Willian P. Amorim

"Novas Abordagens de Aprendizado Semisupervisionado por Conectividade Ótima"

Campo Grande - MS
2016

Universidade Federal de Mato Grosso do Sul
Faculdade de Computação

Willian P. Amorim

"Novas Abordagens de Aprendizado Semisupervisionado por Conectividade Ótima"

Orientador(a): **Prof. Dr. Marcelo H. Carvalho**

Co-Orientador(a): **Prof. Dr. Alexandre X. Falcão**

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Ciência da Computação do Faculdade de Computação da Universidade Federal de Mato Grosso do Sul para obtenção do título de Doutor em Ciência da Computação.

Este exemplar corresponde à versão da Tese apresentada à banca examinadora por Willian P. Amorim, sob orientação de Prof. Dr. Marcelo H. Carvalho.

Assinatura do Orientador(a)

Campo Grande - MS
2016

Faculdade de Computação
Universidade Federal de Mato Grosso do Sul

Novas Abordagens de Aprendizado Semisupervisionado por Conectividade Ótima

Willian P. Amorim

19 de dezembro de 2016

Banca Examinadora:

- Prof. Dr. Marcelo H. Carvalho (*Orientador*)
- Prof. Dr. Alexandre X. Falcão (*Co-Orientador*)
  IC-Unicamp
- Prof. Dr. Moacir Ponte Jr.
  ICMC-USP
- Prof. Dr. Flávio Keidi Miyazawa
  IC-Unicamp
- Prof. Dr. Hemerson Pistori
  UCDB
- Edson Takashi Matsubara
  FACOM-UFMS
- Prof. Dr. Aparecido Nilceu Marana
  Unesp (*Suplente*)
- Prof. Dr. Bruno Magalhães Nogueira
  FACOM-UFMS (*Suplente*)

Abstract

The annotation of large data sets by a classifier is a problem whose challenge increases as the number of supervised samples available to train the classifier reduces in comparison to the number of unsupervised samples. In this context, semi-supervised learning methods aim at discovering and propagating labels to informative samples among the unsupervised ones, such that their addition to the correct class in the training set can improve the classification performance. This PhD thesis presents a series of novel semi-supervised learning approaches based on the Optimum-Path Forest (OPF) methodology. This methodology interprets the pattern recognition problem as a graph search problem, where the nodes are the training samples, the arcs are defined by a given adjacency relation, and the paths are assessed by some connectivity function. It identifies key nodes (prototypes) among the training samples and performs a competition process among them, such that each sample is conquered by the prototype that offers an optimum path to it. The result is a classifier — optimum-path forest rooted at the prototype set — which assigns labels to new samples by assessing extended paths to them. Classifiers can be created by one or multiple executions of the OPF algorithm for different graphs and connectivity functions. We present two approaches (OPFSEMI and its optimized version, OPFSEMI$_{mst}$) for the single-label problem, which differ from one another with respect to the final prototypes and number of executions of the OPF algorithm. We also propose a semi-supervised approach more suitable for the multi-label problem than the previous ones. This is a challenging problem, especially when it relies on the transformation of multi-label data into single-label data, which might affect performance at the boundary between classes. To resolve this problem, we improve the multi-label assignment by adding a final step in the training process of OPFSEMI$_{mst}$. The method, called OPFSEMI$_{mst+knn}$, creates an optimum-path forest rooted at the maxima of a probability density function, as estimated from a k-NN graph. Finally, we propose an active learning approach based on OPFSEMI$_{mst}$ (OPFSEMI). The method selects informative samples for expert supervision, such that the number of active learning iterations (user effort) is reduced.

Resumo

A anotação de grandes bases de dados por um classificador é um problema cujo desafio aumenta à medida que o número de amostras supervisionadas usadas para treinar o classificador reduz em comparação com o número de amostras não supervisionadas. Neste contexto, métodos de aprendizagem semisupervisionados visam a descoberta e propagação de rótulos para amostras informativas entre as não supervisionadas, de tal forma que a sua adição à classe correta no conjunto de treinamento possa melhorar o desempenho de classificação. Esta tese de doutorado apresenta uma série de novas abordagens de aprendizado semisupervisionado com base na metodologia adotada por Floresta de Caminhos Ótimos (OPF). Esta metodologia interpreta o problema de reconhecimento de padrões como um problema de busca em grafo, onde os nós são amostras de treinamento, os arcos são definidos por uma dada relação de adjacência, e os caminhos são avaliados por alguma função de conectividade. Nós protótipos são identificados entre as amostras de treinamento e a competição entre eles faz com que cada amostra seja conquistada (rotulada) pelo protótipo que lhe oferece um caminho ótimo. O resultado é um classificador — floresta de caminhos ótimos enraizado no conjunto de protótipos. Classificadores podem ser criados por uma ou múltiplas execuções do algoritmo OPF para diferentes grafos e funções de conectividade. Apresentamos duas abordagens (OPFSEMI e OPFSEMI$_{mst}$) para o problema de rótulo único, que diferem entre si em relação aos protótipos finais e ao número de execuções do algoritmo OPF. Também propomos uma abordagem semisupervisionada mais adequada para o problema multirótulos do que as anteriores. Este é um problema desafiador, especialmente quando a solução adota a transformação de dados de multirótulos em dados de rótulo único, o que pode afetar o desempenho na fronteira entre classes. Para resolver este problema, melhoramos a atribuição de multitótulos adicionando uma etapa final no processo de treinamento de OPFSEMI$_{mst}$. O método, chamado OPFSEMI$_{mst+knn}$, cria uma floresta de caminhos ótimos enraizada nos máximos de uma função de densidade de probabilidade, estimada a partir de um grafo k-NN. Finalmente, propomos uma abordagem de aprendizagem ativa baseada em OPFSEMI$_{mst}$ (OPFSEMI). O método seleciona amostras informativas para a supervisão de especialistas, de modo que o número de iterações no aprendizado ativo (esforço do usuário) é reduzido.

Agradecimentos

Inicialmente, gostaria de agradecer a Deus por me guiar, iluminar e me dar tranquilidade para seguir em frente com os meus objetivos e não desanimar com as dificuldades. Agradeço também a Nossa Senhora Aparecida, minha protetora de todos os dias.

Agradeço imensamente ao meu orientador Prof. Dr. Marcelo H. Carvalho e co-orientador Prof. Dr. Alexandre X. Falcão, pela ajuda, confiança, disponibilidade e amizade. Professores a qual criei um enorme respeito e admiração pela sua ética, seu conhecimento, mas principalmente pela maneira humilde de ensinar e acreditar no trabalho desenvolvido.

Agradeço a minha esposa Natália. Obrigado por estar sempre ao meu lado e por toda ajuda nos momentos de ansiedade em tempos de estudos, concursos, graduação, mestrado e doutorado. Obrigado por ser minha sustentação e por todo carinho. Te amo muito. Para você meu filho Olavo. Você foi a obra mais especial da minha vida. No dia da defesa do doutorado, acreditava que cairia em lágrimas de felicidade pela conquista do término. Mas não queria chorar, queria apenas voltar para casa e encontrar você e sua mãe para ficarmos juntos. Tinha certeza que a minha maior conquista já estava nos meus braços, e não nasceu no dia da defesa, e sim nasceu no dia 25 de abril de 2016. Que você possa no futuro seguir seus sonhos, e sempre em uma vida cristã, filho de Deus, um discípulo de Cristo. Amo imensamente você, a nossa família.

Agradeço a toda a minha família, (pais, irmãs, tios, tias, primos e primas) em especial a minha mãe (Euzani Ribeiro Paraguassu) pelo apoio, torcida e confiança que sempre depositam em mim, pelos momentos que não estivemos juntos e souberam entender. Obrigado!

Às demais pessoas que contribuíram direta ou indiretamente na elaboração deste trabalho ou participaram da minha vida, e que, por ventura, eu tenha me esquecido de agradecer. Obrigado a todos.

Sumário

Lista de Tabelas

Lista de Figuras

Capítulo 1

Introdução

1.1 Contexto e motivação

A organização, previsão e recuperação de dados são componentes cruciais em muitas aplicações, principalmente quando os conjuntos de dados são totalmente supervisionados. Em reconhecimento de padrões, presume-se que um classificador supervisionado possa classificar um conjunto de dados, quando um especialista fornece amostras de treinamento supervisionadas de todas as classes. No entanto, como os conjuntos de dados crescem em tamanho devido aos avanços dos sistemas de aquisição e armazenamento de dados [34], o projeto padrão de um classificador torna-se mais sensível à dimensão limitada do conjunto de treinamento e à escolha de suas amostras manualmente rotuladas. O problema nesta área então se refere ao custo de oferecer um conjunto de dados representativo e com padrões relevantes em um determinado domínio.

Esse custo é frequentemente associado à dificuldade de obtenção de amostras, ao tempo e complexidade do processo de rotulação, que são itens diretamente relacionados à experiência do especialista para tratar o problema. Por outro lado, há amostras não supervisionadas, que são geralmente disponíveis em grandes quantidades com um custo relativamente baixo para ser obtido. O problema com os métodos tradicionais de classificação é que eles não podem usar dados não supervisionados para treinar os classificadores. Neste contexto, uma pergunta que surge naturalmente é: *Podemos melhorar a eficácia do classificador, explorando a maior quantidade de dados não supervisionados?* Abordagens de aprendizado semisupervisionado têm buscado responder a esta pergunta.

Dada esta importante questão, esta tese de doutorado tem o objetivo de apresentar uma série de novas abordagens de aprendizado semisupervisionado baseada na metodolodia de Floresta de Caminhos Ótimos (OPF), inicialmente proposta para o projeto de operadores de processamento de imagens [29] e, posteriormente, estendida para o agrupamento e classificação [69, 72, 18, 66]. Na metodologia OPF, dados de amostras (por

exemplo: pixels, superpixels, imagens, objetos ou outras entidades) são interpretados como nós de um grafo, cujas arestas conectam amostras com base em alguma *relação de adjacência*. Para um dado problema, que pode ser direta ou indiretamente relacionado à partição ideal do conjunto de treinamento, uma adequada *função de conectividade* atribui um valor a qualquer caminho no grafo, incluindo os formatos triviais formados por nó único. A partir do mapa inicial de conectividade, onde todos os caminhos são triviais, o algoritmo propaga em uma ordem não decrescente do valor de caminho das propriedades dos dados a partir dos mínimos do mapa trivial para os nós restantes, de tal forma que cada nó seja conquistado por algum nó que ofereça o melhor caminho mínimo possível. O resultado é uma Floresta de Caminhos Ótimos enraizada pelos mínimos de um mapa de conectividade definitiva, ou seja, um mapa predecessor acíclico que atribui a cada nó seu predecessor no caminho ótimo ou um marcador distinto quando o nó é uma raiz do mapa. Atributos como valor do caminho ótimo e rótulo da raiz também são adicionados a cada nó, e o problema é reduzido a um processamento local desses atributos. Um classificador é então uma Floresta de Caminhos Ótimos, o qual pode propagar os rótulos para novas amostras, avaliando os caminhos estendidos.

Para entedermos melhor a lógica no aprendizado semisupervisionado baseado em Floresta de Caminhos Ótimos, e força de conectividade aplicada sobre as amostras, a Figura 1.1 apresenta um simples exemplo, comparando com o desempenho na propagação de rótulos no conjunto não supervisionado sobre três métodos supervisionados mais populares. Inicialmente, apresentamos um conjunto de treinamento totalmente não supervisionado (Figura 1.1a) e algumas amostras rotuladas selecionadas (Figura 1.1b). Esse conjunto de amostras supervisionadas é plausível, dado que um especialista deve selecionar e rotular amostras a partir de dados inicialmente não supervisionados com distribuição desconhecida no espaço de atributos[1]. Na aprendizagem supervisionada, um classificador é treinado usando apenas as amostras manualmente rotuladas, ou seja, isso limita sua capacidade de propagação correta dos rótulos para novas amostras. Os resultados dos classificadores, SVM com núcleo Radial Basis Function (RBF) [16], 1-NN [28], e o classificador mais popular supervisionado baseado em Floresta de Caminhos Ótimos [66], são apresentados nas Figuras 1.1c-e.

Podemos notar por esse exemplo que a estrutura das propostas a serem apresentadas no decorrer desta tese baseada em Floresta de Caminhos Ótimos no aprendizado semisupervisionado, apresenta uma considerável força de conectividade entre as amostras, permitindo assim uma correta propagação dos rótulos para as amostras não supervisionadas (Figura 1.1f). Deve ficar claro que, se a hipótese acima se confirmar, então os

[1]É comum a seleção de amostras uniformemente distribuídas de cada classe para o treinamento conjunto ao comparar classificadores. Note-se que esta opção não é muitas vezes possível, na prática, uma vez que os dados adquiridos são inicialmente não supervisionados.

algoritmos propostos neste trabalho também serão robustos para a seleção das amostras supervisionadas, ou seja, as amostras mais representativas. Quando não for este o caso, é necessária uma escolha correta de amostras supervisionadas, como será discutido mais à frente. Na prática, o número de amostras corretamente rotuladas no conjunto de treinamento pode aumentar consideravelmente, tornando possível melhorar o desempenho de classificação em novas amostras, em comparação com a abordagem supervisionada.

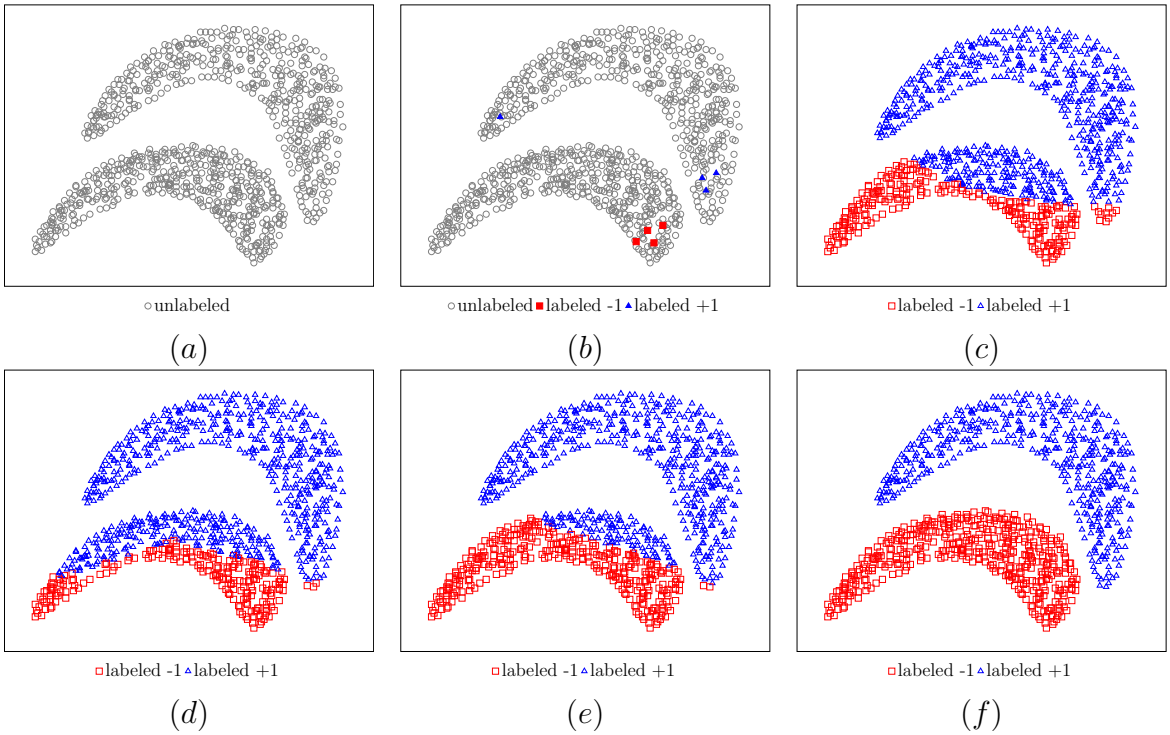


Figura 1.1: (a) Exemplo de um conjunto de treinamento com duas classes em forma de meia-lua. (b) Algumas amostras manualmente rotuladas para cada classe. Propagação dos rótulos para as amostras restantes por (c) SVM com RBF, (d) 1-NN, (e) OPF supervisionado e (f) modelo semisupervisionado das abordagens a serem propostas.

Os métodos aqui propostos têm sido extensivamente avaliados usando diferentes tipos de classificadores supervisionados e semisupervisionados, conjuntos de dados a partir de aplicações distintas e abordagens de aprendizagem *baseline* [22, 91, 10, 42, 50, 61]. Finalmente, a tese demonstra que novas melhorias são obtidas com a estratégia de aprendizagem ativa. Esta tese é baseada nos trabalhos científicos apresentados ou publicados, e as principais partes dos documentos são usadas direta ou indiretamente na síntese da explicação da tese.

Considerando as propostas elaboradas, apresentaremos a seguir uma visão geral dos principais objetivos e contribuições desta tese.

1.2 Objetivos e metodologia

O objetivo deste trabalho é propor uma estratégia geral para o uso de amostras não supervisionadas com base no aprendizado semisupervisionado, melhorando assim o desempenho do classificador baseado em Floresta de Caminhos Ótimos. Dada a importância da avaliação das propostas aqui apresentadas, avaliamos o desempenho do OPF semisupervisionado baseado em diferentes contextos. Todos os métodos recebem como entrada um conjunto de treinamento com amostras supervisionadas e não supervisionadas. Eles diferem com relação à escolha dos protótipos (raízes da floresta que vão propagar rótulos para as demais amostras de treinamento), função de conectividade, topologia do grafo, e número de execuções do algoritmo geral de floresta de caminhos ótimos para diferentes escolhas desses parâmetros

Inicialmente, apresentamos dois métodos semisupervisionados baseados em OPF, especificamente para classificação dos dados de único rótulo[2], chamados de OPFSEMI [5] e OPFSEMI$_{mst}$ [6]. Em OPFSEMI, o algoritmo geral OPF [69] é executado 4 vezes. A primeira para encontrar uma Árvore Geradora Mímina (Minimum Spanning Tree - MST) e identificar protótipos entre as amostras supervisionadas. A segunda para gerar uma floresta de caminhos ótimos enraizadas nesses protótipos, propagando rótulos para as amostras não supervisionadas. A terceira repete o primeiro procedimento agora com todas as amostras de treinamento, visando identificar mais protótipos (que podem incluir amostras não supervisionadas), e a quarta (última) repete o segundo para encontrar a floresta de caminhos ótimos final (classificador) com raízes nos novos protótipos.

Mais recentemente, propomos uma melhoria significativa da nossa abordagem anterior OPFSEMI. O método, chamado OPFSEMI$_{mst}$ [6], também explora a conectividade entre amostras supervisionadas e não supervisionadas. Em OPFSEMI$_{mst}$ reduzimos os quatro procedimentos de OPFSEMI a dois. A primeira execução calcula uma MST com todas as amostras e a segunda usa todas as amostras supervisionadas como protótipos e calcula a floresta final (classificador) sobre a topologia da MST, com raízes nas amostras supervisionadas. Este tipo de estrutura prevê que as amostras não supervisionadas com caminhos ótimos para outras amostras não supervisionadas serão conquistadas por amostras supervisionadas. Em geral, OPFSEMI$_{mst}$ é consideravelmente mais eficiente e preciso do que OPFSEMI, como vamos demonstrar neste trabalho.

Uma nova proposta semisupervisionada também foi desenvolvida para problemas multirótulos. Aprendizagem multirótulos se refere a problemas de classificação em que a mesma amostra pode ser atribuída a mais do que uma classe. Métodos para a classificação multirótulos existentes caem em duas categorias principais: *métodos de transformação* e *métodos de adaptação*. A primeira abordagem transforma o problema de atribuição mul-

[2]Cada amostra é associada a uma única classe.

tirótulos em várias tarefas de classificação de único rótulo. Basicamente, esta abordagem serve para transformar o problema multirótulos para uma visão de classificação binária[3] de vários conjuntos de dados ou a união de rótulos para a classificação multiclasse[4]. Realizando testes iniciais com a abordagem semisupervisionado OPFSEMI em problemas multirótulos, verificamos que na maioria das situações houve dificuldades na propagação dos rótulos para amostras não supervisionadas, devido à questão da transformação de multirótulos para único rótulo. O problema principal destas técnicas de transformação é que na maioria das situações o limite entre as classes de informações fica comprometido e cada classe é tratada individualmente, ignorando as possíveis relações entre elas. Assim, quando testada com OPFSEMI, houve uma grande perda de desempenho, pois, no momento da propagação do rótulo para as amostras não supervisionadas, era propagado um rótulo que não representava a relação real entre as classes, ou seja, o erro era potencializado.

