificador D_i é calculada (ς_i) e o seu peso é atualizado de forma a aumentar aqueles que são relacionados aos dados incorretamente classificados. Ao final de L rodadas, o algoritmo combina todas as hipóteses intermediárias D_i consideradas "fracas" em uma hipótese final "forte" D_{final}. Na abordagem tradicional proposta por Schapire (SCHAPIRE, 1990), a atribuição do rótulo para um novo padrão

de entrada é determinada pela votação ponderada das L hipóteses fracas considerando a sua importância ς.

Algoritmo 2 – ADABOOST

ENTRADA: O tamanho L do conjunto de classificadores, conjunto de treinamento $\mathscr{X}$ de tamanho N rotulado, classe $\omega_i \in \{-1,+1\}$, e pesos w_i inicializados com distribuição uniforme, ou seja, $w_i = \frac{1}{N}, i = 1,\ldots,L$.

1. **Para** $i = 1,\ldots,L$, **Faça**
2. *Treina o classificador D_i utilizando a distribuição w_i.*
3. *Calcula o erro* $\varepsilon_i \leftarrow P_{w_i}(D_i(z) \neq \omega)$
4. **Se** $\varepsilon_i \geq 0.5$, **Então**
5. *Encerra o algoritmo ou treina outro classificador-base.*
6. *Calcula a relevância do classificador D_i utilizando* $\varsigma_i \leftarrow \frac{1}{2}\ln\left(\frac{1-\varepsilon_i}{\varepsilon_i}\right)$
7. *Atualiza os pesos* $w_{i+1} \leftarrow \frac{w_{i+1}}{R_i}\exp^{(-\varsigma_i \omega_i D_i(z_i))}$,
8. *em que R_i é fator de normalização utilizado para assegurar que o somatório dos pesos seja 1.*
9. **Para** *cada novo padrão* $\boldsymbol{x}$ *de entrada* **Faça**
10. $D_{final}(x) \leftarrow sign\left(\sum_{i=1}^{L} \varsigma_i D_i(x)\right)$

A abordagem *Boosting* e, consequentemente a *AdaBoost*, ganharam destaque na literatura por serem consideradas técnicas relativamente simples de serem implementadas e com ganhos significativos para construção de múltiplos classificadores (MARTÍNEZ-MUÑOZ; HERNÁNDEZ-LOBATO; SUÁREZ, 2009; QUINLAN, 1996; DIETTERICH, 2000; BAUER; KOHAVI, 1999). Entretanto, em alguns problemas de classificação, principalmente aqueles com alta taxa de ruídos, esses algoritmos podem apresentar um desempenho inferior, prejudicando a generalização dos dados (QUINLAN, 1996; BAUER; KOHAVI, 1999; DIETTERICH, 2000; CARUANA; NICULESCU-MIZIL, 2006; MARTÍNEZ-MUÑOZ; HERNÁNDEZ-LOBATO; SUÁREZ, 2009). Além disso, em determinados casos quando se tem um grande conjunto de classificadores combinados, pode ocorrer o supertreinamento dos dados, isto é, quando um modelo estatístico se ajusta ao erro, prejudicando a generalização para diferentes dados (RÄTSCH; ONODA; MÜLLER, 2001; MARTÍNEZ-MUÑOZ; HERNÁNDEZ-LOBATO; SUÁREZ, 2009). Apesar disso, o método *Boosting* e suas variações têm se apresentado promissores para muitas aplicações.

2.3.3 Subespaços Aleatórios

O método de subespaços aleatórios (*Random Subspaces Method* - RSM) (HO, 1998) cria vários classificadores usando diferentes espaços de características para o seu treinamento. A ideia é que diferentes subespaços forneçam diferentes formas de tratar os dados. A ideia consiste em criar classificadores diversificados que são complementares. Evidências experimentais mostraram que, para dados com grande quantidade de características redundantes, o RSM tem se comportado de maneira eficaz e eficiente (ZHOU, 2012). É válido notar que esse método é melhor explorado quando há uma grande quantidade de dados a serem considerados.

O RSM é semelhante ao *Bagging*, porém ao invés de construir classificadores selecionando subconjuntos, o RSM executa um tipo de amostragem das características sem substituição, visto que seria desnecessário incluir a mesma característica mais de uma vez. Em geral, a abordagem RSM, conforme apresentado no Algoritmo 3, constrói um classificador utilizando um vetor de características menor do que o espaço dimensional original fornecido por $\mathscr{Z}$ utilizando t componentes obtidos randomicamente.

Algoritmo 3 – RANDOM SUBSPACES

ENTRADA: O tamanho L do conjunto de classificadores, conjunto de treinamento $\mathscr{Z}$ de tamanho N, e T o número de características, sendo t_i a quantidade de características para cada classificador D_i em que $t_i < T, \quad i = 1,\ldots,L$.

1. **Para** $i = 1,\ldots,L$, **Faça**
2. $\quad$ *$\mathscr{Z}_i \leftarrow (\mathscr{Z}, t_i)$ para t_i características escolhidas aleatoriamente de $\mathscr{Z}$*
3. $\quad$ *Treina o classificador D_i sobre $\mathscr{Z}_i$.*
4. **Para** *cada novo padrão de entrada* $\boldsymbol{x}$ **Faça**
5. $\quad$ *Classificar $\boldsymbol{x}$ utilizando $D_i, i = 1,\ldots,D_L$*
6. $\quad$ **Se** *as saídas forem contínuas,* **Então**
7. $\quad\quad$ *Calcular a média das decisões de $D_i, i = 1,\ldots,D_L$*
8. $\quad$ **Caso contrário, Se** *as saídas forem rótulos de classes,* **Então**
9. $\quad\quad$ *Computar o voto da maioria utilizando $D_i, i = 1,\ldots,D_L$*

Capítulo 3

CLASSIFICAÇÃO POR FLORESTA DE CAMINHOS ÓTIMOS

Este capítulo tem por objetivo apresentar a teoria de classificadores OPF para o modelo de aprendizado supervisionado. Embora a biblioteca do classificador OPF [1] apresente ambas versões supervisionadas e não supervisionadas, este trabalho concentrou-se apenas no modelo supervisionado, o qual foi utilizado para o desenvolvimento da abordagem proposta nesse trabalho. Atualmente, existem duas versões distintas do OPF para aprendizado supervisionado: (i) uma que faz o uso de um grafo completo (PAPA; FALCÃO; SUZUKI, 2009; PAPA et al., 2012), e (ii) outra versão que utiliza um grafo k-vizinhos mais próximos (PAPA; FALCÃO, 2008, 2009). Nas próximas seções, serão apresentadas ambas técnicas.

3.1 OPF com grafo completo

O classificador OPF com grafo completo usa uma generalização do algoritmo de Dijkstra (DIJKSTRA, 1959) para múltiplas fontes em conjunto com uma função de custo de caminho. Seus autores demonstraram que o classificador OPF tem apresentado resultados interessantes em termos de eficiência e eficácia, sendo em alguns casos comparáveis aos obtidos pelas Máquinas de Vetores de Suporte, mas, geralmente, mais rápido para a etapa de treinamento, pois a técnica OPF com grafo completo não necessita de parâmetros e seu treinamento leva $\theta(n^2)$, onde n representa o tamanho do conjunto dos dados para treinamento[2] .

O classificador OPF modela o problema de reconhecimento de padrões como sendo um problema de particionamento em um grafo induzido pelo conjunto de dados. Cada amostra da

[1] https://github.com/jppbsi/LibOPF

[2] Na Seção 3.1 todas as referências ao classificador OPF dizem respeito à sua versão com grafo completo.

base de dados, a qual é representada pelo seu vetor de características, é tratada como sendo o nó de um grafo completo, sendo que as arestas entre elas são ponderadas pela distância entre seus respectivos vetores de características. Em seguida, amostras *protótipos* de cada classe são escolhidas e competem entre si com intuito de conquistar as demais amostras do conjunto de dados. Após esse processo de competição, cada protótipo será a raiz de uma árvore de caminhos ótimos, a qual contém as amostras mais fortemente conexas à este protótipo. A coleção dessas árvores nos remete a uma floresta de caminhos ótimos, que dá o nome ao referido classificador. Segue, abaixo, uma fundamentação teórica do mesmo.

3.1.1 Etapa de treinamento

Considere $\mathscr{Z}$ uma base de dados λ-rotulada e $\mathscr{Z}_1$ e $\mathscr{Z}_2$ os conjuntos de treinamento e teste, respectivamente, com $|\mathscr{Z}_1|$ e $|\mathscr{Z}_2|$ amostras, tal que $\mathscr{Z} = \mathscr{Z}_1 \cup \mathscr{Z}_2$. Adicionalmente, considere $\mathbf{s} \in \mathscr{Z}$ uma amostra n-dimensional e $d(\mathbf{s}, \mathbf{v})$ uma função que calcula a distância entre duas amostras $\mathbf{s}$ e $\mathbf{v}$, sendo $\mathbf{v} \in \mathscr{Z}$. Considere $\mathscr{G}^{tr} = (\mathscr{Z}_1, \mathscr{A})$ um grafo derivado do conjunto de treinamento tal que cada nó $\mathbf{v} \in \mathscr{Z}_1$ encontra-se conectado com todos os outros nós em $\mathscr{Z}_1 \backslash \{\mathbf{v}\}$, ou seja, $\mathscr{A}$ define uma relação de adjacência conhecida como **grafo completo** (a Figura 3.1a apresenta o grafo de treinamento), em que os arcos são ponderados pela função $d(\cdot, \cdot)$. Por conseguinte, um caminho π_s define uma sequência de nós adjacentes e distintos em $\mathscr{G}^{tr}$ com término no nó $\mathbf{s} \in \mathscr{Z}_1$. Observe que um **caminho trivial** é denotado por $\langle s \rangle$, isto é, um caminho composto por uma única amostra.

Considere $f(\pi_s)$ uma função de custo de caminho que atribui um valor real e positivo para um determinado caminho π_s, e $\mathscr{S}$ como sendo um conjunto de nós protótipos. A grosso modo, a técnica OPF visa resolver o seguinte problema de otimização:

$$\min f(\pi_s), \ \forall \, \mathbf{s} \in \mathscr{Z}_1. \tag{3.1}$$

A única regra para resolver a Equação 3.1 implica que todos os caminhos devem estar associados a $\mathscr{S}$. Portanto, duas considerações são levantadas: como calcular $\mathscr{S}$ (heurística para definir as amostras protótipos) e $f(\pi)$ (função de custo de caminho).

Visto que os protótipos desempenham um papel importante no processo de conquista, Papa et al. (PAPA; FALCÃO; SUZUKI, 2009) propuseram selecionar os protótipos das regiões que apresentam maior probabilidade de ocorrerem erros de classificação, ou seja, na fronteira entre amostras de classes diferentes. De fato, os protótipos são escolhidos com base na menor distância entre amostras de classes diferentes, as quais podem ser encontradas por meio de uma Árvore de Espalhamento Mínima (*Minimum Spanning Tree* - MST) sobre $\mathscr{G}^{tr}$. Além disso,

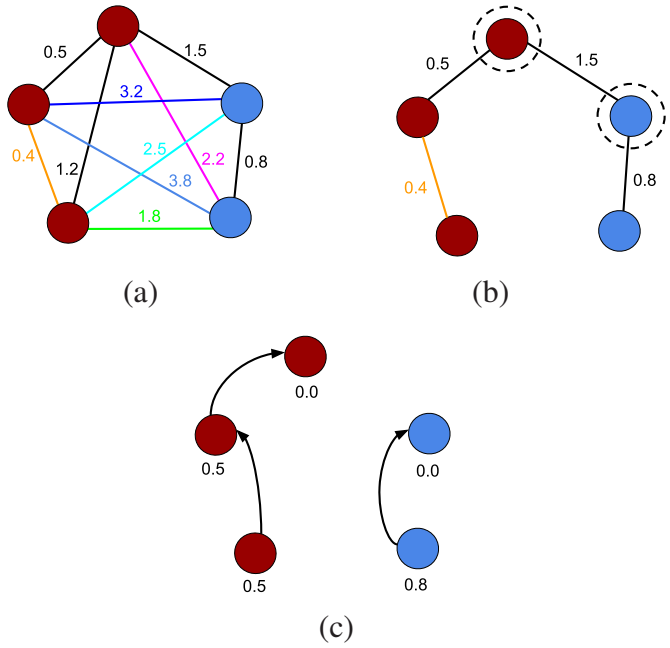


Figura 3.1: Ilustração do procedimento utilizado no OPF: (a) um grafo de treinamento com duas classes (rótulo vermelho e rótulo azul) e arestas ponderadas, (b) uma MST com protótipos destacados, e (c) uma floresta de caminhos ótimos gerada durante a fase de treinamento com os custos sobre os nós (observe que os protótipos têm custo zero).

quando todos os arcos ponderados forem diferentes uns dos outros, a MST garante que a etapa de treinamento da técnica OPF ocorrerá sem erros (ALLÈNE et al., 2010). A Figura 3.1b apresenta uma MST com os protótipos destacados.

Finalmente, com respeito à função de custo de caminho, o algoritmo OPF pode ser utilizado com qualquer função de valor de caminho suave (FALCÃO; STOLFI; LOTUFO, 2004). Basicamente, a função f_{max} consiste em calcular a distância máxima utilizando as arestas ponderadas pelo caminho, definida conforme segue:

$$\begin{aligned} f_{max}(\langle s \rangle) &= \begin{cases} 0 & \text{se } \mathbf{s} \in \mathscr{S} \\ +\infty & \text{caso contrário,} \end{cases} \\ f_{max}(\pi_s \cdot (\mathbf{s},\mathbf{t})) &= \max\{f_{max}(\pi_s), d(\mathbf{s},\mathbf{t})\}, \end{aligned} \tag{3.2}$$

em que $\pi_s \cdot (\mathbf{s},\mathbf{t})$ representa a concatenação entre o caminho π_s e arco $(\mathbf{s},\mathbf{t}) \in \mathscr{A}$ e $f_{max}(\pi_s)$ calcula a distância máxima entre amostras adjacentes de π_s, quando π_s não é um caminho trivial. Em suma, utilizando a Equação 3.2 para toda amostra $\mathbf{s} \in \mathscr{Z}_1$, obtém-se uma coleção de Árvores de Caminhos Ótimos (*Optimum-Path Trees* - OPTs) conduzidas por intermédio de $\mathscr{S}$, de modo que a coleção de árvores forneça uma floresta de caminhos ótimos. Portanto, a etapa de treinamento do classificador OPF objetiva resolver a Equação 3.2 a fim de construir a floresta

de caminhos ótimos, como apresentado na Figura 3.1c. Esse procedimento pode ser descrito pelo Algoritmo 4.

Algoritmo 4 – OPF COM GRAFO COMPLETO - ALGORITMO DE TREINAMENTO

ENTRADA:	Um conjunto de treinamento $\mathscr{Z}_1$ λ-rotulado e uma função de distância d.
SAÍDA:	Floresta de caminhos ótimos P, mapa de rótulos L, mapa de custos C e um conjunto ordenado $\hat{\mathscr{Z}}_1$.
ESTRUTURAS AUXILIARES:	Fila de prioridade Q, conjunto $\mathscr{S}$ de protótipos e variável de custo *cst*

1. *Define* $\hat{\mathscr{Z}}_1 \longleftarrow \emptyset$ *e computa o conjunto de protótipos* $\mathscr{S} \subset \mathscr{Z}_1$ *por meio da MST.*
2. **Para todo** $\boldsymbol{s} \in \mathscr{Z}_1 \setminus \mathscr{S}$, **Faça**
3. $\quad C(\boldsymbol{s}) \leftarrow +\infty$.
4. **Para todo** $\boldsymbol{s} \in \mathscr{S}$, **Faça**
5. $\quad C(\boldsymbol{s}) \leftarrow 0$, $P(\boldsymbol{s}) \leftarrow nil$, $L(\boldsymbol{s}) \longleftarrow \lambda(\boldsymbol{s})$ *e insere* $\boldsymbol{s}$ *em* Q.
6. **Enquanto** $Q \neq \emptyset$, **Faça**
7. $\quad$ *Remova de* Q *uma amostra* $\boldsymbol{s}$ *tal que* $C(\boldsymbol{s})$ *é mínimo.*
8. $\quad$ *Insere* $\boldsymbol{s}$ *em* $\hat{\mathscr{Z}}_1$
9. $\quad$ **Para todo** $\boldsymbol{v} \in \mathscr{Z}_1$ *tal que* $C(\boldsymbol{v}) > C(\boldsymbol{s})$, **Faça**
10. $\quad\quad$ *Computa* $cst \leftarrow \max\{C(\boldsymbol{s}), d(\boldsymbol{s},\boldsymbol{v})\}$.
11. $\quad\quad$ **Se** $cst < C(\boldsymbol{v})$, **Então**
12. $\quad\quad\quad$ **Se** $C(\boldsymbol{v}) \neq +\infty$, **Então**
13. $\quad\quad\quad\quad$ *Remova* $\boldsymbol{v}$ *de* Q.
14. $\quad\quad\quad$ $P(\boldsymbol{v}) \leftarrow \boldsymbol{s}$, $L(\boldsymbol{v}) \leftarrow \lambda(\boldsymbol{s})$, $C(\boldsymbol{v}) \leftarrow cst$.
15. $\quad\quad\quad$ *Insere* $\boldsymbol{v}$ *em* Q.
16. **Retorna** $[P, L, C, \hat{\mathscr{Z}}_1]$

A Linha 1 computa o conjunto de protótipos, conforme Figura 3.1b, e, na sequência, as Linhas $2-5$ inicializam os mapas de custo e de rótulos e insere os protótipos em Q, respectivamente, onde $C(\mathbf{s})$[3] representa o custo da amostra $\mathbf{s}$, $P(\mathbf{s})$ denota seu predecessor na floresta de caminhos ótimos, $L(\mathbf{s})$ compreende o rótulo da amostra $\mathbf{s}$ (a função $\lambda(\cdot)$ na linha 5 aloca o rótulo verdadeiro para cada amostra de treinamento), e Q representa uma fila de prioridade baseada no conjunto de entrada e no custo de cada amostra.

O laço principal das Linhas $6-15$ compreende o processo de competição do classificador OPF. A cada iteração, uma amostra $\mathbf{s}$ é removida da fila de prioridade tal que o seu custo seja

[3] O mapa de custo C armazena o caminho ótimo de cada amostra de treinamento

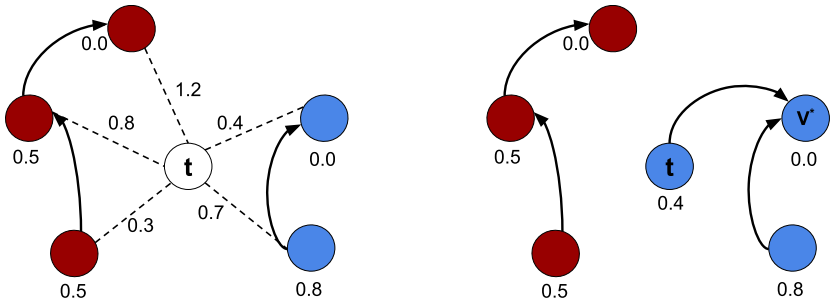
mínimo (Linha 7). O laço interno (Linhas 9 – 15) avalia todos os vizinhos de **s** com o objetivo de conquistá-los, ao passo que a Linha 10 é responsável por calcular f_{max} conforme descrito pela Equação 3.2. Quando uma amostra **v** é conquistada por **s** (Linhas 11 – 15), o seu predecessor, o mapa de rótulos, e o mapa de custos (Linha 14) de **v** são atualizados, bem como **v** recebe o rótulo da amostra que o conquistou. Note que $C(\mathbf{v}) > C(\mathbf{s})$ na Linha 9 é falso quando **v** for removido de Q e, adiante na Linha 12, $C(\mathbf{v}) \neq +\infty$ é verdadeiro somente quando $\mathbf{v} \in Q$.

3.1.2 Etapa de classificação

O próximo passo remete à etapa de teste, onde cada amostra $\mathbf{t} \in \mathscr{Z}_2$ é classificada individualmente. Para isso, a amostra **t** é conectada à todos os nós de treinamento de uma floresta de caminhos ótimos construída durante a etapa de treinamento (Figura 3.2a) e, em seguida, o nó $\mathbf{v}^* \in \mathscr{Z}_1$ que conquistou **t** é avaliado, conforme a equação:

$$C(\mathbf{t}) = \arg\min_{\mathbf{v} \in \mathscr{Z}_1} \max\{C(\mathbf{v}), d(\mathbf{v}, \mathbf{t})\}. \tag{3.3}$$

O procedimento de classificação atribui $L(\mathbf{t}) = \lambda(\mathbf{v}^*)$, como representado na Figura 3.2b. Portanto, o objetivo da etapa de teste consiste em encontrar o nó de treinamento **v** que minimiza $C(\mathbf{t})$.



(a) (b)

Figura 3.2: Ilustração do procedimento de classificação do OPF onde: (a) a amostra t está conectada com todos os nós de treinamento, e (b) t é conquistado por v*, recebendo, assim, o rótulo "azul".

Cabe destacar que, conforme o exemplo apresentado na Figura 3.2, embora **t** encontra-se posicionada mais próxima da classe "vermelha", ela acaba sendo rotulada por outra classe, o que enfatiza que OPF não é um classificador baseado em distância, mas sim um classificador que usa a influência da conectividade entre as amostras. A técnica OPF com grafo completo degenera-se para um classificador de vizinhos mais próximos somente quando todas as amostras de treinamento forem protótipos. Na verdade, tal situação é consideravelmente difícil de

ocorrer, indicando, assim, um elevado grau de sobreposição entre as amostras.

O Algoritmo 5 implementa o procedimento de classificação do OPF. O laço principal (Linhas 1 - 8) executa a classificação de todos as amostras de $\mathscr{Z}_2$. O laço mais interno (Linhas 3 - 8) visita cada amostra $\mathbf{k}_{i+1} \in \hat{\mathscr{Z}}_1, i = 1,2,\cdots,|\hat{\mathscr{Z}}_1|-1$ até um caminho ótimo $\pi_{\mathbf{k}_{i+1}} \cdot \langle \mathbf{k}_{i+1}, \mathbf{t} \rangle$ for encontrado. No pior caso, o algoritmo visita todas as amostras em $\hat{\mathscr{Z}}_1$. A Linha 4 avalia $f_{max}(\pi_{\mathbf{k}_{i+1}} \cdot \langle \mathbf{k}_{i+1}, \mathbf{t} \rangle)$, e as linhas 6 - 7 atualizam o custo, o rótulo e o predecessor de $\mathbf{t}$ quando o custo do caminho $\pi_{\mathbf{k}_{i+1}} \cdot \langle \mathbf{k}_{i+1}, \mathbf{t} \rangle$ for melhor do que o custo do caminho atual $\pi_{\mathbf{t}}$.

Algoritmo 5 – OPF COM GRAFO COMPLETO - ALGORITMO DE CLASSIFICAÇÃO

ENTRADA:	Classificador $[P, C, L, \hat{\mathscr{Z}}_1]$ e um conjunto de teste $\mathscr{Z}_2$.
SAÍDA:	Mapa de rótulos $\hat{L}$, mapa de custos $\hat{C}$ e um mapa de predecessores $\hat{P}$ definido por $\mathscr{Z}_2$.
ESTRUTURAS AUXILIARES:	Variáveis de custo *cst* e *mincst*.

1. **Para todo** $t \in \mathscr{Z}_2$, **Faça**
2. $\quad i \longleftarrow 0$, $mincost \longleftarrow max\{C(\boldsymbol{k}_i), d(\boldsymbol{k}_i, \boldsymbol{t})\}$, $\hat{L}(\boldsymbol{t}) \longleftarrow L(\boldsymbol{k}_i)$, $\hat{P}(\boldsymbol{t}) \longleftarrow \boldsymbol{k}_i$ *e* $\hat{C}(\boldsymbol{t}) \longleftarrow mincost$.
3. $\quad$ **Enquanto** $i < |\hat{\mathscr{Z}}_1|$ *e* $mincost > C(\boldsymbol{k}_{i+1})$, **Faça**
4. $\quad\quad$ *Calcula* $cst \longleftarrow max\{C(\boldsymbol{k}_{i+1}), d(\boldsymbol{k}_{i+1}, \boldsymbol{t})\}$.
5. $\quad\quad$ **Se** $cst < mincost$, **Então**
6. $\quad\quad\quad$ $mincost \longleftarrow cst$ *e* $\hat{C}(\boldsymbol{t}) \longleftarrow mincost$.
7. $\quad\quad\quad$ $\hat{L}(\boldsymbol{t}) \longleftarrow L(\boldsymbol{k}_{i+1})$ *e* $\hat{P}(\boldsymbol{t}) \longleftarrow \boldsymbol{k}_{i+1}$.
8. $\quad\quad$ $i \longleftarrow i+1$.
9. **Retorna** $[\hat{P}, \hat{L}, \hat{C}]$

3.2 OPF com grafo *k*-vizinhos mais próximos

Uma outra abordagem para o classificador OPF é por meio do grafo *k*-vizinhos mais próximos (*k*-nn) (PAPA; FALCÃO, 2008, 2009), ao invés do grafo completo como descrito anteriormente. A fim de lidar com essa relação de adjacência, dois princípios devem ser alterados: a função de custo de caminho e a heurística para definir as amostras protótipos. Uma estratégia utilizada para estimar os protótipos em um grafo com adjacência *k*-nn, conforme apresentado por Papa e Falcão (PAPA; FALCÃO, 2008), usa as regiões mais densas, isto é, eleger protótipos próximos aos centros dos agrupamentos. Portanto, OPF com grafo *k*-nn pode ser interpretado como uma versão estendida do OPF com grafo completo (Equação 3.1), visto que visa maximizar o custo de cada amostra, conforme descrito a seguir:

$$\max f(\pi_s),\ \forall\ \mathbf{s} \in \mathscr{Z}_1. \tag{3.4}$$

Uma das principais diferenças da versão OPF com grafo k-nn diz respeito aos nós serem ponderados, visto que a versão do OPF com grafo completo, conforme apresentada anteriormente, utiliza pesos somente nos arcos entre amostras. Redefinindo os grafos de treinamento e teste como: $\mathscr{G}^{tr} = (\mathscr{Z}_1, \mathscr{A}_k)$, e $\mathscr{G}^{ts} = (\mathscr{Z}_2, \mathscr{A}_k)$, onde $\mathscr{A}_k$ representa uma relação de adjacência baseada em k-vizinhos mais próximos. Adicionalmente, considere $\rho(\mathbf{s})$ como uma função de densidade de probabilidade (*probability density function* - pdf) de uma determinada amostra $\mathbf{s} \in \mathscr{Z}_1$ calculada conforme segue:

$$\rho(\mathbf{s}) = \frac{1}{\sqrt{2\pi\sigma^2}k} \sum_{\forall \mathbf{t} \in \mathscr{A}_k(\mathbf{s})} \exp\left(\frac{-d(\mathbf{s},\mathbf{t})}{2\sigma^2}\right), \tag{3.5}$$

onde $\mathscr{A}_k(\mathbf{s})$ representa os k-vizinhos mais próximos da amostra $\mathbf{s}$, e $\sigma = d_{max}/3$. Nesse caso, d_{max} denota o máximo arco ponderado em $\mathscr{G}^{tr}$. Observe que $\rho(\mathbf{s})$ considera todos os nós adjacentes para efeito do cálculo da probabilidade, uma vez que a função Gaussiana cobre 99.7% das amostras dentro $d(\mathbf{s},\mathbf{t}) \in [0, 3\sigma]$.

Depois de calculada a pdf para cada nó de treinamento, o processo de competição entre os protótipos ocorre por meio da função de custo de caminho f_{min}, como segue:

$$\begin{aligned} f_{min}(\langle \mathbf{t} \rangle) &= \begin{cases} \rho(\mathbf{t}) & \text{se } \mathbf{t} \in \mathscr{S} \\ \rho(\mathbf{t}) - 1 & \text{caso contrário} \end{cases} \\ f_{min}(\pi_s \cdot (\mathbf{s},\mathbf{t})) &= \min\{f_{min}(\pi_s), \rho(\mathbf{t})\}. \end{aligned} \tag{3.6}$$

A Equação 3.6 pode ser explicada em duas etapas: (i) a primeira (equação superior) está relacionada com o cálculo do custo inicial de cada amostra em $\mathscr{Z}_1$ usando a pdf, e (ii) a segunda parte (equação inferior) propaga o caminho ótimo para as amostras. Com respeito à etapa (i), no início da fase de treinamento, OPF com grafo k-nn calcula a pdf para cada amostra em $\mathscr{Z}_1$ e armazena esses valores em uma fila de prioridade Q (semelhante a linha 10 no Algoritmo 4). A fim de evitar super-agrupamentos (protótipos localizados em um platô de densidades), quando uma nova amostra é retirada de Q, verifica se existe predecessor para esse amostra. Caso não exista um predecessor, significa que essa determinada amostra é um protótipo (observe que, agora, para o OPF com grafo k-nn são removidas as amostras com maiores custos da fila Q) e, portanto, mantém o seu valor de densidade original; caso contrário, subtrai em uma unidade o seu valor de densidade.

Referente à etapa (ii), considere que a amostra **s** está em um processo de conquista da amostra **t**. A ideia consiste em **s** oferecer um valor mínimo entre o seu custo $f_{min}(\pi_{\mathbf{s}})$ e o valor da pdf de **t**, isto é, $\rho(\mathbf{t})$. Uma vez que o protótipo detém o maior custo de sua árvore de caminhos ótimos, o propósito aqui é conquistar amostras com menor custo. Finalmente, a amostra que maximiza f_{min} para **t** será seu "conquistador". O Algoritmo 6 implementa a etapa de treinamento do OPF com grafo k-nn.

Algoritmo 6 – OPF COM GRAFO k-NN - ALGORITMO DE TREINAMENTO

ENTRADA: Um grafo de treinamento $\mathscr{G}^{tr} = (\mathscr{Z}_1, \mathscr{A}_k)$ λ-rotulado e uma função de distância d.

SAÍDA: Floresta de caminhos ótimos P, mapa de rótulos L e mapa de custos C

1. **Para todo** $s \in \mathscr{Z}_1$, **Faça**
2. $\quad C(s) \leftarrow \rho(s)$.
3. $\quad P(s) \leftarrow nil$.
4. $Q \leftarrow Construa\ uma\ Fila\ de\ Prioridade(\mathscr{Z}_1, C)$.
5. **Enquanto** $Q \neq \emptyset$, **Faça**
6. $\quad$ *Remova de Q uma amostra* **s** *tal que* $C(s)$ *é máximo.*
7. $\quad$ **Se** $P(s) \neq nil$, **Então**
8. $\quad\quad C(s) \leftarrow C(s) - 1$.
9. $\quad$ **Para todo** $v \in \mathscr{A}_k(s)$, **Faça**
10. $\quad\quad tmp \leftarrow \min\{C(s), \rho(v)\}$.
11. $\quad\quad$ **Se** $(tmp > C(v))$, **Então**
12. $\quad\quad\quad P(v) \leftarrow s$.
13. $\quad\quad\quad L(v) \leftarrow \lambda(s)$.
14. $\quad\quad\quad C(v) \leftarrow tmp$.
15. **Retorna** $[P, L, C]$

As Linhas $1-3$ inicializam o mapa de custos e o mapa de predecessores para todas as amostras de treinamento, enquanto que na Linha 4 é criada uma fila de prioridade Q. O laço principal (Linhas $5-14$) representa o processo de competição. Na Linha 6, uma amostra de Q é retirada, e na Linha 7 é verificada se a amostra é um protótipo (não possui um nó predecessor). Caso possua um nó predecessor, ou seja, amostra não protótipo, o seu custo original é decrementado por uma unidade a fim de evitar platôs de valores de densidade (Linha 8). O laço mais interno (Linhas $9-14$) executa o processo de conquista para todas as amostras da vizinhança de **s**, ao passo que na Linha 10 é computada a função de custo de caminho f_{min}. No

caso da amostra **v** ser conquistada por **s**, seu predecessor, mapa de rótulos e mapa de custos são atualizados nas Linhas 12 – 14.

Finalmente, a etapa de classificação sobre $\mathscr{Z}_2$ é conduzida similarmente ao processo de conquista, isto é, dada uma amostra $\mathbf{t} \in \mathscr{Z}_2$, buscam-se os k-vizinhos mais próximos em $\mathscr{Z}_1$ para o cálculo de $\rho(\mathbf{t})$. Além disso, verifica se o nó $\mathbf{v}^* \in \mathscr{Z}_1$ satisfaz a seguinte equação:

$$C(\mathbf{t}) = \arg\max_{\mathbf{v} \in \mathscr{Z}_1} \min\{C(\mathbf{v}), \rho(\mathbf{t})\}. \tag{3.7}$$

Um ponto importante da técnica OPF com grafo k-nn é o tamanho da vizinhança, ou seja, o parâmetro k. Papa e Falcão (PAPA; FALCÃO, 2009) propuseram uma abordagem para o aprendizado do tamanho da vizinhança de $k \in [1, k_{max}]$ de modo que k^* maximiza a acurácia da classificação sobre um conjunto de validação $\mathscr{Z}_v$. Considere, agora, um novo conjunto de treinamento formado por $\hat{\mathscr{Z}}_1 = \mathscr{Z}_1 \setminus \mathscr{Z}_v$. O Algoritmo 7 implementa o procedimento acima.

Algoritmo 7 – OPF COM GRAFO k-NN - ALGORITMO DE APRENDIZAGEM

ENTRADA: Um conjunto de treinamento $\hat{\mathscr{Z}}_1$ e um conjunto de validação $\mathscr{Z}_v$, valor de k_{max} e uma função de distância d.