O mesmo problema foi encontrado para OPFSEMI$_{mst}$, no entanto observamos que a propagação erra menos nas amostras que estão em regiões de alta densidade de probabilidade, possivelmente porque essas regiões possuem mais representantes entre as amostras supervisionadas. Para resolver este problema, melhoramos a atribuição multirótulo, adicionando um último passo para o processo de treinamento de OPFSEMI$_{mst}$, usando grafo k-vizinhos mais próximos e os máximos da função de densidade de probabilidade (pdf) como protótipos. Visto que esses máximos têm maior probabilidade de estarem corretamente rotulados, quando são amostras não supervisionadas, esta última execução repropaga seus rótulos para as demais amostras, corrigindo os erros de propagação do OPFSEMI$_{mst}$, e consequentemente melhorando significativamente o desempenho em problemas de atribuição multirótulos.

A partir das duas propostas semisupervisionadas baseadas em OPF (OPFSEMI e OPFSEMI$_{mst}$), assumimos que o conjunto das amostras supervisionadas é pré-determinado e fixo. Na prática, pode fazer sentido o uso da aprendizagem ativa em conjunto com o aprendizado semisupervisionado. Isto é, podemos permitir que o algoritmo de aprendizado escolha amostras não supervisionadas para serem rotuladas por um especialista do domínio do problema. Esse especialista então irá retornar o rótulo que, em seguida, será utilizado para aumentar o conjunto de dados supervisionados. Em outras palavras, se precisamos rotular alguns exemplos para o uso do aprendizado semisupervisionado, pode ser atraente permitir que o algoritmo de aprendizagem diga-nos quais as amostras a serem rotuladas, ao invés de selecionarmos aleatoriamente. Assim, uma nova contribuição deste trabalho chama-se Aprendizado Semisupervisionado Ativo usando OPF (ASSL-OPF) [74]. Esta proposta difere do aprendizado típico semisupervisionado em que os métodos pre-

[3]Na classificação binária, só existem duas possibilidades de classes.
[4]Na classificação multiclasse, existem várias possibilidades de classes.

cisam esperar pela escolha de todas as amostras supervisionadas e não supervisionadas antes de iniciar o processo de aprendizagem. Além disso, difere drasticamente do aprendizado ativo padrão no qual todas as amostras do conjunto de dados têm de ser classificadas e/ou reorganizadas em cada iteração de aprendizagem, ou seja, a proposta é capaz de selecionar mais rapidamente as amostras de todas as classes e manter uma interação mínima do usuário.

Resumindo, as principais contribuições deste trabalho são:

- As abordagens semisupervisionadas (OPFSEMI e OPFSEMI$_{mst}$) com base na metodologia por Floresta de Caminhos Ótimos;
- Uma abordagem de aprendizado semisupervisionado usando Floresta de Caminhos Ótimos, chamada OPFSEMI$_{mst+knn}$ sobre o contexto multirótulo;
- Uma forma de integrar aprendizado semisupervisionado e aprendizagem ativa usando classificadores de Floresta de Caminhos Ótimos;
- Nos trabalhos acima nós discutimos os prós e contras e validamos os métodos estatisticamente pelos testes de Friedman e Nemenyi.

1.3 Publicações

A lista abaixo fornece um resumo das principais publicações e trabalhos submetidos em processo de revisão, originadas durante o desenvolvimento desta tese:

- A. X. Falcão, J. P. Papa and W. P. Amorim, The newest version of LibOPF 3.0 (2016), (*in development*).
- W. P. Amorim, A. X. Falcão, J. P. Papa, and M. H. d. Carvalho, Improving semi-supervised learning through optimum connectivity, Pattern Recognition 60 (2016) 72–85.
- W. P. Amorim, A. X. Falcão and J. P. Papa, Multi-Label Semi-Supervised Classification Through Optimum-Path Forest Methodology, Information Sciences, 2016, (*in review*).
- W. P. Amorim, A. X. Falcão, and M. H. Carvalho. Semi-Supervised Pattern Classification Using Optimum-Path Forest. XXVII SIBGRAPI - Conference on Graphics, Patterns and Images, doi: 10.1109/SIBGRAPI.2014.45, Rio de Janeiro, RJ, pp. 111-118, 2014.

- P. T. M. Saito; W. P. Amorim.;, A. X. Falcão; P. J. Rezende; C. T. N. Suzuki; J. F. Gomes; M. H. Carvalho. Active Semi-Supervised Learning using Optimum-Path Forest. In: 22nd International Conference on Pattern Recognition (ICPR), 2014, Estocolmo. 22nd International Conference on Pattern Recognition (ICPR), 2014.

- J. P. Papa, W. P. Amorim, A. X. Falcão, and J. M. R. S. Tavares; Recent Advances on Optimum-Path Forest for data classification: Supervised, Semi-Supervised, and Unsupervised Learning. Handbook of Pattern Recognition and Computer Vision: 5th, pp. 109-123, (2016).

- P. T. M. Saito, R. Y. M. Nakamura, W. P. Amorim, J. P. Papa, P. J. de Rezende, A. X. Falcão. Choosing the most effective pattern classification model under learning-time constraint. PloS one, 2015.

- W. P. Amorim; M. H. Carvalho; V. V. V. A. Odakura. Face Recognition using Optimum-path Forest Local Analysis. In: Brazilian Conference on Intelligent Systems, 2013, Fortaleza. Proceedings of the 2013. Brazilian Conference on Intelligent Systems, 2013. v. 1. p. 242-248.

- L. A. M. Pereira; J. P. Papa; P. A. Jurandy; R. S. Torres; W. P. Amorim. A Multiple Labeling-Based Optimum-Path Forest for Video Content Classification. In: 2013 XXVI SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), 2013, Arequipa. 2013 XXVI Conference on Graphics, Patterns and Images, 2013. p. 334.

- L. N. B. Quinta; W. P. Amorim; M. H. Carvalho; M. P. Cereda; H. Pistori. Floresta de Caminhos Ótimos na Classificação de Pólen. In: WVC 2012 - Workshop de Visão Computacional, 2012, Goiânia. WVC 2012 - Workshop de Visão Computacional, 2012.

- W. P. Amorim; M. H. Carvalho. Supervised Learning Using Local Analysis in an Optimal-Path Forest. In: 2012 XXV SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), 2012, Ouro Preto. 2012 25th SIBGRAPI Conference on Graphics, Patterns and Images. p. 330.

1.4 Organização do trabalho

Esta tese de doutorado está organizada da seguinte forma: no Capítulo 2 apresentamos uma revisão dos principais fundamentos em aprendizado supervisionado, não supervisionado, semisupervisionado, ativo, classificação multirótulos e uma revisão bibliográfica abrangendo os principais trabalhos correlatos. Apresentamos também uma metodologia

geral sobre Floresta de Caminhos Ótimos, a qual auxiliará o entendimento das propostas apresentadas no Capítulo 3. Os resultados comparativos são apresentados no Capítulo 4. Finalmente, as conclusões e trabalhos futuros são apresentados no Capítulo 5.

Capítulo 2

Fundamentos e Trabalhos Relacionados

2.1 Considerações iniciais

Este capítulo resume os principais trabalhos e conceitos no campo de aprendizado e reconhecimento de padrões, bem como discute brevemente a metodologia geral para criação de classificadores baseados em Floresta de Caminhos Ótimos que serão necessários para uma compreensão adequada das propostas apresentadas.

2.2 Dados, treinamento, classificação e validação

Seja o conjunto $\mathcal{Z}$ de dados, cujos elementos são amostras (exemplos) s que provêm de alguma aplicação. As amostras podem ser imagens, objetos de uma imagem, regiões de uma imagem, pixels de uma imagem, sinais de voz, textos, etc. O importante é que cada amostra deve ser representada por um vetor de características no $\Re^n$ (n medidas) adequado à aplicação, de modo que amostras de uma mesma categoria (classe, padrão) sejam mapeadas em pontos próximos no $\Re^n$, da melhor forma possível. Na tese estamos interessados em dois tipos de problemas de reconhecimento de padrões: (1) classificação de amostra por único rótulo e (2) classificação de amostra multirótulos. No primeiro, cada amostra s tem um rótulo $\lambda(s)$ que determina uma única classe em um conjunto $\mathcal{C}$ de c possíveis classes. No segundo, uma amostra pode ser atribuída a múltiplos rótulos em $\mathcal{C}$. O problema em (1) consiste em treinar (projetar) um modelo (classificador) de reconhecimento de padrões capaz de associar um rótulo $\mathcal{L}(s)$ a uma amostra s de modo que $\mathcal{L} = \lambda(s)$. No caso (2), o problema pode ser transformado em múltiplos problemas do tipo (1), usando diversas estratégias, e depois as soluções podem ser combinadas

para resolver o problema (2). Com dispomos de um único conjunto de dados $\mathcal{Z}$ para cada aplicação e precisamos saber se o classificador pode ser generalizado para amostras que não foram vistas no treinamento, a base $\mathcal{Z}$ é normalmente dividida em três partes: $\mathcal{Z}_1$ (treinamento), $\mathcal{Z}_2$ (validação), e $\mathcal{Z}_3$ (teste). A finalidade de $\mathcal{Z}_2$ é o aprendizado de parâmetros do classificador. Neste trabalho, porém, os parâmetros (quando é o caso) são aprendidos em $\mathcal{Z}_1$, de modo que a base $\mathcal{Z}$ é dividida em $\mathcal{Z}_1$ (treinamento) e $\mathcal{Z}_2$ (teste). O propósito de $\mathcal{Z}_1$ é o projeto do classificador e o propósito de $\mathcal{Z}_2$ é avaliar sua capacidade de generalizar para amostras ainda não vistas. Ademais, os experimentos devem repetir para conjuntos aleatórios $\mathcal{Z}_1$ e $\mathcal{Z}_2$ de modo a levantar uma estatística sobre o desempenho do classificador. Neste trabalho, o conjunto $\mathcal{Z}_1$ é ainda constituindo de duas partes, $\mathcal{Z}_1^l$ e $\mathcal{Z}_1^u$, tal que a função λ é conhecida apenas para as amostras de $\mathcal{Z}_1^l$.

Com objetivo de facilitar a compreensão do aprendizado semisupervisionado, iremos iniciar uma explicação sobre aprendizagem supervisionada e não supervisionada.

2.3 Classificação supervisionada e não supervisionada

Tradicionalmente, existem dois tipos de tarefas em aprendizado de máquina: supervisionado e não supervisionado. Basicamente, a diferença entre as formas de aprendizado reside na disponibilidade ou não de uma amostra de casos com características observáveis conhecidas, em que a resposta também é conhecida.

No aprendizado supervisionado qualquer amostra de treinamento consiste no par $\mathcal{Z}_1^l$ e λ, sendo que cada amostra s tem um rótulo $\lambda(s)$. Pode-se pensar que $\lambda(s)$ como o rótulo de $\mathcal{Z}_1^l$ fornecido por um especialista ou supervisor, daí o nome de aprendizado supervisionado. Os pares (amostra, rótulo) são chamados de dados supervisionados, enquanto amostras sem rótulo são chamadas de dados não supervisionados. O objetivo então é: dado um conjunto de treinamento $\mathcal{Z}_1^l$ no domínio das amostras e rótulos $\mathcal{Z}_1^l$ x λ, encontrar uma função de treinamento de aprendizado supervisionado $f : \mathcal{Z}_1^l \mapsto \mathcal{L}$, referente à família de funções $\mathcal{F}$, com objetivo de $f(s)$ predizer o real rótulo λ sobre s. Ou seja, a classificação é um problema de aprendizado supervisionado com rótulos discretos $\mathcal{L}$. A função f é chamada de classificador. Entre os principais métodos de aprendizado supervisionado para classificação encontram-se as árvores de decisão [71], aprendizado baseado em regras [51], redes neurais [13], aprendizado bayesiano [11], máquinas de vetores de suporte [23], floresta de caminhos ótimos [66], entre outras.

No caso de algoritmos de aprendizado não supervisionado, estes trabalham diretamente com amostras de treinamento de $\mathcal{Z}_1^u$, sendo que não existe um especialista que forneça uma rotulação dos casos individuais de como as amostras devem ser rotuladas. Esta propriedade define a principal característica no aprendizado não supervisionado. Entre suas principais tarefas, podemos citar a clusterização, onde o objetivo é separar as instâncias

em diferentes grupos; detecção de novidades (casos em que apresentam diferenças com relação a maioria das instâncias); redução de dimensão, representando cada instância com uma dimensão inferior de maneira a não impactar em seu desempenho. Entre as principais técnicas de aprendizado não supervisionado podemos citar as regras de associação [3], análise de agrupamento ou clustering [53], análise de componentes [8], usando árvore geradora mínima [83], baseados em regiões de influência [49], soluções hierárquicas single-linkage [45], floresta de caminhos ótimos [72], entre outras.

2.4 Classificação semisupervisionada

Aprendizado semisupervisionado é uma abordagem que usa tanto amostras supervisionadas como não supervisionadas no treinamento. Seu principal objetivo é tirar proveito de grandes quantidades de amostras de custo baixo, resultado de um subproduto de processos ordinários. Usando este número de amostras, não é difícil inferir propriedades do domínio do problema que pode ser bastante útil no desenvolvimento de esquemas de treinamento eficazes. Aprendizado semisupervisionado inicialmente é motivado por seu valor prático em aprender mais rápido, melhor e mais barato.

Em muitas aplicações no mundo real, é relativamente fácil de adquirir uma grande quantidade de dados não supervisionados. Por exemplo, os documentos podem ser rastreados a partir da rede mundial de computadores, as imagens podem ser obtidas a partir de câmeras de vigilância, e áudios podem ser coletados a partir da emissão e transmissão de sons. No entanto, seus rótulos correspondentes para a tarefa de previsão, tais como orientação de posição, detecção de intrusão e transcrição fonética, muitas vezes requerem uma anotação (rotulação) humana extremamente lenta e caros experimentos de laboratório. Este gargalo de rotulagem resulta em uma escassez de dados supervisionados e um excedente de dados não supervisionados. Por isso, ser capaz de utilizar os dados não supervisionados em excesso é extremamente desejável.

A Figura 2.1 mostra um exemplo simples de aprendizado semisupervisionado. Cada instância é representada por uma característica unidimensional $s \in \mathbb{R}$. Existem duas classes: positivo e negativo. Considere o cenário a seguir:

Na aprendizagem supervisionada, usamos apenas duas amostras de treinamento rotuladas $(s_1, \lambda_1) = (-1, -)$ e $(s_2, \lambda_2) = (+1, +)$ sendo representados pelos simbolos $\bullet$ (rótulo negativo) e $\blacksquare$ (rótulo positivo), respectivamente. A melhor estimativa do limiar de decisão é $s = 0$; todas as instâncias com $s < 0$ devem ser classificadas como $\lambda = -$, enquanto que $s \geq 0$ como $\lambda = +$. Além disso, inserimos uma quantidade de amostras não supervisionadas. Os reais rótulos dessas amostras não supervisionadas são desconhecidas. No entanto, observamos que elas formam dois grupos. Sob a hipótese de que as instâncias em cada classe formam um grupo coerente, essas amostras não supervisionadas nos oferecem mais

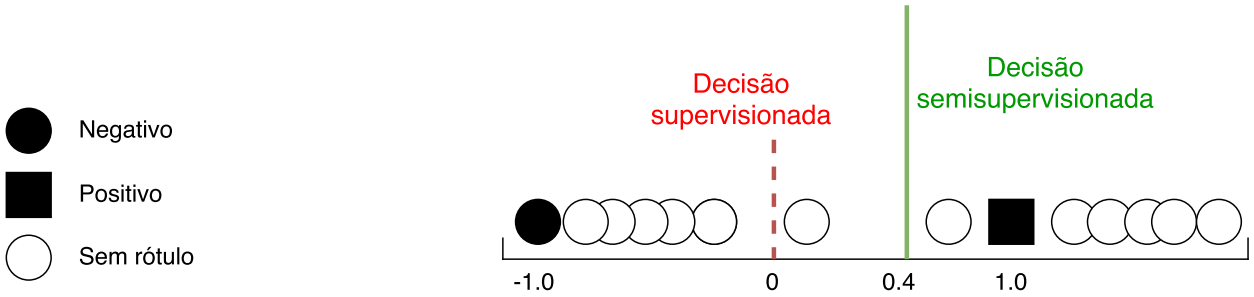


Figura 2.1: Exemplo de aprendizagem semisupervisionado.

informações sobre o real limite de decisão. Assim, nossa estimativa semisupervisionada da fronteira de decisão deve estar entre os dois grupos, aproximadamente $s \approx 0.4$.

2.4.1 Trabalhos relacionados

Embora abordagens heurísticas para o aprendizado semisupervisionado, como a auto formação, datam a partir da década de 1960 [19], esta categoria parece ser mais promissora para abordar a recente variedade de problemas que envolvem grandes quantidades de dados não supervisionados através da identificação e rotulação de algumas amostras representativas. Estratégias de aprendizado ativo combinados com o aprendizado semisupervisionado são promissores exemplos de aplicação [92]. Aprendizado semisupervisionado obteve uma explosão de interesse desde os anos 90, com o desenvolvimento de novos algoritmos, como co-formação e máquinas de vetores de suporte transdutivo, desenvolvendo novas aplicações em processamento de linguagem natural, visão computacional e novas análises teóricas.

A primeira abordagem para o aprendizado semisupervisionado baseado em Floresta de Caminhos Ótimos foi proposta por Amorim et. al [5] e otimizada com resultados promissores em [6]. Todos estes trabalhos serão apresentados nos próximos capítulos. Recentemente, métodos de aprendizagem semisupervisionada baseados em grafos têm atraído muita atenção. Métodos baseados em grafos iniciam a etapa de treinamento, a partir de um grafo onde os nós são as amostras supervisionadas e não supervisionadas, e arestas (ponderadas) refletem a semelhança entre nós. A suposição é de que nós ligados por uma aresta de maior peso tendem a ter o mesmo rótulo, os quais são propagados ao longo do grafo. Uma abordagem baseada em grafo para o aprendizado semisupervisionado foi implementada em SemiL[1] — uma ferramenta usando função harmônica [91] para a solução de problemas de grande escala de inferência transdutiva e aplicado com sucesso em

[1] http://www.support-vector.ws/html/semil.html

diferentes domínios [84, 87, 39, 36]. A idéia é predizer o rótulo de um vértice, calculando a média harmônica dos rótulos da vizinhança, utilizando o peso das arestas para ponderar a função, e devido à sua popularidade, é uma das principais técnicas usadas em nossos experimentos.

Outra solução é o algoritmo de Propagação de Rótulo [90], proposta que utiliza conceitos da função harmônica para propagar rótulos entre os componentes, sendo aqueles vértices que têm a maior probabilidade de pertencer a uma classe serão propagados. O grafo é então atualizado em um processo iterativo, buscando rotular o grafo completamente. Podemos citar também o método baseado em grafos conhecido como algoritmos de Corte Mínimo ou *Mincut* [14]. Sua idéia é gerar componentes no grafo de similaridade de forma que instâncias similares tenham valores com alto peso nas arestas. O algoritmo se baseia na técnica de corte mínimo em grafos com pesos. Para isso, busca encontrar a mínima quantidade de arestas que devem ser retiradas da rede para separá-los em componentes. Por fim, classifica as amostras não supervisionadas de cada componente com o valor do rótulo majoritário.

Basu et al. [9] propuseram dois métodos de aprendizado semisupervisionado, baseado no algoritmo de agrupamento k-means. Em ambos os métodos, as amostras supervisionadas são usadas para estimar as k amostras iniciais mais representativas como clusters. As amostras restantes, incluindo as não supervisionadas, são atribuídas ao conjunto das amostras representativas mais próximas. Os clusters são recalculados e o processo se repete até que se encontre uma convergência. No trabalho, não está claro se os métodos garantem pelo menos um cluster por classe, mas as alternativas para o problema têm sido propostas em [79].

Rosenberg et al. [73] apresentou uma estratégia baseada na auto formação [89]. Inicialmente, o classificador é treinado a partir das amostras supervisionadas realizando a rotulação (classificação) para as amostras não supervisionadas. As amostras classificadas com maior confiança (ou seja, que possuírem a maior chance de serem a real classificação) são adicionadas ao conjunto de treinamento e o processo começa novamente até atingir um critério de convergência. Outra abordagem que vem ganhando destaque com resultados promissores, mas que depende diretamente do classificador padrão a ser usado é YATSI [27]. O algoritmo YATSI é semelhante ao conceito auto formação, uma vez que pode ser encapsulado em torno de qualquer classificador e utiliza as suas próprias previsões no processo de formação. YATSI funciona em duas etapas: primeiramente, um classificador base é treinado sobre as amostras supervisionadas e, em seguida, as amostras não supervisionadas são "pré-rotuladas". A estas amostras pré-rotuladas são atribuídos pesos de confiança e usados pelo classificador k-vizinhos mais próximos para melhorar a classificação inicial.

Blum et al. [15] apresentou o método de co-formação, no qual as amostras não su-

pervisionadas são divididas em dois subgrupos. Dois classificadores supervisionados são treinados em cada subconjunto e cada classificador faz a sua previsão sobre as amostras não supervisionadas, ensinando ao outro classificador seus rótulos. As amostras classificadas com maior confiança são usadas para aumentar o conjunto de treinamento do outro classificador, e o processo de treinamento reinicia até atingir um critério de convergência.

Belkin et al. [10] propôs uma família de algoritmos de aprendizagem baseada em uma nova forma de regularização em variedades (*manifold regularization*) que permite explorar a geometria da distribuição marginal com foco sobre problemas de aprendizado semisupervisionado. A idéia é explorar a geometria da distribuição de probabilidade gerando dados que serão incorporados como um termo de regularização adicional. Um estudo realizado por Niyogi [64] apresenta trabalhos [12, 54] que sugerem que quando os dados obtém uma *manifold* de dimensão baixa, pode ser possível obter boas taxas no aprendizado usando métodos clássicos adequadamente adaptados. Mas alguns trabalhos (como exemplo, [54]) sugerem que *manifold regularization* não fornece nenhuma vantagem particular sobre os métodos tradicionais. Outro trabalho como (Goldberg, Zhu et al. [38]) apresenta uma análise teórica e prática para um algoritmo de aprendizado semisupervisionado com a suposição de multi-manifold.

Bridle e Zhu [17] propuseram o algoritmo p–voltages. Objetivo da proposta é rotular nós em um grafo baseado em suas tensões teóricas em um sistema reformulado de eletricidade. O trabalho prova que a solução p–voltages possui propriedades desejáveis para a aprendizagem semisupervisionada. Entretanto, seus experimentos e análises deixam claro que o algoritmo não consegue superar seus principais concorrentes, além de também ser uma proposta estritamente para problemas de classes binárias.

Mais recentemente, Iosifidis et al. [47] propôs um novo método com o objetivo de abordar a classificação de vídeos de ação *multi-view* semisupervisionada. A técnica é inspirada na aprendizagem usando subespaço discriminante e o algoritmo *Extreme Learning Machine* (ELM) [43] para treinamento de redes neurais *feedfoward* de única camada escondida. Em [42], os autores propõem as técnicas USELM e SSELM, que são extensões de ELMs para lidar com problemas de aprendizagem não supervisionada e semisupervisionada, respectivamente. As propostas também usam *manifold regularization* com objetivo de controlar o impacto no desempenho com o ausência ou escassez de dados supervisionados.

Máquinas de Vetores de Suporte (SVM) estão entre as abordagens supervisionadas e semisupervisionadas mais populares, e seu sucesso é atribuído à teoria margem de maximização [78]. A formulação de SVM maximiza a margem entre classes diferentes, levando a um modelo esparso dependendo das amostras de treinamento. Li et al. [56] apresentou *SVM-KNN* — uma abordagem semisupervisionada híbrida baseado em Máquinas de Vetores de Suporte e k-NN. O treinamento se inicia com um classificador SVM formado

a partir das amostras supervisionadas e propaga os demais rótulos para as amostras não supervisionadas. As amostras pertencentes às regiões de fronteira chamadas de *vetores de fronteira* são reclassificados pelo algoritmo k-NN, utilizando as amostras restantes rotuladas por SVM como conjunto de treinamento. Esse processo busca minimizar as chances de erros sobre regiões informativas usando a estratégia de k-vizinhos mais próximos.

Joachims [50] propôs as Máquinas de Vetores de Suporte Transdutivas (TSVM) [55]. O objetivo é encontrar uma rotulação das amostras não supervisionadas, de modo que a fronteira linear possa obter a margem máxima de separação em ambas amostras supervisionadas e não supervisionadas. No entanto, os hiperplanos de separação máxima do SVM entre as classes também devem satisfazer um segundo critério de estar longe de amostras não supervisionadas. A idéia não funciona bem para um grande número de amostras não supervisionadas [35], o que é geralmente o caso. Collobert et al. [22] então apresentou o procedimento de otimização côncava-convexa (CCP) para melhorar TSVM [5, 80, 58, 60, 81, 57, 33, 2, 59].

Podemos observar em geral que os métodos existentes em aprendizado semisupervisionado aprendem por duas formas: (a) pela propagação dos rótulos para amostras de treinamento não supervisionadas, uma a uma, através de classificação supervisionada [9, 56, 73, 15], ou (b) exploram a distribuição espacial das amostras supervisionadas e não supervisionadas do conjunto de treinamento no espaço de atributos para propagação do rótulo [5, 50]. Em ambas as categorias, o processo de propagação de rótulo pode repetir algumas vezes e um classificador final é então criado a partir do conjunto completamente rotulado (por exemplo, a partir de suas amostras rotuladas de maior confiança). Podemos até mesmo dizer que o classificador supervisionado [66], o qual chamaremos agora de OPFSUP, pode ser usado para propagar os rótulos para as amostras não supervisionadas, uma a uma, sem explorar a distribuição espacial das amostras não supervisionadas (categoria (a) acima) e, em seguida, ser treinado a partir do conjunto completamente rotulado. Neste trabalho, todas as propostas semisupervisionadas a serem apresentadas se enquadram na categoria (b) acima, apresentando precisões muito mais elevadas do que o de OPFSUP para a propagação rótulo, como também já apresentado pela Figura 1.1.

Pelos trabalhos apresentados verificamos também que cada um faz diferentes suposições. Estes métodos incluem desde auto formação, modelos probabilísticos generativos, co-formação, modelos baseados em grafos, máquinas de vetores de suporte semisupervisionado, e assim por diante. No entanto, é importante ressaltar que cegamente selecionando um método semisupervisionado de aprendizagem para uma tarefa específica não vai necessariamente melhorar o desempenho ao longo do aprendizado. Na verdade, os dados não supervisionados podem levar a um pior desempenho com suposições erradas. Para facilitar esse entendimento, iremos a seguir apresentar alguns pressupostos do aprendizado semisupervisionado.

2.4.2 Pressupostos do aprendizado semisupervisionado

Deve ficar claro que nem sempre será vantajoso utilizar dados não supervisionados. Por isso é necessário entender quando é possível combinar dados supervisionados e não supervisionados e melhorar o processo de aprendizado, ou seja, precisamos identificar se os dados não supervisionados poderão ser utilizados para ajudar no processo de aprendizado e que apresente informação útil para ajudar na classificação.

Como podemos saber se as informações não supervisionadas são úteis para o processo de aprendizado? Não existe uma resposta direta para essa questão. Por esse motivo, métodos de aprendizado semisupervisionado fazem alguns pressupostos sobre a relação entre a distribuição dos dados de treinamento [19].

O primeiro pressuposto é a (i) Suavidade (*Smoothness Assumption*). Esse pressuposto define que os rótulos dos exemplos devem variar de uma forma suave em áreas de alta densidade. Ou seja, se temos duas amostras próximas (s_1 e s_2), suas classificações devem ser similares (λ_1 e λ_2). Note então que se dois pontos estão ligados por um caminho de alta densidade (por exemplo, se eles pertencem ao mesmo grupo ou *cluster*), suas saídas são provavelmente próximas. Se, por outro lado, são separadas por uma região de baixa densidade, suas saídas não necessariamente estão próximas.

O segundo pressuposto é a (ii) Formação de Grupos (*Cluster Assumption*). Esse pressuposto assume que amostras que formam um mesmo grupo tendem a pertencer à uma mesma classe. Esse pressuposto pode ser visto também como uma extensão do pressuposto de suavidade, com a diferença que este pressuposto define que a formação de grupos implica áreas de separação de baixa densidade entre eles, o que não é explicitamente definido pelo pressuposto de suavidade. Importante destacar que a suposição de grupos não implica que cada grupo esteja contido em uma única classe. Isso apenas significa que, geralmente, não se observa amostras de classes distintas no mesmo grupo.

O pressuposto de formação de grupos também pode ser formulado de modo equivalente ao pressuposto de (iii) Separação de baixa densidade (*Low density separation*). Isto significa que a fronteira de decisão deverá ser em uma região de baixa densidade de amostras, visto que a fronteira de decisão em uma região de alta densidade poderia cortar um grupo em duas classes diferentes.

Por fim, o pressuposto de (iv) Geração de coleções (*Manifold assumption*). Esse pressuposto aponta que métodos baseados em grafos tipicamente formam coleções contínuas de dados em curvas hiper-dimensionais. Grafos de dimensionalidade geralmente têm grande capacidade de modelagem dos dados quando esse pressuposto é cumprido. Esse pressuposto é de grande importância para este trabalho, pelo fato de que métodos baseados em grafos visam estimar uma melhor função f que satisfaça duas restrições: concordar em grande parte com os rótulos dos vértices supervisionados e deve cumprir o pressuposto de suavidade, ou seja, o importante é a construção de um grafo de boa qualidade informativa.

2.5 Metodologia de aprendizado de máquina por Floresta de Caminhos Ótimos

Nesta seção, iremos apresentar a metodologia de aprendizagem baseada em Floresta de Caminhos Ótimos (OPF). OPF é uma metodologia para projeto de classificadores de padrões (não supervisionado, supervisionado, e agora semisupervisionado) a partir de um conjunto de treinamento, buscando explorar a *conectividade ótima* entre as amostras [66] e que tem sido aplicado com sucesso em problemas de processamento e análise de imagens [66, 48, 18, 72, 25].

O conjunto de treinamento é interpretado como um grafo, cujos nós são as amostras, as arestas são definidas por uma *relação de adjacência* e o classificador é então uma Floresta de Caminhos Ótimos no grafo com raízes em amostras *protótipos*. Portanto, as novas amostras podem ser rotuladas pela avaliação do caminho ótimo a partir do conjunto de protótipos de forma incremental, ou seja, a amostra recebe o rótulo do protótipo mais fortemente conexo. Para conjuntos de treinamento supervisionados, os protótipos são os defensores de suas classes [69], e podem ser estimados como as amostras de classes distintas mais próximas. Para uma *função de custo de caminho* adequada, cada classe torna-se uma Floresta de Caminhos Ótimos enraizadas em seus protótipos. Para conjuntos de treinamento não supervisionados, protótipos podem ser estimados como os máximos de uma função de densidade de probabilidade (fdp) das amostras de treinamento [72]. Para esse caso, uma *função de custo de caminho* adequada, cada região da fdp irá se tornar um cluster (árvore de caminho ótimo) enraizada no seu respectivo máximo.

2.5.1 Fundamentação teórica

O algoritmo de Floresta de Caminhos Ótimos é essencialmente o procedimento de Dijkstra para o cálculo de caminhos mínimos a partir de uma única fonte, ligeiramente modificado para permitir múltiplas fontes e funções de valor de caminho mais genéricas [29]. OPF é composto por dois diferentes classificadores supervisionados: o primeiro utiliza um grafo completo como relação de adjacência e escolhe os protótipos na fronteira entre as classes com diferentes rótulos [69, 66]. Na versão mais recente do OPF, o grafo é visto como uma estrutura k-NN (OPF_{kNN}) modelando também as amostras de treinamento como nós de um grafo e arestas ponderadas pela distância entre os correspondentes vetores de atributos [65]. A diferença básica entre OPF_{kNN} e o algoritmo padrão OPF é o fato de que sua estimativa dos protótipos se estabelece na fronteira entre as classes e uso de uma árvore geradora de custo mínimo (MST), resultado do cômputo de OPF com função de custo de caminho específica, enquanto OPF_{kNN} estima os protótipos nas regiões com alta concentração de amostras e uma estrutura k-NN.

O classificador OPF supervisionado tem como vantagem um baixo custo computacional de treinamento [75], uma vez que não há necessidade de otimização de parâmetros. Papa et al. [69] mostrou que sua fase de treinamento pode ser consideravelmente mais rápida do que as fases de treinamento de algoritmos como SVMs (Support Vector Machines) e ANNs (Artificial Neural Networks), com uma precisão melhor ou equivalente aos obtidos por essas abordagens. O classificador OPF tem sido amplamente utilizado em diversas aplicações, tais como sensoriamento remoto, reconhecimento de emoções através de processamento de fala, classificação automática de vogais, biometria, monitoramento de perfuração de poços de petróleo, segmentação de imagens médicas, e rastreamento de objetos [67, 46, 70, 20, 68, 40, 62]. Melhorias consideráveis têm sido continuamente apresentadas para fazer o classificador OPF mais eficiente para grandes conjuntos de dados [66, 48].

2.5.2 Definições

Seja $\mathcal{Z}$ um conjunto de dados tal que cada amostra $s \in \mathcal{Z}$ é representada por um vetor de atributos $\vec{v}(s) \in \Re^n$. Iremos dividir aleatoriamente $\mathcal{Z}$ em subconjuntos $\mathcal{Z}_1$ para concepção do classificador (treinamento), $\mathcal{Z}_2$ para teste de sua capacidade de generalização e $\mathcal{Z}_3$ para uso auxiliar no aprendizado. O conjunto $\mathcal{Z}_1 = \mathcal{Z}_1^l \cup \mathcal{Z}_1^u$ consiste de um subconjunto de amostras supervisionadas $\mathcal{Z}_1^l$ e não supervisionadas $\mathcal{Z}_1^u$, e $\mathcal{Z}_3$ como sendo um conjunto supervisionado maior do que $\mathcal{Z}_1^l$. A finalidade é avaliar o impacto da aprendizagem de amostras mais representativas para $\mathcal{Z}_1^l$, ao classificar amostras de $\mathcal{Z}_3$ e substituindo aleatoriamente por amostras não protótipos de $\mathcal{Z}_1^l$ com amostras classificadas com erro de $\mathcal{Z}_2$. Depois de aprender a partir de erros em $\mathcal{Z}_3$, espera-se que o classificador melhore a precisão em $\mathcal{Z}_2$ e diminua o erro de propagação de rótulo em $\mathcal{Z}_1^u$.

Além disso, seja $\lambda(s) \in \{1, 2, \ldots, \mathcal{L}\}$ representando o real rótulo de cada amostra $s \in \mathcal{Z}$, sendo $\mathcal{L}$ todos os possíveis rótulos, e $d(s,t) \geq 0$ uma função de distância simétrica entre amostras de acordo com os seus vetores de atributos, tal como $d(s,t) = \|\vec{v}(t) - \vec{v}(s)\|$.

2.5.3 Treinamento

Na metodologia por Floresta de Caminhos Ótimos, pode-se criar um classificador supervisionado a partir de $\mathcal{Z}_1^l$, um classificador não supervisionado a partir de $\mathcal{Z}_1^u$, ou um classificador semisupervisionado a partir de $\mathcal{Z}_1$ definido por uma *relação de adjacência* $\mathcal{A}$ e uma *função de conectividade* f.

A relação de adjacência $\mathcal{A}$ diz como amostras de treinamento devem estar conectadas no espaço de atributos, formando um grafo ponderado $(\mathcal{N}, \mathcal{A}, d)$, onde cada par $(s,t) \in \mathcal{A}$ é um arco ponderado pela distância $d(s,t)$ entre seus nós $s, t \in \mathcal{N} \subseteq \mathcal{Z}_1$. Nós também usamos $t \in \mathcal{A}(s)$ para indicar um elemento do conjunto de nós adjacentes de s. A

função de conectividade f atribui um valor $f(\pi_t)$ a qualquer sequência de nós distintos $\langle s_1, s_2, \ldots, s_n = t\rangle$ (um caminho simples), $(s_i, s_{i+1}) \in \mathcal{A}$, $i = 1, 2, \ldots, n-1$, com terminal em t no grafo, incluindo os caminhos triviais $\pi_t = \langle t\rangle$. O valor do caminho trivial indica o custo de iniciar um caminho a partir de t, enquanto o valor de um caminho simples com terminal em $s_n = t$ indica o custo de conquistar t por um caminho que comece em algum nó $s_1 \neq t$. Um caminho π_t é ótimo para uma dada função de conectividade quando $f(\pi_t) \leq f(\tau_t)$ para qualquer outro caminho τ_t com terminal em t no grafo, independentemente de sua origem. Podemos escrever também $\pi_t = \pi_s \cdot \langle s, t\rangle$ para indicar a extensão de um caminho π_s pelo arco (s, t) no grafo.

Para uma função de conectividade f não decrescente, o algoritmo começa a partir de caminhos triviais, de modo que os mínimos do mapa de custo inicial de $C_0(t) = f(\langle t\rangle)$, $\forall t \in \mathcal{N}$, competem entre si, oferecendo caminhos de menores custos uns aos outros e para os nós restantes. Este processo de relaxamento cria uma *Floresta de Caminhos Ótimos* enraizada nos mínimos do mapa de conectividade final $C(t) = \min_{\forall \pi_t \in \Pi_t}\{f(\pi_t)\} \leq C_0(t)$, onde Π_t é o conjunto de todos os caminhos possíveis com terminal em t. A floresta é um mapa de predecessores acíclico P que atribui a cada nó t o seu predecessor $P(t)$ no caminho ótimo ou um marcador distinto $P(t) = nil$, quando o nó é uma raiz (mínimo) do mapa.

2.5.4 Classificação

Ao atribuir $L(s) \leftarrow \lambda(s)$ (rótulo de um cluster) para uma raiz s da floresta, pode-se atribuir o nó raiz $R(t) \leftarrow s$ e seu rótulo $L(t) \leftarrow L(s)$ a cada nó t conquistado por s na floresta. A floresta de caminhos ótimos P é então um classificador, que pode propagar rótulos de classe/cluster para novas amostras $t \in Z_2$, considerando:

$$C(t) = \min_{\forall s \in \mathcal{N}}\{f(\pi_s \cdot \langle s, t\rangle)\} \tag{2.1}$$

o custo de estender um caminho de π_s por um segmento de $\langle s, t\rangle$. Considere $s^* \in \mathcal{N}$ o nó que satisfaz esta equação, então o classificador atribui $L(t) \leftarrow L(s^*)$. Ocorre um erro de classificação quando $\lambda(t) \neq \lambda(R(s^*))$.

Operações distintas podem exigir outras variantes. Por exemplo, um processo semelhante pode ser definido para funções de conectividade de uma ordem decrescente e raízes nos máximos de mapa de conectividade. A seguir, nas Seções 2.6, 2.7 e 2.8 estaremos apresentando soluções de aprendizado com base na metodologia OPF supervisionada usando grafo completo, não supervisionada e supervisionada usando grafo k-NN, respectivamente.

2.6 Classificação supervisionada por OPF usando grafo completo

Floresta de Caminhos Ótimos com grafo completo considera $(\mathcal{Z}_1^l, \mathcal{A}, d)$ como um grafo completo e ponderado, cujos nós são as amostras supervisionadas $s \in \mathcal{Z}_1^l$, as arestas definidas entre todos os pares de amostras distintas $(s,t) \in \mathcal{A} = \mathcal{Z}_1^l \times \mathcal{Z}_1^l$, e o peso das arestas $(s,t) \in \mathcal{A}$ são dadas por $d(s,t)$. OPF com grafo completo requer uma função de conectividade $f_{\max}(\pi_t)$ definidos por caminhos simples, bem como para os caminhos triviais $\pi_t = \langle t \rangle$, como:

$$\begin{aligned} f_{\max}(\langle s \rangle) &= \begin{cases} 0 & \text{se } s \in \mathcal{S}^*, \\ +\infty & \text{caso contrário,} \end{cases} \\ f_{\max}(\pi_s \cdot \langle s,t \rangle) &= \max\{f_{\max}(\pi_s), d(s,t)\}. \end{aligned} \tag{2.2}$$

O conjunto de sementes (raízes da floresta ou protótipos) $\mathcal{S}^* \subset \mathcal{Z}_1^l$ são obtidos calculando uma árvore geradora mínima (MST) em $(\mathcal{Z}_1^l, \mathcal{A}, d)$ — ou seja, produzindo um grafo acíclico ponderado $\mathcal{B}$ com custo mínimo total nas arestas. Durante o cálculo MST, todos os pares de amostras $(s,t) \in \mathcal{B}$ tal que $\lambda(s) \neq \lambda(t)$ são considerados como protótipos em $\mathcal{S}^*$.

OPF assume que $(\mathcal{Z}_1^l, \mathcal{B}, d)$ e $\mathcal{S}^*$ produz uma Floresta de Caminhos Ótimos para $f_{\max}$ – ou seja, um mapa acíclico P que atribui ao predecessor $P(t) \in \mathcal{Z}_1^l$ o melhor caminho π_t com término em t para todos os nós $t \in \mathcal{Z}_1^l \backslash \mathcal{S}^*$ e $P(t) = nil \notin \mathcal{Z}_1^l$ quando $t \in \mathcal{S}^*$. Também emite um rótulo $L(t)$ com raiz $R(t)$, e o custo $C(t)$ do caminho ótimo π_t para todos nós $t \in \mathcal{Z}_1^l$.