SAÍDA: Floresta de caminhos ótimos P, mapa de rótulos L e mapa de custos C

1. $MaxAcc \leftarrow 0$, $k \leftarrow 1$, $k^* \leftarrow 0$
2. **Enquanto** $k \leq k_{max}$, **Faça**
3. *Cria um grafo* $\hat{\mathscr{G}}^{tr} = (\hat{\mathscr{Z}}_1, \mathscr{A}_k)$
4. $[P, L, C] \leftarrow$ `Algoritmo 6`$(\hat{\mathscr{G}}^{tr}, d)$
5. **Para todo** $t \in \mathscr{Z}_v$, **Faça**
6. *Encontre* $\mathbf{v}^* \in \hat{\mathscr{Z}}_1$ *de acordo com a Equação 3.7.*
7. $\hat{L}(\mathbf{t}) \leftarrow \lambda(\mathbf{v}^*)$.
8. $Acc \leftarrow$ `Calcula Acurácia`$(\hat{L})$
9. **Se** $(MaxAcc \geq Acc)$, **Então**
10. $MaxAcc \leftarrow Acc$.
11. $k^* \leftarrow k$.
12. $k \leftarrow k + 1$.
13. *Cria uma gafo* $\hat{\mathscr{G}}^{tr} = (\hat{\mathscr{Z}}_1, \mathscr{A}_{k^*})$ `(Note que para esse procedimento foi utilizado` $\hat{f}_{min}$`)`
14. $[P, L, C] \leftarrow$ `Algoritmo 6`$(\hat{\mathscr{G}}^{tr}, d)$
15. **Retorna** $[P, L, C]$

O laço principal (Linhas 2 – 15) executa uma busca exaustiva para k^* dentro do intervalo

$[1, k_{max}]$, e a Linha 4 executa o treinamento do OPF com grafo k-nn. O laço mais interno (Linhas $5-7$) classifica o conjunto de validação, ao passo que a Linha 8 calcula a sua acurácia. Na sequência, atualizam-se $MaxAcc$ e k^*, respectivamente, caso o k avaliado possuir melhor eficácia. Por fim, (Linha 13) é criado um novo grafo de treinamento usando k^*.

Entretanto, ao invés de treinar sobre $\hat{\mathscr{G}}^{tr}$ mais uma vez (Linha 14), é necessário examinar uma abordagem diferente. Considerando a Equação 3.6, ela não garante um protótipo por classe (pelo menos), uma vez que o grafo pode ser dividido em vários agrupamentos quando considerada uma relação de adjacência k-nn. Esse problema pode ser resolvido, segundo os autores, penalizando os arcos ponderados entre amostras de diferentes classes, como abordado por $\hat{f}_{min}$:

$$\begin{aligned} \hat{f}_{min}(\langle \mathbf{t} \rangle) &= \begin{cases} \rho(\mathbf{t}) & \text{se } \mathbf{t} \in \mathscr{S} \\ \rho(\mathbf{t}) - 1 & \text{caso contrário} \end{cases} \\ \hat{f}_{min}(\pi_s \cdot (\mathbf{s}, \mathbf{t})) &= \begin{cases} -\infty & \text{se } \lambda(\mathbf{t}) \neq \lambda(\mathbf{s}) \\ \min\{\hat{f}_{min}(\pi_s), \rho(\mathbf{t})\} & \text{caso contrário.} \end{cases} \end{aligned} \tag{3.8}$$

Utilizar $\hat{f}_{min}$ somente no final do algoritmo e não desde o início do processo evita que problemas de *overfitting* possam surgir, visto que $\hat{f}_{min}$ penalizaria todos os arcos das amostras de classes diferentes, forçando, assim, um processo de competição não natural. Referente à esse ajuste, os autores afirmam que usar f_{min} no início da etapa de aprendizado não configura um problema, uma vez que é muito provável que as amostras de todas as classes encontrem-se dentro do intervalo $k \in [1, k_{max}]$. A modificação anteriormente apresentada pode ser implementada substituindo a Linha 10 do Algoritmo 6 pelo seguinte procedimento:

```
Se λ(s) ≠ λ(v), Então
        tmp ← −∞;
caso contrário
        tmp ← min{C(s), ρ(v)};
```

Capítulo 4

Medida de Confiança para Floresta de Caminhos Ótimos

Este capítulo tem por objetivo apresentar um processo de aprendizado do OPF utilizando as amostras de treinamento e uma medida de confiança para uso posterior na fase de classificação, conforme descrito por Fernandes et al. (FERNANDES et al., 2015).

Conforme discutido no Capítulo 3, a abordagem proposta por Papa et. al. (PAPA; FALCÃO; SUZUKI, 2009; PAPA et al., 2012) elege os nós protótipos como sendo as amostras mais próximas de classes diferentes, as quais podem ser encontrados por meio da MST[1]. Em caso de múltiplas MSTs, diferentes conjuntos de protótipos ótimos poderão ser encontrados, sendo que os custos oferecidos por eles serão os mesmos (ambos conjuntos são ótimos). Assim, a principal preocupação diz respeito às regiões de empate, ou seja, regiões onde há um conjunto de amostras de treinamento que oferecem o mesmo custo ótimo para um determinado nó. Portanto, este cenário pode levar o OPF a ser mais propenso à erros no conjunto de treinamento.

A ideia proposta neste capítulo é considerar não apenas o valor de caminho ótimo de uma determinada amostra no processo de classificação, mas também o seu **valor de confiança**, que é calculado por meio de um índice de pontuação por meio de um processo de aprendizagem ao longo de um conjunto de validação. Assim, a ideia consiste em penalizar as amostras de treinamento que não tenham um valor "confiável". Mostrou-se, mediante testes empíricos, que essa abordagem pode superar o modelo tradicional OPF em vários conjuntos de dados, mesmo em conjuntos de treinamento menores, e, além disso, essa abordagem pode ser mais eficiente em determinas situações pois pode realizar o treinamento mais rápido do que sua versão original.

De maneira geral, a estratégia proposta consiste em diminuir possíveis regiões de empate

[1]Note que a técnica a ser utilizada nesse capítulo consiste naquela que faz uso do grafo completo (Seção 3.1). Assim, para facilitar a apresentação, generalizamos a sua chamada no texto desse capítulo para OPF apenas.

que possam surgir mediante um determinado conjunto de treinamento. O restante do capítulo está organizado da seguinte forma: A Seção 4.1 apresenta a abordagem para o cálculo da medida de confiança baseada em pontuações, e a Seção 4.2 descreve a metodologia e os resultados experimentais. Finalmente, a Seção 4.3 apresenta as conclusões.

4.1 Aprendizado por níveis de confiança

A classificação usando o nível de confiança sustenta a ideia de atribuir uma pontuação para todos os nós do conjunto de treinamento utilizando uma etapa de aprendizado sobre um conjunto de validação. Para isso, é necessário particionar o conjunto de dados $\mathscr{Z}$ em três subconjuntos, ou seja, $\mathscr{Z} = \mathscr{Z}_1 \cup \mathscr{Z}_v \cup \mathscr{Z}_2$, em que $\mathscr{Z}_1$, $\mathscr{Z}_v$ e $\mathscr{Z}_2$ representam o conjunto de treinamento, validação e teste, respectivamente. É importante notar que todos os subconjuntos têm a sua respectiva representação em grafo dado por $\mathscr{G}^{tr}(\mathscr{Z}_1, \mathscr{A}, d)$, $\mathscr{G}^{vl}(\mathscr{Z}_v, \mathscr{A}, d)$ e $\mathscr{G}^{ts}(\mathscr{Z}_2, \mathscr{A}, d)$, respectivamente, como definido no Capítulo 3.

A abordagem proposta para aprendizado utilizando confiança visa treinar o classificador OPF sobre $\mathscr{Z}_1$ para posterior classificação de $\mathscr{Z}_v$ utilizando a mesma metodologia descrita no Capítulo 3 (em específico Seção 3.1). A principal diferença agora é que, para cada amostra de treinamento $\mathbf{s}$, há um nível de **confiança** $\phi(\mathbf{s})$, que é calculado por meio do seu desempenho individual em termos de sua taxa de reconhecimento sobre o conjunto de validação. De maneira geral, a ideia consiste em encontrar o valor de $\phi(\mathbf{s}), \forall \mathbf{s} \in \mathscr{Z}_1$, contabilizando as amostras classificadas corretamente. Ao final, esse nível de confiança é dividido pelo total de amostras "conquistadas" por $\mathbf{s}$.

No entanto, considerando a metodologia apresentada na Seção 3.1, uma amostra $\mathbf{s} \in \mathscr{Z}_1$ que não participou de qualquer processo de classificação seria pontuada com $\phi(\mathbf{s}) = 0$, sendo penalizada nesse caso, uma vez que quanto maior a pontuação, mais confiável é a amostra. Portanto, para tais amostras, é atribuído $\phi(\mathbf{s}) \rightarrow 1$ para dar-lhes a chance de realizar um "bom trabalho" durante a classificação sobre um novo padrão de entrada (conjunto de teste). Assim, após o processo de classificação sobre o conjunto de validação $\mathscr{Z}_v$, temos uma medida de pontuação $\phi(\mathbf{s}) \in [0,1], \forall \mathbf{s} \in \mathscr{Z}_1$, que pode ser utilizada como nível de confiança da referida amostra. Em suma, existem três níveis possíveis:

- $\phi(\mathbf{s}) = 0$: significa que a amostra $\mathbf{s}$ não realizou um "bom trabalho" na classificação das amostras em $\mathscr{Z}_v$, uma vez que classificou incorretamente todas as amostras. Portanto, as amostras com pontuações iguais a 0 **podem não ser confiáveis**;

- $0 < \phi(\mathbf{s}) < 1$: significa que a amostra **s** classificou incorretamente algumas amostras de $\mathscr{Z}_v$, mas também atribuiu corretamente os rótulos para outras amostras. Observe que, quanto maior o erro, menor é a confiabilidade da amostra. As amostras com resultados que se enquadram nesta faixa, **podem ser confiáveis**; e

- $\phi(\mathbf{s}) = 1$: significa que amostra **s** não participou de qualquer processo de classificação, ou **s** atribuiu o rótulo corretamente para todas as suas amostras conquistadas, o que significa que **s** é uma **amostra confiável**, de acordo com a nossa definição.

O Algoritmo 8 implementa o procedimento de aprendizagem das medidas de confiança conforme descrito acima.

Algoritmo 8 – ALGORITMO PARA APRENDIZADO DE CONFIANÇA

ENTRADA:	Um conjunto de treinamento e validação λ-rotulados, isto é, $\mathscr{Z}_1$ e $\mathscr{Z}_v$, respectivamente.
SAÍDA:	Nível de confiança $\phi(\mathbf{s})$, $\forall \mathbf{s} \in \mathscr{Z}_1$.
ESTRUTURAS AUXILIARES:	Matriz $n(\cdot)$ e $e(\cdot)$.

1. **Para todo** $s \in \mathscr{Z}_1$, **Faça**
2. $\quad n(s) = 0.$
3. $\quad e(s) = 0.$
4. $\quad \phi(s) = 0.$
5. *Treinar o OPF sobre* $\mathscr{Z}_1$ *de acordo com o Algoritmo 4.*
6. **Para todo** $t \in \mathscr{Z}_v$, **Faça**
7. $\quad$ *Seja* $s^* \in \mathscr{Z}_1$ *a amostra que classificou* **t** *com o rótulo* $L(t)$ *de acordo com a Equação 3.3.*
8. $\quad n(s^*) \leftarrow n(s^*) + 1.$
9. $\quad$ **Se** $\lambda(t) \neq L(t)$, **Então**
10. $\quad\quad e(s^*) \leftarrow e(s^*) - 1.$
11. **Para todo** $s \in \mathscr{Z}_1$, **Faça**
12. $\quad$ **Se** $n(s) = 0$, **Então**
13. $\quad\quad \phi(s) \leftarrow 1.$
14. $\quad\quad$ **Caso contrário**
15. $\quad\quad \phi(s) \leftarrow \frac{n(s)+e(s)}{n(s)}.$

As Linhas $1-4$ inicializam as pontuações de cada amostra de treinamento, e a Linha 5 executa a etapa de treinamento do OPF sobre $\mathscr{Z}_1$. A parte principal do algoritmo é realizada nas Linhas $6-15$. Na linha 7 é realizada a classificação de uma amostra de validação **t** por meio do

algoritmo tradicional de classificação do OPF. Considere $\mathbf{s}^* \in \mathscr{Z}_1$ uma amostra que conquistou $\mathbf{t}$: neste caso, a variável $n(\mathbf{s}^*)$ é incrementada, sinalizando a quantidade de "conquistas" que $\mathbf{s}^*$ realizou no decorrer de $\mathscr{Z}_v$, conforme a Linha 8. Além disso, caso $\mathbf{t}$ seja classificada incorretamente por $\mathbf{s}^*$, a variável $e(\mathbf{s}^*)$ é decrementada, conforme a Linha 10. Após isso, no laço seguinte (Linhas $11-15$) é calculada a pontuação final (confiança) para cada amostra de treinamento $\mathbf{s} \in \mathscr{Z}_1$. As Linhas $12-13$ definem as pontuações para amostras que não participaram de qualquer processo de classificação como $\phi(\mathbf{s}) = 1$, conforme mencionado anteriormente.

Após o aprendizado dos níveis de confiança para cada amostra de treinamento, é necessário modificar o procedimento tradicional de classificação do OPF a fim de considerar esta informação durante a etapa final de classificação para um novo padrão de dados. Para cumprir essa finalidade, foi proposta uma modificação do procedimento de classificação do OPF (Equação 3.3) como segue:

$$C(\mathbf{t}) = \arg\min_{\forall \mathbf{s} \in \mathscr{Z}_1} \left\{ \left(\frac{1}{\phi(\mathbf{s}) + \varepsilon} \right) * \max\{C(\mathbf{s}), d(\mathbf{s},\mathbf{t})\} \right\}, \tag{4.1}$$

em que $\varepsilon = 10^{-4}$ é empregado para evitar instabilidades numéricas. Portanto, a ideia do primeiro termo na Equação 4.1 é penalizar amostras com baixo valor de **confiança**, elevando, assim, o seu custo. Em suma, o valor da penalidade é inversamente proporcional ao nível de confiança de uma amostra.

Para melhor compreender a relevância das confianças, considere OPF* um classificador treinado sobre $\mathscr{Z}_1 \cup \mathscr{Z}_v$, e OPF$_c$ o modelo utilizando confiança na sua etapa de classificação, conforme proposto nesse capítulo. A ideia é mostrar as situações em que a abordagem que utiliza níveis de confiança pode superar o OPF tradicional fazendo uso da confiabilidade de uma dada amostra de treinamento ao classificar um novo conjunto de entrada. Na Figura 4.1 é apresentado o conjunto de treinamento (representado por hexágonos) sobre o conjunto de validação (amostras restantes) com respeito à base de dados "synthetic1" (Tabela 4.1), que compreende duas classes (quadrados e círculos).

Considere a região destacada na Figura 4.1, a qual é ampliada conforme representado nas Figuras 4.2a, 4.2b e 4.2c, para OPF, OPF* e OPF$_c$, respectivamente[2]. As amostras '**A**', '**B**' e '**C**' pertencem ao conjunto de treinamento, enquanto que a amostra '**D**' pertence ao conjunto de validação. As setas indicam o processo de conquista da amostra de teste, sendo que essa pode ser classificada pelas amostras '**A**', '**B**' ou '**C**' (o processo de competição será enfatizado somente entre as amostras '**A**' e '**B**'). Para OPF tradicional (Figura 4.2a), pode ser observado

[2]No caso, OPF corresponde ao classificador original treinado em $\mathscr{Z}_1$ apenas.

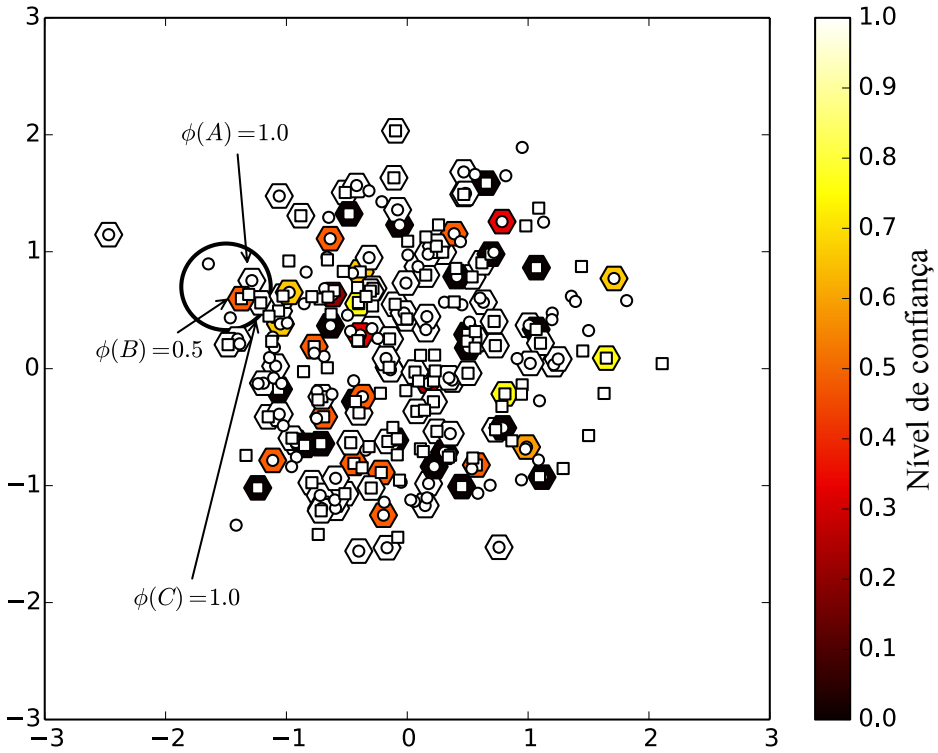


Figura 4.1: Representação gráfica do conjunto de dados "synthetic1" (Tabela 4.1) de cada amostra de treinamento de acordo com seu nível de confiança.

que a amostra '**B**' (arco escuro) proporciona um caminho de melhor custo comparado com a amostra '**A**' (arco claro), conquistando, assim, a amostra de teste e, consequentemente, realizando uma classificação incorreta, visto que o rótulo correto é definido como "círculo", ou seja, o mesmo rótulo da amostra '**A**'. Essa mesma situação pode ser observada para OPF*, conforme apresentado na Figura 4.2b. Esse fato evidencia que, mesmo utilizando conjuntos maiores para treinamento (conforme mencionado, OPF* é treinado sobre $\mathscr{Z}_1 \cup \mathscr{Z}_v$), tal "vantagem" pode não ser significativa para o processo de aprendizagem quando há um número elevado de "regiões de empate".

Entretanto, para o caso do OPF$_c$, conforme apresentado na Figura 4.2c, devido à baixa confiança da amostra '**B**' em relação à amostra '**A**', o custo fornecido para o novo padrão de teste pode ser melhor representado por '**A**' (arco escuro), visto que '**B**' foi penalizada pelo seu valor de confiança mais baixo. Portanto, a classificação apoiada na confiabilidade das amostras de treinamento pode melhorar a eficácia final em determinadas situações, principalmente em regiões com alta sobreposição de amostras, isto é, regiões de empate.

Uma forma de observar o domínio do valor de confiança das amostras de treinamento é por meio do interpolador por vizinhos naturais (HUNTER, 2007), conforme apresentado na Figura 4.3. As regiões mais escuras (valor de confiança próximo à zero) remete às amostras

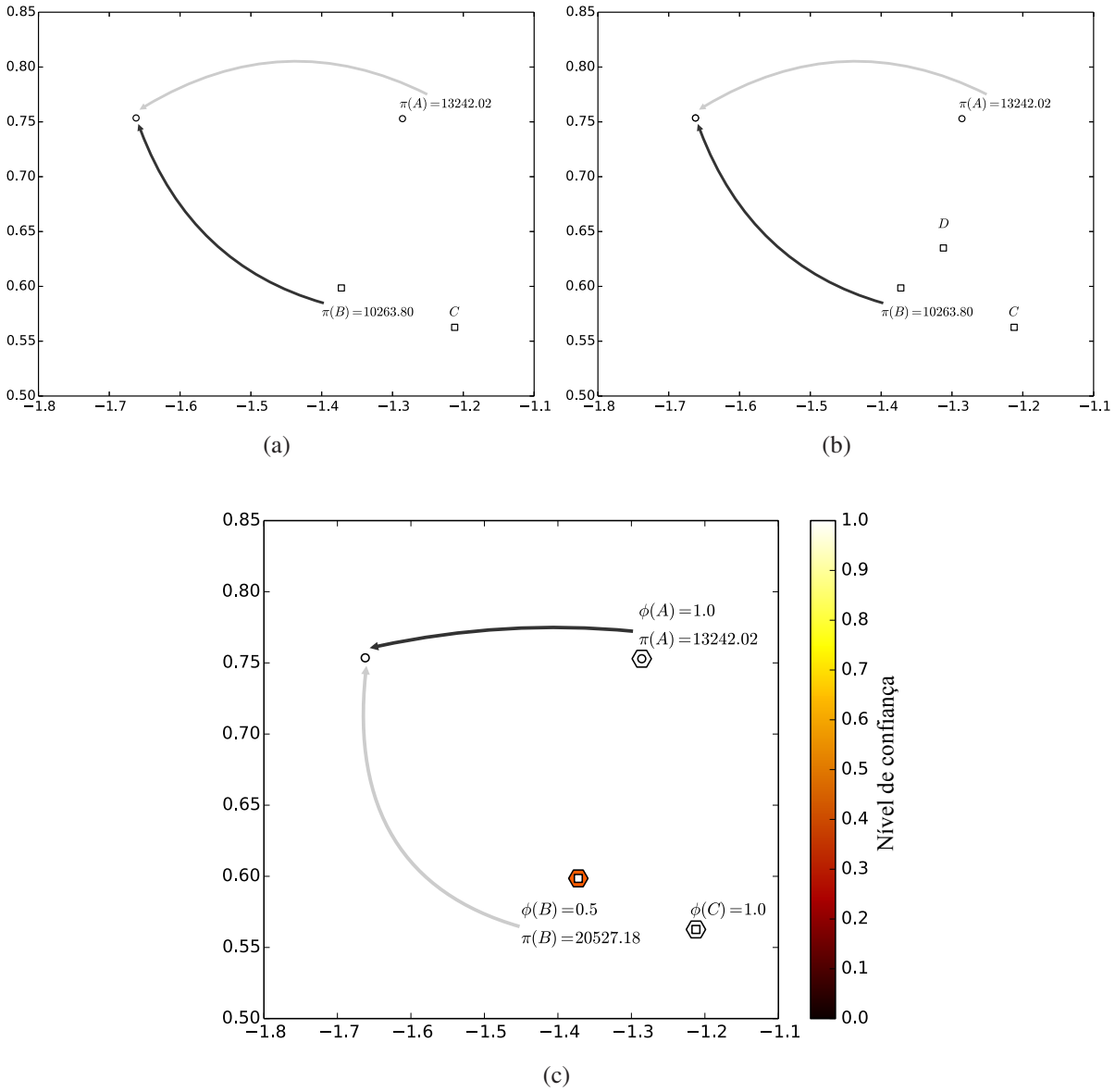


Figura 4.2: Exemplo de classificação sobre o conjunto teste para (a) OPF, (b) OPF* e (C) OPF$_c$.

com maior índice de empate; logo, amostras de treinamento que se enquadram nessas regiões podem não ser confiáveis o suficiente para classificar outras, bem como amostras de treinamento que estão localizadas nas proximidades dos *outliers* apresentam alta probabilidade de serem classificadas incorretamente utilizando técnicas tradicionais de reconhecimento de padrões. Portanto, se uma amostra de treinamento classifica incorretamente um *outlier* do conjunto de validação para o OPF$_c$, o seu nível de confiança irá diminuir, aumentando, assim, o seu custo de classificação para um novo padrão de entrada de dados.

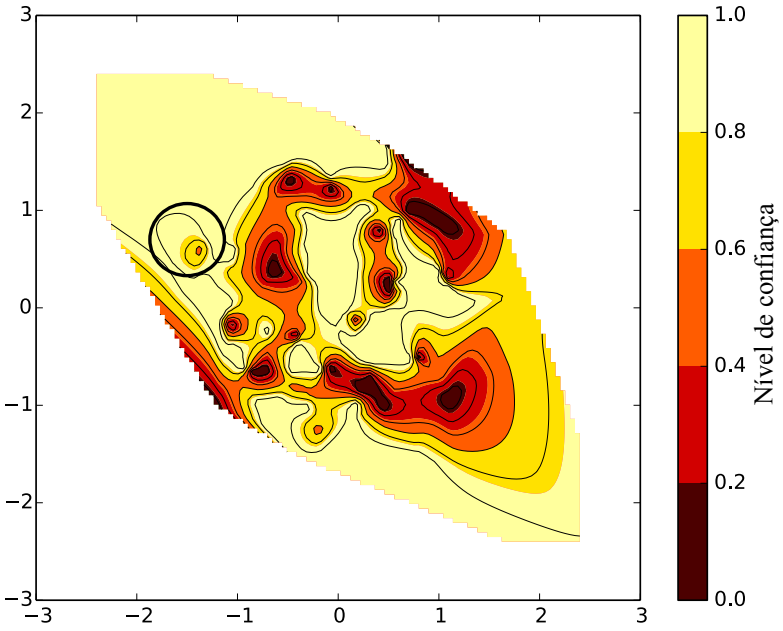


Figura 4.3: Representação gráfica das zonas de dispersão do conjunto de dados "synthetic1" (Tabela 4.1) de acordo com seu nível de confiança

4.2 Metodologia e Resultados Experimentais

A fim de avaliar a eficiência e a eficácia da abordagem baseada na confiança para o classificador OPF, foram realizados experimentos com vinte bases de dados de classificação (reais e sintéticas)[3][4][5], cujas características são apresentadas na Tabela 4.1. A escolha de algumas dessas bases foi motivada pelo seu nível de complexidade (amostras sobrepostas), o que torna a etapa de classificação mais suscetível à erros.[6]

Para cada base de dados, foi realizado o procedimento de validação cruzada com 15 rodadas, sendo cada uma delas dividida da seguinte forma: 30% das amostras foram utilizadas para compor o conjunto de treinamento, e o restante distribuído entre validação e teste da seguinte forma: $10\% - 60\%, 20\% - 50\%, \ldots, 50\% - 20\%$. Estas porcentagens foram escolhidas empiricamente, sendo mais intuitivo fornecer um conjunto maior de validação para o aprendizado de confiança. Além disso, foi calculada a média dos resultados (acurácia) e o tempo de processamento para cada técnica comparada, isto é, OPF tradicional e a abordagem proposta com a classificação baseada em confiança (OPF_c). A fim de permitir uma comparação justa, adicionamos também o OPF tradicional, mas com a etapa de treinamento sobre $\mathscr{Z}_1 \cup \mathscr{Z}_v$ (OPF*).

[3] http://mldata.org
[4] http://archive.ics.uci.edu/ml
[5] http://lrs.icg.tugraz.at/research/aflw
[6] Os experimentos foram realizados em um computador com processador Pentium Intel core *i*3 ® processador de 3.07Ghz, 4 GB de memória RAM e Linux Ubuntu desktop LTS 12.04 como o sistema operacional.

Na Tabela 4.1, conforme mencionado anteriormente, é apresentada a média dos resultados de classificação para todos os conjuntos de dados. Com a intenção de proporcionar uma análise mais robusta, foi realizado o teste estatístico não paramétrico de Friedman, que é utilizado para ranquear os algoritmos de cada conjunto de dados separadamente. No caso do teste de Friedman fornecer resultados significativos para rejeitar uma hipótese nula (h_0: todas as técnicas são equivalentes), mais adiante pode ser realizado um teste *post-hoc*. Para esse caso, foi utilizado o teste de Nemenyi, proposto por Nemenyi (NEMENYI, 1963) e descrito por Demsar (DEMSAR, 2006), que permite verificar se existe uma diferença crítica (DC) entre as técnicas. Os resultados do teste Nemenyi podem ser representados por meio de um diagrama simples, em que o ranque dos métodos comparados são representadas graficamente por um eixo horizontal. Nesse eixo, o menor valor apontado indica a melhor técnica. Além disso, os grupos que não apresentam diferenças significativas são conectados por uma linha horizontal. As melhores técnicas com relação a acurácia, de acordo com o teste de Nemenyi, são destacadas em negrito na Tabela 4.1.

Tabela 4.1: Acurácia média: os valores em negrito representam as técnicas mais eficazes. As taxas de reconhecimento foram calculadas de acordo com Papa et al. (PAPA; FALCÃO; SUZUKI, 2009), os quais consideram conjuntos de dados não balanceados em sua formação.

Base de dados	OPF	OPF*	OPF$_c$	# amostras	# características	# classes
a1a	65,74	65,59	**69,05**	32.561	123	2
aloi	95,31	**96,92**	95,09	108.000	128	1.000
connect-4	**63,32**	63,05	63,10	67.557	126	3
synthetic1	53,05	52,44	**56,14**	500	2	2
synthetic4	50,69	**50,78**	50,72	100.000	100	1.000
synthetic5	85,29	85,56	**87,33**	100.000	4	4
synthetic6	89,55	89,70	**91,14**	100.000	4	4
dmoz-web-directory-topics	59,16	**62,06**	56,72	1.329	10.629	5
dna	83,80	**88,99**	85,02	5.186	180	3
duke-breast-cancer	80,37	**91,15**	79,46	86	7.129	2
ijcnn1	93,78	**96,46**	94,13	191.681	22	2
Statlog-Letter	97,31	**98,58**	97,58	35.000	16	26
Leukemia	71,47	**76,90**	69,63	72	7.129	2
mushrooms	93,68	92,61	**96,93**	8.124	112	2
scene-classification	66,04	**67,78**	66,60	2.407	294	15
shuttle	94,48	**97,25**	95,09	101.500	9	7
usps	97,24	**97,93**	97,28	9.298	256	10
w1a	80,54	80,15	**80,68**	49.749	300	4
yahoo-web-directory-topics	50,54	51,77	**56,36**	1.106	10.629	4
aflw	88,00	**89,48**	88,93	8.193	4.096	2

Pode-se observar que OPF$_c$ obteve os melhores resultados em 7 de 20 conjuntos de dados, e com resultados muito próximos dos melhores em outros 7 conjuntos de dados. Os piores resultados foram obtidos sobre "duke-breast-cancer" e "Leukemia" , visto que estes são conjuntos de dados pequenos proporcionando, assim, um conjunto de validação que não foi suficiente para aprender bons níveis de confiança. No entanto, mesmo nesses conjuntos de dados, a taxa de reconhecimento do OPF$_c$ foi próxima da abordagem tradicional. Como OPF* utilizou conjuntos de dados maiores, é normal que tenha obtido resultados mais eficazes em algumas bases.

Não foi possível estabelecer alguma situação específica (considerando a configuração do conjunto de dados, tais como o número de classes e número de características, por exemplo), em que OPF_c pode ser considerado melhor do que OPF e OPF^*, embora acredita-se que a abordagem proposta obteve os melhores resultados em conjuntos de dados de alta dimensionalidade, com exceção da "dmoz-web-directory-topics". Se considerarmos uma margem de erro de cerca de 3%, a abordagem proposta obteve resultados semelhantes em 17 de 20 conjuntos de dados, podendo ser considerada um abordagem relevante para melhorar o classificador OPF.

A premissa acima pode ser reforçada se considerarmos o esforço computacional das técnicas. Como esperado, OPF tradicional apresentou-se mais rápido do que OPF_c e OPF^* com respeito à etapa de treinamento (treinamento e aprendizado das pontuações), uma vez que não precisa calcular o nível de confiança para cada amostra de treinamento. No entanto, o teste estatístico de Nemenyi apontou que OPF_c foi mais rápido do que OPF^* para a etapa de treinamento (Figura 4.4a), sendo semelhante no que diz respeito à etapa de classificação, tal como apresentado na Figura 4.4b. Em média, isto é, considerando todos os 20 conjuntos de dados, OPF tradicional mostrou-se aproximadamente $2,108$ vezes mais rápido do que OPF_c e OPF^*.

(a) (b)

Figura 4.4: Teste estatístico de Nemenyi com respeito à carga computacional para a (a) etapa de treinamento (treinamento e aprendizado das pontuações) e (b) etapa de teste. Grupos de abordagens semelhantes são conectados uns com os outros.

4.3 Conclusões

Neste capítulo foi introduzido um algoritmo de aprendizagem baseado em confiança para melhorar os resultados de classificação da técnica OPF. A ideia é penalizar as amostras de treinamento que classificam erroneamente outras amostras em um etapa de classificação sobre um conjunto de validação. O algoritmo proposto tem como objetivo aprender os níveis de confiança para cada amostra de treinamento para, em seguida, serem utilizados em uma versão modificada do procedimento de classificação padrão empregado pelo OPF.

Experimentos em vinte bases de dados mostraram que a abordagem utilizando confiança

obteve os melhores resultados em 7 conjuntos de dados, sendo também muito próxima dos melhores em outros 7 conjuntos de dados. Além disso, OPF_c pôde melhorar os resultados do OPF tradicional ainda que utilizando conjuntos de treinamento menores, sendo também mais rápido do que OPF utilizando a união do conjunto de treinamento e validação. Assim sendo, a principal contribuição desse capítulo foi apresentar uma abordagem de aprendizado de medidas de confiança para melhorar o processo de aprendizado da técnica OPF.

Capítulo 5

Combinação de Floresta de Caminhos Ótimos Utilizando Níveis de Confiança Baseados em Pontuações

Este capítulo tem por objetivo apresentar um estudo das medidas de confiança utilizando combinação de classificadores OPFs, conforme descrito por Fernandes e Papa (FERNANDES; PAPA, 2017a). A ideia dessa proposta consiste em estender o uso das medidas de confiança apresentadas por Fernandes et al. (FERNANDES et al., 2015) no Capítulo 4 em um conjunto de classificadores OPFs, ou seja, explorar a combinação de OPFs por meio do caminho de custo ótimo que considera valores de confiança oriundos de diferentes classificadores. Esse trabalho apresenta também um refinamento dos resultados apresentados por Fernandes et al. (FERNANDES et al., 2015), bem como estende tal abordagem de confiança para o classificador OPF com grafo k-nn[1].

O restante do capítulo está organizado da seguinte forma. A Seção 5.1 apresenta a abordagem proposta para a classificação orientada por combinação de classificadores com confiança baseada em pontuações, e a Seção 5.2 descreve a metodologia e os resultados experimentais. Finalmente, a Seção 5.3 apresenta as conclusões.

[1]Para facilitar a apresentação, generalizamos a chamada da técnica OPF que utiliza grafo k-nn (Seção 3.2) para OPF$_{knn}$, e a versão que utiliza confiança como OPF$_{knnC}$.

5.1 Combinação de Classificadores Utilizando Níveis de Confiança Baseada em Pontuações

Nesta seção, uma nova abordagem é apresentada para combinação de classificadores OPFs utilizando níveis de confiança baseados em pontuações para melhorar a eficácia final. É válido lembrar que o classificador OPF, como mencionado anteriormente, emprega o método de saída abstrata ao classificar amostras, isto é, a saída do classificador é representada por um único rótulo. Conforme definido por Xu et al. (XU; KRZYZAK; SUEN, 1992), uma forma de combinar as saídas abstratas de L classificadores é por meio da atribuição dos rótulos para cada amostra do conjunto de dados e, em seguida, uma coleção de L possíveis saídas para cada amostra é gerada.

Considere um conjunto de L classificadores OPFs utilizando diferentes amostras de dados para treinamento. Dados $\mathscr{D} = \{D_1, D_2, \ldots, D_L\}$ um conjunto de L classificadores e $\Omega = \{\omega_1, \omega_2, \ldots, \omega_K\}$ um conjunto de rótulos de K classes, cada classificador associa um rótulo a um vetor n-dimensional, ou seja, $D_i : \Re^n \rightarrow \Omega$, $i = 1, 2, \ldots, L$. Portanto, para qualquer novo padrão de entrada de dados $\mathbf{z}$, o conjunto de classificadores gera uma coleção $d_i(\mathbf{z}) = [d_1(\mathbf{z}), d_2(\mathbf{z}), \ldots, d_L(\mathbf{z})]$ de possíveis saídas, onde $d_i(\mathbf{z})$ representa a saída do i-ésimo classificador considerando a amostra $\mathbf{z}$.

A ideia é particionar o conjunto de treinamento $\mathscr{Z}_1$ em L subconjuntos disjuntos $\mathscr{Z}_1^j$, ou seja, $\mathscr{Z}_1 = \mathscr{Z}_1^1 \cup \mathscr{Z}_1^2 \cup \ldots \cup \mathscr{Z}_1^L$, de tal forma que cada classificador D_j seja treinado sobre $\mathscr{Z}_1^j, j = 1, 2, \ldots, L$. A abordagem proposta emprega o uso dos níveis de confiança baseados em pontuações para cada classificador D_j treinado sobre o conjunto de validação $\mathscr{Z}_v$, conforme apresentado na Seção 4.1. Com isso, será atribuído um nível de pontuação para cada amostra dos diferentes subconjuntos de treinamento. Depois do aprendizado de confiança utilizando o conjunto de validação, a classificação é realizada pela Equação 4.1, e as possíveis saídas são atribuídas à cada amostra $\mathbf{s} \in \mathscr{Z}_2$. A decisão final é realizada por meio da votação por maioria. Essa ideia é ilustrada na Figura 5.1.

Para essa proposta de combinação de classificadores OPFs utilizando confiança, foram avaliadas duas versões distintas do classificador OPF: OPF utilizando grafo completo (PAPA; FALCÃO; SUZUKI, 2009) e com grafo k-nn, isto é, OPF$_{knn}$ (PAPA; FALCÃO, 2008). Além disso, foi aplicada a mesma ideia dos níveis de confiança na etapa de classificação do OPF$_{knn}$, uma vez que o trabalho apresentado por Fernandes et al. (FERNANDES et al., 2015) utiliza somente OPF com grafo completo.

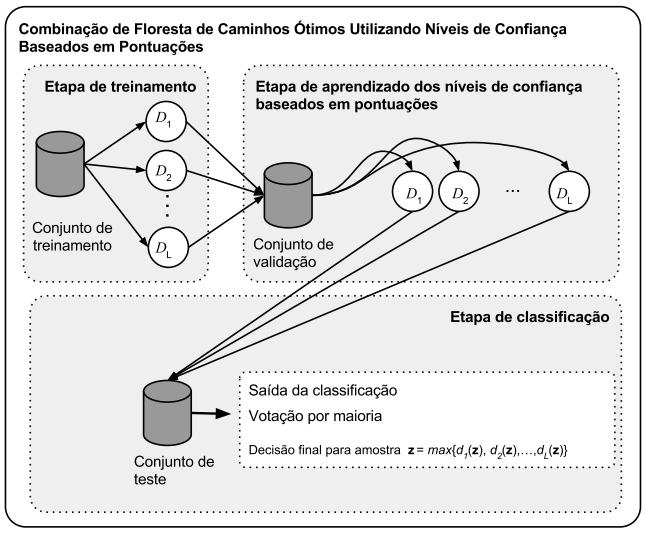


Figura 5.1: Abordagem proposta utilizando combinação de classificadores com medidas de confiança.

5.2 Metodologia e Resultados Experimentais

Visando avaliar a eficiência e a eficácia da abordagem de combinação de classificadores OPFs utilizando níveis de confiança, foi realizada uma comparação com a técnica OPF tradicional utilizando vinte bases de dados de classificação reais e sintéticas[2][3][4]. Na Tabela 5.1 são apresentadas as características das bases de dados utilizadas.[5]

A fim de obter resultados estatisticamente significativos, três diferentes porcentagens de treinamento, validação e teste foram utilizadas: (i) em uma primeira fase, cada conjunto de dados foi dividido em três subgrupos: treinamento com 50%, validação com 10% e teste com 40%, a seguir indicados como 50:10:40; (ii) em uma segunda fase, os conjuntos de dados foram divididos em 45:20:35; e (iii) na última etapa, os conjuntos de dados foram divididos em 40:30:30. Para cada intervalo, os conjuntos de treinamento, validação e teste foram selecionados aleatoriamente e o processo foi repetido vinte vezes via um procedimento de validação cruzada.[6].

Visando uma comparação justa, OPF tradicional foi treinado sobre $\mathscr{Z}_1 \cup \mathscr{Z}_v$ (isto é OPF*) considerando os três estágios mencionados anteriormente, ou seja, 60:40, 65:35 e 70:30. Além disso, foram calculadas a média dos resultados (acurácia) e o tempo de processamento para cada

[2]http://mldata.org

[3]http://archive.ics.uci.edu/ml

[4]http://lrs.icg.tugraz.at/research/aflw

[5]Os experimentos foram realizados em um computador com processador Pentium Intel core *i*3 ® processador de 3.07Ghz, 4 GB de memória RAM e Linux Ubuntu desktop LTS 12.04 como o sistema operacional.

[6]As porcentagens foram empiricamente escolhidas, sendo mais intuitivo fornecer um conjunto de validação maior para o aprendizado dos níveis de confiança

Tabela 5.1: Descrição das bases de dados.

Base de dados	# amostras	# características	# classes
aflw	8.193	4.096	2
Colon-cancer	62	2.000	2
Pima-Indians-Diabetes	768	8	2
Statlog-Australian	690	14	2
Statlog-dna	5.186	180	3
Statlog-Heart	270	13	2
Statlog-Letter	35.000	16	26
Synthetic1	500	2	2
Synthetic2	1.000	2	2
Synthetic3	200	2	2
Synthetic4	100.000	100	1.000
Synthetic5	100.000	4	4
UCI-a1a	32.561	123	2
UCI-Connect-4	67.557	126	3
UCI-Ionosphere	351	34	2
UCI-Liver-disorders	345	6	2
UCI-Mushrooms	8.124	112	2
usps	9.298	256	10
w1a	49.749	300	4
yahoo-web-directory-topics	1.106	10.629	4

técnica comparada. Observe que a intenção em usar diferentes porcentagens de $\mathscr{Z}_1$, $\mathscr{Z}_v$ e $\mathscr{Z}_2$ é motivada pelo intuito de investigar a eficácia da abordagem proposta em diferentes cenários. Ademais, os resultados finais foram avaliados utilizando o teste de Wilcoxon (*signed-rank*) com significância de $0,05$ (WILCOXON, 1945). Para a implementação dessa proposta, foi utilizada a biblioteca de código aberto LibOPF[7].

O teste empírico foi realizado comparando as seguintes técnicas: (a) apenas um classificador OPF tradicional (OPF*); (b) apenas um classificador OPF$_c$, descrito no Capítulo 4; (c) uso de três classificadores OPF$_c$, combinados com a estratégia proposta neste capítulo, definido como **combinação OPF$_c$**; (d) combinação de OPF$_{knn}$ com OPF$_c$ usando a estratégia proposta neste capítulo, ou seja, duas abordagens OPF$_c$ em conjunto com uma abordagem OPF$_{knn}$ (a seguir denominado como **combinação OPF$_c$+OPF$_{knn}$**); e (e) uma última configuração de combinação a qual compreende o uso do OPF$_{knn}$ usando a mesma ideia de confiança do OPF$_c$, porém adaptado para um grafo k-vizinhos (definido como OPF$_{knnC}$). Para esse último caso, a combinação também é composta por três técnicas de classificação, sendo uma OPF$_{knnC}$ e as outras duas OPF$_c$ (definido como **combinação OPF$_c$+OPF$_{knnC}$**). Para todas as configurações

[7]https://github.com/jppbsi/LibOPF.git

que utilizam combinação de classificadores, as decisões são combinadas por meio da votação por maioria. A configuração dos experimentos é ilustrada na Figura 5.2

Para as configurações que utilizaram combinação de classificadores, foram utilizadas somente três técnicas-base para cada esquema de combinação, visto que não foram observados ganhos significativos utilizando mais classificadores. O raciocínio que suporta essa ideia está relacionado com o número de amostras disponíveis para o aprendizado de cada classificador-base, um vez que quanto mais classificadores são utilizados, menor serão os subconjuntos de treinamento. A Tabela 5.2 apresenta a média e o desvio padrão dos resultados de cada abordagem avaliada. As taxas de reconhecimento foram calculadas de acordo com Papa et al. (PAPA; FALCÃO; SUZUKI, 2009), e os valores em destaque representam as técnicas mais eficazes de acordo com o teste de Wilcoxon (*signed-rank*).

Tabela 5.2: Acurácia média dos resultados: os valores em negrito representam as técnicas mais eficazes.

Base de dados	**OPF***	**OPF***c*	*combinação* **OPF***c*	*combinação* **OPF***c* **+** **OPF**$_{knn}$	*combinação* **OPF***c* **+** **OPF**$_{knnC}$
aflw	$89,73 \pm 0,47$	$89,42 \pm 0,48$	$\mathbf{90,37} \pm 0,47$	$90,24 \pm 0,51$	$\mathbf{90,33} \pm 0,44$
Colon-cancer	$68,54 \pm 0,00$	$64,03 \pm 3,60$	$\mathbf{72,80} \pm 8,90$	$68,33 \pm 8,04$	$69,84 \pm 8,27$
Pima-Indians-Diabetes	$63,90 \pm 2,48$	$64,48 \pm 1,79$	$\mathbf{66,23} \pm 1,16$	$\mathbf{66,72} \pm 2,53$	$\mathbf{66,91} \pm 2,61$
Statlog-Australian	$64,88 \pm 2,39$	$64,42 \pm 2,14$	$\mathbf{67,05} \pm 1,34$	$65,62 \pm 2,46$	$65,95 \pm 2,57$
Statlog-dna	$\mathbf{89,95} \pm 0,61$	$87,72 \pm 0,78$	$86,63 \pm 0,86$	$86,84 \pm 0,81$	$86,75 \pm 0,83$
Statlog-Heart	$58,19 \pm 3,35$	$62,76 \pm 1,86$	$\mathbf{64,20} \pm 1,29$	$60,56 \pm 3,98$	$61,20 \pm 3,58$
Statlog-Letter	$\mathbf{98,79} \pm 0,12$	$98,28 \pm 0,14$	$97,30 \pm 0,11$	$97,37 \pm 0,10$	$97,35 \pm 0,11$
Synthetic1	$53,83 \pm 3,01$	$54,00 \pm 0,12$	$\mathbf{58,55} \pm 1,64$	$56,15 \pm 3,56$	$56,32 \pm 3,76$
Synthetic2	$72,62 \pm 1,92$	$73,23 \pm 1,40$	$\mathbf{77,94} \pm 0,68$	$77,83 \pm 1,77$	$\mathbf{78,14} \pm 1,77$
Synthetic3	$92,72 \pm 2,57$	$\mathbf{93,94} \pm 0,68$	$92,42 \pm 0,59$	$\mathbf{94,22} \pm 2,15$	$\mathbf{94,11} \pm 2,27$
Synthetic4	$\mathbf{50,80} \pm 0,03$	$\mathbf{50,79} \pm 0,03$	$50,64 \pm 0,03$	$50,64 \pm 0,03$	$50,64 \pm 0,03$
Synthetic5	$85,58 \pm 0,11$	$86,66 \pm 0,14$	$\mathbf{88,57} \pm 0,11$	$88,40 \pm 0,10$	$88,42 \pm 0,10$
UCI-a1a	$65,32 \pm 1,03$	$68,18 \pm 1,22$	$\mathbf{72,31} \pm 0,71$	$\mathbf{72,24} \pm 0,53$	$\mathbf{72,31} \pm 0,48$
UCI-Connect-4	$63,16 \pm 0,30$	$63,26 \pm 0,30$	$64,98 \pm 0,23$	$\mathbf{65,82} \pm 0,29$	$65,63 \pm 0,32$
UCI-Ionosphere	$\mathbf{83,12} \pm 2,60$	$\mathbf{82,75} \pm 1,12$	$80,45 \pm 1,77$	$80,05 \pm 3,52$	$80,14 \pm 3,58$
UCI-Liver-disorders	$58,47 \pm 3,79$	$62,30 \pm 0,62$	$\mathbf{63,79} \pm 1,92$	$\mathbf{62,46} \pm 3,83$	$\mathbf{62,40} \pm 4,00$
UCI-Mushrooms	$92,95 \pm 8,38$	$97,99 \pm 3,38$	$99,66 \pm 0,73$	$\mathbf{99,90} \pm 0,19$	$\mathbf{99,33} \pm 2,68$
usps	$\mathbf{98,03} \pm 0,17$	$97,73 \pm 0,16$	$97,14 \pm 0,24$	$97,25 \pm 0,22$	$97,24 \pm 0,22$
w1a	$80,42 \pm 1,01$	$\mathbf{80,86} \pm 0,93$	$80,19 \pm 0,59$	$\mathbf{80,61} \pm 0,59$	$80,56 \pm 0,58$
yahoo-web-directory-topics	$51,98 \pm 2,00$	$\mathbf{58,92} \pm 5,76$	$\mathbf{57,04} \pm 5,79$	$53,97 \pm 6,52$	$54,11 \pm 6,81$

Com base nos resultados experimentais, a abordagem proposta (configurações que utilizam combinação de classificadores) obteve os melhores resultados (com relação à eficácia) em treze das vinte bases de dados. Dentre as razões que podem ser apontadas para essa eficácia da abordagem proposta, uma delas está relacionada com as regiões que apresentam alta sobreposição de amostras. Para isso, considere as Figuras 5.3a (base de dados “Synthetic1”) e 5.3b (base de da-

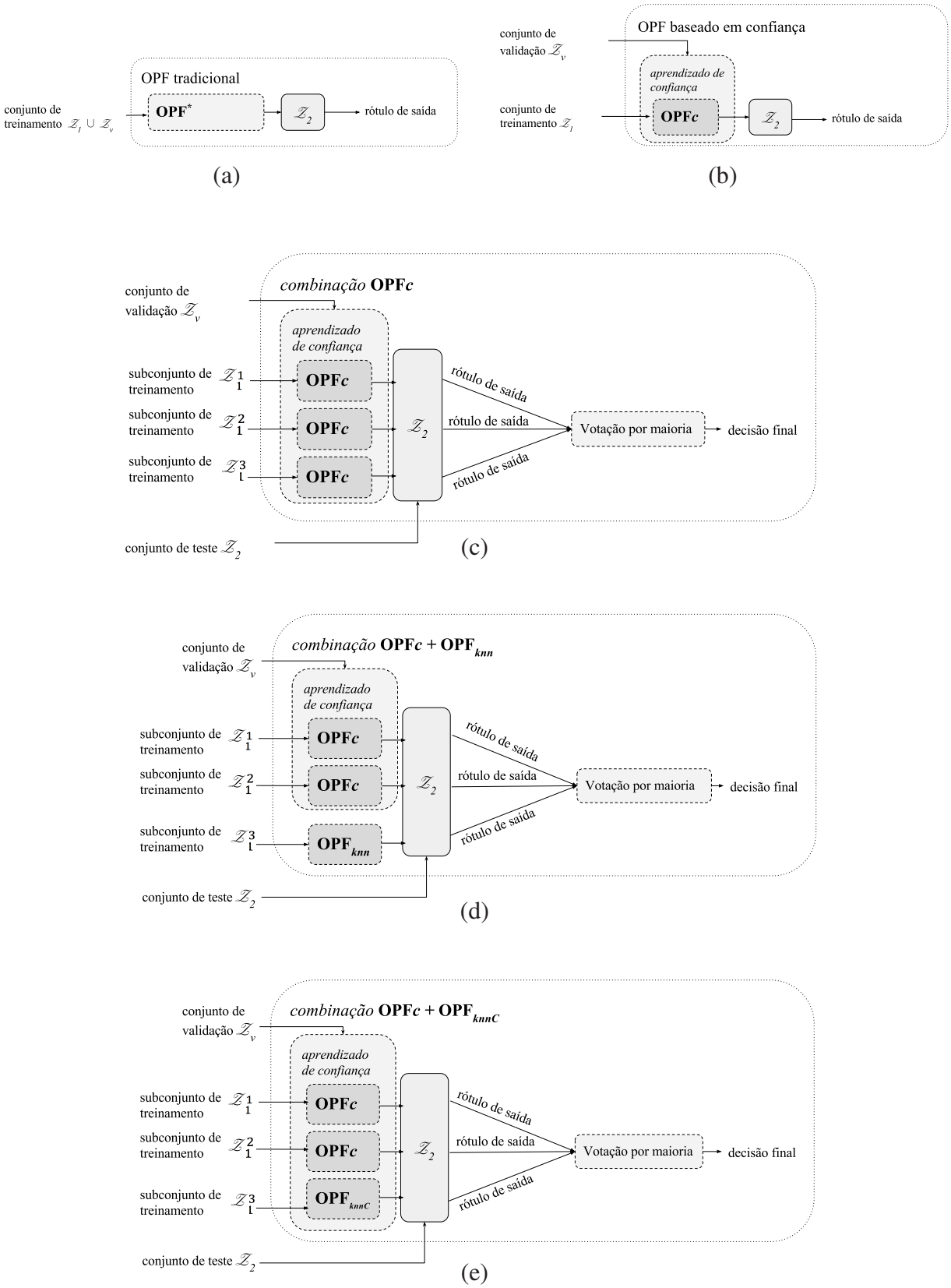
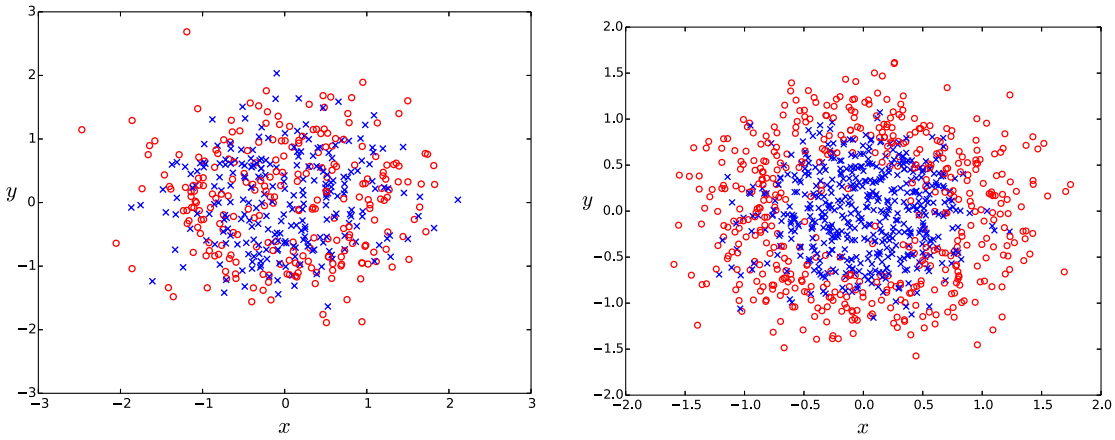


Figura 5.2: Configuração dos experimentos para: (a) OPF tradicional (OPF*) e (b) abordagem OPF$_c$, (c) a abordagem proposta usando combinação de classificadores OPF$_c$, (d) usando combinação com OPF$_c$ e OPF$_{knn}$, e (e) combinação utilizando OPF$_c$, com OPF$_{knnC}$ (OPF$_{knn}$ com aprendizado de confiança).

dos “Synthetic2”), as quais apresentam certa quantidade de amostras sobrepostas, ou seja, existe uma maior probabilidade de ocorrerem regiões de empate para a técnica OPF, portanto, sendo mais útil aprender com OPF$_c$ e, consequentemente, mais eficaz quando realizada a combinação de classificadores, visto que o espaço de características é dividido em diferentes sub-regiões.



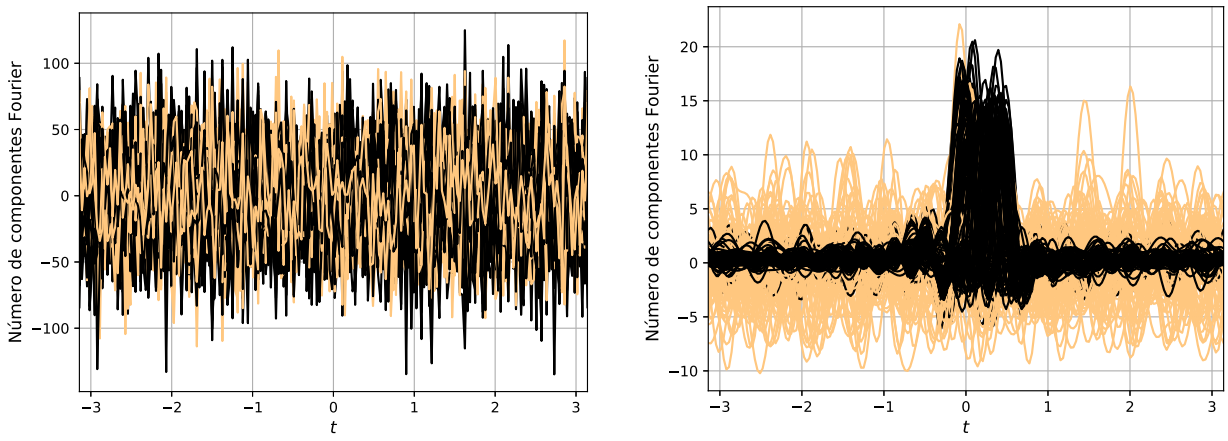
(a) (b)

Figura 5.3: Representação gráfica contendo todas as amostras da base dados (a) “Synthetic1” e (b) “Synthetic2”.

Entretanto, a situação mencionada acima geralmente não ocorre em bases de dados com pouca sobreposição de amostras, ou quando seus dados são considerados “bem-comportados”. Tais problemas de classificação, conforme mencionado anteriormente, podem ser melhores classificados pela técnica tradicional do classificador OPF ao invés de utilizar aprendizado por confiança ou até mesmo a combinação de classificadores.

Um outro aspecto a ser considerado ocorre quando a técnica OPF$_c$ não oferece melhor eficácia do que a abordagem OPF tradicional, porém a **combinação OPF$_c$** fornece, como observado na base de dados “Colon-cancer”, por exemplo. Com base nisso, para melhor compreender a relação de contiguidade dos dados, foi utilizado o método denominado de Curvas de Andrews (*Andrews curves*) (ANDREWS, 1972), o qual fornece uma representação do espaço de características de alta dimensionalidade utilizando séries de Fourier. A transformação procura preservar algumas propriedades dos dados, tornando possível, assim, identificar alguns comportamentos dos mesmos (KOZIOL; HACKE, 1991). Cada linha fornecida pela saída das *Andrews curves* representa um amostra, e cada cor corresponde à uma determinada classe. Considerando as bases de dados “Colon-cancer” e “UCI-Ionosphere”, as suas representações são apresentadas nas Figuras 5.4a e 5.4b, respectivamente.

Conforme apresentado na Figura 5.4, pode-se observar que a complexidade dos padrões favorece a incidência de sobreposição de amostras, sendo que para padrões mais complexos os



(a) (b)

Figura 5.4: Representação dos dados utilizando o método *Andrews curves* com respeito à base (a) "Colon-cancer" e (b) "UCI-Ionosphere" no intervalo de $-\pi < t < \pi$.

ganhos com a combinação utilizando OPF$_c$ em comparação com a abordagem sem combinação de classificadores são mais perceptíveis. Erros durante o processo de classificação são altamente associados com as regiões de empate, as quais, conforme mencionado anteriormente, representam situações onde tem-se um conjunto de amostras de treinamento que oferecem o mesmo custo ótimo para um determinado nó. A teoria de base do OPF elege os nós protótipos como sendo as amostras mais próximas de diferentes classes, as quais podem ser encontradas mediante MST sobre o conjunto de treinamento. Atualmente, caso exista uma única MST, os custos das arestas serão diferentes entre elas, portanto, o erro de classificação para OPF sobre o conjunto de treinamento seria zero, visto que o caminho ótimo do nó protótipo para o restante das amostras segue o formato da MST. Assim, como os protótipos escolhidos estão posicionados nas fronteiras de diferentes classes, não é possível para uma amostra de uma determinada classe conquistar outra amostra de outra classe.

Todavia, a situação mencionada acima não ocorre na prática, pois a probabilidade de ocorrerem múltiplas MSTs em bases grandes são altas. Na implementação padrão do OPF, caso o caminho ótimo oferecido por diferentes classes seja o mesmo para uma determinada amostra, o que for oferecido primeiro irá conquistá-la. Para a combinação utilizando OPF$_c$, quando subconjuntos do conjunto original de treinamento são utilizados em vez de todo o conjunto original, múltiplas MSTs produzem processos distintos de conquistas que, associados ao aprendizado das medidas de confiança, melhoram a eficácia da fase de classificação.

Além do teste de Wilcoxon utilizado para destacar as diferenças das técnicas comparadas na Tabela 5.2, utilizou-se, também, o teste estatístico de Friedman para ranquear as técnicas, bem como o teste de Nemenyi para indicar a diferença crítica entre as abordagens. A Figura 5.5

apresenta a análise estatística considerando a acurácia do conjunto de teste. Como pode ser observado, as propostas que utilizam combinação OPF_c + OPF_{knnC} e combinação OPF_c podem ser consideradas as mais eficazes. Por último, no segundo grupo, estão as técnicas OPF_c e OPF (OPF^*). Esse teste reflete que, de fato, a abordagem utilizando combinação OPF_c mostrou-se mais eficaz na maioria das bases de dados avaliadas. Entretanto, o teste estatístico não apontou qualquer diferença crítica entre as abordagens que utilizam combinação, isto é, combinação OPF_c, combinação OPF_c + OPF_{knn} e combinação OPF_c + OPF_{knnC}. Desta forma, essas abordagens são consideradas similares.

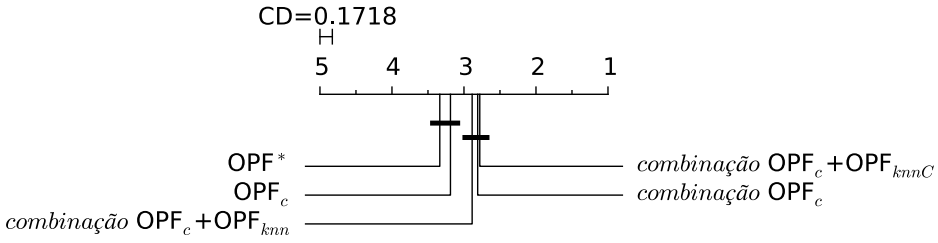


Figura 5.5: Comparação entre todas as abordagens de acordo com a acurácia utilizando o teste de Nemenyi. Grupos similares ($p = 0,05$) são conectados.

Com relação ao tempo de processamento, a Figura 5.6 apresenta a média da carga computacional no que diz respeito à fase de treinamento (treinamento com aprendizado dos valores de confiança). Como esperado, a combinação baseada em classificadores OPF mostrou-se mais eficiente em comparação com as técnicas OPF_c e OPF^* pois, conforme Ponti e Papa (PONTI; PAPA, 2011), treinar utilizando pequenas sub-regiões é mais rápido do que treinar utilizando todo o conjunto de treinamento. Em média, isto é, considerando todas as vinte bases de dados, a combinação OPF_c foi cerca de $2,444$ vezes mais rápida do que OPF^* (é válido lembrar que a abordagem OPF^* compreende OPF tradicional utilizando $\mathscr{Z}_1 \cup \mathscr{Z}_v$ para treinamento), e $2,288$ vezes mais rápida do que OPF_c.

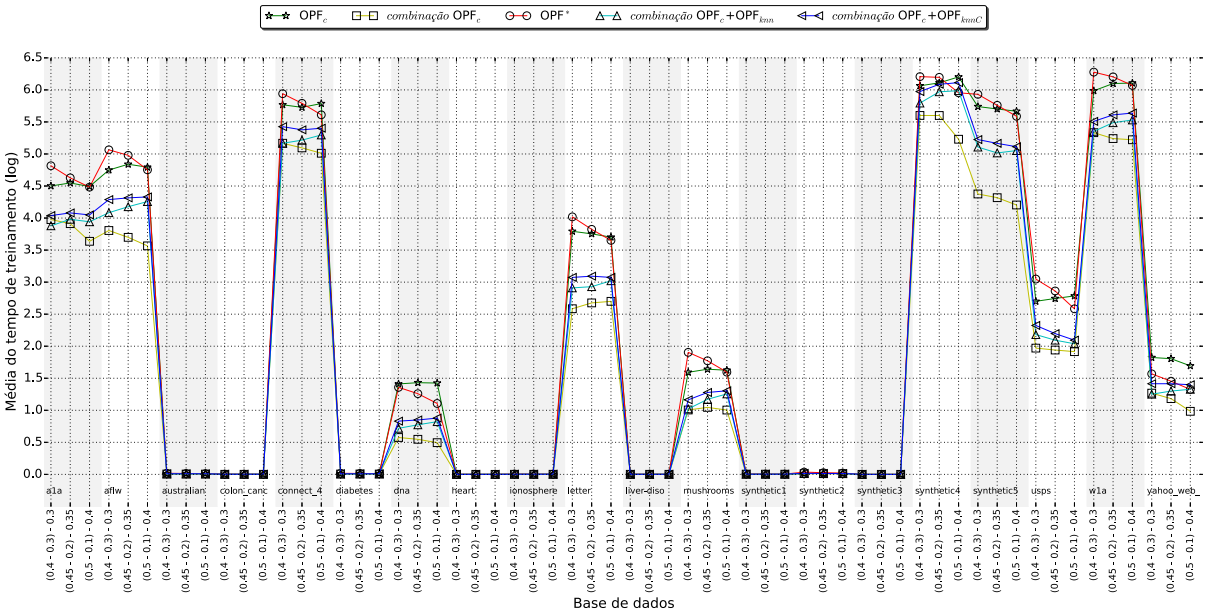
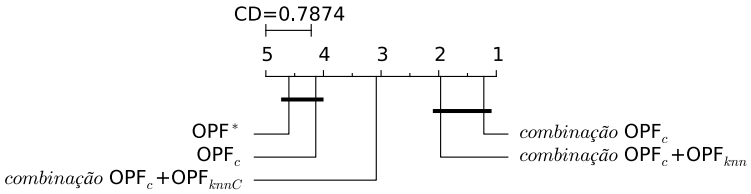
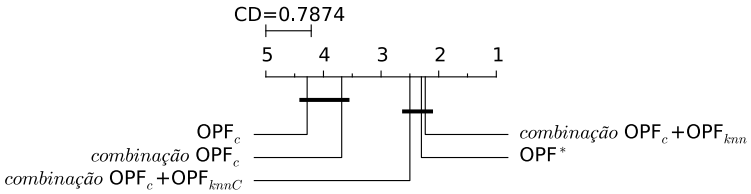


Figura 5.6: Tempo de processamento considerando a fase de treinamento (treinamento com aprendizado dos valores de confiança).