Para a classificação, a abordagem OPF calcula o custo mínimo:

$$C(t) = \min_{\forall s \in \mathcal{Z}_1^l} \{\max\{C(s), d(s,t)\}\} \tag{2.3}$$

estendendo um caminho ótimo π_s por uma aresta (s,t) para uma nova amostra $t \in \mathcal{Z}_2$. Seja $s^* \in \mathcal{Z}_1^l$ um nó que satisfaça a Equação 2.3, $L(s^*) = \lambda(R(s^*))$, ou seja, a classe referente ao protótipo mais fortemente conectado $R(s^*) \in \mathcal{S}^*$ será atribuído como a classe $L(t) \in \{1, 2, \ldots, c\}$.

A Figura 2.2(a) apresenta uma MST com os protótipos selecionados na fronteira (linhas pontilhadas). Em seguida, podemos iniciar o processo de competição entre os protótipos a fim de construir a Floresta de Caminhos Ótimos. A fase de classificação é realizada tomando uma amostra do conjunto de teste (triângulo na Figura 2.2(b)) e conectá-la a todas as amostras de treinamento. A distância para todos os nós de treinamento é calculada e usada para ponderar as arestas. Por fim, cada nó de treinamento oferece para

a amostra de teste um custo determinado por uma função de custo de caminho (aresta com maior peso ao longo de um caminho), e o nó de treinamento que ofereceu o caminho de custo mínimo vai conquistar a amostra de teste. Este procedimento é apresentado na Figura 2.2(c).

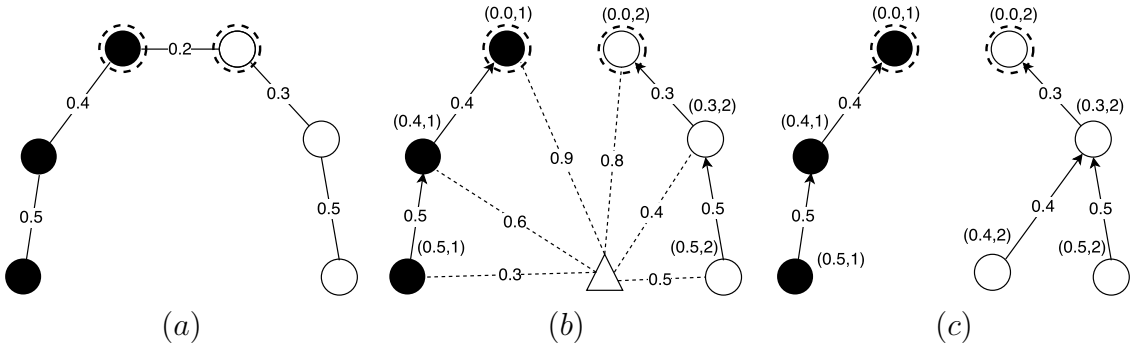


Figura 2.2: MST referente a um grafo completo ponderado nas arestas para um determinado conjunto de treinamento, e os protótipos selecionados (linhas pontilhadas) a partir da heurística de amostras mais próximas de classes distintas (• amostras supervisionadas da classe 1, ∘ amostras supervisionadas da classe 2). (b) Uma Floresta de Caminhos Ótimos enraizadas em seus respectivos protótipos, (△) uma amostra de teste, e suas possíveis conexões com todos os elementos no grafo de treinamento. Os identificadores (x, y) acima dos nós são, respectivamente, o custo e o rótulo dos mesmos. A seta indica o nó predecessor no caminho ótimo. (c) Caminho ótimo do protótipo mais fortemente conexo.

Para facilitar a compreensão da abordagem supervisionada e não supervisionada, iremos ilustrar o desempenho prático de cada técnica com base em dados de difícil classificação chamado Boat[2], principalmente para técnicas que dependem da localização das projeções lineares buscando maximizar a separação entre as classes. Basicamente, a base de dados possui duas classes com regiões em cluster no centro, sendo coberta por uma terceira classe. Considerando o conjunto de dados originais (Figura 2.3(a)), dividimos aleatoriamente em 3 outros conjuntos de dados, sendo uma supervisionada (Figura 2.3(b)), não supervisionada (Figura 2.3(c)) e um conjunto de teste (Figura 2.3(d)).

A partir da Figura 2.4, podemos entender melhor o comportamento prático desta abordagem supervisionada usando grafo completo. Dado o conjunto supervisionado selecionado como ilustrado na Figura 2.4(a), após o processo de treinamento de OPF (Figura 2.4(b)) em conjunto com protótipos selecionados, conseguimos produzir a classificação referente às amostras de testes (Figura 2.4(c)), e seu respectivo resultado (Figura 2.4(d)). Podemos observar o posicionamento dos protótipos selecionados nas regiões

[2]`http://pages.bangor.ac.uk/mas00a/activities/artificial_data.htm`

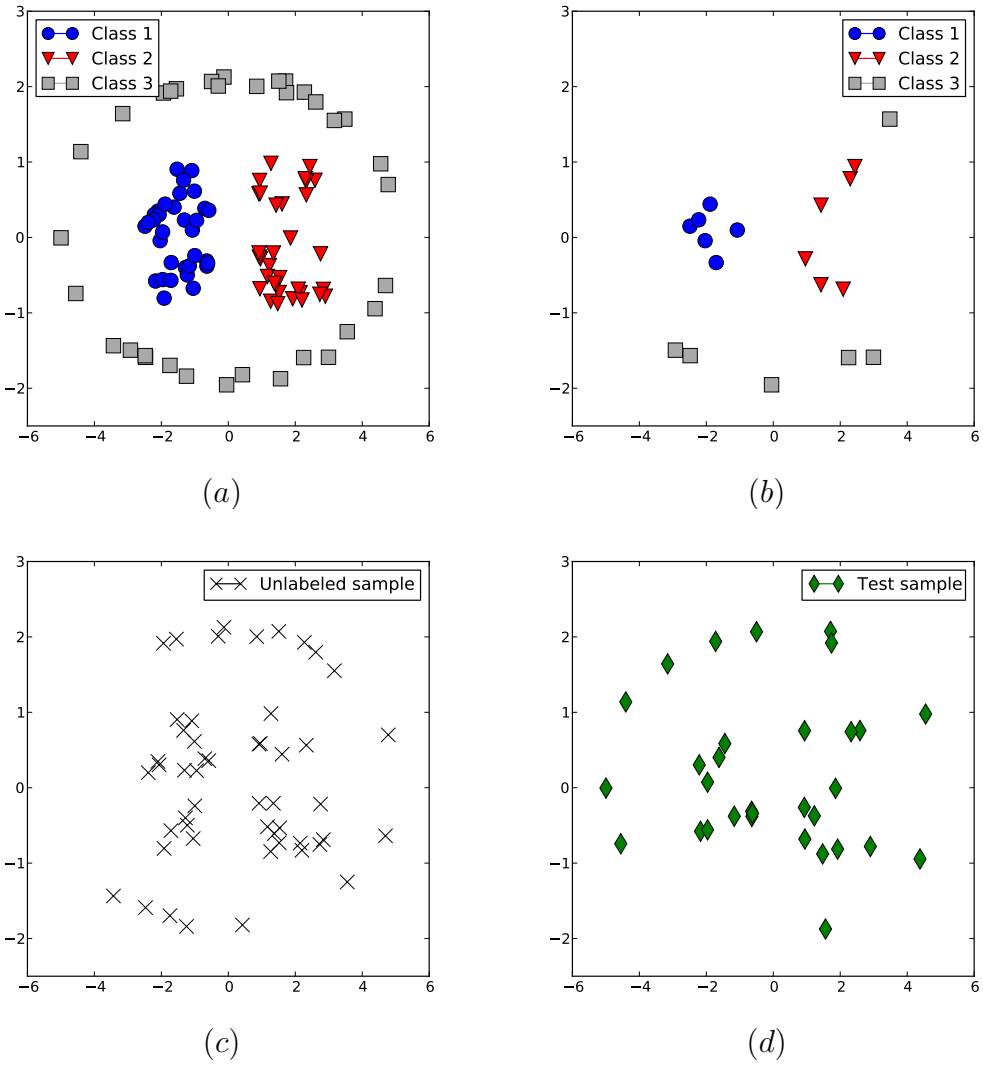


Figura 2.3: Conjunto de dados particionados. (a) conjunto de dados originais, (b) conjunto de dados supervisionados, (c) conjunto de dados não supervisionados e (d) amostras de teste.

de fronteira entre as classes e o seu impacto no resultado da classificação. Mesmo com a força de conectividade do OPF, é muito importante a seleção de amostras representativas supervisionadas para que possa melhor estimar os protótipos, e evitar dificuldades na classificação, como mostrado pela classe 1 e a classe 3.

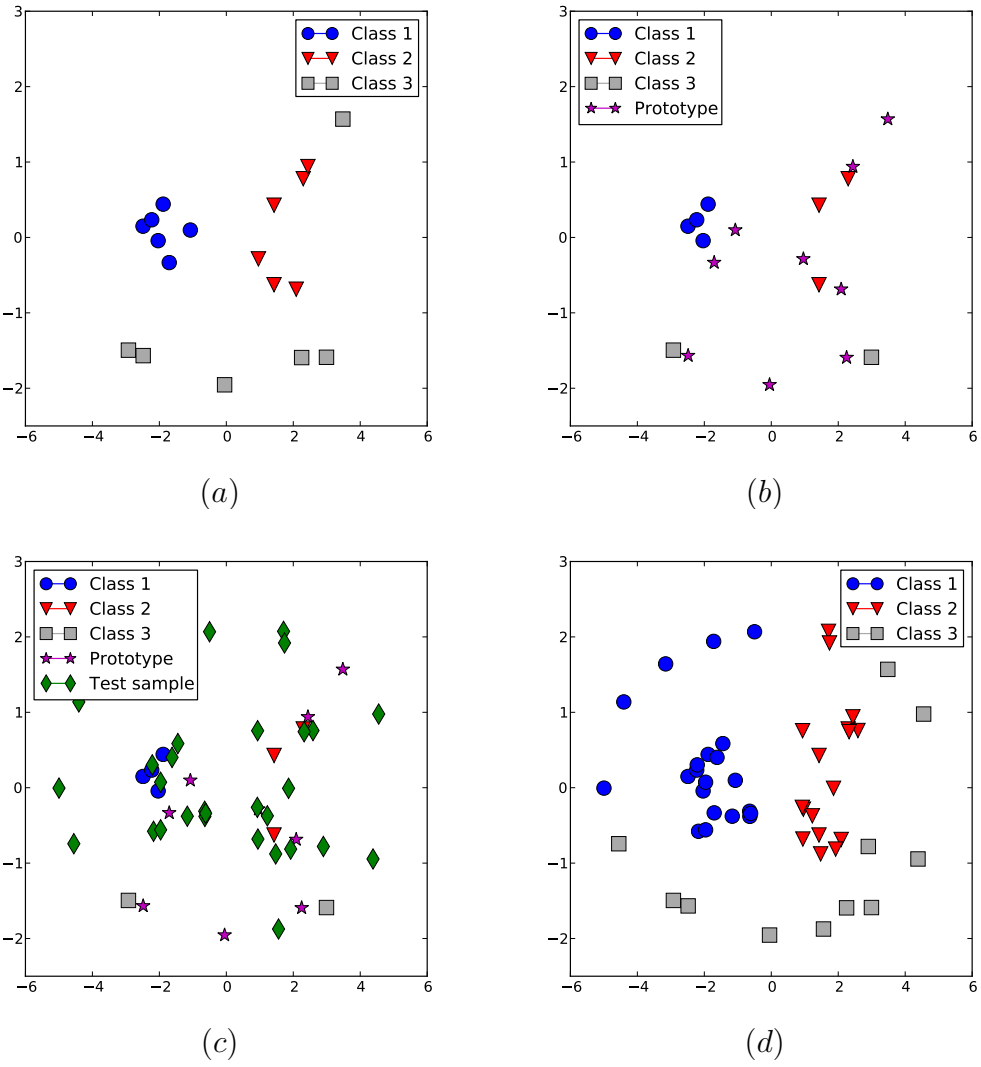


Figura 2.4: Comportamento prático de OPF (grafo completo). (a) conjunto de dados supervisionados, (b) protótipos selecionados, (c) amostras a serem classificadas em conjunto com os protótipos selecionados e (d) resultado da classificação pela abordagem OPF.

2.7 Classificação não supervisionada por OPF usando grafo k-NN

A aprendizagem não supervisionada baseado em Floresta de Caminhos Ótimos, foi proposta inicialmente por Rocha et al. [72], o qual foi desenvolvido com objetivo de identificar clusters como sendo as árvores de uma Floresta de Caminhos Ótimos, em que amostras

de treinamento são não supervisionadas e conectadas aos k vizinhos mais próximos no espaço de atributos.

O grafo é ponderado nos nós por valores de densidades originando, assim, uma função de densidade de probabilidade (pdf), a qual é calculada levando-se em consideração as distâncias (peso dos arcos) entre os vetores de atributos de amostras adjacentes. O valor do melhor k é encontrado minimizando uma medida de corte em grafo, pelos resultados de clustering para $k^* \in [1, k_{max}]$ e a maximização de uma função de valor de caminho origina uma floresta de caminhos ótimos, onde cada árvore (cluster) é enraizada em um máximo da pdf.

Seja $\mathcal{A}_k$ uma relação de adjacência k-NN que inicialmente conecta cada amostra $s \in \mathcal{Z}_1$ com seus k vizinhos mais próximos de acordo com uma função distância d. Isso define um grafo $(\mathcal{Z}_1, \mathcal{A}_k, \rho)$ ponderados nos nós $s \in \mathcal{Z}_1$ por uma função densidade de probabilidade (pdf) $\rho(s)$, que é calculada com base na distância $d(s,t)$ entre os vetores de atributos $\vec{v}(s)$ e $\vec{v}(t)$, para todas as amostras adjacente $t \in \mathcal{A}_k(s)$ de s.

O valor pdf representado por $\rho(s)$ de cada nó $s \in \mathcal{Z}_1$ é definido por:

$$\rho(s) \quad = \quad \frac{1}{\sqrt{2\pi\sigma^2 k}} \sum_{\forall t \in \mathcal{A}_k(s)} exp\left(\frac{-d^2(s,t)}{2\sigma^2}\right), \tag{2.4}$$

onde $\sigma = \frac{d_f}{3}$ e $d_f = \max_{\forall (s,t) \in \mathcal{A}_k}\{d(s,t)\}$.

Quanto mais perto estão os nós adjacentes, maior é o valor de pdf de um nó e quanto mais alto é o valor de k teremos menos clusters. O rótulo de cada cluster é propagado em conjunto com os caminhos ótimos a partir dos nós raizes. Esta abordagem tem sido aplicada com sucesso para segmentação de imagens do cérebro [18], aprendizagem ativa [25], entre outros.

A Figura 2.5, ilustra o comportamento prático no uso de Floresta de Caminhos Ótimos no agrupamento de amostras. Dado o conjunto original Figura 2.5(a), representando um conjunto de dados com duas classes. A Figura 2.5(b) apresenta os valores obtidos pela função de densidade de probabilidade (usando $k_{max} = 10\%$ do número de amostras e distância Euclidiana), coloridas com tons do arco íris, sendo valores com baixa densidade com coloração em azul e alta densidade coloração em vermelho. É possível notar a existência de quatro regiões com os maiores máximos. A Figura 2.5(c), mostra o resultado do agrupamento (rótulos dos grupos) sobre os quatro máximos encontrados, e a Figura 2.5(d), mostra o resultado da propagação quando associamos o rótulo da classe, referente a Figura 2.5(a), para cada uma das 4 raízes (máximos), deixamos elas propagar os rótulos para as demais amostras.

O k_{max} funciona como um fator de escala que depende da aplicação. Se k_{max} for muito alto, o número de grupos é 1, significando que estamos observando as amostras de

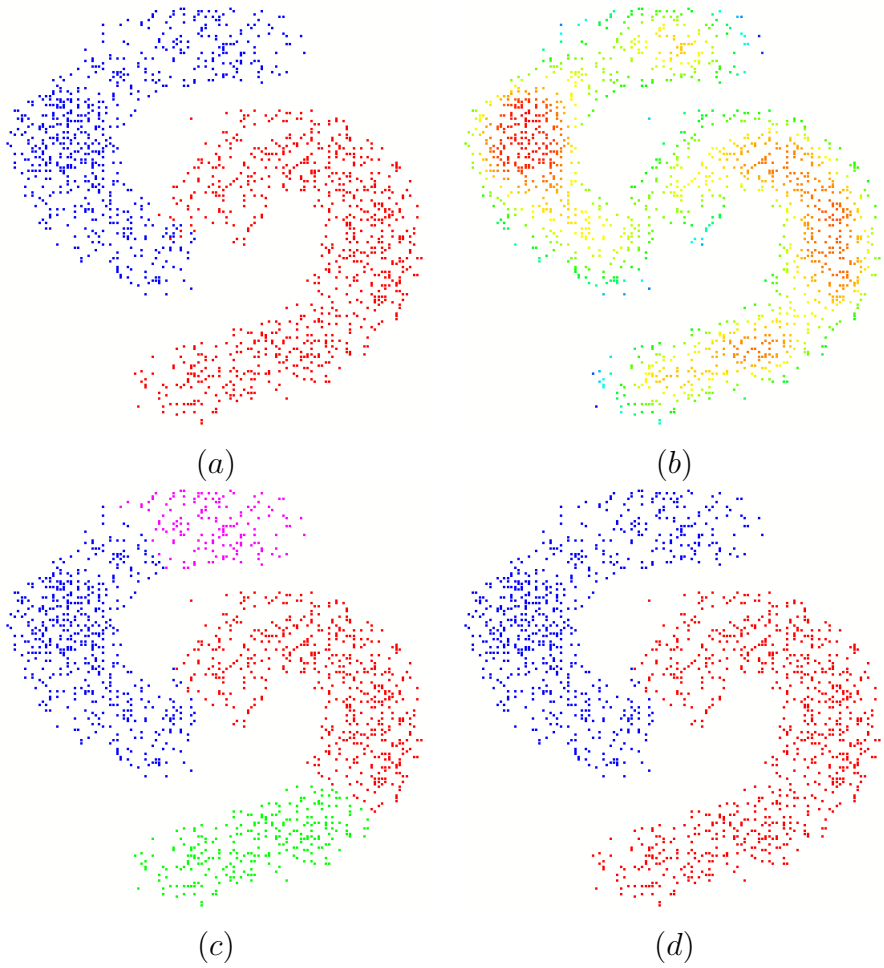


Figura 2.5: Comportamento prático de classificação não supervisionada por OPF. (a) conjunto de amostras original, (b) resultado da função de densidade de probabilidade, (c) resultado de agrupamento sobre os máximos encontrados, (d) resultado de agrupamento sobre os máximos encontrados, usando o rótulo da raiz da base original.

uma distância infinita. Se o k_{max} for muito baixo, teremos muitos grupos, significando que estamos olhando as amostras mais de perto. A Figura 2.6 mostra o resultado de agrupamento para valores ótimos de k quando modificamos o k_{max}. Dado o conjunto original Figura 2.6(a), contendo cinco classes. A Figura 2.6(b), mostra o resultado de agrupamento com quatro grupos obtidos (usando $k_{max} = 15\%$ do número de amostras e distância Euclidiana), e Figura 2.6(c) com 5 grupos (usando $k_{max} = 10\%$ do número de amostras e distância Euclidiana). A Figura 2.6(d), mostra o resultado da propagação quando associamos o rótulo da classe, referente a Figura 2.6(a), usando o agrupamento com 5 grupos (Figura 2.6(c)). Nesses exemplos fica evidente que os grupos mais próximos

se unem quando aumentamos o k_{max}.