As análises estatísticas de Nemenyi para o tempo de treinamento e de teste são apresentadas na Figura 5.7a e 5.7b, respectivamente.



(a)



(b)

Figura 5.7: Teste estatístico de Nemenyi com relação ao tempo de processamento para (a) treinamento (treinamento com aprendizado dos valores de confiança) e (b) teste. Grupos similares ($p = 0,05$) são conectados.

A respeito da Figura 5.7a, o teste estabelece a técnica combinação OPF$_c$ como a mais rápida mas, no entanto, não apresentou diferença crítica em comparação com a combinação OPF$_c$ + OPF$_{knn}$. Em seguida, a técnica de combinação OPF$_c$ + OPF$_{knnC}$ mostrou-se com desempenho intermediário e, por fim, no último grupo, as técnicas OPF$_c$ e OPF* apresentaram-se

como a mais custosas dentre todas as comparadas. Na sequência, com relação à Figura 5.7b, em média, OPF* foi aproximadamente $1,267$ vezes mais rápido do que a combinação OPF$_c$ na fase de teste, visto que existe mais de um classificador para executar a etapa de classificação. Entretanto, o teste de Nemenyi não apontou diferenças quando combinados OPF$_c$ + OPF$_{knn}$ (abordagem pontuada como a mais rápida para classificação). Sendo assim, alguns conclusões podem ser elencadas:

- a abordagem proposta demonstrou melhorias significativas utilizando a combinação de classificadores OPF com aprendizado por níveis de confiança; e
- o modelo proposto fornece uma etapa de treinamento menos custosa.

5.3 Conclusões

Nesse capítulo foi introduzida a ideia de combinar classificadores OPFs apoiados em níveis de confiança baseados em pontuações. A premissa é construir um conjunto de classificadores que utilizam o conceito de aprendizado baseado em confiança proposto por Fernandes et al. (FERNANDES et al., 2015), ou seja, queremos explorar a combinação de classificadores por meio da votação por maioria enquanto valores de confiança indicam a confiabilidade de uma amostra de treinamento sobre um conjunto de validação. A abordagem proposta foi avaliada em duas variantes do classificador OPF.

Testes empíricos sobre vinte bases de dados evidenciaram resultados relevantes, sendo que a abordagem proposta obteve os melhores resultados em treze das vintes bases de dados de acordo com teste de Wilcoxon (utilizando uma significância de $0,05$). Adicionalmente, a técnica apresentada nesse capítulo mostrou-se mais rápida na fase de treinamento quando comparada com o modelo tradicional OPF (OPF*) e a abordagem OPF$_c$. Além disso, a abordagem por combinação OPF$_c$ + OPF$_{knn}$ evidenciou a melhor relação custo-benefício entre eficácia e eficiência. Em suma, as principais contribuições desse capítulo foram:

- apresentar uma abordagem de combinação de classificadores OPF utilizando aprendizado por níveis de confiança para amostras de treinamento e votação por maioria; e
- propor o uso de confiança para o classificador OPF com grafo k-nn.

Capítulo 6

Poda de Conjunto de Classificadores de Floresta de Caminhos Ótimos Utilizando Otimização Baseada em Quatérnions

Este capítulo tem por objetivo apresentar um estudo sobre combinação de classificadores OPF utilizando poda de conjunto de acordo com abordagens baseadas em meta-heurísticas, conforme descrito por Fernandes e Papa (FERNANDES; PAPA, 2017b).

O uso de múltiplos classificadores tornou-se uma área de grande interesse no reconhecimento de padrões, sendo motivado principalmente devido à semelhança na inclusão de outros classificadores durante o processo de decisão como uma boa abordagem para melhorar a capacidade de generalização do conjunto. No entanto, a combinação de um grande número de classificadores requer uma grande quantidade de memória ao preço de uma fase de classificação mais lenta. Uma forma de acelerar tal processo é a seleção de um subconjunto dos classificadores do conjunto inicial. Tal abordagem, também conhecida como **poda de conjunto** (*ensemble pruning*), tem proporcionado vários benefícios em diferentes abordagens (ZHOU; WU; TANG, 2002; LI et al., 2009). Normalmente, grandes conjuntos de classificadores podem compreender modelos de alta e baixa eficácia (MARKATOPOULOU; TSOUMAKAS; VLAHAVAS, 2010). Assim, podar esses modelos de baixa eficácia para um determinado problema de classificação e manter os modelos restantes, podem contribuir para melhorar o desempenho de classificação global de todo o conjunto. Em geral, a poda de conjunto procura selecionar o subconjunto mais adequado de classificadores em favor da eficiência pela redução do tamanho do conjunto antes da combinação dos modelos.

É válido destacar que encontrar o subconjunto ideal é um problema difícil, cuja solução é computacionalmente custosa (MARTÍNEZ-MUÑOZ; HERNÁNDEZ-LOBATO; SUÁREZ, 2009). Por-

tanto, uma forma interessante de lidar com esse problema é modelar a poda de conjunto como uma tarefa de otimização guiada por modelos meta-heurísticos, tal como por meio de Programação Genética (ZHOU; WU; TANG, 2002). Em geral, é esperado que a programação genética possa proporcionar uma seleção quase-ótima do conjunto com uma redução significativa do número de classificadores. Uma série de trabalhos que também empregaram meta-heurísticas para podar conjuntos de classificadores, tais como Krawczyk (KRAWCZYK, 2015) e Jodavi et al. (JODAVI; ABADI; PARHIZKAR, 2015), têm apresentado resultados promissores.

Com relação ao classificador OPF, há poucos trabalhos que lidam com o problema da combinação (PONTI; PAPA, 2011; PONTI; PAPA; LEVADA, 2011). Além disso, até o momento, não foi observado qualquer trabalho que se proponha realizar poda de conjunto para classificadores baseados em OPF. Portanto, as principais contribuições deste capítulo são: (i) investigar o uso de métodos de poda de classificadores com base em técnicas de meta-heurísticas considerando o classificador OPF, bem como (ii) avaliar a eficácia e a eficiência da poda de conjunto quando modelada como uma tarefa de otimização em um espaço guiado por quatérnions. Nesse trabalho, foram avaliados cinco diferentes algoritmos de otimização para poda de conjunto, sendo eles: Otimização por Enxame de Partículas (KENNEDY; EBERHART, 2001), Busca Harmônia (GEEM, 2009), Busca Harmônica baseada em Quatérnion (*Quaternion-based Harmony Search* - QHS) (PAPA et al., 2016), Busca baseada no reprodução dos Pássaros Cuco (*Cuckoo Search* - CS) (YANG; DEB, 2010) e Algoritmo do Vagalume (*Firefly Algorithm* - FFA) (YANG, 2010).

O restante do capítulo está organizado da seguinte forma. A Seção 6.1 apresenta a fundamentação teórica dos quatérnions, e a Seção 6.2 discute a abordagem proposta para poda de conjunto. Já a Seção 6.3 descreve a metodologia e os resultados experimentais e, finalmente, a Seção 4.3 apresenta as conclusões.

6.1 Álgebra dos Quatérnions

Um quatérnion q é composto de números reais e complexos, ou seja, $q = x_0 + x_1 i + x_2 j + x_3 k$, onde $x_0, x_1, x_2, x_3 \in \Re$ e i, j, k são números imaginários conforme o conjunto de equações:

$$ij = k,\ jk = i,\ ki = j,\ ji = -k,\ kj = -i, \tag{6.1}$$

e

$$ik = -j,\ i^2 = j^2 = k^2 = 1. \tag{6.2}$$

A grosso modo, um quatérnion q é representado por um espaço quadridimensional sobre os números reais, ou seja, $\Re^4$. Na verdade, podemos considerar somente os números reais, uma vez que boa parte das aplicações não consideram a parte imaginária, como o abordado neste trabalho.

Dados dois quatérnions $q_1 = x_0 + x_1 i + x_2 j + x_3 k$ e $q_2 = y_0 + y_1 i + y_2 j + y_3 k$, a sua álgebra é regida por um conjunto de operações principais (EBERLY, 2002), tal como adição e subtração, dentre outras. A adição, por exemplo, pode ser definida como:

$$\begin{aligned} q_1 + q_2 &= (x_0 + x_1 i + x_2 j + x_3 k) + (y_0 + y_1 i + y_2 j + y_3 k) \\ &= (x_0 + y_0) + (x_1 + y_1)i + (x_2 + y_2)j + (x_3 + y_3)k, \end{aligned} \tag{6.3}$$

enquanto que a subtração é definida como segue:

$$\begin{aligned} q_1 - q_2 &= (x_0 + x_1 i + x_2 j + x_3 k) - (y_0 + y_1 i + y_2 j + y_3 k) \\ &= (x_0 - y_0) + (x_1 - y_1)i + (x_2 - y_2)j + (x_3 - y_3)k. \end{aligned} \tag{6.4}$$

No entanto, a operação mais importante usado neste trabalho é a **norma**, a qual mapeia um determinado quatérnion para um número de valor real, como se segue:

$$\begin{aligned} N(q_1) &= N(x_0 + x_1 i + x_2 j + x_3 k) \\ &= \sqrt{x_0^2 + x_1^2 + x_2^2 + x_3^2}. \end{aligned} \tag{6.5}$$

A operação acima é muito importante e será discutida posteriormente na Seção 6.3 para realizar as otimizações com quatérnions.

Além disso, Fister et al. (FISTER et al., 2013, 2015) introduziram duas outras operações, *qrand* e *qzero*. A primeira operação inicializa um determinado quatérnion com valores retirados de uma distribuição Gaussiana, conforme definida a seguir:

$$qrand() = \{x_i = \mathcal{N}(0,1) | i \in \{0,1,2,3\}\}. \tag{6.6}$$

A segunda operação inicializa um quatérnion com valores iguais a zero, conforme definida a seguir:

$$qzero() = \{x_i = 0 | i \in \{0,1,2,3\}\}. \tag{6.7}$$

Embora existam outras operações, essas são consideradas as principais.

6.2 Poda de Conjunto Utilizando Otimização por Meta-heurísticas

Considere um conjunto de L classificadores OPFs utilizando a abordagem *bagging*, ou seja, os classificadores são agregados utilizando diferentes amostras de dados para treinamento. No entanto, em vez de considerar as saídas de todos os classificadores, o conceito de poda consiste em selecionar um subconjunto $\mathscr{D}' \subset \mathscr{D}$ tal que a taxa de reconhecimento sobre o conjunto de validação seja maximizada. Considere $\mathscr{Z}_v \subset \mathscr{Z}_1$ um subconjunto de validação, e $\hat{\mathscr{Z}}_1 = \mathscr{Z}_1 \setminus \mathscr{Z}_v$ como um conjunto a ser derivado em L partes com reposição tal que $\hat{\mathscr{Z}}_1 = \hat{\mathscr{Z}}_1^1 \cup \hat{\mathscr{Z}}_1^2 \cup \ldots \cup \hat{\mathscr{Z}}_1^L$. A principal ideia é treinar cada classificador D_j sobre $\hat{\mathscr{Z}}_1^j$ e, em seguida, classificar $\mathscr{Z}_v$ usando a votação por maioria. Em essência, o objetivo é transformar em "habilitado" ou "inabilitado" cada classificador em $\mathscr{D}$ para construir $\mathscr{D}'$ e, por conseguinte, classificar $\mathscr{Z}_v$. A tarefa de considerar ou não um determinado classificador é realizada pela técnica de meta-heurística, que visa essencialmente aprender qual classificador será "habilitado" ou "inabilitado". Portanto, essa escolha será guiada pela acurácia máxima sobre $\mathscr{Z}_v$.

Para guiar o processo de escolha de qual classificador será selecionado ou não, é associado um peso $w_j \in [0,1]$ para cada um deles, sendo posteriormente selecionado para compor $\mathscr{D}'$ se o seu peso $w_j > \tau$, onde τ representa um limiar adaptativo (LARKINS; MAYO, 2008) que é atualizado da seguinte forma:

$$\tau = \rho - \sigma, \tag{6.8}$$

onde ρ representa a média dos pesos e σ é calculado como segue:

$$\sigma = \sqrt{\frac{1}{M}\sum_{j=1}^{M}(w_j - \rho)^2} \quad \forall w_j < \rho, \tag{6.9}$$

em que M é o número de classificadores cujo peso é menor do que ρ. Observe que a Equação 6.9 considera um classificador D_j somente quando o seu peso w_j é menor do que ρ. Em suma, a proposta deste capítulo é modelar cada possível solução do espaço de busca como um conjunto de pesos, e a técnica de otimização tem por finalidade encontrar os melhores valores para esses pesos que maximizam a eficácia do OPF sobre $\mathscr{Z}_v$.

6.2.1 Otimização com Quatérnions

Geralmente, espera-se que a otimização por quatérnions possa produzir funções de aptidão (*fitness landscapes*) mais suaves, projetando, assim, superfícies com menos ótimos locais (PAPA

et al., 2016). A ideia é modelar cada variável real de decisão a ser otimizada (isto é, cada peso dos classificadores no conjunto) como sendo uma outra variável de valor real, mas agora em $\Re^4$, uma vez que um quatérnion compreende quatro variáveis reais, como discutido na Seção 6.1. Portanto, temos um **tensor** $T_{4\times L}$ no espaço de otimização para cada solução possível, pois temos um quatérnion quadridimensional para cada um dos L classificadores.

A norma de cada quatérnion (Equação 6.5) é utilizada para mapeá-lo para uma variável de valor real com o objetivo de realizar uma otimização padrão. No entanto, é importante destacar que vamos otimizar os valores no espaço de quatérnions, para depois mapeá-los para um espaço Euclidiano. Isso significa que, inicialmente, estamos interessados em encontrar boas representações no espaço de quatérnions.

6.3 Metodologia e Resultados Experimentais

Para analisar a eficiência e eficácia da proposta que usa poda de conjunto baseada em abordagem meta-heurística utilizando o classificador OPF, a mesma foi comparada com OPF tradicional em seis bases de dados reais e sintéticas[1][2], cujas principais características são apresentadas na Tabela 6.1.

Tabela 6.1: Descrição das bases de dados.

Base de dados	# amostras	# características	# classes
Pima-Indians-Diabetes	768	8	2
Statlog-Australian	690	14	2
Statlog-Heart	270	13	2
Synthetic1	500	2	2
Synthetic2	1.000	2	2
UCI-Breast-Cancer	683	10	2

Com o intuito de obter resultados estatisticamente significativos, três diferentes faixas de treinamento, validação e de teste foram utilizadas: (i) em uma primeira fase, cada conjunto de dados foi dividido em três subgrupos: treinamento com 50%, validação com 10% e o teste com 40%, a seguir indicados como 50:10:40; (ii) em uma segunda fase, os conjuntos de dados foram divididos em 45:20:35; e (iii) na última etapa, os conjuntos de dados foram divididos em 40:30:30. Para cada intervalo, os conjuntos de treinamento, validação e de teste foram selecionados aleatoriamente e o processo foi repetido 20 vezes via validação cruzada[3].

[1] http://archive.ics.uci.edu/ml

[2] https://www.csie.ntu.edu.tw/∼cjlin/libsvmtools/datasets/

[3] As porcentagens foram empiricamente escolhidas, sendo mais intuitivo fornecer um conjunto de validação maior para o aprendizado da poda de conjunto.

Para garantir uma comparação justa, OPF tradicional foi treinado sobre $\mathscr{Z}_1 \cup \mathscr{Z}_v$ considerando os três estágios mencionados anteriormente, denotado aqui por OPF*. Além disso, foram calculadas a média dos resultados (acurácia) e o tempo de processamento para cada técnica comparada. Observe que a intenção em usar diferentes porcentagens de $\mathscr{Z}_1$, $\mathscr{Z}_v$ e $\mathscr{Z}_2$ é motivada pelo intuito de investigar a eficácia da abordagem proposta em diferentes cenários. Ademais, os resultados finais foram avaliados utilizando o teste de Wilcoxon (*signed-rank*) com significância de $0,05$ (WILCOXON, 1945). Para a implementação dessa proposta, foram utilizadas as bibliotecas de código aberto LibOPF[4] e LibOPT[5].

Para avaliar a influência de alguns tamanhos iniciais de conjuntos de classificadores, realizou-se um comparativo com 3, 5, 7 e 9 classificadores. Além disso, no que diz respeito à otimização por meta-heurística, optou-se por empregar as seguintes técnicas:

- **Busca Harmônica**: utilizando 5 harmonias com 20 iterações, $HMCR = 0,7$, $PAR = 0,7$ e $\beta = 10$. As variáveis *HMCR* e *PAR* representam, respectivamente, a "Taxa de Consideração da Memória Harmônica" (*Harmony Memory Considering Rate*) e "Taxa de Ajuste de Pitch" (*Pitch Adjusting Rate*), as quais são usadas para guiar HS dentro do espaço de busca e também para evitar ótimos locais.

- **Busca Harmônica baseada em Quatérnions**: uma versão aprimorada da técnica HS que utiliza quatérnions (PAPA et al., 2016). Tal variante tem a mesma configuração de entrada usada na HS.

- **Otimização por Enxame de Partículas**: utilizando 5 partículas com 20 iterações, $c_1 = 1,4$, $c_2 = 0,6$ e $\xi = 0,7$. Considerando que c_1 e c_2 são parâmetros *ad-hoc*, ξ representa o "peso da inércia" que é usado como uma escala em direção das melhores soluções.

- **Algoritmo do Vaga-Lume**: utilizando como tamanho da população igual a 5 com 20 iterações, $\gamma = 0,3$ e $\mu = 1,0$. As variáveis γ e μ são usadas para controlar a aleatoriedade e a atratividade, respectivamente.

- **Busca baseada no reprodução dos Pássaros Cuco**: com 5 partículas e 20 iterações, e $p_a = 0,25$. A variável p_a é usada para controlar o elitismo e a busca local.

A Tabela 6.2 apresenta a média da acurácia e o desvio padrão para cada conjunto de dados, sendo as taxas de reconhecimento computadas de acordo com Papa et al. (PAPA; FALCÃO; SUZUKI, 2009). Os valores em negrito representam as técnicas mais eficazes de acordo com o teste

[4]https://github.com/jppbsi/LibOPF.git
[5]https://github.com/jppbsi/LibOPT.git

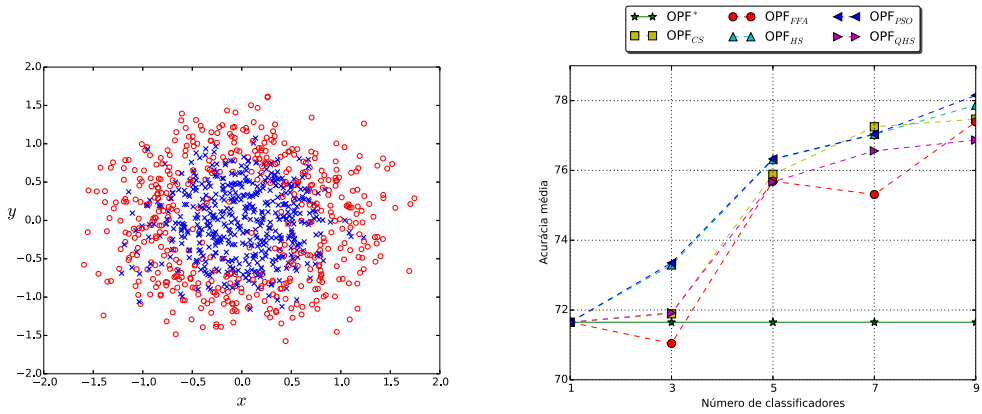
de Wilcoxon (*signed-rank*). Podemos observar que a proposta de poda de conjunto utilizando OPF obteve os melhores resultados em quase todas as bases de dados.

Tabela 6.2: Acurácia média considerando diferentes meta-heurísticas e número de classificadores-base.

Base de dados	**Abordagem**	Conjunto com **3 OPFs**	Conjunto com **5 OPFs**	Conjunto com **7 OPFs**	Conjunto com **9 OPFs**
Pima-Indians-Diabetes	OPF*	$65,50 \pm 2,08$	$65,50 \pm 2,08$	$65,50 \pm 2,08$	$65,50 \pm 2,08$
	OPF_{CS}	$\mathbf{66},\mathbf{87} \pm 2,78$	$\mathbf{68},\mathbf{81} \pm 2,46$	$\mathbf{68},\mathbf{70} \pm 3,30$	$\mathbf{67},\mathbf{65} \pm 2,84$
	OPF_{FFA}	$63,50 \pm 6,01$	$\mathbf{68},\mathbf{14} \pm 5,23$	$66,33 \pm 5,96$	$\mathbf{65},\mathbf{95} \pm 6,00$
	OPF_{HS}	$\mathbf{66},\mathbf{90} \pm 2,61$	$68,73 \pm 2,44$	$\mathbf{68},\mathbf{39} \pm 3,23$	$\mathbf{68},\mathbf{26} \pm 2,52$
	OPF_{PSO}	$\mathbf{67},\mathbf{33} \pm 3,39$	$\mathbf{69},\mathbf{30} \pm 2,72$	$\mathbf{68},\mathbf{09} \pm 3,08$	$\mathbf{67},\mathbf{72} \pm 3,41$
	OPF_{QHS}	$\mathbf{66},\mathbf{90} \pm 2,77$	$67,07 \pm 3,21$	$67,67 \pm 2,96$	$66,88 \pm 2,78$
Statlog-Australian	OPF*	$\mathbf{77},\mathbf{96} \pm 2,04$	$77,96 \pm 2,04$	$77,96 \pm 2,04$	$77,96 \pm 2,04$
	OPF_{CS}	$\mathbf{77},\mathbf{02} \pm 3,21$	$80,65 \pm 2,52$	$\mathbf{82},\mathbf{02} \pm 1,83$	$\mathbf{83},\mathbf{57} \pm 1,79$
	OPF_{FFA}	$75,59 \pm 6,19$	$78,38 \pm 7,38$	$\mathbf{81},\mathbf{66} \pm 4,67$	$80,38 \pm 7,57$
	OPF_{HS}	$\mathbf{78},\mathbf{20} \pm 3,00$	$\mathbf{81},\mathbf{71} \pm 1,93$	$\mathbf{82},\mathbf{82} \pm 2,30$	$\mathbf{83},\mathbf{52} \pm 2,65$
	OPF_{PSO}	$76,64 \pm 3,06$	$81,18 \pm 2,28$	$\mathbf{82},\mathbf{40} \pm 2,20$	$\mathbf{83},\mathbf{86} \pm 2,16$
	OPF_{QHS}	$75,94 \pm 2,95$	$\mathbf{81},\mathbf{28} \pm 1,65$	$\mathbf{81},\mathbf{88} \pm 2,24$	$82,39 \pm 2,68$
Statlog-Heart	OPF*	$74,94 \pm 1,91$	$74,94 \pm 1,91$	$74,94 \pm 1,91$	$74,94 \pm 1,91$
	OPF_{CS}	$\mathbf{78},\mathbf{78} \pm 2,85$	$77,66 \pm 1,63$	$\mathbf{80},\mathbf{12} \pm 3,55$	$\mathbf{81},\mathbf{10} \pm 3,50$
	OPF_{FFA}	$70,89 \pm 6,71$	$\mathbf{79},\mathbf{48} \pm 4,57$	$\mathbf{80},\mathbf{30} \pm 1,59$	$\mathbf{80},\mathbf{64} \pm 3,32$
	OPF_{HS}	$76,18 \pm 2,39$	$\mathbf{80},\mathbf{91} \pm 2,40$	$\mathbf{80},\mathbf{32} \pm 2,81$	$\mathbf{80},\mathbf{91} \pm 2,18$
	OPF_{PSO}	$73,85 \pm 2,99$	$\mathbf{79},\mathbf{76} \pm 2,78$	$\mathbf{80},\mathbf{05} \pm 1,43$	$\mathbf{81},\mathbf{14} \pm 2,28$
	OPF_{QHS}	$76,97 \pm 4,21$	$77,48 \pm 2,73$	$\mathbf{79},\mathbf{53} \pm 3,81$	$\mathbf{80},\mathbf{25} \pm 3,82$
Synthetic1	OPF*	$\mathbf{53},\mathbf{98} \pm 2,89$	$53,98 \pm 2,89$	$53,98 \pm 2,89$	$53,98 \pm 2,89$
	OPF_{CS}	$\mathbf{53},\mathbf{86} \pm 3,62$	$\mathbf{55},\mathbf{84} \pm 2,61$	$\mathbf{55},\mathbf{80} \pm 3,40$	$\mathbf{56},\mathbf{31} \pm 3,02$
	OPF_{FFA}	$\mathbf{54},\mathbf{79} \pm 2,55$	$\mathbf{55},\mathbf{76} \pm 3,76$	$\mathbf{56},\mathbf{04} \pm 3,56$	$54,71 \pm 2,93$
	OPF_{HS}	$52,97 \pm 4,37$	$\mathbf{55},\mathbf{83} \pm 3,21$	$\mathbf{56},\mathbf{33} \pm 3,90$	$\mathbf{55},\mathbf{78} \pm 3,22$
	OPF_{PSO}	$53,38 \pm 4,47$	$\mathbf{54},\mathbf{79} \pm 2,60$	$\mathbf{55},\mathbf{78} \pm 4,26$	$\mathbf{56},\mathbf{29} \pm 2,71$
	OPF_{QHS}	$53,51 \pm 3,18$	$\mathbf{55},\mathbf{52} \pm 3,16$	$\mathbf{55},\mathbf{16} \pm 3,79$	$\mathbf{55},\mathbf{23} \pm 3,69$
Synthetic2	OPF*	$71,65 \pm 2,05$	$71,65 \pm 2,05$	$71,65 \pm 2,05$	$71,65 \pm 2,05$
	OPF_{CS}	$71,89 \pm 2,89$	$\mathbf{75},\mathbf{89} \pm 2,03$	$\mathbf{77},\mathbf{25} \pm 2,19$	$\mathbf{77},\mathbf{47} \pm 2,60$
	OPF_{FFA}	$71,04 \pm 5,90$	$\mathbf{75},\mathbf{69} \pm 3,61$	$75,31 \pm 5,31$	$\mathbf{77},\mathbf{39} \pm 3,34$
	OPF_{HS}	$\mathbf{73},\mathbf{29} \pm 2,43$	$\mathbf{76},\mathbf{31} \pm 2,37$	$\mathbf{77},\mathbf{03} \pm 1,91$	$\mathbf{77},\mathbf{87} \pm 1,81$
	OPF_{PSO}	$\mathbf{73},\mathbf{35} \pm 2,45$	$\mathbf{76},\mathbf{33} \pm 2,42$	$\mathbf{77},\mathbf{04} \pm 2,09$	$\mathbf{78},\mathbf{15} \pm 1,68$
	OPF_{QHS}	$71,92 \pm 2,50$	$\mathbf{75},\mathbf{67} \pm 2,65$	$\mathbf{76},\mathbf{56} \pm 2,29$	$76,87 \pm 2,37$
UCI-Breast-Cancer	OPF*	$\mathbf{94},\mathbf{42} \pm 1,05$	$94,42 \pm 1,05$	$94,42 \pm 1,05$	$94,42 \pm 1,05$
	OPF_{CS}	$92,33 \pm 2,95$	$94,85 \pm 1,63$	$95,50 \pm 1,49$	$95,57 \pm 1,31$
	OPF_{FFA}	$88,11 \pm 14,48$	$92,18 \pm 6,60$	$\mathbf{93},\mathbf{66} \pm 6,29$	$92,73 \pm 6,71$
	OPF_{HS}	$\mathbf{94},\mathbf{10} \pm 2,68$	$\mathbf{95},\mathbf{42} \pm 1,20$	$\mathbf{96},\mathbf{17} \pm 1,19$	$\mathbf{96},\mathbf{09} \pm 0,97$
	OPF_{PSO}	$91,76 \pm 3,36$	$94,94 \pm 1,22$	$\mathbf{96},\mathbf{23} \pm 1,07$	$95,70 \pm 0,87$
	OPF_{QHS}	$91,14 \pm 3,16$	$94,89 \pm 1,87$	$95,05 \pm 2,09$	$95,55 \pm 1,64$

Considere a base de dados "Synthetic2" (Figura 6.1), a qual apresenta uma região com elevado grau de sobreposição entre amostras de classes diferentes. Nesse caso, a poda de conjunto foi consideravelmente mais eficaz. Por conseguinte, a utilização de múltiplos classificadores pode ser mais eficaz quando aplicados a problemas mais complexos.

Além do teste de Wilcoxon utilizado para destacar as diferenças das técnicas comparadas



(a) (b)

Figura 6.1: Problema de classificação gerado sinteticamente: (a) com a representação gráfica de todas as amostras, e (b) seu desempenho preditivo comparando a poda de conjunto utilizando o OPF com diferentes abordagens de meta-heurísticas contra o OPF tradicional (OPF*).

na Tabela 6.2, utilizou-se, também, o teste estatístico de Friedman para ranquear as técnicas, bem como o teste de Nemenyi para indicar a diferença crítica entre as abordagens. Na Figura 6.2, é apresentada a análise estatística considerando os resultados da acurácia para poda de conjunto usando nove classificadores. Conforme apontado pelo teste estatístico de Nemenyi, as abordagens que utilizam poda são consideradas as mais eficazes. Isso significa que a poda de conjunto utilizando o classificador OPF mostrou-se eficaz na maioria dos conjuntos de dados analisados. Além disso, a abordagem que utiliza a meta-heurística HS pode ser considerada a técnica mais eficaz para realizar o corte do conjunto de classificadores, isto é, selecionar um subconjunto ótimo de classificadores. Entretanto, o teste estatístico não apontou uma diferença crítica entre as meta-heurísticas HS, PSO e CS no primeiro grupo, ou seja, isso significa que elas apresentaram um comportamento semelhante.

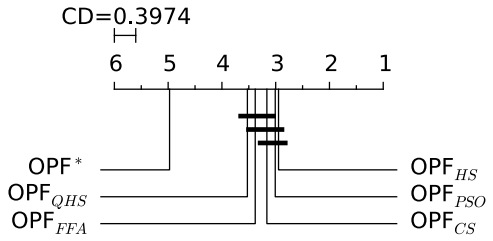
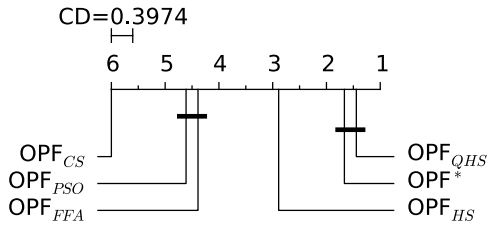


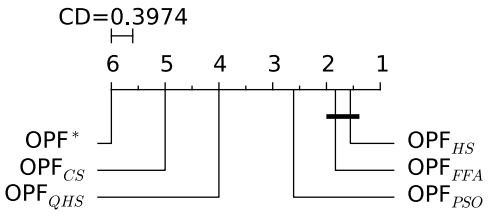
Figura 6.2: Teste estatístico de Nemenyi comparando OPF tradicional (OPF*) e OPF utilizando poda de conjunto para nove classificadores por meio das técnicas meta-heurísticas de acordo com a acurácia.

Na Figura 6.3a é apresentado o tempo de processamento exigido para o treinamento (treinamento e validação) de acordo com o teste de Nemenyi para poda de conjunto usando nove clas-

sificadores. Dentre as técnicas consideradas mais rápidas na etapa de treinamento e validação, a abordagem tradicional OPF e a poda de conjunto utilizando a técnica QHS foram as melhores pontuadas, sendo que elas não apresentam uma diferença crítica; logo, são similares. Em média, a poda usando QHS foi cerca de $1,351$ vezes mais rápida do que OPF tradicional na etapa de treinamento e validação, uma vez que a abordagem QHS mostrou uma rápida convergência quando comparada com as outras técnicas meta-heurísticas, bem como foi mais rápida para o treinamento em comparação com a abordagem tradicional (observe que um conjunto de treinamento maior - $\mathscr{Z}_1 \cup \mathscr{Z}_v$ - no OPF tradicional requer um esforço computacional maior). Com relação à etapa de teste, tanto HS quanto FFA pontuaram como as mais rápidas, uma vez que ambas técnicas "podam" mais classificadores em relação ao conjunto original, isto é, um menor número de classificadores são empregados na etapa de teste. Cabe destacar que a abordagem QHS demostrou resultados interessantes no teste estatístico (Figura 6.2) de acordo com a taxa de reconhecimento, além de ser considerada a mais rápida na etapa de treinamento e validação. Além disso, QHS foi cerca de $1,525$ vezes mais rápida do que OPF tradicional na etapa de teste.



(a)



(b)

Figura 6.3: Teste estatístico de Nemenyi sobre o tempo de processamento para: (a) etapa de treinamento (treinamento e validação) e (b) etapa de teste.

6.4 Conclusões

Neste trabalho, foi introduzida a ideia de poda de conjunto considerando o classificador OPF, bem como uma técnica baseada em quatérnions. A grosso modo, a ideia é construir um conjunto reduzido de classificadores OPFs guiados por algoritmos meta-heurísticos de otimização, tais como HS, PSO, CS, FFA e QHS. Testes empíricos em conjuntos de dados reais e sintéticos evidenciaram que a abordagem proposta alcançou resultados significativos, pontuando-se com os melhores resultados em quase todas as bases avaliadas. A abordagem proposta demonstrou ganhos relevantes para OPF tradicional na etapa de classificação, do mesmo modo que QHS mostrou a melhor relação custo-benefício entre eficácia e eficiência. Assim sendo, as principais contribuições desse capítulo foram:

- apresentar uma abordagem de poda de classificadores OPF; e
- modelar o problema de poda de conjunto de classificadores como sendo uma otimização no espaço de quatérnions.

Capítulo 7

Poda de Conjunto de Classificadores de Floresta de Caminhos Ótimos Utilizando Otimização Meta-heurística para Classificação de Cobertura da Terra

Este capítulo tem por objetivo apresentar uma extensão do estudo sobre combinação de classificadores OPFs utilizando poda de conjunto (Capítulo 6) para imagens de sensoriamento remoto, conforme descrito por Fernandes et al. (FERNANDES et al., 2017).

Consoante ao Capítulo 6, a ideia de utilizar combinação com poda de conjunto privilegia dois pontos: (i) melhorar a capacidade de generalização do conjunto com a inclusão de mais classificadores; e (ii) proporcionar eficiência e eficácia pela seleção de um subconjunto quase-ótimo de classificadores como uma tarefa de otimização guiada por modelos meta-heurísticos.

Uma outra área de grande interesse que tem se beneficiado da estratégia de poda de conjunto é a classificação de cobertura de terra, uma importante aplicação para sensoriamento remoto de imagens. Alguns estudos demostram melhorias significativas nessa área, como exemplo Abe, Gidudu e Marwal (ABE; GIDUDU; MARWAL, 2010), que utilizaram técnicas de seleção de características para criar uma diversidade de classificadores na estratégia de combinação; Tinoco et al. (TINOCO et al., 2013), que mostraram ser eficiente o uso de combinação de classificadores em relação à análise de cobertura de solo utilizando imagens hiperespectrais; e por fim Li e Yin (LI; YIN, 2013) que propuseram análise variável de componentes independentes Bayesiano juntamente com o as Máquinas de Vetores de Suporte (*Variational Bayesian independent component analysis together with Support Vector Machine* - VBICA-SVM) para imagens de sensoriamento remoto de alta resolução espacial com a finalidade de melhorar sua classificação.

Portanto, as principais contribuições deste capítulo são: (i) investigar o uso de métodos de poda de classificadores com base em técnicas de meta-heurísticas considerando o classificador OPF na área de classificação de imagens de sensoriamento remoto, bem como (ii) avaliar a eficiência e eficácia da poda de conjunto em comparação com a estratégia de combinação de classificadores OPFs (sem poda) e, por fim, (iii) aferir a abordagem proposta em comparação com as SVMs. Para esse estudo, foram avaliados quatro diferentes algoritmos de otimização conhecidos para poda de conjunto: CS (YANG; DEB, 2010), FFA (YANG, 2010), HS (GEEM, 2009) e PSO (KENNEDY; EBERHART, 2001).

O restante do capítulo está organizado da seguinte forma. A Seção 7.1 apresenta a metodologia e os experimentos empregados no contexto de classificação de imagens em sensoriamento remoto. A Seção 7.2 discute a abordagem proposta em comparação com o modelo SVM, e, respectivamente, a Seção 7.3 apresenta as conclusões.

7.1 Metodologia e Resultados Experimentais

A estratégia de poda deste capítulo segue a mesma metodologia apresentada na Seção 6.2 do Capítulo 6, isto é, por meio da abordagem *bagging* construir L classificadores com reposição de modo a encontrar o subconjunto $\mathscr{D}' \subset \mathscr{D}$ utilizando uma tarefa de otimização guiada por modelos meta-heurísticos em que a taxa de reconhecimento sobre o conjunto de validação seja maximizada. A Figura 7.1 exemplifica esta abordagem.

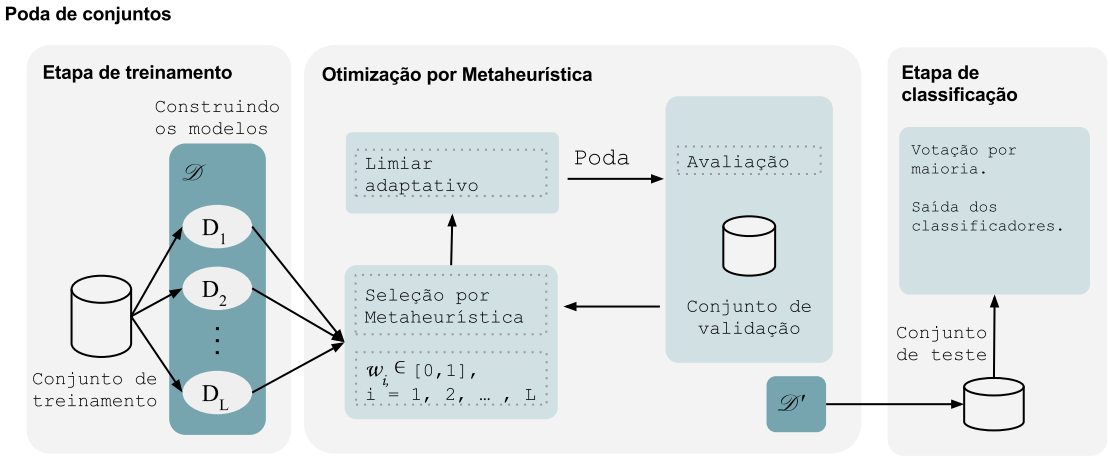


Figura 7.1: Abordagem proposta baseada na poda de conjunto utilizando otimização meta-heurística

Para essa proposta, no contexto de classificação de imagens de sensoriamento remoto, foram utilizadas cinco bases de imagens (PISANI et al., 2014; PISANI, 2016; ROMAY, 2016). As

imagens foram obtidas dos satélites CBERS-2B CCD (20m, sensor R2G3B4) e LANDSAT-5 *Thematic Mapper* - TM (30m, sensor R4G3B5), cobrindo a área de Itatinga, São Paulo - Brasil, e imagens obtidas dos satélites IKONOS-2 *Multispectral* - MS (sensor R4G3B2) e GEOEYE (sensor R5G4B3), cobrindo a área Duque de Caxias, Rio de Janeiro - Brasil. Por fim, utilizou-se o conjunto de dados Indian Pines, o qual foi obtido utilizando o sensor AVIRIS sobre a região Noroeste de Indiana - EUA. As principais características são apresentadas na Tabela 7.1, e nas Tabelas 7.2, 7.3, 7.4, 7.5 e 7.6 são mostradas as descrições das classes das imagens (*ground truth*)[1].

Tabela 7.1: Descrição das bases de imagens de satélite.

Base de dados	# amostras (pixels)	# características	# classes
CBERS-2B	526×492	3	6
LANDSAT-5 TM	526×492	3	6
IKONOS-2 MS	258×250	3	9
GEOEYE	268×250	3	9
Indian Pines	145×145	200	17

Tabela 7.2: Descrição das classes da base CBERS-2B.

Número	Classe	Número de amostras
1	Plantações	71434
2	Arbustos	53499
3	Barragens	605
4	Pastagens	57268
5	Reflorestamento	72866
6	Estradas	3120

Tabela 7.3: Descrição das classes da base LANDSAT-5 TM.

Número	Classe	Número de amostras
1	Plantações	62327
2	Arbustos	47985
3	Pastagens	59890
4	Reflorestamento	85189
5	Barragens	464
6	Estradas	2937

[1]Note que as imagens IKONOS-2 MS e GEOEYE foram obtidas por meio de um processo de fusão entre os sensores multispectral (4m) e pancromático (1m) utilizando o método *pan-sharpening*. A imagem resultante tem uma resolução espacial de 1m.

Tabela 7.4: Descrição das classes da base IKONOS-2 MS.

Número	Classe	Número de amostras
1	Cobertura de árvores	5914
2	Sombras	6481
3	Pastagens	12054
4	Cobertura da tonalidade escura	3578
5	Estradas	22871
6	Solo descoberto - úmido	4417
7	Solo descoberto - claro	7400
8	Cobertura de tonalidade clara	1738
9	Cobertura de tonalidade média	47

Tabela 7.5: Descrição das classes da base GEOEYE.

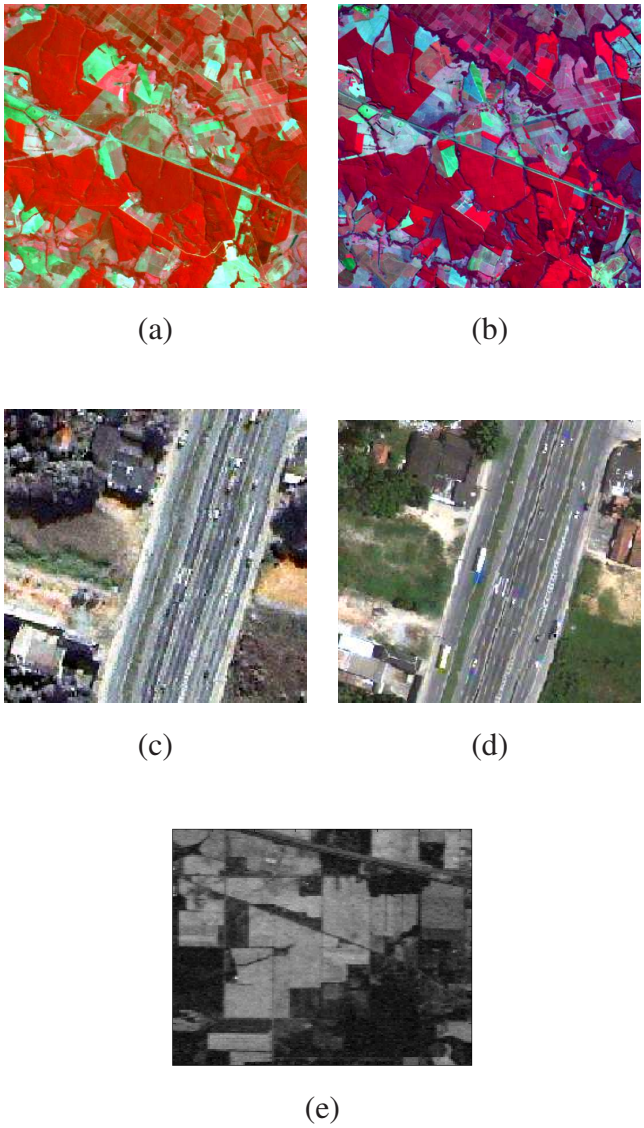
Número	Classe	Número de amostras
1	Solo descoberto - úmido	2380
2	Cobertura de árvores	6132
3	Pastagens	19370
4	Solo descoberto - claro	4490
5	Sombras	2822
6	Cobertura de tonalidade escura	5073
7	Estradas	22924
8	Cobertura de tonalidade clara	1026
9	Cobertura de tonalidade média	2783

Tabela 7.6: Descrição das classes da base Indian Pines.

Número	Classe	Número de amostras
1	Fundo	10776
2	Alfafa	46
3	Milho - Plantio direto (*notill*)	1428
4	Milho - Manipulação mínima de solo (*mintill*)	830
5	Milho	237
6	Pastagens	483
7	Gramas	730
8	Pastagem cortada	28
9	Feno Enrolado	478
10	Aveia	20
11	Soja - Plantio direto (*notill*)	972
12	Soja - Manipulação mínima de solo (*mintill*)	2455
13	Soja	593
14	Trigo	205
15	Mata	1265
16	Edifícios-Grama-Árvores-Condutores	386
17	Pedra-Aço-Torres	93

A Figura 7.2 mostra essas imagens enquanto que a Figura 7.3 mostra as imagens temáticas da verdade terrestre. Os conjuntos de dados foram particionados em três subconjuntos, 20% para treinamento ($\mathscr{Z}_1$), 10% para validação ($\mathscr{Z}_v$) e 70% para teste ($\mathscr{Z}_2$), a seguir denominado

como 20:10:70. Para cada intervalo, os conjuntos de treinamento, validação e de teste foram selecionados aleatoriamente e o processo foi repetido 15 vezes via validação cruzada[2]. Além disso, no que diz respeito aos parâmetros do processo de otimização por meta-heurística, foram utilizados 9 classificadores com 10 harmonias e 20 iterações, com os respectivos parâmetros: para a técnica CS $p_a = 0{,}25$; para a técnica FFA $\gamma = 0{,}3$ e $\mu = 1{,}0$; para a técnica HS $HMCR = 0{,}7$, $PAR = 0{,}7$ e $\beta = 10$; para a técnica PSO $c_1 = 1{,}4$, $c_2 = 0{,}6$ e $\xi = 0{,}7$.



(a) (b)

(c) (d)

(e)

Figura 7.2: Imagens de satélite utilizadas no experimento: cobertura da área de Itatinga, São Paulo - Brasil, adquirida pelo sensor (a) CBERS-2B e pelo sensor (b) LANDSAT-5 TM, cobertura da área de Duque de Caxias, Rio de Janeiro - Brasil, adquirida pelo sensor (c) IKONOS-2 MS e pelo sensor (d) GEOEYE, e (e) cobertura da região Noroeste de Indiana - EUA

[2] As porcentagens foram empiricamente escolhidas

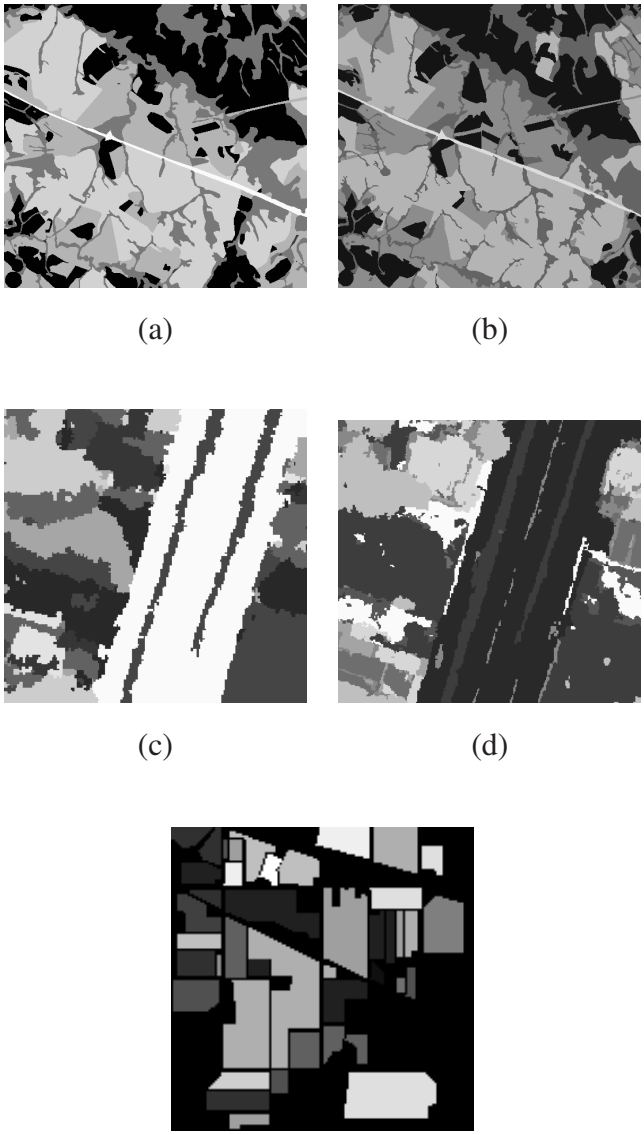


Figura 7.3: Imagens rotuladas usadas no experimento: (a) e (b) correspondem as imagens conforme mostradas na Figura 7.2a e Figura 7.2b, respectivamente, (c) e (d) correspondem as imagens mostradas na Figura 7.2c e Figura 7.2d, respectivamente, e (e) corresponde a imagem mostrada na Figura 7.2e.

Ademais, a proposta foi comparada com o classificador OPF padrão (denominado como OPF*) e também comparada com a combinação de classificadores OPFs sem poda de conjunto utilizando 9 classificadores (denominado como *combinação* OPF). Cabe destacar que essas duas abordagens foram treinadas sobre o conjunto de treinamento e validação, isto é, $\mathscr{Z}_1 \cup \mathscr{Z}_v$.

A Tabela 7.7 apresenta a média da acurácia e o desvio padrão para cada conjunto de dados, sendo as taxas de reconhecimento computadas de acordo com Papa et al. (PAPA; FALCÃO; SUZUKI, 2009). Além disso, na Tabela 7.8 foi empregado outro critério de avaliação para análise

dos resultados, denominada *F-measure*, também conhecida como *F-score*, para uma análise mais refinada (FAWCETT, 2004; GODBOLE; SARAWAGI, 2004). Os valores em negrito representam as técnicas mais eficazes de acordo com o teste de Wilcoxon (*signed-rank*) com significância de $0,05$.

Tabela 7.7: Acurácia média dos resultados (%) e seu desvio padrão para todas as bases considerando o OPF*, *combinação* OPF e poda de conjunto sob diferentes técnicas de otimização. As técnicas mais precisas são destacadas em negrito, conforme teste de Wilcoxon.

Abordagem	**Base de dados**				
	CBERS-2B	GEOEYE	IKONOS-2 MS	LANDSAT-5 TM	Indian Pines
OPF*	$64,28 \pm 0,76$	$71,40 \pm 1,68$	**77,85** $\pm 1,07$	$60,82 \pm 0,78$	**60,22** $\pm 0,38$
combinação OPF	**73,28** $\pm 1,02$	**73,89** $\pm 1,30$	$70,01 \pm 0,62$	$71,35 \pm 0,55$	$56,14 \pm 0,12$
OPF_{CS}	**73,72** $\pm 0,95$	**74,95** $\pm 1,59$	$68,87 \pm 0,62$	**72,17** $\pm 0,48$	$56,02 \pm 0,33$
OPF_{FFA}	**73,26** $\pm 1,93$	**75,17** $\pm 1,57$	$69,02 \pm 0,71$	**72,02** $\pm 0,39$	$56,37 \pm 0,84$
OPF_{HS}	**73,93** $\pm 0,40$	**75,24** $\pm 1,50$	$69,22 \pm 0,55$	**72,16** $\pm 0,63$	$56,10 \pm 0,10$
OPF_{PSO}	**73,73** $\pm 0,59$	**75,06** $\pm 1,58$	$68,97 \pm 0,73$	**72,15** $\pm 0,47$	$56,21 \pm 0,20$

Tabela 7.8: Valores *F-measure* e seu desvio padrão para todas as bases considerando o OPF*, *combinação* OPF e poda de conjunto sob diferentes técnicas de otimização. As técnicas mais precisas são destacadas em negrito, conforme teste de Wilcoxon.

Abordagem	**Base de dados**				
	CBERS-2B	GEOEYE	IKONOS-2 MS	LANDSAT-5 TM	Indian Pines
OPF*	$0,4186 \pm 0,02$	$0,5947 \pm 0,01$	**0,6766** $\pm 0,00$	$0,4139 \pm 0,03$	$0,4114 \pm 0,06$
combinação OPF	$0,6210 \pm 0,02$	**0,7053** $\pm 0,00$	$0,6206 \pm 0,00$	$0,6809 \pm 0,02$	**0,4571** $\pm 0,03$
OPF_{CS}	**0,6365** $\pm 0,02$	**0,7027** $\pm 0,01$	$0,6021 \pm 0,00$	**0,7112** $\pm 0,02$	**0,4599** $\pm 0,00$
OPF_{FFA}	**0,6311** $\pm 0,04$	**0,7035** $\pm 0,01$	$0,6020 \pm 0,01$	**0,7045** $\pm 0,01$	**0,4613** $\pm 0,01$
OPF_{HS}	**0,6430** $\pm 0,01$	**0,7040** $\pm 0,01$	$0,6046 \pm 0,00$	**0,7087** $\pm 0,02$	$0,4416 \pm 0,03$
OPF_{PSO}	**0,6405** $\pm 0,01$	**0,7042** $\pm 0,01$	$0,6016 \pm 0,00$	**0,7122** $\pm 0,01$	**0,4619** $\pm 0,00$

Como pode ser observado, a poda de conjunto OPFs obteve os melhores resultados em alguns conjuntos de dados, exceto para as bases IKONOS-2 MS e Indian Pines, que são melhores classificadas pelo OPF*, como pode ser observado na Tabela 7.7. No entanto, com relação à análise estatística *F-measure*, a poda de conjunto OPFs obteve os melhores resultados em quase todas as bases de dados, exceto o conjunto de dados IKONOS-2 MS, como pode ser observado na Tabela 7.8. Cabe destacar que todas as bases de dados das imagens que foram utilizadas para essa proposta apresentam classes desbalanceadas e, diante disso, as técnicas podem ser melhor aferidas quando avaliadas por outros métodos, tal como *F-measure*. É valido notar que a estratégia de poda de conjunto pode ser mais eficaz quando aplicada em problemas mais

complexos, caso contrário, os resultados podem mostrar baixa generalização dos dados, como observado na base de dados IKONOS-2 MS.

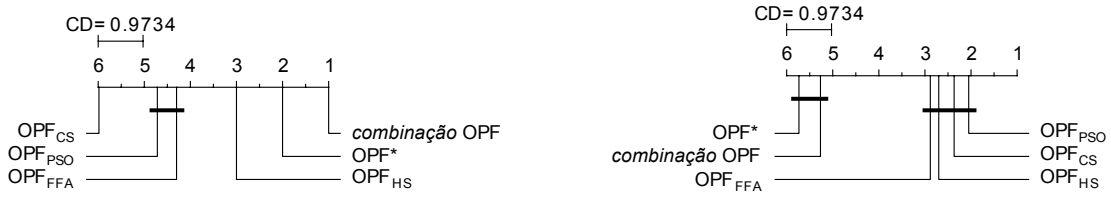
Além do teste de Wilcoxon utilizado para destacar as diferenças das técnicas comparadas nas Tabelas 7.7 e 7.8, utilizou-se, também, o teste estatístico de Friedman para ranquear as técnicas, bem como o teste de Nemenyi para indicar a diferença crítica entre as abordagens. Na Figura 7.4, é apresentada a análise estatística considerando os resultados da acurácia e análise *F-measure* para a abordagem proposta. Conforme apontado pelo teste estatístico de Nemenyi na Figura 7.4a as abordagens que utilizam poda são consideradas as mais eficazes. Isso reflete que a poda de conjunto utilizando classificadores OPFs alcançou uma das melhores taxas de acurácia na maioria dos conjuntos de dados. Além disso, a abordagem HS pode ser considerada a técnica mais precisa. Entretanto, o teste estatístico não apontou uma diferença crítica entre as abordagens meta-heurísticas com a *combinação* OPF e o OPF*, ou seja, isso significa que elas apresentaram um comportamento semelhante. Com respeito aos valores *F-measure* (Figura 7.4b), as técnicas PSO e HS são apontadas como as mais eficazes, destacadas pelo primeiro grupo. Na sequência, no segundo grupo, as técnicas HS, CS, FFA e *combinação* OPF são consideradas similares. Por fim, o OPF* mostrou-se o menos preciso comparado com as outras abordagens considerando os valores de *F-measure*.

(a) (b)

Figura 7.4: Resultados do teste de Nemenyi para uma comparação do OPF* e *combinação* OPF contra a poda de conjunto utilizando classificadores OPFs e suas variações sob diferentes técnicas de otimização com base nos resultados da (a) acurácia e nos valores de (b) *F-measure* em todos os conjuntos de dados de imagem. Grupos similares (p = 0,05) são conectados.

A Figura 7.5a apresenta o tempo de processamento exigido para treinamento e validação de acordo com o teste de Nemenyi. Dentre as técnicas avaliadas, FFA e PSO são consideradas similares e OPF utilizando CS mostrou-se a mais custosa computacionalmente. Na média, a estratégia de poda utilizando HS foi aproximadamente $3,5327$ vezes mais lenta que o OPF* na etapa de treinamento com validação. No que diz respeito ao tempo de classificação (Figura 7.5b), o grupo formado pelas técnicas de otimização pontuaram como as mais rápidas, sendo a técnica PSO a mais eficiente quanto à classificação, visto que um número menor de classificadores em relação ao conjunto original foram considerados "habilitados". E, na sequência,

o último grupo formado pelo OPF* e *combinação* OPF são relacionados como os menos eficientes para etapa de teste. Na média, HS foi cerca de $2,6457$ vezes mais rápido do que o OPF*, ou seja, menos classificadores foram utilizados. É válido destacar que a abordagem utilizando HS demonstrou resultados interessantes no teste estatístico de acordo com a taxa de reconhecimento (Figura 7.4), além de apresentar uma boa eficiência na etapa de treinamento (treinamento com validação) e na etapa de classificação.



(a) (b)

Figura 7.5: Resultados do teste de Nemenyi para uma comparação do OPF* e *combinação* OPF contra a poda de conjunto utilizando classificadores OPFs e suas variações sob diferentes técnicas de otimização com base no tempo de processamento para: (a) etapa treinamento (treinamento e validação) e (b) etapa de teste. Grupos similares (p = 0,05) são conectados.

As Figuras 7.6, 7.7 ,7.8, 7.9 e 7.10 apresentam algumas imagens classificadas com o OPF*, *combinação* OPF e poda de conjunto utilizando HS (uma das técnicas considerada a mais precisa na maioria das bases avaliadas), considerando CBERS-2B, LANDSAT-5 TM, IKONOS-2 MS, GEOEYE e Indian Pines, respectivamente. Além disso, para cada imagem é apresentada a sua matriz de confusão correspondente, o que esclarece que as abordagens por *combinação* OPF e sua versão com "poda" podem melhorar os resultados de classificação. É evidente que os resultados das técnicas que utilizam meta-heurística e *combinação* OPF são mais assertivas do que o OPF* quando se considera as imagens rotuladas (Figura 7.3), exceto para a Figura 7.8 e Figura 7.10 (base KONOS-2 MS e Indian Pines), em que o OPF* foi mais efetivo.

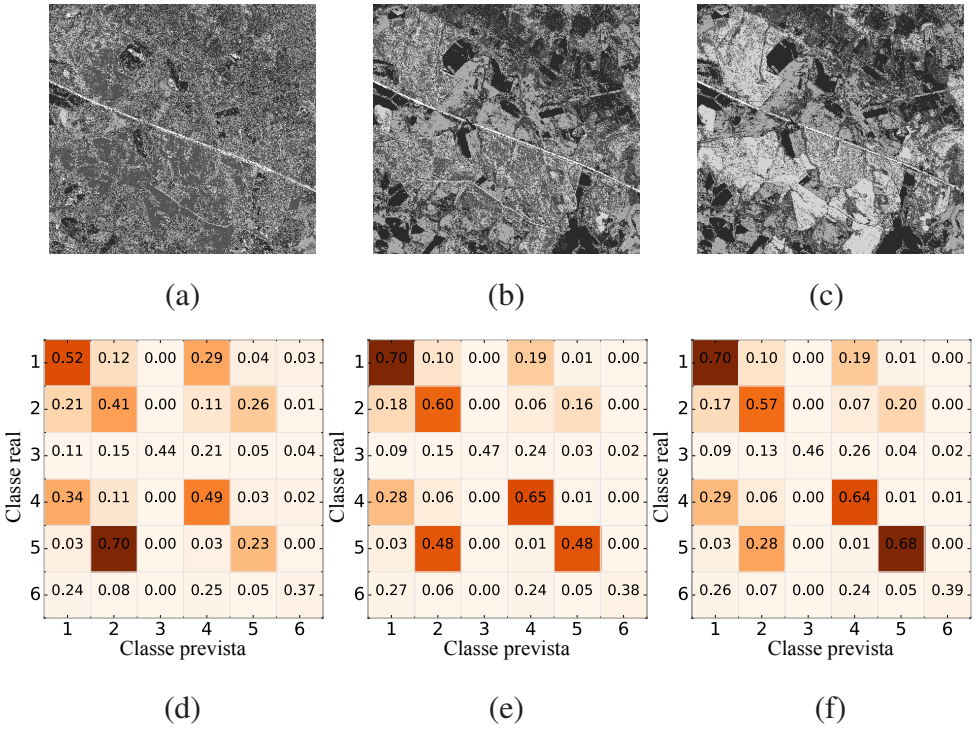


Figura 7.6: Imagens do satélite CBERS-2B classificadas usando (a) OPF*, (b) *combinação* OPF e (c) poda usando HS, e as partes (d), (e) e (f) correspondem as suas matrizes de confusão, respectivamente.

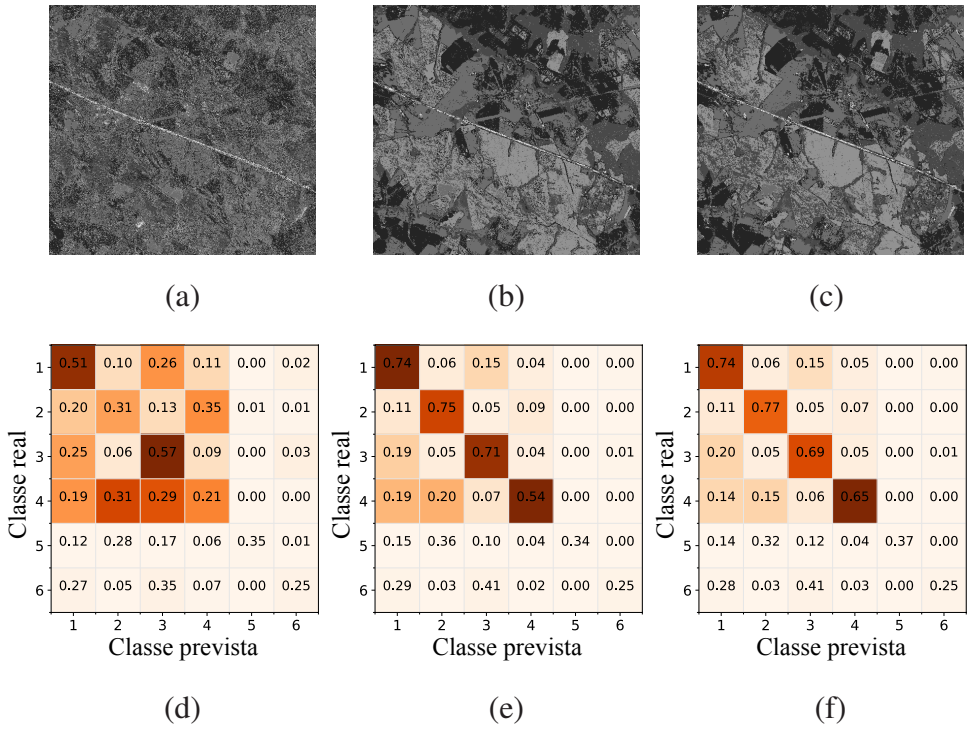


Figura 7.7: Imagens do satélite LANDSAT-5 TM classificadas usando (a) OPF*, (b) *combinação* OPF e (c) poda usando HS, e as partes (d), (e) e (f) correspondem as suas matrizes de confusão, respectivamente.

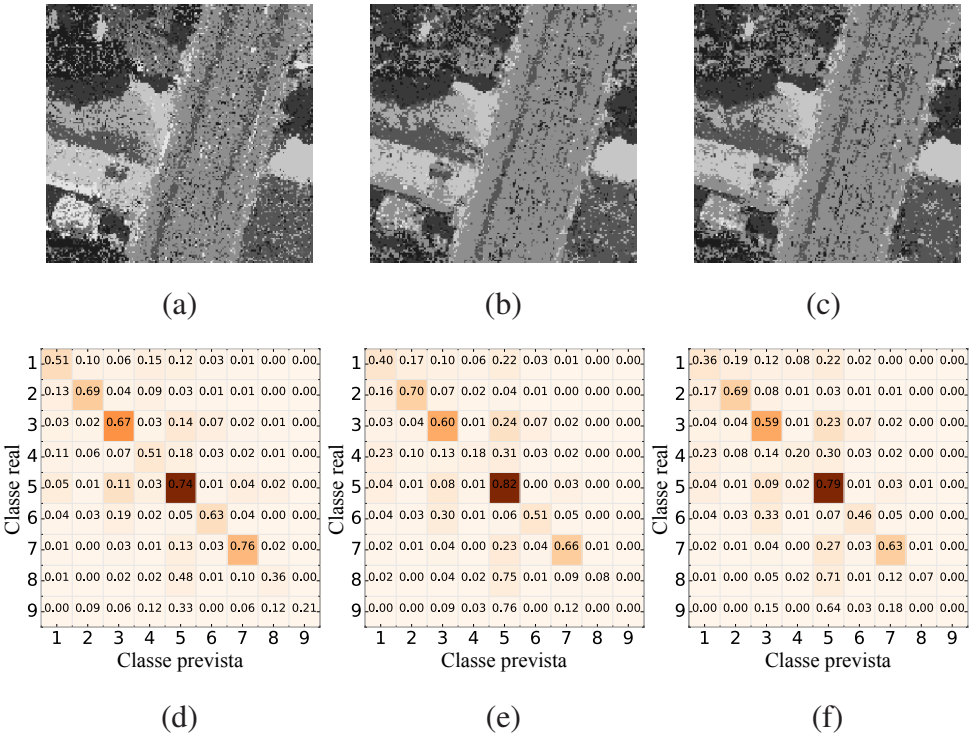


Figura 7.8: Imagens do satélite IKONOS-2 MS classificadas usando (a) OPF*, (b) *combinação* OPF e (c) poda usando HS, e as partes (d), (e) e (f) correspondem as suas matrizes de confusão, respectivamente.