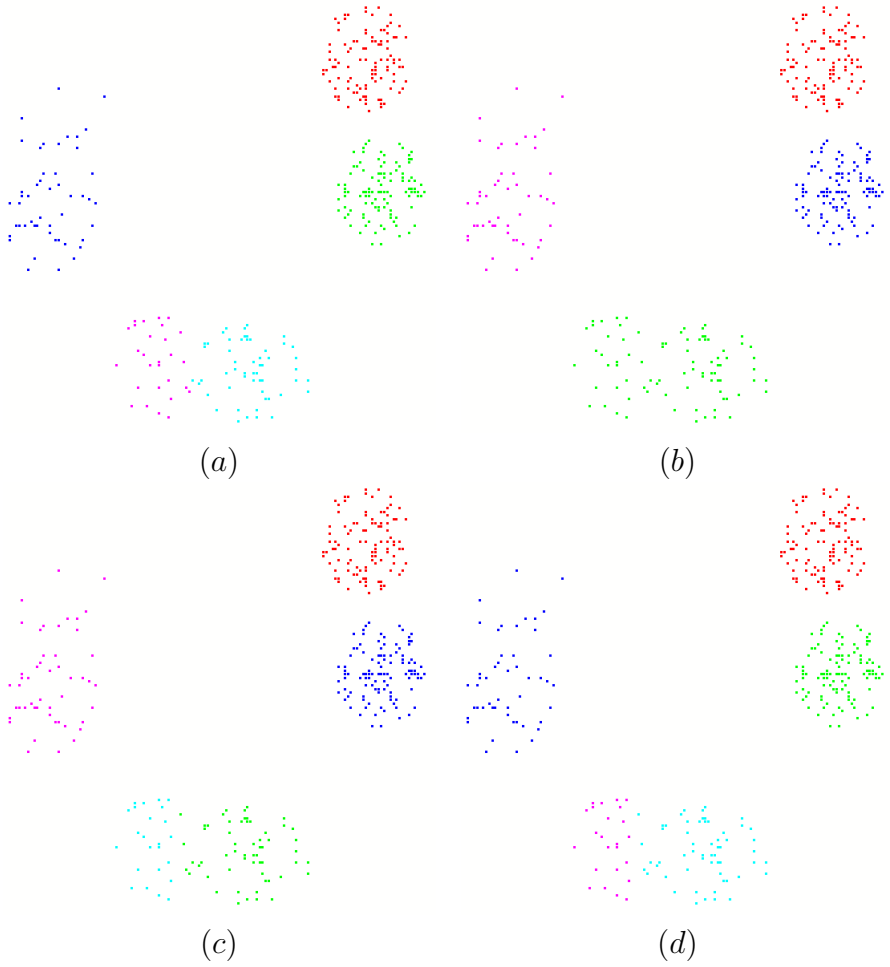


Figura 2.6: Comportamento prático de classificação não supervisionada por OPF. (a) conjunto de amostras original, (b) resultado de agrupamento usando $k_{max} = 15\%$ do número de amostras, (c) resultado de agrupamento usando $k_{max} = 10\%$ do número de amostras, (d) resultado de agrupamento sobre os máximos encontrado ($k_{max} = 10\%$), usando o rótulo da raiz da base original.

2.8 Classificação supervisionada por OPF usando grafo k-NN

Seja $\mathcal{A}_k$ uma relação de adjacência k-NN que inicialmente conecta cada amostra $s \in \mathcal{Z}_1^l$ com seus k vizinhos mais próximos de acordo com uma função distância d. Isso define um

grafo $(\mathcal{Z}_1^l, \mathcal{A}_k, \rho)$ ponderados nos nós $s \in \mathcal{Z}_1^l$ por uma função densidade de probabilidade (pdf) $\rho(s)$ (Equação 2.4), que é calculada com base na distância $d(s,t)$ entre os vetores de atributos $\vec{v}(s)$ e $\vec{v}(t)$, para todas as amostras adjacente $t \in \mathcal{A}_k(s)$ de s. Basicamente, a etapa de treinamento da abordagem OPF supervisionado usando grafo k-NN [65] consiste de duas etapas para cada valor avaliado de k: (i) a estimativa de $\rho(s)$, $\forall s \in \mathcal{Z}_1^l$, e (ii) o algoritmo OPF determinará um processo de competição entre os protótipos (ou seja, buscando o máximo em $\rho(s)$), a fim de conquistar os nós restantes, oferecendo-lhes os melhores caminhos. A função densidade de probabilidade é calculada atribuindo os valores mais elevados para amostras cujos k-vizinhos estão mais próximos. Em [65], $\rho(s)$ é estimado através da busca pelo valor máximo de $1 \leq k \leq k_{\max}$ que mantém os erros de propagação de $\mathcal{Z}_1^l$ abaixo de um determinado limiar reduzido.

Ao final, o algoritmo elege um único protótipo por máxima da pdf (identificação em tempo real), e então propaga seu rótulo para as amostras restantes sobre sua respectiva região, promovendo uma competição entre eles. O caminho de valor máximo para cada amostra s, a partir de um conjunto de elementos protótipos (raízes), particiona o conjunto de treinamento $\mathcal{Z}_1^l$ em uma floresta de caminhos ótimos. As raízes da floresta formam um subconjunto dos máximos da fdp onde, cada raiz, define uma árvore de caminhos ótimos (zona de influência do respectivo máximo) composta pelas suas amostras mais fortemente conexas, as quais recebem o mesmo rótulo de sua raiz. A idéia é que os melhores caminhos que vêm do conjunto de nós protótipos $\mathcal{S}$ sejam conquistados no mesmo cluster em uma ordem não-crescente de valor de caminho até caminhos de clusters distintos.

Dado o conjunto supervisionado apresentado na Figura 2.4(a), aplicando OPF_{kNN} com k=5, a Figura 2.4(b) apresenta os protótipos selecionados, seguidos das amostras a serem classificadas (Figura 2.4(c)) e o resultado de classificação na Figura 2.4(d). A mudança no formato da seleção dos protótipos é dada pela estrutura e abordagem utilizada. É possível verificar que os protótipos são os elementos que representam as regiões de maior densidade de amostras. Da mesma forma, é possível notar a diferença em comparação com a classificação utilizando OPF (grafo completo), mas ainda com dificuldades na separação entre as classes, especialmente entre classe 2 e classe 3.

2.9 Aprendizado ativo

Como apresentado anteriormente, em diversas áreas das ciências e engenharias, a quantidade de informações vem aumentando consideravelmente e a obtenção de classificadores apropriados requer que os dados empregados para treinamento sejam representativos da população. Contudo, a rotulação das amostras de treinamento pode ser custosa devido à necessidade de consultar um especialista da área ou realizar ensaios laboratoriais. Em tais casos, justifica-se a utilização de técnicas para seleção das amostras a serem rotuladas

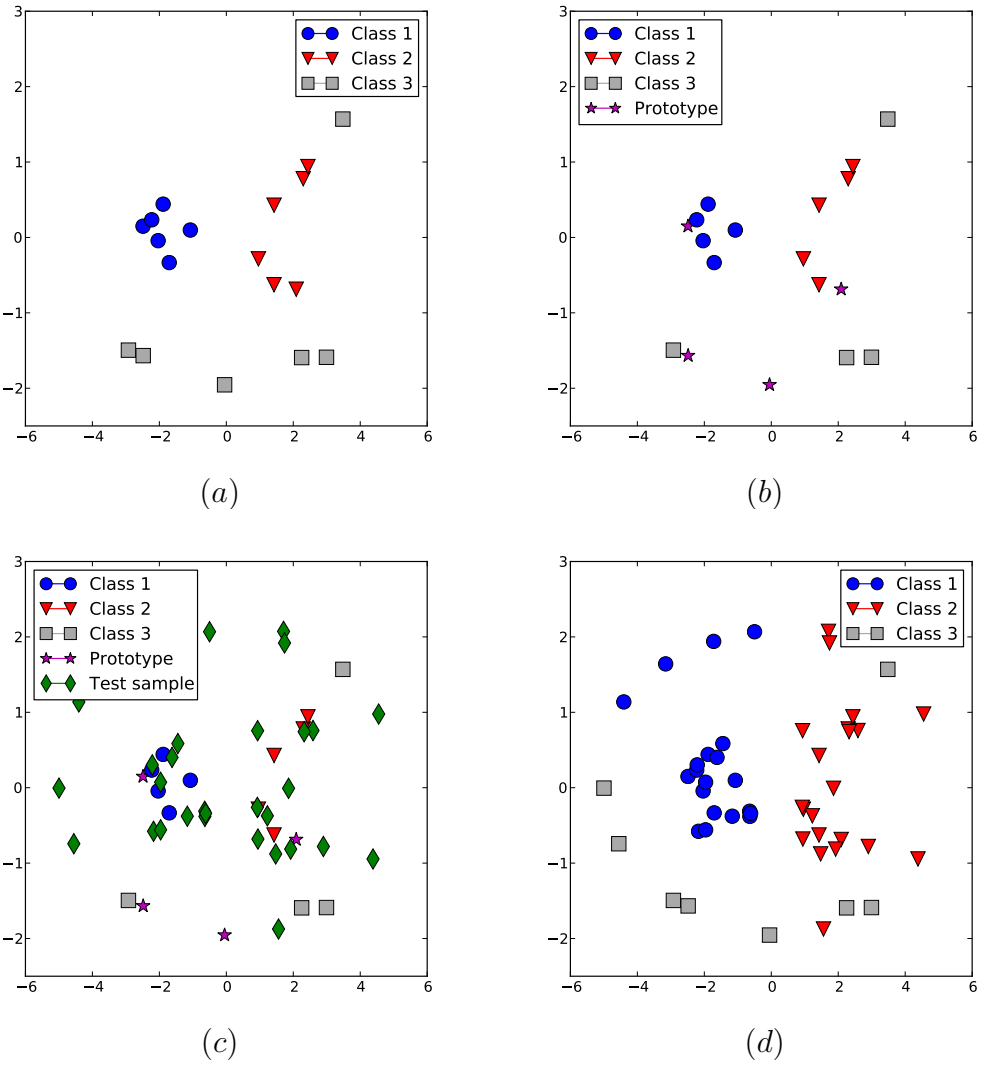


Figura 2.7: Comportamento prático de OPF_{kNN} (grafo kNN). (a) conjunto de dados supervisionados, (b) protótipos selecionados, (c) amostras a serem classificadas em conjunto com os protótipos selecionados e (d) resultado da classificação pela abordagem OPF_{kNN}.

para minimizar o envolvimento do ser humano e ao mesmo tempo garantir o sucesso do aprendizado. Neste contexto, uma possível abordagem consiste no uso de Aprendizado Ativo.

No Aprendizado Ativo o algoritmo de classificação passa de uma condição passiva, no qual só recebe amostras para treinamento, para uma condição ativa, de controle sobre o aprendizado. Nesse caso, o algoritmo passa a escolher as amostras cuja classe verdadeira deve ser determinada. Com isso, espera-se reduzir o trabalho associado à classificação

prévia das amostras por um especialista ou método laboratorial de referência. Aplicações recentes que envolvem grandes quantidades de dados não supervisionados certamente beneficiam de uma combinação de estratégias de aprendizado ativo e aprendizado semi-supervisionado, pois isso permitiria a identificação e rotulação de um pequeno número de amostras mais representativas. Esta abordagem permitiria também que o algoritmo de aprendizagem pudesse aumentar o seu próprio conjunto supervisionado, selecionando um conjunto de amostras não supervisionadas para serem anotadas por um especialista do domínio do problema.

Quando somos obrigados a rotular as amostras no aprendizado semisupervisionado, é mais eficaz permitir que o algoritmo de aprendizagem nos diga quais as amostras devemos rotular ao invés de uma escolha aleatória. Algumas tentativas foram feitas para investigar a utilização conjunta de aprendizado ativo e semisupervisionado sobre diversas aplicações e diferentes domínios de problemas. Entre eles: compressão de imagem [41], reconhecimento de voz [82], reconhecimento de caracteres [37], reconhecimento de dígitos manuscritos, tarefas de classificação de texto [93], classificação de páginas web [63], entre outros. No entanto, a maioria dos trabalhos foram realizados para classificação binária.

2.10 Classificação multirótulos

Atribuição multirótulo é necessária em vários cenários, como exemplo categorização de textos e sistemas de diagnóstico médico assistido por computador. No primeiro caso, um artigo de revista pode ser classificado como pertencendo a religião e artes, enquanto que, no segundo caso, o paciente pode ser afetado por várias doenças simultaneamente. Portanto, uma amostra pode ser atribuída a um ou vários rótulos em cada cenário. Os métodos existentes buscam reduzir o problema em outros vários problemas de classificação de único rótulo ou criar uma metaclasse para cada combinação de rótulo possível, sendo dividido nas seguintes estratégias respectivamente: (i) *métodos de transformação* e (ii) *métodos de adaptação*. Em ambos os casos, as amostras são remarcadas de acordo com a estratégia escolhida, a fim de utilizar um ou vários modelos de classificação conhecidos. Métodos de adaptação modificam as abordagens de aprendizado supervisionado, como o k-vizinhos mais próximos [86], árvores de decisão [21], e redes neurais artificiais [85] para identificar as metaclasses da amostras com base em informações de ranking e estatística. Da mesma forma, existem soluções que exploram informações de correlação entre os rótulos [52, 88]. Em resumo, cada categoria traz suas vantagens, mas também desvantagens que devem ser conhecidas com antecedência antes de seu uso.

Uma vez que a estratégia de métodos de transformação escolhida cria conjuntos de dados com amostras de único rótulo a partir do conjunto de dados multirótulos original, pode-se construir um classificador para cada novo conjunto de dados com base no aprendi-

zado supervisionado e usá-lo para atribuir rótulos individuais para novas amostras. Essas amostras podem então ser atribuídas a várias classes pelo o processo de transformação inversa de dados. No entanto, a classificação multirótulos, muitas vezes conta com um pequeno número de amostras supervisionadas, favorecendo abordagens que fazem uso de amostras supervisionadas e não supervisionadas no conjunto de treinamento — os métodos de aprendizagem semisupervisionada. Para estratégias de métodos de adaptação, deve-se usar uma abordagem de auto formação para treinar as amostras supervisionadas em amostras não supervisionadas.

As principais abordagens para o aprendizado semisupervisionado exploram a distribuição espacial das amostras de treinamento supervisionadas e não supervisionadas no espaço de atributos para a propagação do rótulo [6, 5, 17, 47, 38, 50, 10]. Este processo pode-se repetir algumas vezes, de tal forma que um classificador final é criado a partir das amostras de maior confiança do conjunto completamente rotulado.

2.10.1 Métodos de classificação multirótulos

Nesta seção, apresentamos uma breve revisão sobre as duas categorias de aprendizagem multirótulos: (i) métodos de adaptação e (ii) métodos de transformação, assim como também discutimos a forma de propagação dos rótulos para amostras não supervisionadas considerando cada estratégia.

Métodos de adaptação

O foco dos métodos de adaptação está em modificar algoritmos existentes para que eles possam lidar com amostras multirótulo, sem necessidade de qualquer pré-processamento. Como o número de trabalhos que lidam com esse problema tem aumentado consideravelmente, vamos nos concentrar apenas na explicação sobre os métodos usados na seção experimental. O primeiro método é a versão multirótulo do algoritmo clássico k-NN [86], diferenciado pelo uso de probabilidades a priori e a posteriori. Inicialmente o método identifica, para cada instância na base de treinamento, os seus k vizinhos mais próximos. Então, baseado na informação estatística a priori obtida do conjunto de rótulos destes vizinhos, é utilizado o princípio *Maximum a Posteriori* (MAP) para determinar o conjunto de rótulos da instância de teste. Outro método tradicional na área é a aprendizagem multirótulo por *back-Propagation* [85] (BPMLL). O algoritmo BPMLL é uma adaptação do algoritmo de redes neurais *back-propagation* para aprendizado multirótulo, sendo a principal modificação desse algoritmo é a introdução de uma nova função de erro que considera múltiplos rótulos.

Uma técnica comum e simples usada para o aprendizado semisupervisionado, baseado na estratégia de métodos de adaptação é conhecido como “auto formação”, em que o

classificador utiliza as suas próprias previsões para ensinar a si mesmo. Nesta abordagem, um classificador subjacente é primeiro treinado com um pequeno número de amostras supervisionadas chamado de conjunto "inicial"de treinamento. Em seguida, o classificador subjacente é utilizado para classificar os dados não supervisionados, que é usado para aumentar o conjunto de treino a ser ainda usado em um passo de re-treinamento. Este procedimento se repete até que todas as instâncias não supervisionadas forem transferidas para o conjunto de treinamento supervisionado. Neste trabalho, usamos a abordagem de auto formação para a avaliação de ambos os métodos MLkNN e MPMLL.

Métodos de transformação

A fim de compreender as diferenças entre as estratégias de métodos de transformação, seja $\mathcal{Z}$ o espaço de atributos d-dimensional e $\mathcal{Y} = \{y_1, y_2, \ldots, y_{\mathcal{L}}\}$ um espaço de rótulos com $\mathcal{L}$ possíveis rótulos de classe. A cada amostra $s_i \in \mathcal{Z}$ pode ser atribuído um conjunto de rótulos $\mathcal{Y}_i \subseteq \mathcal{Y}$ através da especificação aos valores binários $y_k^i \in \{0, 1\}$, $1 \leq k \leq \mathcal{L}$, onde $y_k^i = 1$ indica a amostra s_i pertencendo à classe k e $y_k^i = 0$ representando a situação oposta[3]. Dado um conjunto de treinamento $\mathcal{Z}$ com l amostras supervisionadas e u amostras não supervisionadas, definimos por $\mathcal{Z} = \mathcal{Z}^l \cup \mathcal{Z}^u$, onde $\mathcal{Z}^l = \{(s_1, \mathcal{Y}_1), \ldots, (s_l, \mathcal{Y}_l)\}$ e $\mathcal{Z}^u = \{s_{l+1}, \ldots, s_{l+u}\}$ representando os conjuntos de amostras supervisionadas e não supervisionadas, respectivamente. Neste caso, o problema de aprendizagem visa encontrar a partir de $\mathcal{Z}$ uma família de funções reais $f_i : \mathcal{Z} \times \mathcal{Y} \to \Re$, $i = 1, 2, ..., \mathcal{L}$ tal que $f_i(s_i, \mathcal{Y}_i)$ é a confiança que $\mathcal{Y}_i$ seja o verdadeiro conjunto de rótulos de s_i. A seguir estaremos apresentando os quatro principais métodos de transformação de dados multirótulo para único rótulo que estaremos usando em nossos experimentos, que são: Binary Relevance, Label Powerset, Classifier Chain e Hierarchy of Multi-Label Classifiers [61].

Binary Relevance (BR) decompõe o conjunto original $\mathcal{Z}^l$ multirótulo em $\mathcal{L}$ bases de dados de único rótulo $\mathcal{Z}^l[k] = \{(\mathbf{s}_1, y_1^k), \ldots, (\mathbf{s}_l, y_l^k)\}$, e também classificadores binários independentes de treinamento f_k em $\mathcal{Z}^l[k]$ a fim de prever $y_i^k \in \{0, 1\}$, $k = 1, 2, \ldots, \mathcal{L}$, onde $y_i[k]$ indica o rótulo da amostra i com relação à classe k. Por exemplo, em um problema multirótulo com $\mathcal{L} = 3$ classes, $\mathcal{Z}^l[1]$ representa o conjunto no qual amostras de dados que pertencem à classe 1 são rotuladas como positiva, e as restantes (isto é, aqueles a partir de classes 2 e 3) estão rotuladas como as amostras negativas. Label Powerset (LP) considera cada possível combinação de rótulo $\mathcal{L}$ como um único rótulo, transformando o problema no modelo múltiplas classes. LP transforma $\mathcal{Z}^l$ em uma nova base de dados $\hat{\mathcal{Z}}^l = \{(\mathbf{s}_1, c_1), ..., (\mathbf{s}_l, c_l)\}$ onde $c_i = g(Y_i)$ e $g : \{0, 1\}^{\mathcal{L}} \to \{1, ..., 2^{\mathcal{L}}\}$, $i = 1, 2, \ldots, l$, qual representa uma função que mapeia cada combinação de único rótulo para uma nova representação de rótulos. Um classificador multirótulo é então treinado

[3]Note que $\mathcal{Y}_i = \{y_i^1, y_i^2, \ldots, y_i^{\mathcal{L}}\}$, $i = 1, 2, \ldots, l$.

com o novo conjunto de dados para prever o rótulo de cada nova amostra.

Classifier Chain (CC) e Hierarchy of Multi-Label Classifiers (HOMER) são extensões otimizadas de BR e LP, respectivamente. Classifier Chain realiza o mapeamento em $\mathcal{L}$ conjuntos de dados binários como BR, mas também amplia o espaço de atributos para cada $\mathcal{Z}^l[k]$ adicionando o rótulo 0/1 a partir de $\mathcal{Z}^l[k-1]$. Isto é, $\mathcal{Z}^l[k] = \{(\mathbf{s}'_1, y_1^k), \ldots, (\mathbf{s}'_l, y_l^k)\}$ e $\mathbf{s}'_i = (\mathbf{s}_i \cdot y_i^{k-1})$, onde $\cdot$ representa o operador[4] de concatenação, $i = 1, 2, \ldots, l$. HOMER transforma um classificador multirótulo em uma hierarquia de simples classificadores multirótulo, de tal modo que o classificador de um nó filho trata com um conjunto menor de rótulos do que o classificador do nó pai. O nó raiz trata com $\mathcal{L}$ rótulos, que são agrupados em $k \leq \mathcal{L}$ nós filhos disjuntos. O processo de agrupamento repete para cada nó de uma forma em profundidade até que os nós se tornam folhas com $\mathcal{L}$ classificadores de único rótulo.