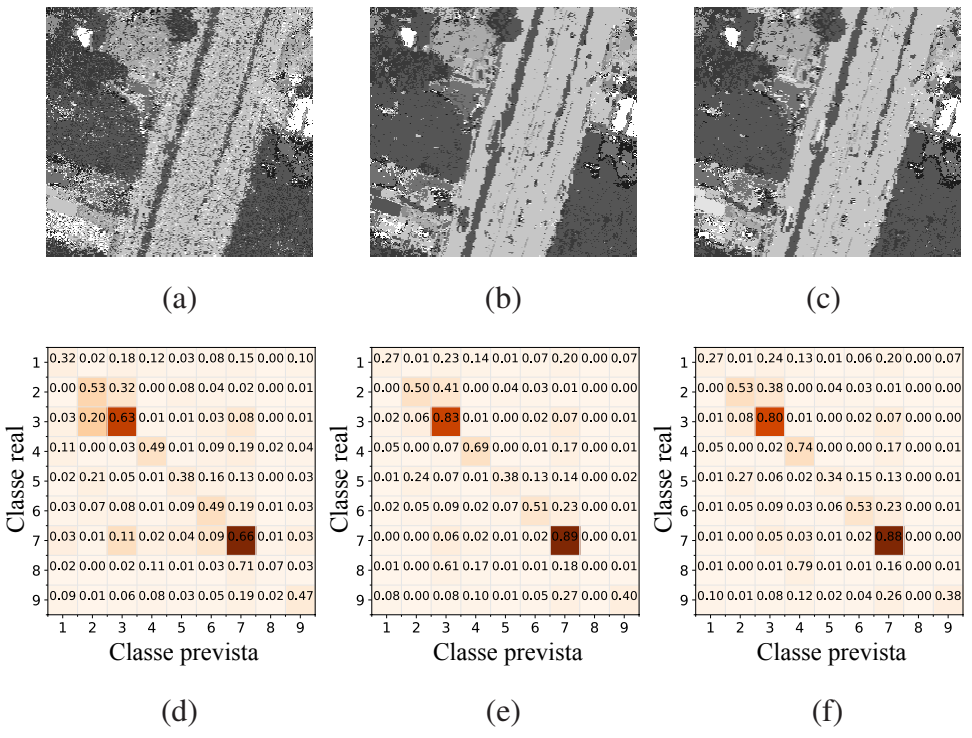


Figura 7.9: Imagens do satélite GEOEYE classificadas usando (a) OPF*, (b) *combinação* OPF e (c) poda usando HS, e as partes (d), (e) e (f) correspondem as suas matrizes de confusão, respectivamente.

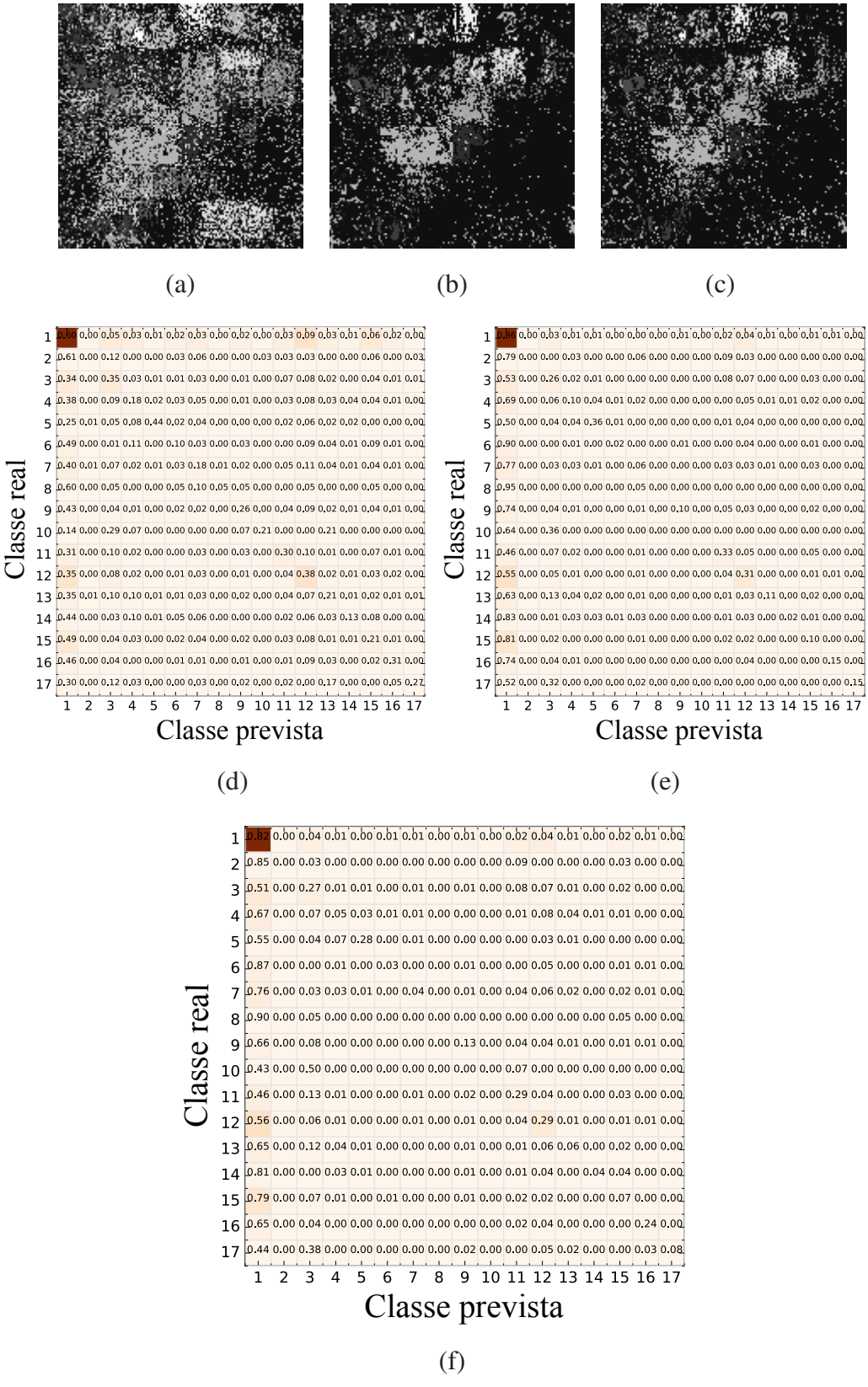
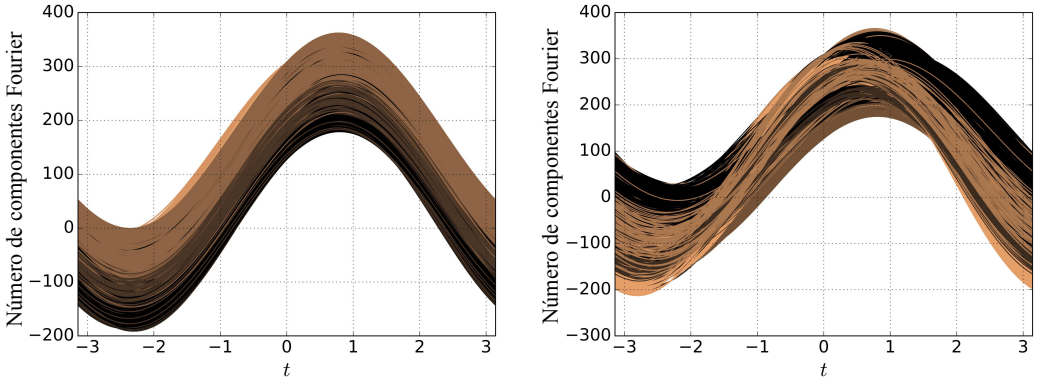


Figura 7.10: Imagens do satélite Indian Pines classificadas usando (a) OPF*, (b) *combinação* OPF e (c) poda usando HS, e as partes (d), (e) e (f) correspondem as suas matrizes de confusão, respectivamente.

Além disso, uma outra análise que pode ser explorada é o método *Andrews curves*, o qual permite compreender a contiguidade dos dados em um espaço n-dimensional e identificar padrões distintos entre as classes, conforme representado pela Figura 7.11. Observe que na Figura 7.11a (base IKONOS-2 MS) os padrões apresentam um comportamento menos complexo que pode ser melhor classificado sem a estratégia de combinação ou poda de conjunto de classificadores, como observado pelo OPF* nas Tabelas 7.7 e 7.8. Na sequência, a Figura 7.11b (base CBERS-2B) apresenta um padrão mais complexo, sendo melhor classificado por meio da estratégia de poda de classificadores. Com base nisso, pode-se estabelecer que conjuntos de dados complexos e com considerável sobreposição de amostras de diferentes classes podem ser melhores classificados pela abordagem de poda usando meta-heurística, isso devido ao subconjunto quase-ótimo de classificadores fornecido pelo processo de otimização.



(a) (b)

Figura 7.11: Representação dos dados utilizando o método *Andrews curves* com respeito à base (a) IKONOS-2 MS e (b) CBERS-2B no intervalo de $-\pi < t < \pi$

7.2 Comparação entre Máquina de Vetores de Suporte e OPF

Além disso, foi comparada a abordagem proposta com a técnica SVM e SVM usando a mesma estratégia de poda de conjunto proposto nesse capítulo. A ideia é realizar o mesmo procedimento usando o processo de otimização por meta-heurística para SVM com kernel RBF. Para cumprir esta finalidade, foi aplicada a mesma metodologia, ou seja, intervalo de 20:10:70 para treinamento, validação e teste dos conjuntos, respectivamente, repetidos 15 vezes (validação cruzada). Os parâmetros para SVM foram otimizados em um conjunto de validação usando uma busca em grade (*grid-search*) ($\gamma \in \{0.001, 0.01, 0.1, 1\}$ e $C \in \{1, 10, 100, 1000\}$). Com relação às técnicas de meta-heurísticas, foi considerada a poda SVM usando apenas HS, uma vez que esta abordagem foi considerada a mais precisa na maioria dos conjuntos de dados. Os parâmetros da HS para a poda de conjunto SVM foram os mesmos usados na Seção 7.1. As Tabelas 7.9 e 7.10 apresentam a taxa de reconhecimento e os resultados dos valores de *F-measure*, respectivamente. Os valores em negrito representam as técnicas mais precisas de acordo com o teste Wilcoxon.

Tabela 7.9: Acurácia média dos resultados (%) e seu desvio padrão para todas as bases considerando o SVM padrão e a estratégia de poda com classificadores OPFs e SVMs utilizando a técnica HS. As técnicas mais precisas são destacadas em negrito, conforme teste de Wilcoxon.

Abordagem	**Base de dados**				
	CBERS-2B	GEOEYE	IKONOS-2 MS	LANDSAT-5 TM	Indian Pines
OPF_{HS}	$73,93 \pm 0,40$	$75,24 \pm 1,50$	**69,22** $\pm 0,55$	$72,16 \pm 0,63$	**56,10** $\pm 0,10$
SVM_{HS}	$76,15 \pm 0,48$	**76,38** $\pm 0,24$	$68,28 \pm 0,33$	$73,92 \pm 0,72$	$50,00 \pm 0,00$
SVM	**76,77** $\pm 0,30$	**76,51** $\pm 0,13$	$68,64 \pm 0,48$	**74,58** $\pm 0,37$	$50,00 \pm 0,00$

Tabela 7.10: Média dos valores de *F-measure* e seu desvio padrão para todas as bases considerando o SVM padrão e a estratégia de poda com classificadores OPFs e SVMs utilizando a técnica HS. As técnicas mais precisas são destacadas em negrito, conforme teste de Wilcoxon.

Abordagem	**Base de dados**				
	CBERS-2B	GEOEYE	IKONOS-2 MS	LANDSAT-5 TM	Indian Pines
OPF_{HS}	**0,6430** $\pm 0,01$	**0,7040** $\pm 0,01$	**0,6046** $\pm 0,00$	**0,7087** $\pm 0,02$	**0,4416** $\pm 0,03$
SVM_{HS}	$0,6311 \pm 0,01$	$0,5884 \pm 0,00$	$0,4307 \pm 0,01$	$0,5449 \pm 0,01$	$0,0399 \pm 0,00$
SVM	**0,6421** $\pm 0,00$	$0,5911 \pm 0,00$	$0,4384 \pm 0,01$	$0,5580 \pm 0,01$	$0,0399 \pm 0,00$

Conforme mostrado pelas Tabelas 7.9 e 7.10, a proposta de poda aplicada ao classificador SVM não apresentou resultados significativos em comparação com as outras abordagens avaliadas, enquanto que o SVM sem poda foi mais bem avaliado considerando a acurácia, como

pode ser observado na Figura 7.12a de acordo com o teste de Nemenyi. Contudo, com relação aos valores de *F-measure* (Tabela 7.10) a abordagem OPF utilizando poda apresentou os melhores resultados em todas as bases de dados avaliadas, como pode ser observado pelo teste de Nemenyi na Figura 7.12b. A baixa generalização das bases IKONOS-2 MS e Indian Pines pode ser devida a baixa complexidade dos conjuntos de dados, bem como um pequeno número de amostras.

(a) (b)

Figura 7.12: Resultados do teste de Nemenyi considerando o SVM padrão e a estratégia de poda com classificadores OPFs e SVMs utilizando a técnica HS para: (a) acurácia e (b) valores d *F-measure*. Grupos similares (p = 0,05) são conectados.

A Figura 7.13 apresenta a matriz de confusão concernente à todas as bases de dados para a abordagem SVM sem poda, visto ser melhor pontuado nos testes estatísticos em comparação com a estratégia de poda utilizando SVM. Como pode ser observado na Figura 7.13e (Indian Pines), o classificador SVM apresentou uma baixa generalização, classificando incorretamente a maioria das amostras. Entretanto, são necessários mais trabalhos para abordar algumas questões importantes relacionadas ao SVM usando a poda de conjunto, como diferentes kernels e outras estratégias.

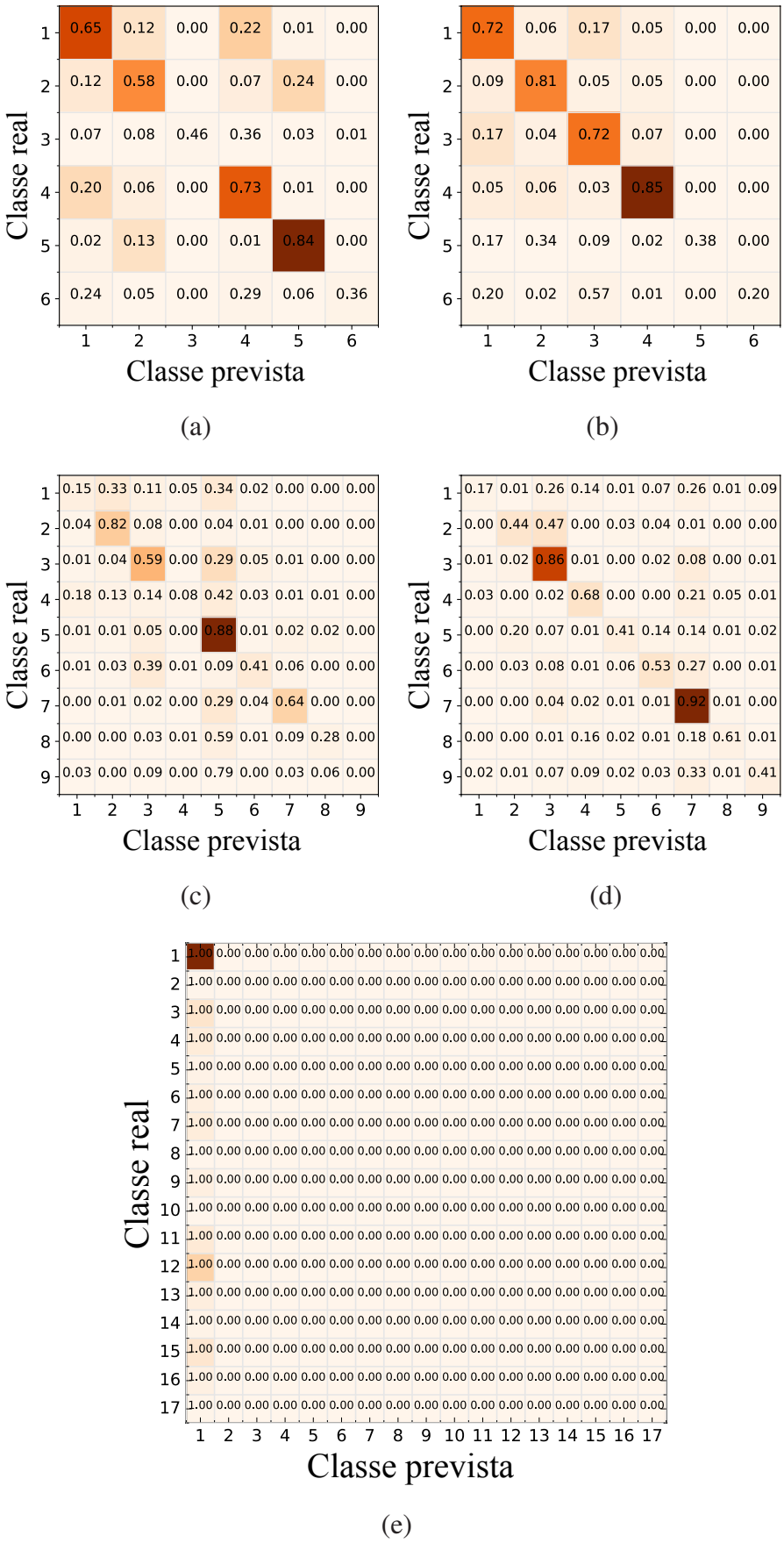


Figura 7.13: Matriz de confusão para SVM sem poda de classificadores para as bases (a) CBERS-2B, (b) LANDSAT-5 TM, (c) IKONOS-2 MS, (d) GEOEYE e (e) Indian Pines, respectivamente.

Com relação à carga computacional, nas Figuras 7.14a e Figuras 7.14b são mostrados o teste estatístico de Nemenyi para o tempo de treinamento (treinamento com validação) e tempo de teste, respectivamente. A etapa de treinamento usando *bagging* com poda de classificadores (SVM_{HS}) mostrou ser eficiente quando comparada ao procedimento de treinamento sobre todo conjunto $\mathscr{Z}_1 \cup \mathscr{Z}_v$ para um único classificador (treinamento do SVM tradicional), como evidenciado na Figura 7.14a. Enquanto isso, na Figura 7.14b é mostrado o teste estatístico para o tempo na etapa de classificação; elencando a abordagem SVM_{HS} como a mais rápida, em segundo lugar a abordagem proposta utilizando classificadores OPFs (OPF_{HS}) e, por último, como a mais custosa computacionalmente desse comparativo, a abordagem SVM tradicional.

(a) (b)

Figura 7.14: Resultados do teste de Nemenyi considerando o SVM padrão e a estratégia de poda com classificadores OPFs e SVMs utilizando a técnica HS para o tempo de (a) treinamento (treinamento com validação) e tempo de (b) teste. Grupos similares (p = 0,05) são conectados.

7.3 Conclusões

Neste capítulo foi apresentada a poda de conjunto considerando o classificador OPF para classificação de imagens de sensoriamento remoto. A ideia consiste em investigar a eficiência e eficácia da proposta de poda de conjunto aplicada em problemas complexos, que requerem um tempo considerável para encontrar um subconjunto adequado. Para esse propósito, utilizou-se a estratégia de *bagging* com nove classificadores OPFs associada a um procedimento de poda modelada como uma tarefa de otimização baseada em algoritmos meta-heurísticos, tal como CS, FFA, HS e PSO. Além disso, para uma análise comparativa da proposta, foi também avaliada a estratégia de poda aplicada em classificadores SVMs.

Conforme investigado nesse capítulo, os testes evidenciaram que a estratégia de poda aplicada em classificadores OPFs mostrou-se significativa, pontuando os melhores resultados em comparação com o OPF padrão e OPF utilizando combinação de classificadores sem poda para a maioria das bases avaliadas. Cabe destacar que a vantagem da abordagem proposta é relevante quando aplicada a problemas considerados complexos com alta sobreposição de amostras. É válido destacar que mesmo as técnicas sendo agrupadas como similares em relação a eficácia,

a abordagem que faz uso da poda com meta-heurística HS foi mais assertiva com um custo computacional menos distante da abordagem tradicional, mostrando um custo-benefício entre eficácia e eficiência.

Com relação ao comparativo utilizando o classificador SVM, foi possível estabelecer que a mesma estratégia de poda aplicada ao classificador SVM não resultou em ganhos significativos em relação ao SVM tradicional. Com relação a média da acurácia, SVM tradicional foi o classificador mais bem pontuado, alcançando os melhores resultados em três bases de cinco. Ademais, considerando os valores de *F-measure*, a proposta de poda utilizando classificadores OPFs foi melhor pontuada do que a estratégia de poda usando SVMs e SVM tradicional, visto que o classificador SVM não apresentou uma boa generalização para algumas bases, como a Indian Pines. Entretanto, conforme destacado anteriormente, são necessários mais trabalhos para abordar algumas questões importantes relacionadas ao classificador SVM com poda de conjunto. Um outro aspecto importante é a ausência de parâmetros do OPF, o que o torna interessante para a generalização dos dados em poda de conjunto para vários problemas. Assim sendo, as principais contribuições desse capítulo foram:

- investigar a estratégia de poda apresentada no Capítulo 6 no contexto de classificação de imagens de sensoriamento remoto; e
- avaliar a eficiência e eficácia do modelo proposto em comparação ao classificador SVM.

Capítulo 8

Classificador de Floresta de Caminhos Ótimos Probabilístico

Este capítulo tem por objetivo apresentar uma nova variante do classificador OPF com saída do tipo confidência, visto que a abordagem OPF utiliza saída do tipo abstrata, conforme descrito por Fernandes et al. (FERNANDES et al., 2017).

Conforme mencionado anteriormente (Seção 2.1.1), a saída dos algoritmos de classificação pode ser categorizada em abstrata, *ranking* e confidência. Classificadores baseados em saídas abstratas compreendem à maioria das técnicas que atribuem um rótulo (geralmente um número inteiro) para cada amostra a ser classificada, enquanto que no modo *ranking* os possíveis rótulos para uma dada amostra são armazenados em uma fila de prioridades de acordo com algum critério. Por outro lado, as técnicas que utilizam confidência atribuem a probabilidade de uma determinada amostra ser rotulada por uma dada classe.

As técnicas que utilizam saídas probabilísticas desempenham um papel importante no aprendizado de máquina, pois estendem o processo de classificação a um alcance maior do que apenas os rótulos. Muitas vezes, depara-se com problemas onde é desejável obter alguma probabilidade. Considerando o problema da identificação de furto em sistemas de distribuição de energia, as companhias de energia elétrica consideram muito mais proveitosas monitorar a probabilidade de um determinado usuário se tornar um possível infrator ao longo do tempo em vez de identificar o transgressor. Com as probabilidades, a empresa pode adotar uma abordagem preventiva ao longo do tempo, o que pode ser mais rentável do que simplesmente punir o cliente.

Além disso, com base no tipo de saída do classificador, diferentes estratégias de combinação podem ser aplicadas, tais como votação e votação ponderada. Cabe destacar que algumas abordagens de combinação de classificadores são baseadas no nível de confidência, como abordado

por Nabavi-Kerizi, Abadi e Kabir (NABAVI-KERIZI; ABADI; KABIR, 2010), que propuseram a combinação linear de Redes Neurais Artificiais utilizando PSO para calcular os pesos da matriz do perfil de decisão. Com a saída do tipo confidência, novas estratégias de combinação podem ser investigadas e ganhos significativos podem ser alcançados.

Felizmente, há uma quantidade considerável de técnicas probabilísticas na literatura. Um trabalho realizado por Platt (PLATT, 1999) ampliou o uso das SVMs, que foram primeiro projetadas para lidar com saídas abstratas, para classificação probabilística. A ideia consiste em usar as saídas do classificador SVM para alimentar uma função logística. Portanto, as saídas iniciais são mapeadas dentro do intervalo $[0,1]$. Contudo, para lidar com problemas relacionados ao número de amostras diferentes por classe (conjunto desbalanceado), o autor utiliza um procedimento de otimização no conjunto de treinamento para encontrar as variáveis que regularizam o processo de mapeamento da probabilidade do rótulo. Essa técnica é, muitas vezes, referida como "Platt Scaling".

Mais tarde, Niculescu-Mizil e Caruana (NICULESCU-MIZIL; CARUANA, 2005) apresentaram uma comparação muito interessante entre Platt Scaling e Regressão Isotônica (*Regression Isotonic*) para obter saídas probabilísticas com SVMs. Seu trabalho foi motivado pelo fato de que as funções logísticas podem funcionar bem para várias situações, mas podem não ser apropriada para outros casos. A grosso modo, a Regressão Isotônica visa aprender uma função que é impelida a crescer de forma monotônica (isotônica), e é alimentada com as saídas dos valores reais do classificador SVM (ou seja, antes de tomar o sinal da função para classificar uma amostra como positiva ou negativa). Os autores concluíram que a Platt Scaling funciona melhor com conjuntos de dados de tamanho pequeno, e como a Regressão Isotônica é mais propensa a *overfitting*, é recomendável que seja aplicada em grandes conjuntos de dados.

Zadrozny e Elkan (ZADROZNY; ELKAN, 2001) propuseram obter estimativas de probabilidade considerando árvores de decisão e classificadores Bayesianos. Com isso, os autores adotaram estimativas mais suaves para as árvores de decisão, ou seja, as ajustaram para serem menos extremas. A suavidade é um meio interessante para lidar com a estimativa de probabilidade, uma vez que em alguns métodos as probabilidades podem ficar fora do intervalo $[0,1]$, e outros ajustam para próximo de $0,5$ (por exemplo, a correção de Laplace), o que pode não ser interessante quando as classes das amostras não são equiprováveis. Em seguida, o mesmo grupo de autores estendeu seu trabalho para lidar com problemas de múltiplas classes (ZADROZNY; ELKAN, 2002). Outros trabalhos recentes podem ser referenciados (NAPOLI et al., 2015; SOUNDARARAJAN; SCHULTZ, 2015; SCHLEIF; GISBRECHT; TINO, 2015), contudo eles se concentram principalmente na aplicação de classificadores probabilísticos ou estudos de comparação e não

em novas teorias ou abordagens.

Assim sendo, a proposta de uma nova variante do algoritmo OPF que trabalha com saídas baseadas em probabilidades preenche uma lacuna importante para o classificador e possibilita novas estratégias de combinação. A proposta, que inicialmente foi projetada para lidar com problemas binários, foi comparada com o OPF padrão e SVM probabilístico em diferentes cenários, mostrando resultados significativos. O restante do capítulo está organizado da seguinte forma: a Seção 8.1 apresenta a base teórica do classificador OPF com abordagem probabilística; a Seção 8.2 descreve a metodologia e os resultados experimentais; e, finalmente, a Seção 8.3 apresenta as conclusões.

8.1 Floresta de Caminhos Ótimos Probabilística

O classificador OPF probabilístico é inspirado na abordagem Platt Scaling, que essencialmente mapeia as saídas do algoritmo SVM para estimativas de probabilidade. Portanto, antes de introduzir a abordagem proposta, é preciso descrever o mecanismo Platt Scaling.

8.1.1 Máquinas de Vetores de Suporte Probabilísticas

Considere o conjunto de dados rotulado $\mathscr{Z}$, em que cada amostra $\mathbf{x}_i \in \mathscr{Z}$ pode ser atribuída à classe $\omega_i \in \{-1,+1\}$, $i = 1,2,\ldots,|\mathscr{Z}|$. Platt propõe aproximar a probabilidade a posteriori $P(\omega_i = 1|\mathbf{x}_i)$, conforme segue (PLATT, 1999):

$$P(\omega_i = 1|\mathbf{x}_i) \approx P_{A,B}(f_i) = \frac{1}{1+\exp(Af_i+B)}, \tag{8.1}$$

em que f_i significa a saída (função de decisão) do algoritmo SVM em relação à amostra $\mathbf{x}_i$. Considere $\theta = (A^*,B^*)$ o melhor conjunto de parâmetros que podem ser determinados pelo seguinte problema de máxima verossimilhança:

$$\arg\min_{\theta} F(\theta) = -\sum_{i=1}^{m}(\omega_i \log(p_i) + (1-\omega_i)\log(1-p_i)), \tag{8.2}$$

em que $p_i = P_{A,B}(f_i)$ e m representa o número de amostras a serem consideradas. Essencialmente, a equação acima representa a função de custo do classificador por Regressão Logística.

Para evitar *overfitting*, Platt propôs regularizar a Equation 8.2 da seguinte maneira:

$$\arg\min_{\theta} F(\theta) = -\sum_{i=1}^{m}(t_i \log(p_i) + (1-t_i)\log(1-p_i)), \tag{8.3}$$

onde t_i é formulado como segue:

$$t_i = \begin{cases} \frac{N_+ + 1}{N_+ + 2} & \text{Se } \omega_i = +1 \\ \frac{1}{N_- + 2} & \text{Se } \omega_i = -1. \end{cases} \tag{8.4}$$

Na formulação acima, N_+ e N_- representam o número de amostras positivas e negativas, respectivamente. Em suma, t_i pode ser usado para lidar com conjuntos de dados desbalanceados.

8.1.2 Máquinas de Vetores de Suporte Probabilísticas Modificada

Aproximadamente uma década depois do trabalho de Platt, Lin et al. (LIN; LIN; WENG, 2007) demostraram algumas instabilidades numéricas relacionadas à Equação 8.3:

- é conhecido que as funções log e exp podem facilmente causar um estouro de memória (*overflow*), uma vez que $\exp(Af_i + B) \to \infty$ quando $Af_i + B$ é grande o suficiente. Além disso, $\log(p_i) \to -\infty$ quando $p_i \to 0$.
- de acordo com Goldberg (GOLDBERG, 1991), $1 - p_i = 1 - \frac{1}{1+\exp(Af_i+B)}$ é um "cancelamento catastrófico" quando p_i é próximo de 1. Esse termo decorre do fato de subtrair dois números relativamente próximos que já são resultados de operações anteriores de ponto flutuante. Lin et al. (LIN; LIN; WENG, 2007) descreveram um exemplo interessante: suponha $f_i = 1$ e $(A,B) = (-64,0)$. Nesse caso, $1 - p_i$ retorna 0, mas a formulação equivalente $\frac{\exp(Af_i+B)}{1+\exp(Af_i+B)}$ fornece um resultado mais preciso. Além disso, o mesmo grupo de autores afirmou que o cancelamento acima mencionado induz a maioria das ocorrências $\log(0)$.

Para lidar com a situação acima mencionada, Lin et al. (LIN; LIN; WENG, 2007) propuseram reformular a função de custo $F(\theta)$ da seguinte maneira:

$$F(\theta) = -\sum_{i=1}^{m}(t_i \log(p_i) + (1-t_i)\log(1-p_i)) \quad (8.5)$$

$$= \sum_{i=1}^{m}((t_i-1)(q_i) + \log(1+\exp(q_i))) \quad (8.6)$$

$$= \sum_{i=1}^{m}(t_i q_i + \log(1+\exp(-Af_i - B))), \quad (8.7)$$

em que $q_i = Af_i + B$. Portanto, considerando a formulação acima, $1-p_i$ e $\log(0)$ não acontecem[1].

No entanto, mesmo usando as Equações 8.6 e 8.7, o problema de *overflow* ainda pode ocorrer. Para lidar com isso, Lin et al. (LIN; LIN; WENG, 2007) propuseram aplicar a Equação 8.7 quando $Af_i + B \geq 0$; caso contrário, deve-se usar a Equação 8.6.

8.1.3 Abordagem Proposta para Saídas Probabilísticas

Nesta seção, são apresentados os conceitos teóricos sobre OPF probabilístico, denominado aqui como P-OPF. Uma vez que para cada amostra durante a etapa de treinamento e de classificação com OPF o custo atribuído é positivo (Equação 3.3), são necessários pequenos ajustes em relação à Equação 8.1, que pode ser reescrita para acomodar os requisitos do OPF:

$$P(\hat{\omega}_i = \omega_i | \mathbf{x}_i) \approx P_{A,B}(C_i) = \frac{1}{1+\exp(A\omega_i C_i + B)}, \quad (8.8)$$

em que C_i representa o custo atribuído à amostra $\mathbf{x}_i$ durante a etapa de treinamento ou classificação do OPF, e $\hat{\omega}_i$ significa o rótulo previsto pelo classificador. Basicamente, substituiu-se f_i por $\omega_i C_i$, uma vez que a função de custo C_i não possui sinal, enquanto $\text{sgn}(f_i) \in \{-,+\}$.

O raciocínio diante da abordagem proposta é assumir que, quanto menor for o custo atribuído à amostra $\mathbf{x}_i$, ou seja, C_i, maior será a probabilidade de essa amostra ser corretamente classificada. Essa ideia que é semelhante a técnica utilizada por Platt, uma vez que quanto maior f_i (mais distante é uma amostra do limite de decisão), mais provável que a amostra pertença à classe $+1$ (lado positivo) ou -1 (lado negativo).

A função de custo $F(\theta)$ precisa ser reformulada para considerar o modelo P-OPF, conforme segue:

[1]Para uma explicação detalhada e completa sobre as formulações matemáticas, considere o trabalho de Lin et al. (LIN; LIN; WENG, 2007).

$$F(\theta) = -\sum_{i=1}^{m}(t_i \log(p_i) + (1-t_i)\log(1-p_i)) \quad (8.9)$$

$$= \sum_{i=1}^{m}((t_i-1)(q_i) + \log(1+\exp(q_i))) \quad (8.10)$$

$$= \sum_{i=1}^{m}(t_i q_i + \log(1+\exp(-A\omega_i C_i - B))), \quad (8.11)$$

em que $p_i = P_{A,B}(C_i)$. Por último, mas não menos importante, é preciso utilizar o artifício proposto por Lin et al. (LIN; LIN; WENG, 2007), isto é, se $A\omega_i C_i + B \geq 0$, então deve-se usar $\frac{\exp(-A\omega_i C_i - B)}{1+\exp(-A\omega_i C_i - B)}$; caso contrário, deve-se usar $\frac{1}{1+\exp(A\omega_i C_i + B)}$.

Depois de aprender os parâmetros A e B, calcula-se a probabilidade de cada amostra pertencer à classe $+1$, ou seja, $P(\omega_i = 1|\mathbf{x}_i)$. Se $P(\omega_i = 1|\mathbf{x}_i) > P(\omega_i = 0|\mathbf{x}_i)$, então P-OPF atribui a classe $+1$ a essa amostra; caso contrário, a amostra é atribuída à classe -1. Neste trabalho foi adotado $\Theta = 0,5$, uma vez que lida com uma única chance. No entanto, pode-se ajustar facilmente esse limite por meio de uma busca linear ou qualquer outro algoritmo de otimização. O Algoritmo 9 implementa o classificador OPF probabilístico.

A linha 1 executa o algoritmo de treinamento OPF apresentado na Seção 3.1.1, retornando, respectivamente, a floresta de caminhos ótimos (P), o mapa de rótulos (L), o mapa dos custos (C) e o conjunto de treinamento ordenado pelo mapa de custos ($\hat{\mathscr{Z}}_1$). A linha 2 classifica o conjunto de validação ($\mathscr{Z}_v$) de acordo com o procedimento descrito na Seção 3.1.2 e o laço nas Linhas $12-25$ é responsável pela otimização dos parâmetros A e B, isto é, visa computar o melhor conjunto de parâmetros θ empregando uma busca em grade nos intervalos $A \in [L_A, U_A]$ e $B \in [L_B, U_B]$, onde L_A e U_A representam os limites inferiores e superiores considerando o parâmetro A. O mesmo pode ser definido para a variável B. Finalmente, a linha 26 armazena o melhor conjunto de parâmetros θ, que são utilizados para calcular a probabilidade de cada amostra do conjunto de teste ($\mathscr{Z}_2$) nas Linhas $27-31$.

Algoritmo 9 – FLORESTA DE CAMINHOS ÓTIMOS UTILIZANDO PROBABILIDADE

ENTRADA: Um conjunto λ-rotulado de treinamento $\mathscr{Z}_1$ e validação $\mathscr{Z}_v$, um conjunto de teste $\mathscr{Z}_2$, e valores no intervalo $[L_A, U_A]$ e $[L_B, U_B]$ concernente aos parâmetros A e B, respectivamente.

SAÍDA: Probabilidade para cada amostra em $\mathscr{Z}_2$.

1. $[P, L, C, \hat{\mathscr{Z}_1}] \leftarrow$ *Algoritmo 4 (*$\mathscr{Z}_1$*).*
2. $[\hat{P}, \hat{L}, \hat{C}] \leftarrow$ *Algoritmo 5 (*$[P, L, C, \hat{\mathscr{Z}_1}], \mathscr{Z}_v$*).*
3. $N_+ \leftarrow 0$ *e* $N_- \leftarrow 0$.
4. **Para cada** $\boldsymbol{v} \in \mathscr{Z}_v$, **Faça**
5. **Se** $\lambda(\boldsymbol{v}) = +1$, **Então**
6. $N_+ \leftarrow N_+ + 1$.
7. **Caso contrário**
8. $N_- \leftarrow N_- + 1$.
9. $t_+ \leftarrow \frac{N_+ + 1}{N_+ + 2}$.
10. $t_- \leftarrow \frac{1}{N_- + 2}$.
11. $A^\star \leftarrow 0$, $B^\star \leftarrow 0$, $minF \leftarrow +\infty$.
12. **Para cada** $A_i \in [L_A, U_A], B_j \in [L_B, U_B]$, **Faça**
13. **Para cada** $\boldsymbol{v} \in \mathscr{Z}_v$, **Faça**
14. **Se** $\lambda(\boldsymbol{v}) = +1$, **Então**
15. $t \leftarrow t_+$.
16. **Caso contrário**
17. $t \leftarrow t_-$.
18. **Se** $A_i C_{\boldsymbol{v}} + B_j \geq 0$, **Então**
19. $F_{ij} \leftarrow F_{ij} + (t(A_i \lambda(\boldsymbol{v}) C_{\boldsymbol{v}} + B_j) + \log(1 + \exp(-A_i \lambda(\boldsymbol{v}) C_{\boldsymbol{v}} - B_j)))$.
20. **Caso contrário**
21. $F_{ij} \leftarrow F_{ij} + (t-1) * (A_i \lambda(\boldsymbol{v}) C_{\boldsymbol{v}} + B_j) + \log(1 + \exp(A_i \lambda(\boldsymbol{v}) C_{\boldsymbol{v}} + B_j))$.
22. **Se** $F_{ij} <$ *minF*, **Então**
23. $minF \leftarrow F_{ij}$.
24. $A^* \leftarrow A_i$.
25. $B^* \leftarrow B_j$.
26. $\theta \leftarrow (A^\star, B^\star)$
27. **Para cada** $\boldsymbol{i} \in \mathscr{Z}_2$, **Faça**
28. **Se** $(A^\star \lambda(\boldsymbol{i}) C_{\boldsymbol{i}} + B^\star) \geq 0$, **Então**
29. $P_{\boldsymbol{i}} \leftarrow \frac{\exp(-A^\star \lambda(\boldsymbol{i}) C_{\boldsymbol{i}} - B^\star)}{1 + \exp(-A^\star \lambda(\boldsymbol{i}) C_{\boldsymbol{i}} - B^\star)}$.
30. **Caso contrário**
31. $P_{\boldsymbol{i}} \leftarrow \frac{1}{1 + \exp(A^\star \lambda(\boldsymbol{i}) C_{\boldsymbol{i}} + B^\star)}$.

8.2 Metodologia e Resultados Experimentais

Esta seção apresenta a metodologia e os experimentos empregados para validar a efetividade e eficiência da abordagem proposta. Para isso, na seção 8.2.1, avalia-se a eficácia do P-OPF contra OPF padrão (denominado como OPF*) em alguns problemas de classificação. Além disso, na seção 8.2.2, é comparada a abordagem proposta contra o classificador SVM probabilístico.

8.2.1 Validação da Proposta sobre conjuntos de dados de propósito geral

Para ajustar os parâmetros A e B, foram utilizadas duas estratégias diferentes, sendo uma delas baseada em meta-heurísticas e outra puramente matemática. Com relação às técnicas meta-heurísticas, optou-se pelo PSO (KENNEDY; EBERHART, 2001), a seguir designado por P-OPF-PSO, e em relação à abordagem matemática usou-se o método de busca linear conhecido como *Newton Backtracking - NB*[2] para resolver a Equação 8.2, a seguir designado por P-OPF-NB, como descrito por Lin et al. (LIN; LIN; WENG, 2007). Para estudar o comportamento do P-OPF em diferentes cenários, foram utilizados dois conjuntos de dados sintéticos (synhetic01, synthesis02), dois conjuntos de dados relativos à detecção de furto de energia (comercial e industrial) (PEREIRA et al., 2016), bem como onze bases de dados públicas[3]. Cabe destacar que esses conjuntos de dados têm sido utilizados com frequência para avaliação de diferentes métodos de classificação. Os conjuntos de dados foram normalizados da seguinte forma:

$$\mathbf{t}' = \frac{\mathbf{t} - \mu}{\rho}, \tag{8.12}$$

onde μ denota a média e ρ significa seu desvio padrão. Além disso, t e t' correspondem às características originais e normalizadas, respectivamente. A Tabela 8.1 apresenta as principais características de cada conjunto de dados.

Com relação à metodologia, cada conjunto de dados foi particionado em três subconjuntos: (i) conjunto de treinamento com 40%; (ii) conjunto de validação com 10%; e (iii) conjunto de teste com 50%, a seguir denominados como 40:10:50. Para cada intervalo, os conjuntos de treinamento, validação e teste foram selecionados aleatoriamente e o processo foi repetido 15 vezes (validação cruzada)[4]. É importante destacar que o OPF* foi treinado utilizando $\mathscr{Z}_1 \cup \mathscr{Z}_v$ considerando as três etapas acima mencionadas. Depois de otimizados os parâmetros, as abordagens

[2]Para mais detalhes do algoritmo *Newton Backtracking* considere o estudo de Lin et al. (LIN; LIN; WENG, 2007).
[3]`http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`
[4]Observe que as porcentagens foram escolhidas empiricamente.

Tabela 8.1: Descrição das bases de dados

Base de dados	# amostras	# características	# classes
Statlog-Australian	690	14	2
Comercial	4.952	8	2
Industrial	3.182	8	2
UCI-Breast-Cancer	683	10	2
Pima-Indians-Diabetes	768	8	2
Statlog-German	1.000	24	2
Statlog-Heart	270	13	2
UCI-Hepatitis	155	19	2
IJCNN1	141.691	22	2
UCI-Ionosphere	351	34	2
UCI-Liver-disorders	345	6	2
UCI-Madelon	2.600	500	2
UCI-Phishing	11.055	68	2
Synthetic01	1.000	2	2
Synthetic02	1.000	2	2

P-OPF-PSO e P-OPF-NB foram treinadas mais uma vez usando o conjunto de treinamento original (ou seja, o mesmo usado pelo OPF*). Para comparar a abordagem proposta, foi calculada a acurácia média e o tempo de execução para, em seguida, serem analisadas por meio do teste estatístico de Wilcoxon (com significância de $0,05$).

Com relação às técnicas de otimização, considerando a técnica PSO, foram utilizos 20 agentes (soluções iniciais), $c_1 = 1{,}4$, $c_2 = 0{,}6$, $\xi = 0{,}5$, e 100 iterações. O espaço de busca $A \times B$ foi definido dentro de intervalo $[-5,5] \times [-5,5]$. Mais uma vez, esses valores foram escolhidos empiricamente. Para o método de Newton, utilizou-se um número máximo de iterações de 100, etapa mínima para busca linear de $1e-07$ e $\sigma = 1e-07$.

Na Tabela 8.2 é apresentada a acurácia média e o desvio padrão para todos os conjuntos de dados avaliados, sendo as taxas de acurácia calculadas de acordo com (PAPA; FALCÃO; SUZUKI, 2009), e na Tabela 8.3 é apresentada a média dos valores de *F-measure* relativos ao mesmo grupo avaliado. As técnicas mais precisas, considerando o teste de Wilcoxon, são destacadas em negrito. Como se pode observar, o OPF* obteve os melhores resultados em quase todos os conjuntos de dados, exceto "Statlog-Australian", que pode ser melhor classificado pelo P-OPF usando o método de Newton. No entanto, é válido observar que o OPF probabilístico proposto obteve resultados competitivos.

Tabela 8.2: Acurácia média dos resultados (%) e seu desvio padrão para todas as bases considerando o OPF* e P-OPF sob diferentes técnicas de otimização.

Base de dados	OPF*	P-OPF-NB	P-OPF-PSO
Statlog-Australian	$77,52 \pm 1,25$	**77,76** $\pm 1,22$	$77,52 \pm 1,25$
Comercial	**81,57** $\pm 1,53$	$61,61 \pm 5,33$	**81,57** $\pm 1,53$
Industrial	**79,77** $\pm 1,75$	$58,89 \pm 3,04$	$79,75 \pm 1,76$
UCI-Breast-Cancer	**93,92** $\pm 1,07$	$93,72 \pm 1,10$	**93,92** $\pm 1,07$
Pima-Indians-Diabetes	**65,72** $\pm 0,97$	**65,61** $\pm 1,02$	**65,72** $\pm 0,97$
Statlog-German	**61,09** $\pm 2,71$	**61,00** $\pm 2,69$	**61,09** $\pm 2,71$
Statlog-Heart	**77,27** $\pm 3,92$	**77,27** $\pm 3,92$	**66,20** $\pm 22,28$
UCI-Hepatitis	**69,80** $\pm 4,36$	**69,80** $\pm 4,36$	$48,16 \pm 20,19$
IJCNN1	**94,08** $\pm 0,23$	$93,75 \pm 0,26$	**94,08** $\pm 0,23$
UCI-Ionosphere	**79,12** $\pm 2,72$	$77,41 \pm 2,97$	**64,22** $\pm 25,56$
UCI-Liver-disorders	**58,70** $\pm 3,38$	**57,80** $\pm 4,33$	$56,61 \pm 5,58$
UCI-Madelon	**53,01** $\pm 1,12$	**53,01** $\pm 1,12$	**52,52** $\pm 1,99$
UCI-Phishing	**92,14** $\pm 0,89$	$88,86 \pm 2,00$	$89,68 \pm 0,89$
Synthetic01	**60,27** $\pm 1,46$	$56,09 \pm 3,90$	**60,27** $\pm 1,46$
Synthetic02	**90,08** $\pm 1,08$	$88,73 \pm 1,54$	**90,08** $\pm 1,08$

Tabela 8.3: Valores *F-measure* e seu desvio padrão para todas as bases considerando o OPF* e P-OPF sob diferentes técnicas de otimização.

Base de dados	OPF*	P-OPF-NB	P-OPF-PSO
Statlog-Australian	$0,8051 \pm 0,0125$	**0,8080** $\pm 0,0118$	$0,8051 \pm 0,0125$
Comercial	**0,9797** $\pm 0,0016$	$0,9726 \pm 0,0011$	**0,9797** $\pm 0,0016$
Industrial	**0,9728** $\pm 0,0021$	$0,9692 \pm 0,0013$	$0,9726 \pm 0,0020$
UCI-Breast-Cancer	**0,9626** $\pm 0,0070$	$0,9616 \pm 0,0070$	**0,9626** $\pm 0,0070$
Pima-Indians-Diabetes	**0,5486** $\pm 0,0165$	**0,5462** $\pm 0,0175$	**0,5486** $\pm 0,0165$
Statlog-German	**0,4543** $\pm 0,0397$	**0,4529** $\pm 0,0395$	**0,4543** $\pm 0,0397$
Statlog-Heart	**0,7464** $\pm 0,0459$	**0,7464** $\pm 0,0459$	**0,6377** $\pm 0,2184$
UCI-Hepatitis	**0,5298** $\pm 0,0740$	**0,5298** $\pm 0,0740$	$0,3410 \pm 0,1591$
IJCNN1	**0,9916** $\pm 0,0002$	$0,9912 \pm 0,0002$	**0,9916** $\pm 0,0002$
UCI-Ionosphere	**0,7344** $\pm 0,0428$	$0,7067 \pm 0,0486$	**0,6111** $\pm 0,2187$
UCI-Liver-disorders	**0,5229** $\pm 0,0382$	**0,5060** $\pm 0,0614$	$0,5095 \pm 0,0413$
UCI-Madelon	**0,5517** $\pm 0,0108$	**0,5517** $\pm 0,0108$	**0,5372** $\pm 0,0388$
UCI-Phishing	**0,9125** $\pm 0,0101$	$0,8748 \pm 0,0245$	$0,8851 \pm 0,0103$
Synthetic01	**0,6108** $\pm 0,0150$	**0,4527** $\pm 0,2738$	**0,6108** $\pm 0,0150$
Synthetic02	**0,9011** $\pm 0,0107$	$0,8867 \pm 0,0161$	**0,9011** $\pm 0,0107$

Referente à análise *F-measure*, o comportamento foi considerado semelhante ao obtido pela acurácia (Tabela 8.2), com exceção do conjunto de dados "Synthetic01", que foi bem classificado por todas as técnicas, como pode ser observado na Tabela 8.3. Ambos classificadores P-OPF e OPF* mostraram resultados próximos em 13 de 15 conjuntos de dados referentes aos resultados da acurácia. Somente as bases "Industrial" e "UCI-Phishing" foram melhor classificadas utilizando OPF*. No geral, o P-OPF usando PSO foi mais eficaz do que a abordagem P-OPF-NB para encontrar o melhor conjunto de parâmetros. Vale ressaltar que o P-OPF não superou o classificador OPF*, mas mostrou resultados interessantes, já que o P-OPF conseguiu explorar as estimativas de probabilidade.

Na Tabela 8.4 é apresentada a carga computacional média (em segundos) para as fases de treinamento e validação relativas ao OPF* e P-OPF com busca de parâmetros. Uma vez que o PSO é uma técnica baseada em enxames, o que significa que todas as soluções possíveis (agentes) são atualizadas a cada iteração, é esperado que apresentará um custo maior do que a técnica NB.

Tabela 8.4: Tempo de treinamento e validação (em segundos) considerando OPF* e P-OPF sob diferentes técnicas de otimização.

Base de dados	OPF*	P-OPF-NB	P-OPF-PSO
Statlog-Australian	$0,0120 \pm 0,0032$	$0,0192 \pm 0,0018$	$0,0376 \pm 0,0033$
Comercial	$0,3701 \pm 0,0017$	$0,6784 \pm 0,0010$	$0,8148 \pm 0,0182$
Industrial	$0,1554 \pm 0,0021$	$0,2816 \pm 0,0021$	$0,3644 \pm 0,0032$
UCI-Breast-Cancer	$0,0125 \pm 0,0006$	$0,0168 \pm 0,0020$	$0,0358 \pm 0,0032$
Pima-Indians-Diabetes	$0,0149 \pm 0,0011$	$0,0203 \pm 0,0020$	$0,0439 \pm 0,0024$
Statlog-German	$0,0224 \pm 0,0018$	$0,0369 \pm 0,0024$	$0,0646 \pm 0,0026$
Statlog-Heart	$0,0028 \pm 0,0007$	$0,0056 \pm 0,0007$	$0,0146 \pm 0,0014$
UCI-Hepatitis	$0,0010 \pm 0,0001$	$0,0021 \pm 0,0003$	$0,0104 \pm 0,0027$
IJCNN1	$441,45 \pm 5,15$	$860,17 \pm 16,93$	$868,63 \pm 21,35$
UCI-Ionosphere	$0,0055 \pm 0,0013$	$0,0089 \pm 0,0024$	$0,0195 \pm 0,0033$
UCI-Liver-disorders	$0,0034 \pm 0,0005$	$0,0054 \pm 0,0013$	$0,0168 \pm 0,0020$
UCI-Madelon	$0,7191 \pm 0,0091$	$1,3565 \pm 0,0100$	$1,4216 \pm 0,0122$
UCI-Phishing	$3,5049 \pm 0,0184$	$6,5220 \pm 0,0193$	$6,7649 \pm 0,0172$
Synthetic01	$0,0180 \pm 0,0020$	$0,0287 \pm 0,0015$	$0,0570 \pm 0,0027$
Synthetic02	$0,0209 \pm 0,0009$	$0,0308 \pm 0,0020$	$0,0600 \pm 0,0031$

Além do teste de Wilcoxon, utilizou-se o teste estatístico de Friedman para ranquear as abordagens, bem como o teste de Nemenyi para indicar a diferença crítica entre elas. Conforme apresentado na Figura 8.1, para acurácia (Figura 8.1a) e valores *F-measure* (Figura 8.1b), o OPF* foi considerado o mais assertivo de acordo com o teste estatístico. Isso reflete que o OPF* alcançou os melhores valores de acurácia na maioria das bases. Contudo, o teste estatístico não apontou uma diferença crítica entre OPF*, P-OPF-PSO e P-OPF-NB, o que significa que as

abordagens comparadas apresentam um comportamento semelhante, evidenciando, assim, que a probabilidade aplicada ao classificador OPF não interfere negativamente no seu comportamento, mostrando ser uma proposta interessante para estratégias mais robustas de combinação de classificadores.

(a) (b)

Figura 8.1: Comparação entre OPF* e P-OPF considerando o teste estatístico de Nemenyi para: (a) acurácia e (b) valores d *F-measure*. Grupos similares (p = 0,05) são conectados.

Na Figura 8.2 é apresentada a análise estatística de Nemenyi da carga computacional para o tempo de treinamento (treinamento com validação). Conforme mencionado anteriormente, a técnica PSO compreende um algoritmo de maior complexidade computacional em comparação com a busca linear NB, por isso pontuou como a mais lenta dentre todas avaliadas. Como esperado, a abordagem OPF* foi considerada a mais rápida em quase todos os conjuntos de dados. Em média, P-OPF usando o método NB foi aproximadamente $1,947$ vezes mais lento do que OPF* nas etapas de treinamento com validação, enquanto que P-OPF usando PSO foi cerca de $1,968$ vezes mais lento do que OPF*.

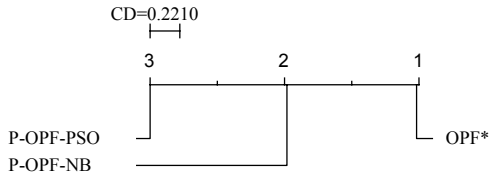


Figura 8.2: Teste estatístico de Nemenyi para a tempo de treinamento (treinamento com validação). Grupos similares (p = 0,05) são conectados.

8.2.2 Comparação entre Máquina de Vetores de Suporte e Floresta de Caminhos Ótimos utilizando classificação probabilística

Nesta seção é comparada a abordagem proposta usando PSO (técnica considerada a mais assertiva na seção anterior) contra o classificador SVM probabilístico. A idéia é realizar a mesma configuração experimental, mas agora considerando um classificador SVM com kernel RBF e parâmetros ajustados por meio da busca em grade no conjunto de validação[5]. Para cumprir este objetivo, foi aplicado o mesmo procedimento, ou seja, um intervalo de 40:10:50 para o conjunto de treinamento, validação e teste, respectivamente, repetidos 15 vezes (validação cruzada). Nas Tabelas 8.5 e 8.6 são apresentados, respectivamente, a acurácia média e os valores de *F-measure*. Os valores em negrito representam as técnicas mais precisas de acordo com o teste Wilcoxon.

Tabela 8.5: Acurácia média dos resultados (%) e seu desvio padrão considerando os classificadores P-OPF-PSO e SVM probabilístico.

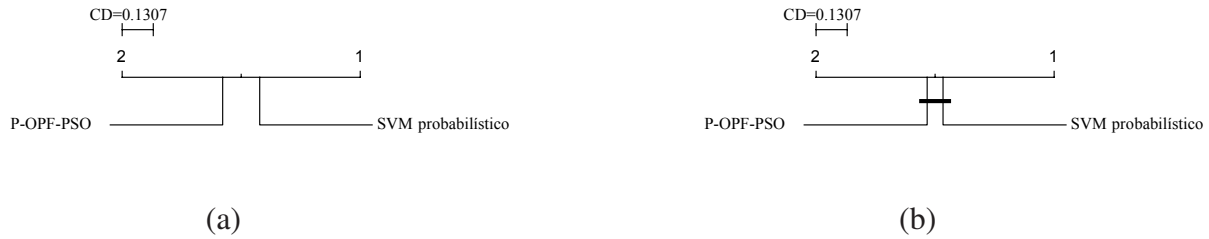
Base de dados	P-OPF-PSO	SVM probabilístico
Statlog-Australian	$77,52 \pm 1,25$	**85,39** $\pm 1,68$
Comercial	**81,57** $\pm 1,53$	$51,85 \pm 2,60$
Industrial	**79,75** $\pm 1,76$	$55,05 \pm 6,07$
UCI-Breast-Cancer	$93,92 \pm 1,07$	**96,08** $\pm 1,36$
Pima-Indians-Diabetes	**65,72** $\pm 0,97$	**68,75** $\pm 3,63$
Statlog-German	**61,09** $\pm 2,71$	**63,47** $\pm 2,28$
Statlog-Heart	$66,20 \pm 22,28$	**81,29** $\pm 1,88$
UCI-Hepatitis	**48,16** $\pm 20,19$	**57,14** $\pm 5,62$
IJCNN1	$94,08 \pm 0,23$	**96,69** $\pm 0,45$
UCI-Ionosphere	$64,22 \pm 25,56$	**86,07** $\pm 5,43$
UCI-Liver-disorders	**56,61** $\pm 5,58$	**60,08** $\pm 6,42$
UCI-Madelon	**52,52** $\pm 1,99$	$50,36 \pm 1,89$
UCI-Phishing	$89,68 \pm 0,89$	**95,54** $\pm 0,52$
Synthetic01	**60,27** $\pm 1,46$	$56,52 \pm 3,77$
Synthetic02	**90,08** $\pm 1,08$	$87,89 \pm 1,72$

[5]Os parâmetros RBF foram optimizados dentro dos intervalos $\gamma \in \{0.01, 0.1, 1\}$ e $C \in \{1, 10, 100\}$.

Tabela 8.6: Valores de *F-measure* para P-OPF-PSO e SVM probabilístico.

Base de dados	P-OPF-PSO	SVM probabilístico
Statlog-Australian	$0,8051 \pm 0,0125$	**0,8595** $\pm 0,0189$
Comercial	**0,9797** $\pm 0,0016$	$0,9727 \pm 0,0015$
Industrial	**0,9726** $\pm 0,0020$	**0,9708** $\pm 0,0035$
UCI-Breast-Cancer	$0,9626 \pm 0,0070$	**0,9733** $\pm 0,0096$
Pima-Indians-Diabetes	**0,5486** $\pm 0,0165$	**0,5698** $\pm 0,0713$
Statlog-German	**0,4543** $\pm 0,0397$	**0,4487** $\pm 0,0544$
Statlog-Heart	$0,6377 \pm 0,2184$	**0,7916** $\pm 0,0208$
UCI-Hepatitis	**0,3410** $\pm 0,1591$	**0,2414** $\pm 0,1583$
IJCNN1	$0,9916 \pm 0,0002$	**0,9947** $\pm 0,0009$
UCI-Ionosphere	$0,6111 \pm 0,2187$	**0,8280** $\pm 0,0665$
UCI-Liver-disorders	**0,5095** $\pm 0,0413$	**0,4010** $\pm 0,2087$
UCI-Madelon	**0,5372** $\pm 0,0388$	$0,1200 \pm 0,1901$
UCI-Phishing	$0,8851 \pm 0,0103$	**0,9506** $\pm 0,0055$
Synthetic01	**0,6108** $\pm 0,0150$	$0,5299 \pm 0,0842$
Synthetic02	**0,9011** $\pm 0,0107$	$0,8787 \pm 0,0158$

Como se pode observar na Tabela 8.5, ambos classificadores apresentaram resultados próximos, dos quais P-OPF obteve as melhores taxas de acurácia em 5 de 15 bases, 4 apresentaram resultados semelhantes e 6 de 15 bases são melhor classificadas com o SVM probabilístico. Com relação à análise estatística considerando os resultados de acurácia (Figura 8.3a) e *F-measure* (Figura 8.3b), SVM probabilístico é indicado como o mais preciso pelo teste de Nemenyi. No entanto, o teste estatístico não apontou uma diferença crítica entre o P-OPF-PSO e SVM em relação aos valores de *F-measure*, o que significa que ambos apresentam um comportamento similar nessa análise.



(a) (b)

Figura 8.3: Comparação entre as abordagens probabilísticas P-OPF-PSO e SVM considerando o teste estatístico de Nemenyi para: (a) acurácia e (b) valores de *F-measure*. Grupos similares (p = 0,05) são conectados.

Na Tabela 8.7 é apresentada a carga computacional média (em segundos) para as etapas de treinamento e validação relativas a P-OPF-PSO e SVM probabilístico.

Tabela 8.7: Tempo de treinamento (em segundos) concernente as abordagens P-OPF-PSO e SVM probabilístico com respeito a etapa de treinamento e validação.

Bases de dados	P-OPF-PSO	SVM probabilístico
Statlog-Australian	$0,0376 \pm 0,0033$	$0,1759 \pm 0,0524$
Comercial	$0,8148 \pm 0,0182$	$0,9505 \pm 0,3703$
Industrial	$0,3644 \pm 0,0032$	$0,6080 \pm 0,1619$
UCI-Breast-Cancer	$0,0358 \pm 0,0032$	$0,1307 \pm 0,0073$
Pima-Indians-Diabetes	$0,0439 \pm 0,0024$	$0,3419 \pm 0,3326$
Statlog-German	$0,0646 \pm 0,0026$	$0,2389 \pm 0,0386$
Statlog-Heart	$0,0146 \pm 0,0014$	$0,1224 \pm 0,0021$
UCI-Hepatitis	$0,0104 \pm 0,0027$	$0,1179 \pm 0,0071$
IJCNN1	$868,63 \pm 21,35$	$740,46 \pm 121,50$
UCI-Ionosphere	$0,0195 \pm 0,0033$	$0,1339 \pm 0,0093$
UCI-Liver-disorders	$0,0168 \pm 0,0020$	$0,1287 \pm 0,0073$
UCI-Madelon	$1,4216 \pm 0,0122$	$35,1377 \pm 48,70$
UCI-Phishing	$6,7649 \pm 0,0172$	$11,8098 \pm 4,2058$
Synthetic01	$0,0570 \pm 0,0027$	$0,2241 \pm 0,0364$
Synthetic02	$0,0600 \pm 0,0031$	$0,1795 \pm 0,0093$

Conforme a Tabela 8.7, a abordagem SVM foi considerada a mais dispendiosa, uma vez que usa a busca em grade para ajustar os hiper-parâmetros C, γ, seguido do procedimento de Platt, que também emprega uma etapa de validação cruzada adicional. Na Figura 8.4 é apresentada a análise estatística de Nemenyi considerando a carga computacional para o tempo de treinamento (treinamento com validação). Pode-se observar que o P-OPF-PSO pontuou como a abordagem mais rápida nas etapas de treinamento com validação.

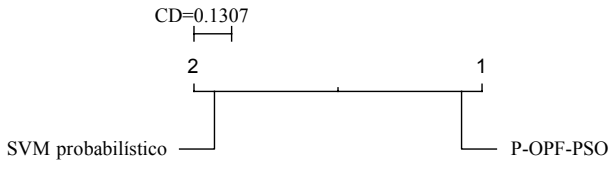


Figura 8.4: Análise estatística de Nemenyi comparando as abordagens P-OPF-PSO e SVM probabilístico com respeito ao tempo de treinamento (treinamento com valiadação).

8.3 Conclusões

A classificação probabilística tem sido um tema de grande interesse com relação à comunidade de aprendizado de máquina, principalmente devido à falta de uma informação mais "flexível" em vez de apenas rótulos. Esse trabalho abordou este tema, propondo uma variação do classificador OPF com saída probabilística para problemas de classificação binários, nomeadamente P-OPF. Os resultados do P-OPF proposto foram comparados com o OPF padrão e o SVM probabilístico em vários conjuntos de dados, obtendo resultados adequados em vários deles.

Com relação aos trabalhos futuros, busca-se ampliar o P-OPF para problemas de classificação de múltiplas classes, bem como considerar outras técnicas de otimização para ajustar os novos parâmetros que ajudam a minimizar a função de custo. Além disso, busca-se fornecer um importante e novo caminho para o classificador OPF em novas estratégias de combinação e resultados mais expressivos. Assim sendo, as principais contribuições desse capítulo foram:

- apresentar um nova variante do classificador OPF com saída do tipo probabilística;
- investigar a influência das probabilidades em relação a abordagem tradicional com saídas abstratas para problemas binários; e
- avaliar a eficiência e eficácia do modelo proposto em comparação ao classificador SVM probabilístico.

Capítulo 9

Conclusões

Este capítulo tem por objetivo apresentar as conclusões do trabalho, bem como apontar alguns trabalhos correlatos ao estudo da tese de doutorado e, por fim, indicar ideias futuras.

Esse trabalho objetivou investigar a estratégia de combinação de classificadores baseados em OPFs. Até o presente momento, poucos estudos foram apresentados sobre combinação de classificadores OPFs objetivando uma eficácia no processo de classificação. Dentre eles, pode-se destacar Ponti e Papa (PONTI; PAPA, 2011), por exemplo, com combinação de classificadores OPFs utilizando subconjuntos disjuntos e, em seguida, Ponti et al. (PONTI; PAPA; LEVADA, 2011) apresentaram um estudo sobre combinação de classificadores OPFs utilizando Teoria dos Jogos e Campos Aleatórios Markovianos.

Sendo assim, as contribuições desse trabalho deram-se por meio dos estudos sobre a técnica OPF como, por exemplo, o aprendizado por níveis de confiança para OPF com grafo completo e OPF com grafo k-nn; combinação de classificadores OPFs utilizando aprendizado por confiança com decisão final baseada em votação por maioria; combinação de classificadores OPFs utilizando poda de conjunto guiados por otimizações meta-heurísticas; combinação de classificadores OPFs utilizando poda de conjunto e otimizações meta-heurísticas para imagens de sensoriamento remoto; e uma nova variante do classificador OPF com saída do tipo probabilística.

Em suma, o aprendizado das confianças tem por objetivo atribuir uma pontuação às amostras de treinamento penalizando, assim, aquelas que não apresentam uma "boa" confiabilidade em relação a um conjunto de validação. Essa ideia, que posteriormente foi aplicada em duas variantes do OPF, mostrou-se uma abordagem interessante para o processo de classificação, diminuindo os erros provocados pelas regiões de empate quando amostras de diferentes classes apresentam o mesmo custo de caminho ótimo.

Em seguida, a ideia mencionada acima foi aplicada sobre uma estratégia de combinação de classificadores utilizando votação por maioria. Testes empíricos sobre esse estudo evidenciaram ganhos significativos na eficácia ao utilizarem a confiança como um indicador de confiabilidade em subconjuntos disjuntos de treinamento em comparação com modelo OPF tradicional e a abordagem que utliza confiança, mas sem combinação. Além disso, essa estratégia demostrou eficiência durante a etapa de treinamento como uma das melhores ranqueadas e sem apresentar um custo elevado durante a etapa de classificação, visto que estratégias de combinação geralmente tem um custo adicional por conta dos L classificadores utilizados e do processo de decisão final.