Portanto, pode-se aplicar uma transformação de dados $\mathcal{T}(\mathcal{Z}^l)$, propagando os rótulos de $\mathcal{T}(\mathcal{Z}^l)$ a $\mathcal{Z}^u$, e reverter o processo de $\mathcal{T}^{-1}(\mathcal{Z}^u)$ para atribuição multirótulo. Ao projetar um classificador semisupervisionado a partir de $\mathcal{T}(\mathcal{Z}^l) \cup \mathcal{Z}^u$ e usá-lo para atribuir rótulos para novas amostras, o inverso $\mathcal{T}^{-1}$ descobrirá suas múltiplas classes. Neste trabalho, usamos os métodos de transformação LP, BR, CC e HOMER para a avaliação de todas as propostas semisupervisionadas.

[4]Observe que quando $k = 1$, o vetor de atributos não está estendido.

Capítulo 3

Aprendizado semisupervisionado por OPF

3.1 Considerações iniciais

Organizamos a apresentação das propostas desta tese de doutorado da seguinte forma. Inicialmente, duas propostas semisupervisionadas serão apresentadas. Por ordem cronológica, a primeira técnica apresentada para a comunidade foi OPFSEMI, a qual propomos um método semisupervisionado de aprendizagem baseado na conectividade ótima entre amostras supervisionadas e não supervisionadas [5] e na sequência propomos uma melhoria significativa na abordagem anterior [6]. O método chamado de OPFSEMI$_{mst}$, também explora a melhor conectividade entre amostras supervisionadas e não supervisionadas para propagação dos rótulo, mas resultando em um classificador final a partir de uma única iteração de propagação, sendo ainda mais eficiente e preciso.

Em seguida, propomos a técnica OPFSEMI$_{(mst+knn)}$, uma nova proposta semisupervisionada a partir da propagação inicial OPFSEMI$_{mst}$ e repropagação com OPF$_{kNN}$, melhorando significativamente o desempenho em problemas de atribuição multirótulo. Ambas as técnicas OPFSEMI$_{mst}$ e OPFSEMI$_{(mst+knn)}$, utilizam a topologia da MST para propagar rótulos das amostras supervisionadas para as não supervisionadas, porém o primeiro utiliza a floresta resultante como o classificador de padrões (que associa rótulo para novas amostras como na Seção 2.6) e o segundo usa os rótulos propagados, mas utiliza o método apresentado na Seção 2.7 para gerar a floresta final. E por fim, a partir das técnicas semisupervisionadas (OPFSEMI e OPFSEMI$_{mst}$), propomos uma integração de aprendizado ativo com semisupervisionado. Essa proposta difere do aprendizado típico ativo e semisupervisionado sendo capaz de selecionar mais rapidamente as amostras de todas as classes e manter uma interação mínima do usuário.

Estaremos adotando em nossas explicações a metodologia de Floresta de Caminhos

Ótimos, a qual tem sido usada para o projeto de classificadores de padrões não supervisionados, supervisionados e semisupervisionados. Como apresentado anteriormente, esta metodologia consiste basicamente na escolha de três componentes principais para a criação de um novo classificador: (a) amostras de treinamento, (b) relação de adjacência, e (c) uma função de conectividade adequada. As amostras de treinamento podem ser supervisionadas, não supervisionadas, ou ambas, caracterizando os processos de aprendizagem supervisionada, não supervisionada, e semisupervisionada, respectivamente. A relação de adjacência visa conectar amostras de treinamento no espaço de atributos como nós de um grafo, a fim de melhor explorar a sua conectividade. A função de conectividade define um valor para qualquer sequência de amostras distintas e adjacentes (caminho simples) no grafo, bem como os caminhos triviais formados por um nó único. Para isso, inicialmente cada nó define um caminho trivial e a minimização do mapa de conectividade calcula caminhos ótimos com terminal em cada nó, de tal modo que (a) as raízes dos caminhos são primeiramente derivadas dos valores mínimos do mapa de conectividade e (b) destas raízes conquistam outros nós. Assim serão oferecidos caminhos ótimos, particionando o grafo em uma floresta de caminhos ótimos (classificador), que atribuirá rótulos para novas amostras avaliando caminhos estendidos a eles.

Para a elaboração de cada proposta, assumimos que em um grande conjunto de dados, a maioria das amostras de uma mesma classe estão mais fortemente ligadas do que amostras de classes distintas. No entanto, se amostras não supervisionadas pertencessem à estrutura inicial de treinamento, ou seja, estivessem conectadas com as amostras supervisionadas, poderia-se diminuir as chances de erros de rotulação gerando um melhor classificador semisupervisionado.

OPF proposto por Papa et al. [66], utiliza apenas as amostras supervisionadas sobre uma estrutura de grafo completo, seleciona as amostras que se situam na proximidade entre as classes como protótipos (amostras mais informativas), ou seja, calcula uma MST de um grafo completo com nós supervisionados por λ para escolher protótipos como amostras de classes distintas que compartilham arco na MST, e os caminhos ótimos são oferecidos pelos protótipos aos demais nós. Com isso, além da estrutura obter um elevado custo de processamento (para grande conjunto de dados), como o uso do grafo completo em todas as fases de treinamento, a rotulação das outras amostras depende diretamente da qualidade das amostras de fronteira entre as classes. No entanto, para uma visão semisupervisionada, este tipo de metodologia torna-se inviável para certas aplicações, devido à necessidade de um melhor uso das amostras supervisionadas (isto é, algumas amostras supervisionadas e um grande volume de amostras não supervisionadas), e a simplicidade na estrutura de treinamento para obter um resultado eficaz e eficiente. A seguir apresentamos os processos de treinamento e classificação das duas propostas semisupervisionadas OPFSEMI$_{mst}$ e OPFSEMI. Acreditamos que esse formato de apresentação poderá facili-

tar o entendimento de cada contribuição, principalmente nas otimizações realizadas por OPFSEMI$_{mst}$.

3.2 Aprendizado semisupervisionado usando OPFSEMI$_{mst}$ e OPFSEMI

Em OPFSEMI$_{mst}$, treinamento semisupervisionado por OPF, consideramos uma relação de adjacência $\mathcal{A} = \mathcal{Z}_1 \times \mathcal{Z}_1$, a qual define um grafo completo e ponderado $(\mathcal{Z}_1, \mathcal{A}, d)$, e criamos uma árvore geradora mínima sobre o conjunto de treinamento usando OPF com função de conectividade f_{mst} (Equação 3.1), a qual irá produzir $(\mathcal{Z}_1, \mathcal{B}, d)$ — uma árvore geradora mínima, ou seja, um grafo conexo acíclico com todas as amostras de treinamento, onde $\sum_{\forall (s,t) \in \mathcal{B} \subseteq \mathcal{A}} \{d(s,t)\}$ é mínimo.

$$\begin{aligned} f_{mst}(\langle s \rangle) &= \begin{cases} 0 & \text{se } s \in \mathcal{Z}_1^l, \\ +\infty & \text{caso contrário,} \end{cases} \\ f_{mst}(\pi_s \cdot \langle s,t \rangle) &= d(s,t), \end{aligned} \tag{3.1}$$

A árvore geradora mínima gerada $(\mathcal{Z}_1, \mathcal{B}, d)$, irá ser usada como grafo de entrada na execução de OPF, mas de tal forma que o conjunto de protótipos $\mathcal{S} \leftarrow \mathcal{Z}_1^l$ seja composto por todas as amostras supervisionadas. Notamos que no contexto semisupervisionado, amostras supervisionadas podem ser mais suscetíveis a erros. Assim se selecionadas apenas amostras na fronteira entre as classes como protótipos (i.e. OPF e OPFSEMI) (amostras mais difíceis de serem classificadas), e essas amostras rotuladas de maneira incorreta, o erro poderá ser propagado para as demais amostras prejudicando o classificador final.

Os arcos em $\mathcal{B}$ já conectam as amostras supervisionadas e não supervisionadas mais próximas, mas um nó $t \in \mathcal{Z}_1^u$ ainda pode ser alcançado por caminhos de nós de $\mathcal{Z}_1^l$ com rótulos distintos. Portanto, amostras supervisionadas irão competir uma com as outras e o rótulo $L_1(t) \leftarrow \lambda(s)$ de t, será atribuído pelo nó mais fortemente conexo de $s \in \mathcal{S}^*$. Um erro ocorre quando a propagação do rótulo $L_1(t) \neq \lambda(t)$.

O Algoritmo 1 ilustra esse processo, através da transformação do grafo completo da Figura 3.1a para uma árvore geradora mínima Figura 3.1b. Note que para gerarmos a saída para a execução de OPF, estaremos aplicando MST sobre todo o conjunto $\mathcal{Z}_1$ e não apenas em $\mathcal{Z}_1^l$ e que $\mathcal{B} \subset \mathcal{A}$ é muito menor do que $\mathcal{A}$. Em seguida, consideramos o grafo MST como entrada usando a função de conectividade $f_{\max}$ (Equação 2.2) para cálculo do caminho ótimo (Figura 3.1d) – Algoritmo 2. Esta função de conectividade define custo 0 somente para caminhos triviais que começam em um conjunto especial $\mathcal{S}$ com amostras sementes. O custo para estender um caminho de π_s por uma aresta (s,t), formando π_t, é

dado pela aresta com peso máximo ao longo de π_t. As sementes devem representar todas as classes e ser informativa o suficiente para oferecer caminhos de custos mínimos para as amostras restantes de sua classe.

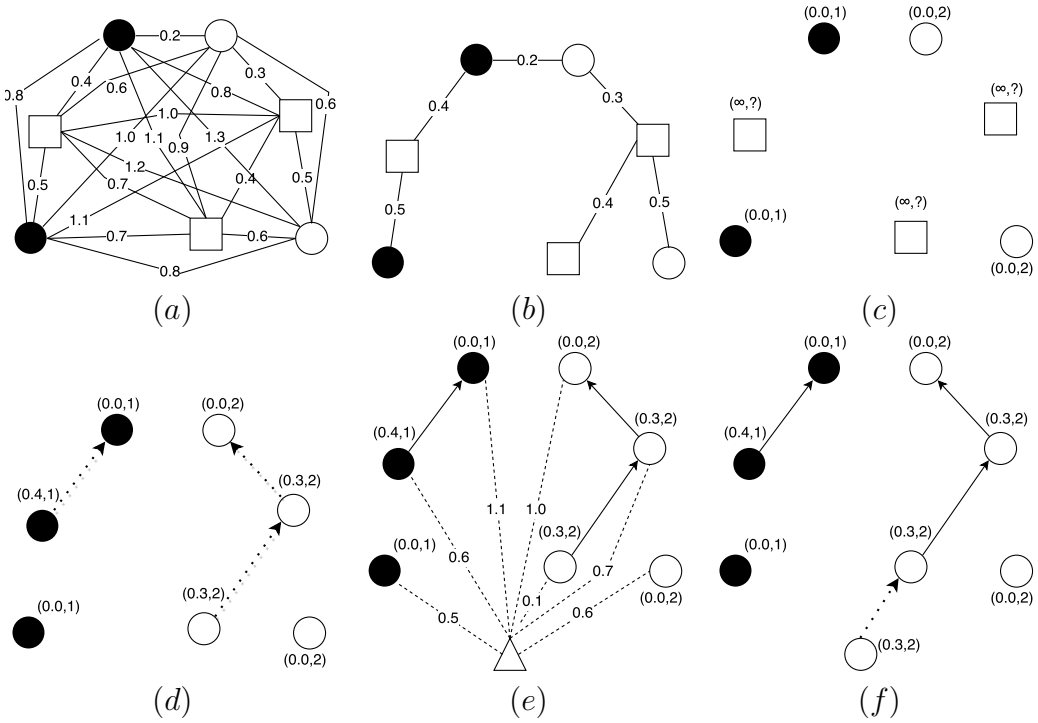


Figura 3.1: Comportamento prático de OPFSEMI$_{mst}$. (a) Grafo completo e ponderado sobre o conjunto de treinamento $\mathcal{Z}_1$ (• amostras supervisionadas da classe 1, ∘ amostras supervisionadas da classe 2 e □ amostras não supervisionadas). (b) Uma árvore geradora mínima de (a) e (c) um mapa de valor de caminho trivial. As amostras supervisionadas são forçadas a ser o mínimo do mapa (isto é, custo 0), e amostras não supervisionadas são atribuídas a um custo infinito. Os identificadores (x, y) acima dos nós são, respectivamente, o custo e o rótulo dos mesmos. (d) Uma Floresta de Caminhos Ótimos e propagação do rótulo sobre (b) a partir do mapa de valor de caminho em (c). A seta indica o nó predecessor no caminho ótimo. (e) Uma amostra de teste △ e (f) e sua respectiva classificação a partir do caminho ótimo do protótipo mais fortemente conexo.

Algoritmo 1 – Algoritmo OPF para f_{mst}

Entrada:	Um grafo completo e ponderado $(\mathcal{Z}_1, \mathcal{A}, d)$.
Saída:	Uma árvore geradora mínima $(\mathcal{Z}_1, \mathcal{B}, d)$.
Auxiliares:	Fila de prioridades Q, variável de custo cst, mapa de valores de custo de caminhos C_1, mapa de predecessores P_1, e $color(s)$ sendo: $(white)$ quando s nunca inserido em Q; $(gray)$ quando $s \in Q$; e $(black)$ quando s removido de Q.

1. $\mathcal{B} \leftarrow \emptyset$.
2. **Para Cada** $t \in \mathcal{Z}_1$ **Faça**
3. $\quad C_1(t) \leftarrow +\infty$, *e* $color(t) \leftarrow white$.
4. *Selecione qualquer nó* $s \in \mathcal{Z}_1$, $C_1(s) \leftarrow 0$, $P_1(s) \leftarrow nil$, $color(s) \leftarrow gray$, *e insira* s *in* Q.
5. **Enquanto** Q *é não vazia,* **Faça**
6. $\quad$ *Remova de* Q *uma amostra* s *tal que* $C_1(s)$ *é mínima, e atualize* $color(s) \leftarrow black$.
7. $\quad$ **Se** $P_1(s) \neq nil$ **Então** $\mathcal{B} \leftarrow \mathcal{B} \cup \{(s, P_1(s)), (P_1(s), s)\}$.
8. $\quad$ **Para Cada** $t \in \mathcal{A}(s)$ **Faça**
9. $\quad\quad$ **Se** $color(t) \neq black$, **Então**
10. $\quad\quad\quad cst \leftarrow d(s,t)$.
11. $\quad\quad\quad$ **Se** $cst < C_1(t)$, **Então**
12. $\quad\quad\quad\quad P_1(t) \leftarrow s$ *e* $C_1(t) \leftarrow cst$.
13. $\quad\quad\quad\quad$ **Se** $color(t) = gray$, **Então** *atualize a posição de* t *em* Q
14. $\quad\quad\quad\quad$ **Senão** *insira* t *em* Q *e atualize* $color(t) \leftarrow gray$.
15. **Retorne** *uma árvore geradora mínima* $(\mathcal{Z}_1, \mathcal{B}, d)$.

O Algoritmo 2 recebe como entrada $(\mathcal{Z}_1, \mathcal{B}, d)$, conjunto de protótipos $\mathcal{S}$, e função de conectividade $f_{\max}$. A idéia é minimizar o mapa de valores de caminhos C_1 (mapa de conectividade), considerando todos os caminhos possíveis com terminal em t e atribuir a t o custo de $C_1(t)$ de um caminho ótimo π_t. Esse caminho pode ser obtido retrocedendo a partir do mapa de predecessores P_1 — Floresta de Caminhos Ótimos: um mapa acíclico definido para todos os nós em $\mathcal{Z}_1$ sendo $P_1(t) = nil$, quando $t \in \mathcal{S}$ sendo uma raiz do mapa, ou $P_1(t) = s \in \mathcal{Z}_1$, quando s é o predecessor de t no caminho ótimo π_t. Cada amostra $s \in \mathcal{S}$ será raiz de uma árvore de caminhos ótimos e cada classe será representada por suas amostras raízes. Os verdadeiros rótulos $\lambda(s)$ das raízes $s \in \mathcal{S}$ podem ser propagados para criar uma mapa de rótulos L_1, onde as amostras não supervisionadas $t \in \mathcal{Z}_1^u$ serão atribuídas ao rótulo $L_1(t) \leftarrow \lambda(s)$, ou seja, com a raiz mais fortemente conexo $s \in \mathcal{S}$. Por fim, os nós t da floresta são armazenados em $\mathcal{Z}_1'$ seguindo uma ordem não-decrescente de valores de caminhos ótimos $C_1(t)$. Isso será útil para acelerar o processo de classificação. Uma observação interessante é que, uma vez que todos os caminhos em $(\mathcal{Z}_1, \mathcal{B}, d)$ são ótimos para $f_{\max}$, as execuções do algoritmo em $(\mathcal{Z}_1, \mathcal{A}, d)$ e $(\mathcal{Z}_1, \mathcal{B}, d)$ produzem saídas semelhantes, sendo o último muito mais eficiente.

Algoritmo 2 – Algoritmo OPF para $f_{\max}$

Entrada:	Grafo $(\mathcal{Z}_1, \mathcal{B}, d)$.
Saída:	Mapa da Floresta de Caminhos Ótimos e seus atributos $[P_1, C_1, L_1, \mathcal{Z}_1']$.
Auxiliares:	Fila de prioridades Q, variável de custo cst, e $color(s)$ sendo: $(white)$ quando s nunca inserido em Q; $(gray)$ quando $s \in Q$; e $(black)$ quando s removido de Q.

1. **Para Cada** $t \in \mathcal{Z}_1$, **Faça**

2. $C_1(t) \leftarrow +\infty$, *e* $color(t) \leftarrow white$.
3. **Se** $t \in \mathcal{S}$, **Então**
4. $C_1(t) \leftarrow 0$, $P_1(t) \leftarrow nil$, $L_1(t) \leftarrow \lambda(t)$, *e* $color(t) \leftarrow gray$.
5. *insira* t *em* Q.
6. **Enquanto** Q *é não vazia,* **Faça**
7. *Remova de* Q *uma amostra* s *tal que* $C_1(s)$ *é mínima, e atualize* $color(s) \leftarrow black$.
8. *Insira* s *em* $\mathcal{Z}_1'$.
9. **Para Cada** $t \in \mathcal{B}(s)$ **Faça**
10. **Se** $color(t) \neq black$, **Então**
11. $cst \leftarrow \max\{C_1(s), d(s,t)\}$.
12. **Se** $cst < C_1(t)$, **Então**
13. $P_1(t) \leftarrow s$, $L_1(t) \leftarrow L_1(s)$, *e* $C_1(t) \leftarrow cst$.
14. **Se** $color(t) = gray$, **Então** *atualize a posição de* t *in* Q
15. **Senão** *insira* t *em* Q *e atualize* $color(t) \leftarrow gray$.
16. **Retorne** *floresta de caminhos ótimos e seus atributos* $[P_1, C_1, L_1, \mathcal{Z}_1']$.

Em OPFSEMI [5] o treinamento também consiste do conjunto de dados com amostras supervisionadas e não supervisionadas, e o classificador requer duas execuções do algoritmo OPF. Na primeira execução, as raízes da floresta são as amostras mais próximas de classes distintas (apenas do conjunto supervisionado inicial) e, na segunda execução, o conjunto raiz é melhorado pela adição de novas amostras representativas (ou seja, o uso das amostras não supervisionadas anteriormente mas agora rotuladas). Basicamente, OPFSEMI considera uma relação de adjacência $\mathcal{A} = \mathcal{Z}_1 \times \mathcal{Z}_1$, a qual define um grafo completo e ponderado $(\mathcal{Z}_1, \mathcal{A}, d)$ (Figura 3.2a), e o treinamento seleciona o conjunto de protótipos $\mathcal{S}$, de tal forma que $\mathcal{S} \subset \mathcal{Z}_1^l$, ou seja, entre as amostras supervisionadas. Esse processo pode ser realizado criando uma árvore geradora mínima sobre o conjunto de treinamento, e selecionando as amostras que compartilham na estrutura de adjacência da MST, arestas de rótulos distintos (Figura 3.2b). Isso garante que todas as classes terão protótipos dentro da fronteira entre as classes com amostras supervisionadas. Esse procedimento pode ser realizado em duas fases: (i) primeiramente considerando o grafo $(\mathcal{Z}_1^l, \mathcal{A}, d)$ com função de conectividade f_{mst} (Equação 3.1), para seleção e marcação dos protótipos somente sobre os dados supervisionados, e em seguida (ii) aplicar novamente a função f_{mst} sobre $(\mathcal{Z}_1, \mathcal{A}, d)$ a qual irá produzir $(\mathcal{Z}_1, \mathcal{B}, d)$. Importante destacar que a primeira etapa usando f_{mst} foi usada apenas para alimentar o conjunto de protótipos iniciais $\mathcal{S}$, e a segunda etapa, para a criação da árvore geradora mínima considerando todas as amostras de treinamento semisupervisionado, que será usado como entrada pelo Algoritmo 2, usando a função de conectividade $f_{\max}$ (Equação 2.2) para cálculo do caminho ótimo (Figura 3.2d). Temos observado que, uma vez que os rótulos são propagados para todas as amostras em $\mathcal{Z}_1$, torna-se mais eficiente o método quando treinado novamente

usando o Algoritmo 2, sobre o novo conjunto de protótipos $\mathcal{S}$ (isto essencialmente melhora o conjunto protótipo), mas agora considerando o conjunto totalmente rotulado.