Além disso, um outro estudo sobre combinação utilizando poda de conjunto de classificadores foi conduzido. Nesse estudo, o problema de otimização da poda de classificadores OPF foi modelado como sendo uma otimização no espaço de quatérnions, mostrando-se como uma abordagem interessante do ponto de vista da eficácia ao selecionar os melhores classificadores de um conjunto original para um determinado problema e, além disso, promovendo eficiência devido à sua rápida convergência quando comparada com outras técnicas. Visto que a premissa da poda de classificadores consiste em fornecer uma abordagem com foco na eficácia utilizando múltiplos classificadores, mas sem comprometer a eficiência por utilizar um grande número de técnicas durante a fase de classificação, utilizar estratégias baseadas em meta-heurísticas para selecionar um subconjunto quase-ótimo torna-se pertinente. Portanto, como parte da proposta desse estudo, além da estratégia de poda de conjunto, procurou-se avaliar algumas técnicas de meta-heurísticas, sendo que a abordagem baseada em quatérnions apresentou a melhor relação custo-benefício para poda de conjunto utilizando classificadores OPF.

Na sequência, o conceito mencionado acima foi estendido no contexto de sensoriamento remoto, área de crescimento na literatura, principalmente devido a sua complexidade, possibilitando explorar o potencial das estratégias de combinação com poda de conjunto. Nessa circustância, foi também avaliado o conceito proposto de poda de conjunto de classificadores OPFs guiados por otimizações meta-heurísticas em comparação com o classificador SVM, igualmente sob a mesma estratégia de poda por meta-heurísticas. Conforme avaliado, testes empíricos evidenciaram resultados relevantes para o OPF, principalmente em situações com elevado grau de sobreposição de amostras.

Por fim, foi apresentada uma nova variante do classificador OPF com saída do tipo probabilística, visto que sua implementação original fornece apenas saídas abstratas. Esse estudo, inicialmente desenvolvido para problemas binários, procura preencher uma área importante no âmbito da classificação, pois estende para um alcance maior do que apenas rótulos, onde, em

determinados casos, é desejável conhecer a probabilidade. Além disso, fornece novas possibilidades de estratégias de combinação de classificadores, permitindo explorar novos conceitos. Experimentos comparando OPF probabilístico e a abordagem tradicional mostraram que ambos modelos são considerados similares, o que assegura sua eficácia utilizando probabilidades.

Além disso, cabe destacar outros trabalhos correlatos com a área de aprendizado de máquina que também foram realizados durante essa tese:

- Aprendizado de Kernels em Máquinas de Vetores de Suporte utilizando Polinômios Potências de Sigmoide (FERNANDES et al., 2014). Esse trabalho propôs o uso dos Polinômios Potências de Sigmoide (MARAR, 1997) como kernel para SVM, o qual mostrou ser uma função interessante para mapeamento, pois não necessita de parâmetros para o seu ajuste e, além disso, apresentou resultados similares aos das funções de base radiais.

- Otimização de parâmetros guiado por meta-heurísticas em Redes Neurais Probabilísticas Aprimoradas (FERNANDES et al., 2016). Esse trabalho propôs uma otimização dos parâmetros necessários para uma Rede Neural Probabílistica Aprimorada (*Enhanced Probabilistic Neural Network* - EPNN), a qual considera a probabilidade de densidade local mediante círculos de decisões locais.

- Floresta de Caminhos Ótimos baseada em k-conectividade: Teoria e Aplicações (PAPA; FERNANDES; FALCÃO, 2017). Esse trabalho propôs um estudo aprofundado sobre OPF utilizando uma relação de adjacência baseada em k-conectividade, bem como um aprimoramento da etapa de classificação em conjunto com otimizações por meta-heurísticas para encontrar o melhor k.

- Identificação de Parâmetros de Borramento utilizando Floresta de Caminhos Ótimos (PIRES; FERNANDES; PAPA, 2017). Esse trabalho propôs modelar o problema de identificação do borramento como uma tarefa de reconhecimento de padrões utilizando o classificador OPF.

Como trabalhos futuros, busca-se ampliar o classificador OPF probabilístico para múltiplas classes, bem como implementar outras estratégias de combinação como, por exemplo, combinação linear e *AdaBoost*.

REFERÊNCIAS

ABE, B.; GIDUDU, A.; MARWAL, T. Investigating the effects of ensemble classification on remotely sensed data for land cover mapping. In: *Geoscience and Remote Sensing Symposium (IGARSS), 2010 IEEE International*. [S.l.: s.n.], 2010. p. 2832–2835. ISSN 2153-6996.

ACOSTA-MENDOZA, N. et al. Learning to assemble classifier via genetic programming. *International Journal of Pattern Recognition and Artificial Intelligence*, v. 28, n. 07, 2014.

ADELI, H.; HUNG, S.-L. *Machine Learning: Neural Networks, Genetic Algorithms, and Fuzzy Systems*. New York, NY, USA: John Wiley & Sons, Inc., 1994. ISBN 0-471-01633-0.

ADELI, H.; PARK, H. S. *Neurocomputing for Design Automation*. 1st. ed. Boca Raton, FL, USA: CRC Press, Inc., 1998. ISBN 0849320925.

AHMADLOU, M.; ADELI, H. Enhanced probabilistic neural network with local decision circles: A robust classifier. *Integrated Computer-Aided Engineering*, IOS Press, Amsterdam, The Netherlands, The Netherlands, v. 17, n. 3, p. 197–210, 2010.

AL-ANI, A.; DERICHE, M. A new technique for combining multiple classifiers using the dempster-shafer theory of evidence. *Journal of Artificial Intelligence Research*, AI Access Foundation, USA, v. 17, n. 1, p. 333–361, 2002. ISSN 1076-9757.

ALLÈNE, C. et al. Some links between extremum spanning forests, watersheds and min-cuts. *Image and Vision Computing*, v. 28, n. 10, p. 1460–1471, 2010. ISSN 0262-8856. Image Analysis and Mathematical Morphology.

ANDREWS, D. F. Plots of High-Dimensional Data. *Biometrics*, International Biometric Society, v. 28, n. 1, 1972. ISSN 0006341X.

BAUER, E.; KOHAVI, R. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, Kluwer Academic Publishers, v. 36, n. 1-2, p. 105–139, 1999. ISSN 0885-6125.

BI, Y. et al. Combining multiple classifiers using dempster's rule for text categorization. *Applied Artificial Intelligence*, Taylor & Francis, Inc., Bristol, PA, USA, v. 21, n. 3, p. 211–239, 2007. ISSN 0883-9514.

BIANCHINI, M.; MAGGINI, M.; JAIN, L. C. *Handbook on Neural Information Processing*. [S.l.]: Springer Publishing Company, Incorporated, 2013. ISBN 3642366562, 9783642366567.

BOLOURCHI, A.; MASRI, S. F.; ALDRAIHEM, O. J. Studies into computational intelligence and evolutionary approaches for model-free identification of hysteretic systems. *Computer-Aided Civil and Infrastructure Engineering*, v. 30, n. 5, p. 330–346, 2015. ISSN 1467-8667.

BREIMAN, L. Bagging predictors. *Machine Learning*, Kluwer Academic Publishers, Hingham, MA, USA, v. 24, n. 2, p. 123–140, 1996.

BREIMAN, L. Arcing classifiers. *Annals of Statistics*, v. 26, n. 3, p. 801–824, 1998.

BUNKE, H.; WANG, P. S.-p. (Ed.). *Handbook of character recognition and document image analysis*. Singapore: World Scientific, 1997. ISBN 981-022270-X.

CARUANA, R.; NICULESCU-MIZIL, A. An empirical comparison of supervised learning algorithms. In: *Proceedings of the 23rd International Conference on Machine Learning*. New York, NY, USA: [s.n.], 2006. (ICML '06), p. 161–168. ISBN 1-59593-383-2.

CHEN, C. H. *Handbook of Pattern Recognition and Computer Vision*. River Edge, NJ, USA: World Scientific Publishing, 2005. ISBN 9812561056.

COLETTA, L. F. S. et al. Using metaheuristics to optimize the combination of classifier and cluster ensembles. *Integrated Computer-Aided Engineering*, IOS Press, v. 22, n. 3, p. 229–242, 2015.

CORTES, C.; VAPNIK, V. Support vector networks. *Machine Learning*, v. 20, p. 273–297, 1995.

DEMPSTER, A. P. Upper and lower probabilities induced by a multivalued mapping. In: . [S.l.]: The Annals of Mathematical Statistics, 1967. p. 325–339.

DEMSAR, J. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, JMLR.org, v. 7, p. 1–30, 2006. ISSN 1532-4435.

DIAO, R.; SHEN, Q. Fuzzy-rough classifier ensemble selection. In: *IEEE International Conference on Fuzzy Systems*. [S.l.: s.n.], 2011. p. 1516–1522. ISSN 1098-7584.

DIETTERICH, T. Ensemble methods in machine learning. In: *Multiple Classifier Systems*. [S.l.]: Springer Berlin / Heidelberg, 2000. (Lecture Notes in Computer Science, v. 1857), p. 1–15.

DIJKSTRA, E. W. A note on two problems in connexion with graphs. *Numerische Mathematik*, v. 1, p. 269–271, 1959.

DUDA, R. O.; HART, P. E.; STORK, D. G. *Pattern Classification (2nd Edition)*. [S.l.]: Wiley-Interscience, 2000. ISBN 0471056693.

EBERLY, D. *Quaternion algebra and calculus*. [S.l.], 2002.

FALCÃO, A. X.; STOLFI, J.; LOTUFO, R. A. The image foresting transform: theory, algorithms, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 26, n. 1, p. 19–29, 2004.

FAWCETT, T. *ROC Graphs: Notes and Practical Considerations for Researchers*. [S.l.], 2004.

FERNANDES, S. E. N.; PAPA, J. P. Improving optimum-path forest learning using bag-of-classifiers and confidence measures. *springer Image Processing - Pattern Analysis and Applications (PAA)*, 2017. (submitted).

FERNANDES, S. E. N.; PAPA, J. P. Pruning optimum-path forest ensembles using quaternion-based optimization. *International Joint Conference on Neural Networks (IJCNN)*, 2017.

FERNANDES, S. E. N. et al. A probabilistic optimum-path forest classifier for binary classification problems. *Neural Processing Letters*, 2017. (submitted).

FERNANDES, S. E. N. et al. Learning kernels for support vector machines with polynomial powers of sigmoid. In: *27th SIBGRAPI Conference on Graphics, Patterns and Images*. [S.l.: s.n.], 2014. p. 259–265. ISSN 1530-1834.

FERNANDES, S. E. N. et al. Progress in pattern recognition, image analysis, computer vision, and applications: 20th iberoamerican congress. In: ____. [S.l.]: Springer International Publishing, 2015. cap. Improving Optimum-Path Forest Classification Using Confidence Measures, p. 619–625. ISBN 978-3-319-25751-8.

FERNANDES, S. E. N. et al. Chapter 2 - fine-tuning enhanced probabilistic neural networks using metaheuristic-driven optimization. In: YANG, X.-S.; PAPA, J. P. (Ed.). *Bio-Inspired Computation and Applications in Image Processing*. [S.l.]: Academic Press, 2016. p. 25 – 45. ISBN 978-0-12-804536-7.

FERNANDES, S. E. N. et al. Pruning optimum-path forest ensembles using metaheuristic optimization for land-cover classification. *International Journal of Remote Sensing*, v. 38, n. 20, p. 5736–5762, 2017.

FISTER, I. et al. Modified bat algorithm with quaternion representation. In: *2015 IEEE Congress on Evolutionary Computation (CEC)*. [S.l.: s.n.], 2015. p. 491–498. ISSN 1089-778X.

FISTER, I. et al. Modified firefly algorithm using quaternion representation. *Expert Systems with Applications*, v. 40, n. 18, p. 7220–7230, 2013. ISSN 0957-4174.

FREUND, Y.; SCHAPIRE, R. E. Experiments with a new boosting algorithm. In: SAITTA, L. (Ed.). *Proceedings of the Thirteenth International Conference on Machine Learning (ICML 1996)*. [S.l.]: Morgan Kaufmann, 1996. p. 148–156. ISBN 1-55860-419-7.

FREUND, Y.; SCHAPIRE, R. E. A short introduction to boosting. In: *In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*. [S.l.]: Morgan Kaufmann, 1999. p. 1401–1406.

FUMERA, G.; FABIO, R.; ALESSANDRA, S. A theoretical analysis of bagging as a linear combination of classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 30, n. 7, p. 1293–1299, 2008. ISSN 0162-8828.

GEEM, Z. W. *Music-Inspired Harmony Search Algorithm: Theory and Applications*. 1. ed. [S.l.]: Springer Publishing Company, Incorporated, 2009. ISBN 364200184X, 9783642001840.

GEMAN, S.; BIENENSTOCK, E.; DOURSAT, R. Neural networks and the bias/variance dilemma. *Neural Comput.*, MIT Press, Cambridge, MA, USA, v. 4, n. 1, p. 1–58, jan. 1992. ISSN 0899-7667.

GODBOLE, S.; SARAWAGI, S. Discriminative methods for multi-labeled classification. In: ____. *Advances in Knowledge Discovery and Data Mining: 8th Pacific-Asia Conference, PAKDD 2004, Sydney, Australia, May 26-28, 2004. Proceedings*. [S.l.]: Springer Berlin Heidelberg, 2004. p. 22–30. ISBN 978-3-540-24775-3.

GOLDBERG, D. What every computer scientist should know about floating-point arithmetic. *ACM Computing Surveys*, ACM, New York, NY, USA, v. 23, n. 1, p. 5–48, 1991. ISSN 0360-0300.

GOLDBERG, D. E. *Genetic Algorithms in Search, Optimization and Machine Learning*. 1st. ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1989. ISBN 0201157675.

GORDON, J.; SHORTLIFFE, E. H. Readings in uncertain reasoning. In: SHAFER, G.; PEARL, J. (Ed.). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1990. cap. The Dempster-Shafer Theory of Evidence, p. 529–539. ISBN 1-55860-125-2.

GÜNTER, S.; BUNKE, H. Optimization of weights in a multiple classifier handwritten word recognition system using a genetic algorithm. *Electronic Letters of Computer Vision and Image Analysis*, v. 3, n. 1, p. 25–44, 2004.

GUO, L.; BOUKIR, S. Fast data selection for SVM training using ensemble margin. *Pattern Recognition Letters*, v. 51, n. 0, p. 112–119, 2015. ISSN 0167-8655.

HANSEN, L. K.; SALAMON, P. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE Computer Society, Washington, DC, USA, v. 12, n. 10, p. 993–1001, 1990. ISSN 0162-8828.

HAYKIN, S. *Neural Networks: A comprehensive foundation*. [S.l.]: Prentice-Hall, 1998.

HAYKIN, S. *Neural Networks: A Comprehensive Foundation (3rd Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2007. ISBN 0131471392.

HO, T. K. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE Computer Society, Washington, DC, USA, v. 20, n. 8, p. 832–844, 1998.

HO, T. K. Multiple classifier systems: Second international workshop, mcs 2001 cambridge, uk, july 2–4, 2001 proceedings. In: ____. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001. cap. Data Complexity Analysis for Classifier Combination, p. 53–67. ISBN 978-3-540-48219-2.

HUANG, Y. S.; SUEN, C. Y. The behavior-knowledge space method for combination of multiple classifiers. In: *Computer Vision and Pattern Recognition, 1993. Proceedings CVPR '93., 1993 IEEE Computer Society Conference on*. [S.l.: s.n.], 1993. p. 347–352. ISSN 1063-6919.

HUNTER, J. D. Matplotlib: A 2d graphics environment. *Computing In Science & Engineering*, IEEE COMPUTER SOC, v. 9, n. 3, p. 90–95, 2007.

JACOBS, R. A. et al. Adaptive mixtures of local experts. *Neural Comput.*, MIT Press, Cambridge, MA, USA, v. 3, n. 1, p. 79–87, 1991. ISSN 0899-7667.

JAIN, A. K.; DUIN, R. P. W.; MAO, J. Statistical pattern recognition: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 22, n. 1, p. 4–37, 2000. ISSN 0162-8828.

JIA, L.; WANG, Y.; FAN, L. Multiobjective bilevel optimization for production-distribution planning problems using hybrid genetic algorithm. *Integrated Computer-Aided Engineering*, v. 21, n. 1, p. 77–90, 2014.

JODAVI, M.; ABADI, M.; PARHIZKAR, E. Jsobfusdetector: A binary pso-based one-class classifier ensemble to detect obfuscated javascript code. In: *International Symposium on Artificial Intelligence and Signal Processing*. [S.l.: s.n.], 2015. p. 322–327.

JOLY, M. M.; VERSTRAETE, T.; PANIAGUA, G. Integrated multifidelity, multidisciplinary evolutionary design optimization of counterrotating compressors. *Integrated Computer-Aided Engineering*, IOS Press, v. 21, n. 3, p. 249–261, 2014. ISSN 1069-2509.

KENNEDY, J.; EBERHART, R. *Swarm Intelligence*. [S.l.]: M. Kaufman, 2001.

KITTLER, J. et al. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE Computer Society, Washington, DC, USA, v. 20, n. 3, p. 226–239, 1998. ISSN 0162-8828.

KOZIOL, J.; HACKE, W. A bivariate version of andrews plots. *Biomedical Engineering, IEEE Transactions on*, v. 38, n. 12, p. 1271–1274, 1991. ISSN 0018-9294.

KRAWCZYK, B. One-class classifier ensemble pruning and weighting with firefly algorithm. *Neurocomputing*, v. 150, Part B, p. 490 – 500, 2015. ISSN 0925-2312.

KROGH, A.; VEDELSBY, J. Neural network ensembles, cross validation, and active learning. In: *Advances in Neural Information Processing Systems*. [S.l.]: MIT Press, 1995. p. 231–238.

KUNCHEVA, L.; SKURICHINA, M.; DUIN, R. P. W. An experimental study on diversity for bagging and boosting with linear classifiers. *Information Fusion*, v. 3, n. 4, p. 245–258, 2002.

KUNCHEVA, L. I. *Combining Pattern Classifiers: Methods and Algorithms*. [S.l.]: Wiley-Interscience, 2004.

LAM, L. Multiple classifier systems: First international workshop, mcs 2000 cagliari, italy, june 21–23, 2000 proceedings. In: ____. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000. cap. Classifier Combinations: Implementations and Theoretical Issues, p. 77–86. ISBN 978-3-540-45014-6.

LAM, L.; SUEN, S. Y. Application of majority voting to pattern recognition: an analysis of its behavior and performance. *IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans*, IEEE Press, Piscataway, NJ, USA, v. 27, n. 5, p. 553–568, 1997. ISSN 1083-4427.

LARKINS, R.; MAYO, M. Adaptive feature thresholding for off-line signature verification. In: *Image and Vision Computing New Zealand, 2008. IVCNZ 2008. 23rd International Conference*. [S.l.: s.n.], 2008. p. 1–6.

LEE, B. et al. A new ensemble learning algorithm using regional classifiers. *International Journal on Artificial Intelligence Tools*, v. 22, n. 04, p. 1350025, 2013.

LI, B. et al. Using ensemble classifier for small bowel ulcer detection in wireless capsule endoscopy images. In: *Robotics and Biomimetics (ROBIO), 2009 IEEE International Conference on*. [S.l.: s.n.], 2009. p. 2326–2331.

LI, C.-F.; YIN, J.-Y. Variational bayesian independent component analysis-support vector machine for remote sensing classification. *Computers & Electrical Engineering*, v. 39, n. 3, p. 717 – 726, 2013. ISSN 0045-7906. Special issue on Image and Video ProcessingSpecial issue on Recent Trends in Communications and Signal Processing.

LIN, H.-T.; LIN, C.-J.; WENG, R. C. A note on platt's probabilistic outputs for support vector machines. *Machine Learning*, v. 68, n. 3, p. 267–276, 2007.

LU, Y. Knowledge integration in a multiple classifier system. *Applied Intelligence*, v. 6, n. 2, p. 75–86, 1996. ISSN 1573-7497.

LUNA, J. M. et al. Reducing gaps in quantitative association rules: A genetic programming free-parameter algorithm. *Integrated Computer-Aided Engineering*, IOS Press, v. 21, n. 4, p. 321–337, out. 2014. ISSN 1069-2509.

MARAR, J. F. *Polinômios Potências de Sigmóides PPS: Uma nova técnica para aproximação de funções e construção de wavenets e suas aplicações em procesamento de imagens e sinais*. Tese (Doutorado) — Universidade Federal de Pernambuco, UFPE, 1997.

MARKATOPOULOU, F.; TSOUMAKAS, G.; VLAHAVAS, I. Instance-based ensemble pruning via multi-label classification. In: *22nd IEEE International Conference on Tools with Artificial Intelligence*. [S.l.: s.n.], 2010. v. 1, p. 401–408.

MARTÍNEZ-MUÑOZ, G.; HERNÁNDEZ-LOBATO, D.; SUÁREZ, A. An analysis of ensemble pruning techniques based on ordered aggregation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 31, n. 2, p. 245–259, 2009. ISSN 0162-8828.

MENDOZA, L. F.; VELLASCO, M.; FIGUEIREDO, K. Intelligent multiagent coordination based on reinforcement hierarchical neuro-fuzzy models. *International Journal of Neural Systems*, v. 24, n. 08, p. 1450031, 2014.

NABAVI-KERIZI, S. H.; ABADI, M.; KABIR, E. A pso-based weighting method for linear combination of neural networks. *Computers & Electrical Engineering*, Pergamon Press, Inc., Tarrytown, NY, USA, v. 36, n. 5, p. 886–894, 2010. ISSN 0045-7906.

NAPOLI, C. et al. Artificial intelligence and soft computing: 14th international conference, proceedings, part i. In: ____. [S.l.]: Springer International Publishing, 2015. (ICAISC '15), cap. Toward Work Groups Classification Based on Probabilistic Neural Network Approach, p. 79–89.

NEMENYI, P. *Distribution-free Multiple Comparisons*. [S.l.]: Princeton University, 1963.

NICULESCU-MIZIL, A.; CARUANA, R. Predicting good probabilities with supervised learning. In: *22nd International Conference on Machine Learning*. New York, NY, USA: ACM, 2005. (ICML '05), p. 625–632.

PAPA, J. et al. On the harmony search using quaternions. In: ____. *Artificial Neural Networks in Pattern Recognition: 7th IAPR TC3 Workshop, ANNPR 2016, Ulm, Germany, September 28–30, 2016, Proceedings*. Cham: Springer International Publishing, 2016. p. 126–137. ISBN 978-3-319-46182-3.

PAPA, J. P.; FALCÃO, A. X. A new variant of the optimum-path forest classifier. In: *Proceedings of the 4th International Symposium on Advances in Visual Computing*. [S.l.]: Springer Berlin Heidelberg, 2008. (Lecture Notes in Computer Science), p. 935–944. ISBN 978-3-540-89638-8.

PAPA, J. P.; FALCÃO, A. X. A learning algorithm for the optimum-path forest classifier. In: TORSELLO, A.; ESCOLANO, F.; BRUN, L. (Ed.). *Graph-Based Representations in Pattern Recognition*. [S.l.]: Springer Berlin Heidelberg, 2009, (Lecture Notes in Computer Science, v. 5534). p. 195–204. ISBN 978-3-642-02123-7.

PAPA, J. P. et al. Efficient supervised optimum-path forest classification for large datasets. *Pattern Recognition*, Elsevier Science Inc., New York, NY, USA, v. 45, n. 1, p. 512–520, 2012.

PAPA, J. P.; FALCÃO, A. X.; SUZUKI, C. T. N. Supervised pattern classification based on optimum-path forest. *International Journal of Imaging Systems and Technology*, John Wiley & Sons, Inc., New York, NY, USA, v. 19, n. 2, p. 120–131, 2009. ISSN 0899-9457.

PAPA, J. P.; FERNANDES, S. E. N.; FALCÃO, A. X. Optimum-path forest based on k-connectivity: Theory and applications. *Pattern Recognition Letters*, v. 87, p. 117 – 126, 2017. ISSN 0167-8655. Advances in Graph-based Pattern Recognition.

PARIS, P. C. D.; PEDRINO, E. C.; NICOLETTI, M. C. Automatic learning of image filters using cartesian genetic programming. *Integrated Computer-Aided Engineering*, IOS Press, v. 22, n. 2, p. 135–151, abr. 2015. ISSN 1069-2509.

PEREIRA, D. R. et al. Social-spider optimization-based support vector machines applied for energy theft detection. *Computers & Electrical Engineering*, v. 49, p. 25–38, 2016. Disponível em: <http://dx.doi.org/10.1016/j.compeleceng.2015.11.001>.

PIRES, R. G.; FERNANDES, S. E. N.; PAPA, J. P. Blur parameter identification through optimum-path forest. In: *17th international Conference on Computer Analysis of Images and Patterns*. [S.l.: s.n.], 2017.

PISANI, R. J. *Remote Sensing Datasets*. 2016. `http://wwwp.fc.unesp.br/~papa/recogna/remote_sensing.html`.

PISANI, R. J. et al. Toward satellite-based land cover classification through optimum-path forest. *IEEE Transactions on Geoscience and Remote Sensing*, v. 52, n. 10, p. 6075–6085, 2014. ISSN 0196-2892.

PLATT, J. C. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In: *Advances in Large Margin Classifiers*. [S.l.]: MIT Press, 1999. p. 61–74.

PONTI, M. P. Combining classifiers: From the creation of ensembles to the decision fusion. In: *Graphics, Patterns and Images Tutorials (SIBGRAPI-T), 2011 24th SIBGRAPI Conference on*. [S.l.: s.n.], 2011. p. 1–10.

PONTI, M. P.; PAPA, J. P. Improving accuracy and speed of optimum-path forest classifier using combination of disjoint training subsets. In: SANSONE, C.; KITTLER, J.; ROLI, F. (Ed.). *Multiple Classifier Systems*. [S.l.]: Springer Berlin / Heidelberg, 2011. (Lecture Notes in Computer Science, v. 6713), p. 237–248. ISBN 978-3-642-21556-8.

PONTI, M. P.; PAPA, J. P.; LEVADA, A. L. M. A markov random field model for combining optimum-path forest classifiers using decision graphs and game strategy approach. In: MARTIN, C. S.; KIM, S.-W. (Ed.). *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*. [S.l.]: Springer Berlin / Heidelberg, 2011, (Lecture Notes in Computer Science, v. 7042). p. 581–590. ISBN 978-3-642-25084-2.

QUINLAN, J. R. Bagging, boosting, and c4.s. In: *Proceedings of the Thirteenth National Conference on Artificial Intelligence - Volume 1*. [S.l.]: AAAI Press, 1996. p. 725–730. ISBN 0-262-51091-X.

RASHIDI, S.; RANJITKAR, P. Bus dwell time modeling using gene expression programming. *Computer-Aided Civil and Infrastructure Engineering*, v. 30, n. 6, p. 478–489, 2015. ISSN 1467-8667.

RÄTSCH, G.; ONODA, T.; MÜLLER, K.-R. Soft margins for adaboost. *Machine Learning*, v. 42, n. 3, p. 287–320, 2001. ISSN 1573-0565.

REYES, O.; MORELL, C.; VENTURA, S. Evolutionary feature weighting to improve the performance of multi-label lazy algorithms. *Integrated Computer-Aided Engineering*, IOS Press, v. 21, n. 4, p. 339–354, 2014. ISSN 1069-2509.

ROCHA, A.; PAPA, J. P.; MEIRA, L. A. A. How far do we get using machine learning black-boxes? *International Journal of Pattern Recognition and Artificial Intelligence*, v. 26, n. 02, p. 1261001, 2012.

ROMAY, M. G. *Hyperspectral Remote Sensing Scenes*. 2016. `http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes`.

SCHAPIRE, R. E. The strength of weak learnability. *Machine Learning*, Kluwer Academic Publishers, Hingham, MA, USA, v. 5, n. 2, p. 197–227, 1990. ISSN 0885-6125.

SCHAPIRE, R. E. et al. Boosting the margin: A new explanation for the effectiveness of voting methods. *The Annals of Statistics*, Institute of Mathematical Statistics, v. 26, n. 5, p. 1651–1686, 1998.

SCHLEIF, F.-M.; GISBRECHT, A.; TINO, P. Probabilistic classification vector machine at large scale. In: *2015 European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*. [S.l.: s.n.], 2015. p. 555–560.

SHABBIR, F.; OMENZETTER, P. Particle swarm optimization with sequential niche technique for dynamic finite element model updating. *Computer-Aided Civil and Infrastructure Engineering*, v. 30, n. 5, p. 359–375, 2015. ISSN 1467-8667.

SHAFER, G. *A Mathematical Theory of Evidence*. Princeton: Princeton University Press, 1976.

SHEEN, S. et al. Hybrid artificial intelligent systems: 7th international conference, hais 2012, salamanca, spain, march 28-30th, 2012. proceedings, part ii. In: ____. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. cap. Ensemble Pruning Using Harmony Search, p. 13–24. ISBN 978-3-642-28931-6.

SIDDIQUE, N.; ADELI, H. *Computational intelligence : synergies of fuzzy logic, neural networks and evolutionary computing*. [S.l.]: John Wiley & Sons, 2013.

SIDDIQUE, N.; ADELI, H. Applications of harmony search algorithms in engineering. *International Journal on Artificial Intelligence Tools*, v. 24, n. 06, p. 1530002, 2015.

SIDDIQUE, N.; ADELI, H. Harmony search algorithm and its variants. *International Journal of Pattern Recognition and Artificial Intelligence*, v. 29, n. 08, p. 1539001, 2015.

SIDDIQUE, N.; ADELI, H. Hybrid harmony search algorithms. *International Journal on Artificial Intelligence Tools*, v. 24, n. 06, p. 1530001, 2015.

SOUNDARARAJAN, K. P.; SCHULTZ, T. Learning probabilistic transfer functions: A comparative study of classifiers. *Computer Graphics Forum*, v. 34, n. 3, p. 111–120, 2015. ISSN 1467-8659.

TINOCO, S. L. J. L. et al. Ensemble of classifiers for remote sensed hyperspectral land cover analysis: An approach based on linear programming and weighted linear combination. In: *2013 IEEE International Geoscience and Remote Sensing Symposium - IGARSS*. [S.l.: s.n.], 2013. p. 4082–4085. ISSN 2153-6996.

TOMAN, H. et al. Generalized weighted majority voting with an application to algorithms having spatial output. In: *Proceedings of the 7th international conference on Hybrid Artificial Intelligent Systems - Volume Part II*. Berlin, Heidelberg: Springer-Verlag, 2012. p. 56–67. ISBN 978-3-642-28930-9.

TSOUMAKAS, G.; PARTALAS, I.; VLAHAVAS, I. Applications of supervised and unsupervised ensemble methods. In: ____. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. cap. An Ensemble Pruning Primer, p. 1–13. ISBN 978-3-642-03999-7.

TUMER, K.; GHOSH, J. Error correlation and error reduction in ensemble classifiers. *Connection Science*, v. 8, n. 3/4, p. 385–404, 1996.

WILCOXON, F. Individual Comparisons by Ranking Methods. *Biometrics Bulletin*, International Biometric Society, v. 1, n. 6, p. 80–83, dez. 1945. ISSN 00994987.

WOLPERT, D. H. Stacked generalization. *Neural Networks*, v. 5, p. 241–259, 1992.

XU, L.; KRZYZAK, A.; SUEN, C. Methods of combining multiple classifiers and their applications to handwriting recognition. *Systems, Man and Cybernetics, IEEE Transactions on*, v. 22, n. 3, p. 418–435, 1992. ISSN 0018-9472.

YANG, X.-S. Firefly algorithm, stochastic test functions and design optimisation. *International Journal Bio-Inspired Computing*, v. 2, n. 2, p. 78–84, 2010.

YANG, X.-S.; DEB, S. Engineering optimisation by cuckoo search. *International Journal of Mathematical Modelling and Numerical Optimisation*, v. 1, p. 330–343, 2010.

ZADROZNY, B.; ELKAN, C. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In: *Proceedings of the 18th International Conference on Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001. (ICML '01), p. 609–616. ISBN 1-55860-778-1.

ZADROZNY, B.; ELKAN, C. Transforming classifier scores into accurate multiclass probability estimates. In: *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2002. (KDD '02), p. 694–699. ISBN 1-58113-567-X.

ZENG, Z. et al. Antithetic method-based particle swarm optimization for a queuing network problem with fuzzy data in concrete transportation systems. *Computer-Aided Civil and Infrastructure Engineering*, v. 29, n. 10, p. 771–800, 2014. ISSN 1467-8667.

ZHOU, Z.-H. *Ensemble Methods: Foundations and Algorithms*. 1st. ed. [S.l.]: Chapman & Hall/CRC, 2012. ISBN 1439830037, 9781439830031.

ZHOU, Z.-H.; WU, J.; TANG, W. Ensembling neural networks: Many could be better than all. *Artificial Intelligence*, v. 137, n. 1–2, p. 239 – 263, 2002. ISSN 0004-3702.

GLOSSÁRIO

***k*-NN** – *k-nearest neighbours*

ANNs – *Artificial Neural Networks*

BKS – *Behavior-Knowledge Space*

CS – *Cuckoo Search*

EPNN – *Enhanced Probabilistic Neural Networks*

FFA – *Firefly Algorithm*

GP – *Genetic Programming*

HS – *Harmony Search*

MCS – *Multiple Classifier Systems*

MST – *Minimum Spanning Tree*

NB – *Newton Backtracking*

NN – *Nearest Neighbor*

OPF – *Optimum-Path Forest*

PSO – *Particle Swarm Optimization*

QHS – *Quaternion-based Harmony Search*

RSM – *Random Subspaces Method*

SVMs – *Support Vector Machines*