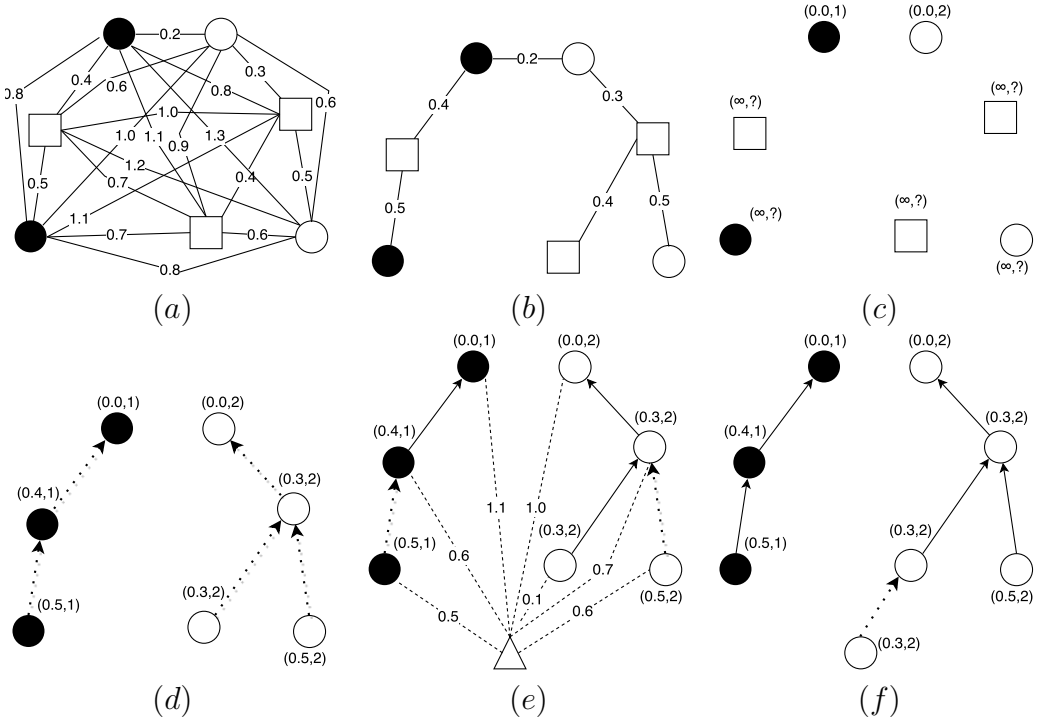


Figura 3.2: Comportamento prático de OPFSEMI. (a) Grafo completo e ponderado sobre o conjunto de treinamento $\mathcal{Z}_1$ ($\bullet$ amostras supervisionadas da classe 1, $\circ$ amostras supervisionadas da classe 2 e $\square$ amostras não supervisionadas). (b) Uma árvore geradora mínima de (a) e (c) um mapa de valor de caminho trivial. As amostras protótipos são forçadas a ser o mínimo do mapa (isto é, custo 0), e as demais amostras supervisionadas e não supervisionadas são atribuídas a um custo infinito. Os identificadores (x, y) acima dos nós são, respectivamente, o custo e o rótulo dos mesmos. (d) Uma Floresta de Caminhos Ótimos e propagação do rótulo sobre (b) a partir do mapa de valor de caminho em (c). A seta indica o nó predecessor no caminho ótimo. (e) Uma amostra de teste $\triangle$ e (f) e sua respectiva classficação a partir do caminho ótimo do protótipo mais fortemente conexo.

Comparando OPFSEMI$_{mst}$ com OPFSEMI podemos chegar nas seguintes conclusões em termos de complexidade. O cálculo da MST sobre $(\mathcal{Z}_1, \mathcal{A}, d)$ tem complexidade de tempo $O(|\mathcal{Z}_1|^2)$, desde que o grafo seja completo, enquanto a complexidade de tempo para geração da floresta de caminhos ótimos de $(\mathcal{Z}_1, \mathcal{B}, d)$ é $O(|\mathcal{Z}_1| \log |\mathcal{Z}_1|)$, uma vez que $|\mathcal{B}| \ll |\mathcal{Z}_1| \log |\mathcal{Z}_1|$. OPFSEMI seleciona os protótipos $\mathcal{S}$ entre amostras que compartilham uma mesma aresta na MST com classes distintas, provocando um custo adicional de $O(n^2)$

para n nós. Em seguida, executado o algoritmo OPF com $f_{\max}$ com custo $O(n \log n)$ na topologia da árvore geradora mínina para propagar os rótulos para as amostras não supervisionadas. Além disso, com objetivo de melhorar o conjunto $\mathcal{S}$ com amostras (agora rotuladas) anteriormente não supervisionadas, o processo como um todo é repetido para o conjunto de treinamento completamente rotulado. Assim, essencialmente OPFSEMI exigirá então duas execuções, com tempo de $O(n^2) + O(n \log n)$ cada. Exceto para a primeira execução, o que pode ser feito com $n = |\mathcal{Z}_1^l|$, e os outros exigindo $n = |\mathcal{Z}_1|$. No caso para OPFSEMI$_{mst}$, a primeira execução do algoritmo OPF leva tempo $O(n^2)$, sendo $n = |\mathcal{Z}_1|$, para computar uma árvore geradora mínima do grafo completo, cujos nós são as amostras supervisionadas e não supervisiondas. A função de conectividade $f_{\max}$ usa $\mathcal{S} = \mathcal{Z}_1^l$, evitando o cálculo na busca pelos protótipos, e uma floresta de caminhos ótimos (classificador) é criado em tempo $O(n \log n)$ na topologia da árvore geradora mínina, sem a necessidade de treinar o classificador sobre um conjunto totalmente rotulado.

O processo de classificação das duas abordagens seguem o mesmo procedimento, como será apresentado a seguir.

3.2.1 Classificação

Para a classificação, os caminhos ótimos de π_s para $s \in \mathcal{Z}_1'$ devem ser estendidos as novas amostras em $t \in \mathcal{Z}_2$ considerando a Equação 3.2, e atribuindo a t o rótulo $L_2(t) \leftarrow L_1(s^*)$ da amostra $s^* \in \mathcal{Z}_1'$ para o qual $\pi_{s^*} \cdot \langle s^*, t \rangle$ é ótimo. As Figuras 3.1e-f e 3.2e-f ilustram o processo de classificação para as técnicas OPFSEMI$_{mst}$ e OPFSEMI, respectivamente.

Note que a classificação considera que t seja conectado a todos os nós em $\mathcal{Z}_1$, em vez de t ser um nó adicional da MST. Portanto, a classificação baseia-se na mesma regra utilizada pelo classificador OPF supervisionado [66] – Algoritmo 3, mas agora usando um conjunto maior $\mathcal{Z}_1'$ do que apenas o conjunto $\mathcal{Z}_1^l$. Além disso, seguindo a ordem de nós em $\mathcal{Z}_1'$, a avaliação de $C_2(t)$ pode parar sempre que $C_1(s) \geq \max\{C_1(s^*), d(s^*, t)\}$ para algum nó $s^* \in \mathcal{Z}_1'$ anterior.

$$C_2(t) = \min_{\forall s \in \mathcal{Z}_1'} \{\max\{C_1(s), d(s,t)\}\}. \tag{3.2}$$

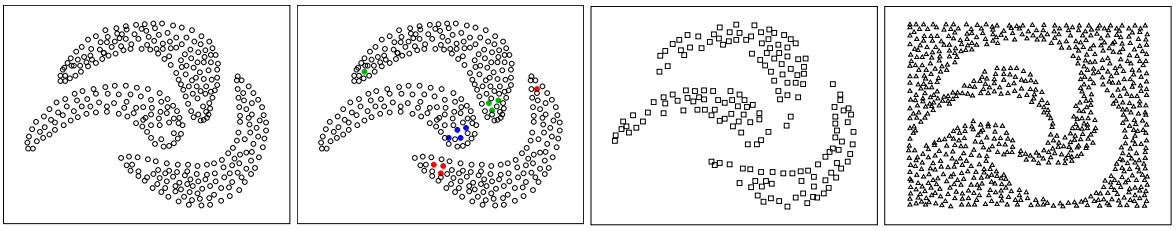
Algoritmo 3 – Algoritmo de Classificação semisupervisionado OPF

Entrada:	Classificador $[P_1, C_1, L_1, \mathcal{Z}_1']$, conjunto de teste $\mathcal{Z}_2$, e o par (v, d) para vetor de atributos e cálculo das distâncias.
Saída:	Mapa de rótulos L_2 e predecessores P_2 definido por $\mathcal{Z}_2$, e o conjunto $E_2 \subset \mathcal{Z}_2$ com as amostras classificadas incorretamente.
Auxiliares:	Variáveis *cst* e *mincost*.

1. $E_2 \leftarrow \emptyset$.

2. ***Para*** $t \in \mathcal{Z}_2$, ***Faça***
3. $\quad i \leftarrow 1$, $mincost \leftarrow max\{C_1(s_i), d(s_i, t)\}$.
4. $\quad L_2(t) \leftarrow L_1(s_i)$ *e* $P_2(t) \leftarrow s_i$.
5. $\quad$ ***Enquanto*** $i < |\mathcal{Z}_1'|$ *e* $mincost > C_1(s_{i+1})$, ***Faça***
6. $\quad\quad$ *Calcule* $cst \leftarrow \max\{C_1(s_{i+1}, d(s_{i+1}, t)\}$.
7. $\quad\quad$ ***Se*** $cst < mincost$, ***Então***
8. $\quad\quad\quad mincost \leftarrow cst$.
9. $\quad\quad\quad L_2(t) \leftarrow L(s_{i+1})$ *e* $P_2(t) \leftarrow s_{i+1}$.
10. $\quad\quad i \leftarrow i + 1$.
11. $\quad$ ***Se*** $\lambda(t) \neq L_2(t)$ ***Então*** *insira* t *em* E_2.
12. ***Retorne*** $[L_2, P_2, E_2]$.

A Figura 3.3 apresenta uma comparação de OPFSEMI$_{mst}$ e OPFSEMI entre os principais métodos semisupervisionados a serem avaliados em um conjunto de dados simples com amostras de treinamento, sendo (Figura 3.3a) conjunto das amostras não supervisionadas, (Figura 3.3b) amostras supervisionadas e não supervisionadas, amostras de teste dentro das regiões das classes (Figura 3.3c) e amostras de teste fora das regiões das classes (Figura 3.3d).



(a) (b) (c) (d)

Figura 3.3: (a) Um conjunto de dados com amostras não supervisionadas, (b) amostras supervisionadas selecionadas, (c) amostras de teste dentro das classes, e (d) amostras de teste fora das regiões das classes.

Os resultados da propagação do rótulo para $\mathcal{Z}_1^u$ e classificação de $\mathcal{Z}_2$ com amostras de teste dentro e fora das classes são apresentados para OPFSEMI$_{mst}$ (para esse caso com os mesmos resultados para OPFSEMI) (Figuras 3.4a–3.4c), SemiL [91] (Figuras 3.4d–3.4f), TSVM [22] (Figuras 3.4g–3.4i), LapSVM [10] (Figuras 3.4j–3.4l) e SSELM [42] (Figuras 3.4m–3.4o), respectivamente. A conectividade entre amostras supervisionadas e não supervisionadas permite uma redução considerável no erro de propagação do rótulo em $\mathcal{Z}_1^u$ e $\mathcal{Z}_2$ para OPFSEMI$_{mst}$, OPFSEMI e *Manifold regularization* usando LapSVM, comparados com SSELM, SemiL e TSVM.

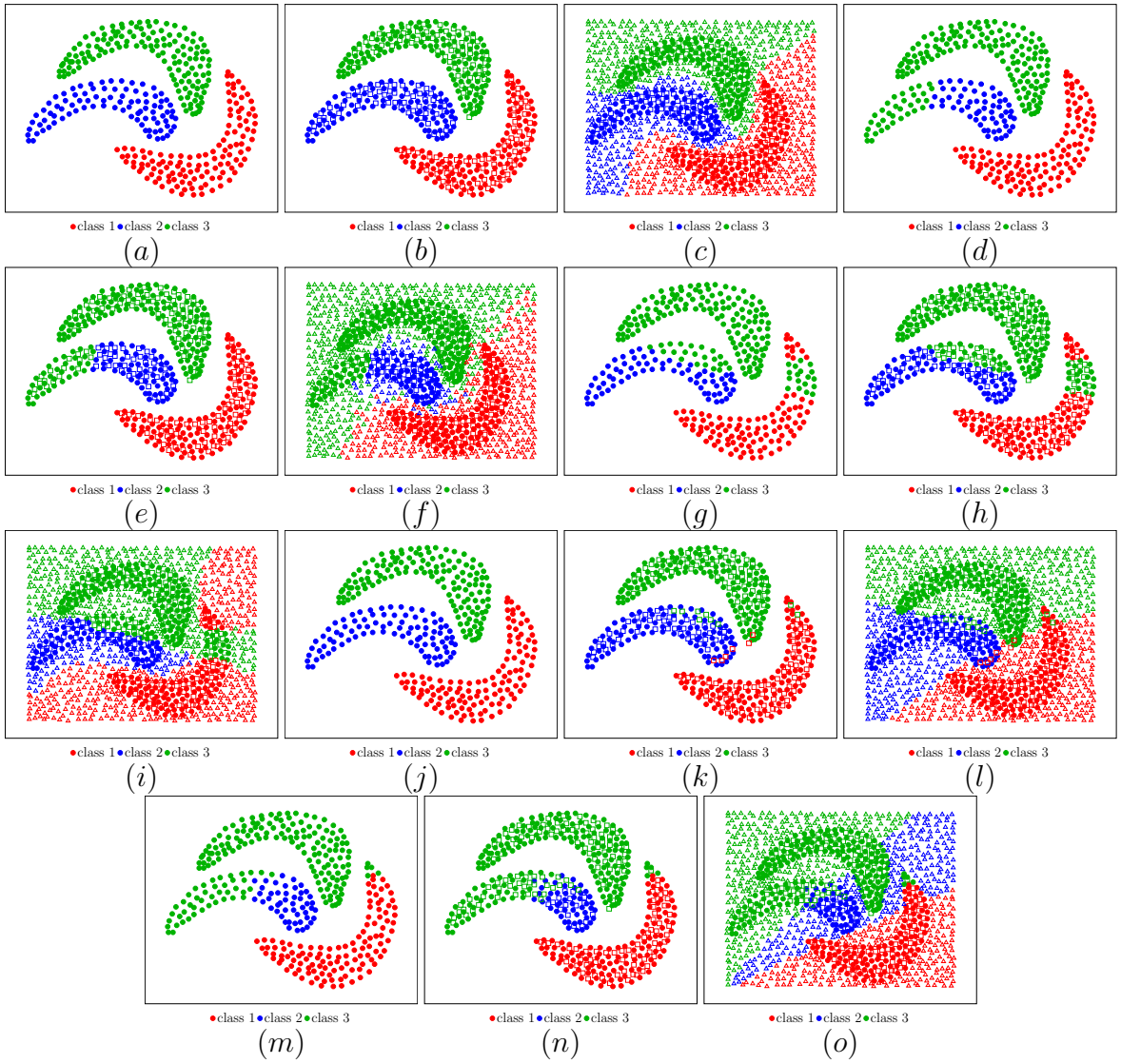


Figura 3.4: Propagação do rótulo e classificação das amostras de teste dentro e fora das regiões das classes para (a–c) OPFSEMI$_{mst}$ (resultado igual para OPFSEMI), (d–f) SemiL, (g–i) TSVM, (j–l) LapSVM e (m–o) SSELM, respectivamente.

Outra importante análise quanto a OPFSEMI$_{mst}$ e OPFSEMI, é quando há sobreposição entre classes, tornando sensível à escolha de amostras de treinamento para uma rotulação manual. Se forem selecionados nas regiões de sobreposição entre classes no espaço de atributos, podem assim proteger suas classes reduzindo os erros de propagação do rótulo e classificação (Figuras 3.5a–3.5c). No entanto, quando isto não for o caso, esses erros podem prejudicar o seu desempenho (Figuras 3.5d–3.5f). Isto essencialmente sugere o uso das abordagens propostas para o aprendizado ativo (Seção 3.4), onde o ob-

jetivo é identificar e selecionar amostras de aprendizagem representativas para rotulação correção/confirmação por um especialista.

Assim podemos destacar que os resultados de OPFSEMI$_{mst}$ e OPFSEMI devem ser equivalentes em teoria, porque qualquer caminho entre dois nós de uma árvore geradora mínima é ótima de acordo com a função de conectividade $f_{\max}$ usada por ambos. No entanto, as suas escolhas das raízes $\mathcal{S}$ da floresta não são as mesmas, e na prática, será possível verificar o poder de ganho de OPFSEMI$_{mst}$ sendo mais preciso e rápido comparado com OPFSEMI. Além disso, os métodos apresentados são (a) livre de parâmetros, (b) tratam problemas multiclasse em uma maneira natural, (c) não fazem suposições sobre as formas das classes, e (d) podem lidar com alguma sobreposição entre as classes, enquanto as raízes da floresta protegem suas respectivas classes.

3.3 Aprendizado semisupervisionado usando OPFSEMI$_{mst+knn}$

Na Seção anterior 3.2, propomos um método semisupervisionado de aprendizagem baseado na conectividade ótima entre amostras supervisionadas e não supervisionadas (OPFSEMI). Este método foi recentemente melhorado em termos de precisão e eficiência através de um novo algoritmo, chamado OPFSEMI$_{mst}$ [6]. Diretamente, podemos usar OPFSEMI$_{mst}$ para atribuir rótulos para novas amostras e inversamente transformar esses rótulos em uma ou várias classes por amostra. Em OPFSEMI$_{mst}$, o classificador é uma floresta de caminhos ótimos calculado sobre a topologia de uma árvore geradora mínima, cujos nós são amostras de treinamento supervisionados e não supervisionados. As raízes da floresta são as amostras supervisionadas e as amostras não supervisionadas são conquistadas e rotulados por sua raiz mais fortemente conexa. O processo de propagação de rótulo pressupõe que as amostras de uma mesma classe estão mais intimamente ligadas do que amostras de classes distintas. Para classificar uma nova amostra, todos os exemplos de treinamento são conectados à uma nova amostra e caminhos estendidos são avaliados para atribuir o rótulo de sua raiz mais fortemente conexa. Este classificador tem se mostrado robusto para um certo nível de erros de propagação de rótulos no conjunto de treinamento para problemas de atribuição de único rótulo [6]. No entanto, o problema da atribuição multirótulo, tendem a aumentar os erros de classificação quando os rótulos errados são inversamente transformados em múltiplas classes.

Essencialmente, o problema de atribuição multirótulo requer um classificador mais conservador do que OPFSEMI$_{mst}$ para melhor lidar com as possíveis sobreposições entre as classes. Nesta proposta, podemos melhorar a atribuição multirótulo, adicionando um último passo para o processo de treinamento de OPFSEMI$_{mst}$. O novo classifica-

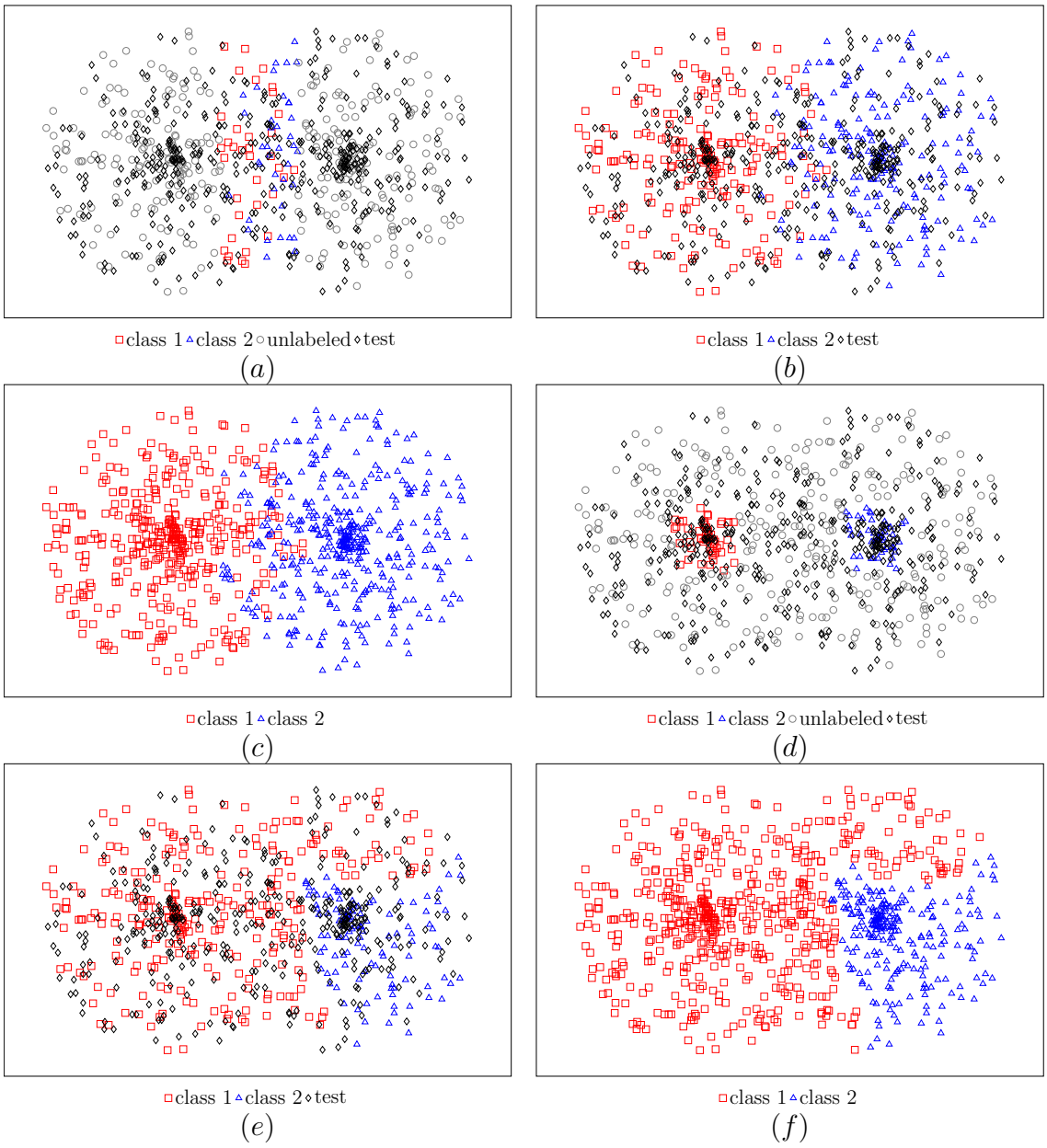


Figura 3.5: Impacto na seleção de amostras representativas. (a) Um conjunto de dados com as classes sobrepostas, com amostras de treinamento (supervisionadas e não supervisionadas) e amostras de teste. Os resultados de (b) propagação de rótulo e (c) classificação para OPFSEMI$_{mst}$, quando as amostras são supervisionadas nas regiões sobrepostas de classes. Em caso contrário (d), os erros de propagação rótulo (e) e classificação (f) tendem a aumentar.

dor semisupervisionado, chamado OPFSEMI$_{mst+knn}$, se tornará uma floresta de caminhos ótimos enraizadas nos máximos de uma função de densidade de probabilidade (pdf), calculado a partir de um grafo k-vizinhos mais próximos cujos nós são as amostras de treinamento. Basicamente, as raízes da floresta (novos protótipos) formam um subconjunto dos máximos da fdp onde, cada raiz, define uma árvore de caminhos ótimos (zona de influência do respectivo máximo) composta pelas suas amostras mais fortemente conexas. Durante a classificação, as amostras de treinamento mais fortemente conexas com suas raízes têm maior prioridade para atribuir rótulos para novas amostras. Uma vez que amostras classificadas erroneamente no conjunto de treinamento tendem a estar na fronteira entre as classes, onde os valores são mais baixos na pdf, OPFSEMI$_{mst+knn}$ pode melhorar significativamente o desempenho no problema de atribuição multirótulo.

A última etapa de OPFSEMI$_{mst+knn}$ é semelhante ao algoritmo proposto para aprendizado supervisionado por [65] e apresentada com mais detalhes na Seção 2.8. No entanto, o melhor valor de k pode ser estimado por diferentes maneiras. Em [72], o valor de k é selecionado como aquele que produz um corte normalizado mínimo sobre o grafo k-NN. Em [65], os autores usam uma outra variante do algoritmo OPF em um grafo k-NN para o aprendizado supervisionado. O valor de k é selecionado como aquele que minimiza os erros de propagação de rótulos sobre o conjunto de treinamento. Neste caso, no contexto supervisionado, esse critério resulta em baixos valores de k (ou seja, muitos clusters), o que também implica em uma estimativa pobre da pdf. No nosso caso, no entanto, OPFSEMI$_{mst}$ pode atribuir rótulos incorretos para as amostras de treinamento. Por isso, estaremos primeiramente estimando a percentagem de erros de propagação sobre a metade das amostras supervisionadas e selecionando o valor máximo de k tal que, quando rótulos são re-propagados para os máximos da pdf, o desacordo de rotulação com OPFSEMI$_{mst}$ seja menor ou igual à esta percentagem. Este critério tende a obter uma melhor estimação da pdf com um maior valor de k, além também de reduzir o erro de propagação sobre as amostras em regiões de menor densidade.

3.3.1 Treinamento

O processo de aprendizado semisupervisionado OPFSEMI$_{mst+knn}$ consiste basicamente na execução da técnica OPFSEMI$_{mst}$, e por fim após todas amostras serem rotuladas criamos um novo classificador a partir do algoritmo proposto para a aprendizagem supervisionada [65]. Ou seja, para a propagação dos rótulos no conjunto de treinamento, a nossa abordagem assume que as amostras de uma mesma classe estão mais fortemente conectadas através da sequência de amostras mais próximas do que amostras de classes distintas, que normalmente é o caso de conjuntos de dados com um *trade-off* razoável entre o número de amostras e a dimensão espaço de atributos. Considerando um conjunto de treinamento com uma grande quantidade de amostras completamente supervisionado,

o método usa uma função de densidade de probabilidade para criar um classificador mais adequado para o problema de atribuição multirótulo. A seguir iremos apresentar o modelo adotado para a geração do classificador final após a execução de OPFSEMI$_{mst}$.

Gerando o classificador final

Podemos pensar em amostras de treinamento a partir de $\mathcal{Z}_1$ como um ponto definido no espaço de atributos, que pode ser observado a partir de diferentes distâncias. A partir do infinito, todos os pontos são procurados como um único cluster. Conforme nos aproximamos, a escala em que os pontos são observados sofre mudanças e vários clusters podem aparecer. A determinação da melhor escala para resolver um problema de agrupamento é uma tarefa que depende da aplicação.

O algoritmo de agrupamento OPF, como proposto por [72], segue o princípio, definindo a escala como um inteiro $1 \leq k < |\mathcal{Z}_1|$. Dado que o conjunto de treinamento $\mathcal{Z}_1$ é não supervisionado em [72], o método estima uma função densidade de probabilidade (pdf) para os pontos em $\mathcal{Z}_1$ e encontra os cluster como a zona de influência do respectivo máximo da pdf. A pdf é interpretada como pesos de nó de um grafo $(\mathcal{Z}_1, \mathcal{A}_k, \rho)$ definida pela relação de adjacência $\mathcal{A}_k$, que conecta cada amostra a seu k-vizinho mais próximo no espaço de atributos.

Ao alterar $k \in [k_{\min}, k_{\max}]$, $k_{\min} \geq 1$ e $k_{\max} < |\mathcal{Z}_1|$, adequado para uma determinada aplicação, pode-se obter uma pdf com mais ou menos regiões de influência (clusters). O algoritmo OPF é executado várias vezes para rotular as regiões da pdf, produzindo um corte no grafo, e a melhor solução de cluster (o melhor valor de k) é escolhido como aquele que minimiza a medida de corte no grafo. Em nosso caso, propomos uma abordagem diferente, como será explicado mais tarde.

Assumimos que, de momento, ao selecionar o melhor valor de k, uma variante do algoritmo OPF (Algoritmo 4) requer uma função de conectividade $f_{\min}$, onde $\delta = min_{\forall (s,t) \in \mathcal{A}_k | \rho(t) \neq \rho(s)} |\rho(t) - \rho(s)|$.

$$\begin{aligned} f_{min}(\langle t \rangle) &= \begin{cases} \rho(t) & \text{se } t \in \mathcal{S} \subset \mathcal{Z}_1, \\ \rho(t) - \delta & \text{caso contrário,} \end{cases} \\ f_{min}(\pi_s \cdot \langle s,t \rangle) &= \min\{f(\pi_s), \rho(t)\}, \end{aligned} \tag{3.3}$$

O algoritmo maximiza o mapa de conectividade $C_2(t) = \max_{\forall \pi_t \in \Pi_t}\{f_{\min}(\pi_t)\}$, de tal forma que o grafo é particionado em uma floresta de caminhos ótimos P_2 (classificador) enraizado no máximo do pdf. Isto é, o algoritmo detecta em tempo real (*on-the-fly*), uma amostra raiz por máximo para compor o conjunto $\mathcal{S}$ assegurando uma única árvore de caminho ótimo por cada zona de influência da pdf. Esta propriedade, no entanto, exige inserirmos em $\mathcal{A}_k(s)$ o nó t, tal que $s \in \mathcal{A}_k(t)$ e $\rho(s) = \rho(t)$, para torná-lo simétricos nos

platôs da pdf. O rótulo $L_1(s)$ de cada raiz $s \in \mathcal{S}$ é propagado para o mapa de rótulo $L_2(t)$ dos nós restantes $t \in \mathcal{Z}_1$.

A execução do algoritmo 4 para $f_{\min}$ sobre o grafo k-NN $(\mathcal{Z}_1, \mathcal{A}_k, \rho)$ cria uma floresta de caminhos ótimos com atributos $[P_2, C_2, L_2]$ (Figuras 3.6 (e) e (f)), e uma lista ordenada $\mathcal{Z}_1'$ de nós em $\mathcal{Z}_1$ com a finalidade de acelerar classificação, conforme descrito na próxima seção.

Algoritmo 4 – Algoritmo OPF para $f_{\min}$ sobre $(\mathcal{Z}_1, \mathcal{A}_k, \rho)$

Entrada:	Grafo $(\mathcal{Z}_1, \mathcal{A}_k, \rho)$ e mapa de rótulos L_1.
Saída:	Mapas da floresta de caminhos ótimos e seus atributos $[P_2, C_2, L_2]$ e lista ordenada $\mathcal{Z}_1'$.
Auxiliares:	Fila de prioridade Q e váriavel de conectividade val.

1. ***Para Cada*** *nó* $t \in \mathcal{Z}_1$, ***Faça***
2. $\quad$ $C_2(t) \leftarrow \rho(t) - \delta$, $P_2(t) \leftarrow nil$, *e insira* t *em* Q.
3. ***Enquanto*** Q *é não vazia,* ***Faça***
4. $\quad$ *Remova de* Q *uma amostra* s *tal que* $C_2(s)$ *é mínimo.*
5. $\quad$ *Insira* s *em* $\mathcal{Z}_1'$.
6. $\quad$ ***Se*** $P_2(s) = nil$, ***Então*** $C_2(s) \leftarrow \rho(s)$ *e* $L_2(s) \leftarrow L_1(s)$
7. $\quad$ ***Para Cada*** $t \in \mathcal{A}_k(s)$ ***Faça***
8. $\quad\quad$ ***Se*** $C_2(t) < C_2(s)$, ***Então***
9. $\quad\quad\quad$ *Calcule* $val \leftarrow \min\{C_2(s), \rho(t)\}$.
10. $\quad\quad\quad$ ***Se*** $val > C_2(t)$, ***Então***
11. $\quad\quad\quad\quad$ *Remova* t *de* Q.
12. $\quad\quad\quad\quad$ $P_2(t) \leftarrow s$, $L_2(t) \leftarrow L_2(s)$,
13. $\quad\quad\quad\quad$ $C_2(t) \leftarrow val$, *e insira* t *em* Q.
14. ***Retorne*** $[P_2, C_2, L_2]$ *e* $\mathcal{Z}_1'$.

Note que o Algoritmo 4 não requer um código de cores para controlar o status dos nós de Q. No início, todos os nós de $\mathcal{Z}_1$ são candidatos a raiz, um nó por máximo da pdf ρ é selecionado (conjunto $\mathcal{S}$) quando $P_2(s) = nil$ como raiz do mapa na Linha 6. Este nó raiz, irá então, propagar o rótulo $L_1(s)$ a todos nós no mesmo platô (uma vez que $\mathcal{A}_k$ são simétricos nos platôs), assim como a mesma zona de influência da pdf. Note que, sempre que um nó s encontrar um nó adjacente t, satisfazendo a condição da Linha 10, o nó t estará em Q e será conquistada pela raiz de s.

Para escolhermos o valor de k, propomos inicialmente estimar a porcentagem de erro de propagação de rótulo $\mathcal{E}$ que OPFSEMI$_{mst}$ possa cometer em determinado conjunto de treinamento. Para isso, executamos apenas sobre o conjunto supervisionado $\mathcal{Z}_1^l$ os Algoritmos 1 e 2, definindo as raízes da floresta na metade das amostras supervisionadas

e a medição de $\mathcal{E}$ na outra metade. Este processo também pode ser repetido algumas vezes para melhor estimar a percentagem de erros de propagação de rótulo. Finalmente, executamos OPFSEMI$_{mst+knn}$, avaliando a saída do Algoritmo 4 sobre o grafo k-NN $(\mathcal{Z}_1, \mathcal{A}_k, \rho)$ para valores de k a partir de $k_{\max}$ até $k_{\min}$, com objetivo de selecionar o maior valor de k que mantenha o percentual de discordância de rotulação entre L_1 e L_2 menor ou igual a $\mathcal{E}$. Este critério tende a obter uma melhor estimativa da pdf com maior valor de k, e que também poderá reduzir erros de rotulação cometidos inicialmente pelo Algoritmo 2, uma vez que amostras em regiões com maior densidade devem conquistar amostras de regiões de menor densidade.

3.3.2 Classificação

Para classificar uma nova amostra $t \in \mathcal{Z}_2$, o algoritmo avalia os caminhos ótimos de maneira incremental da seguinte forma:

$$C_2(t) = \max_{\forall s \in \{\mathcal{Z}_1 \cap \mathcal{A}_k(t)\}} \{\min\{C_2(s), \rho(t)\}\}. \tag{3.4}$$

Se $s^* \in \mathcal{Z}_1$ é o único nó que satisfaz a equação acima, então a classificação simplesmente atribui $L_2(t) \leftarrow L_2(s^*)$.

Em [18], os autores aceleram o processo de propagação de rótulo dos clusters para novas amostras com base na pdf evitando o cálculo de $\mathcal{A}_k(t)$ para todos os nós $t \in \mathcal{Z}_2$. Uma idéia semelhante aplica-se pela Equação 3.4. Durante o treinamento, um raio $\Omega(s)$ pode ser estimado como o máximo entre os valores de distância $d(s,t)$ (o valor mediano torna mais robusto a *outliers*) entre as amostras $s, t \in \mathcal{Z}_1$, tal que $t \in \mathcal{A}_k(s)$. Se a distância $d(s,t) \leq \Omega(s)$ para $s \in \mathcal{Z}_1$ e $t \in \mathcal{Z}_2$, então t está dentro da região definida pela k-adjacência de s.

A lista $\mathcal{Z}_1'$ de amostras de treinamento ordenadas na ordem não-crescente dos valores de caminho torna-se desnecessária o cálculo de $\rho(t)$ e, seguindo a ordem dos nós em $\mathcal{Z}_1'$, definimos $L_2(t) \leftarrow L_2(s^*)$ para o primeiro nó $s^* \in \mathcal{Z}_1'$ que satisfaça $d(s^*, t) \leq \Omega(s^*)$. Note que os nós mais fortemente conexos com suas raízes terão maior prioridade na atribuição de rótulo para novas amostras. Uma vez que amostras classificadas de maneira errada em $\mathcal{Z}_1$, devido à etapa anterior (usando primeiramente OPFSEMI$_{mst}$) são mais propensas a estar na fronteira entre as classes, isso indica que agora estarão mais fracamente conectadas as suas raízes possuindo valores mais baixos da pdf na fronteira entre clusters. Isso justifica a melhoria de OPFSEMI$_{mst+knn}$ sobre OPFSEMI$_{mst}$ para o problema de atribuição multirótulo.

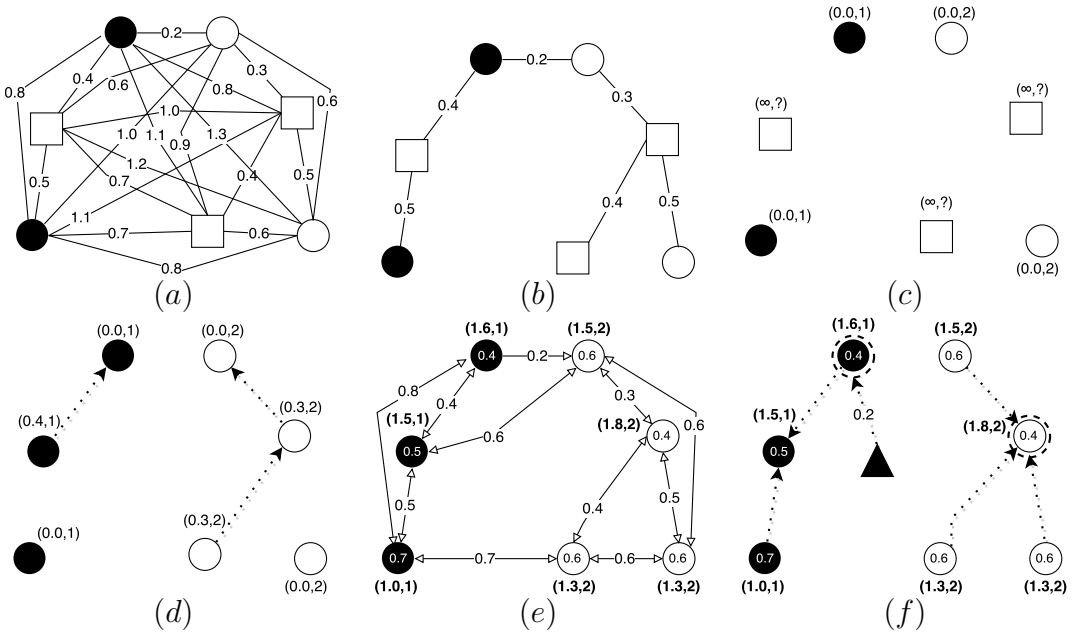


Figura 3.6: (a) Grafo completo e ponderado sobre o conjunto de treinamento $\mathcal{Z}_1$ (• amostras supervisionadas da classe 1, ∘ amostras supervisionadas da classe 2 e □ amostras não supervisionadas). (b) Uma árvore geradora mínima de (a). (c) Um mapa de conectividade trivial para uma Floresta de Caminhos Ótimos calculada usando $f_{\max}$ sobre (b) e $\mathcal{S} = \mathcal{Z}_1^l$. (d) Resultado da Floresta de Caminhos Ótimos. (e) Grafo conectado aos seus k-vizinhos mais próximos ($k = 3$ no exemplo). A seta ($-\triangleright$) aponta para os vizinhos mais próximos. Os identificadores (x, y) acima dos nós são, respectivamente, o seu valor de densidade e o rótulo da classe a qual ele pertence. O valor dentro do nó representa o raio (o valor da mediana). (f) Floresta de Caminhos Ótimos calculada usando $f_{\min}$. Amostra de teste (triângulo) inserida no grafo e o resultado de classificação, tal que $L_2(t) = L_2(s^*)$. Os elementos circulados (tracejados) representam os máximos de cada classe.

Qual é a lógica de OPFSEMI$_{mst+knn}$?

Para o caso de sobreposição entre classes, mostramos na Figura 3.5 que o desempenho de OPFSEMI$_{mst}$ é melhor quando o especialista rotula as amostras de $\mathcal{Z}_1^l$ nas regiões de sobreposição entre classes. No entanto, assumindo uma escolha aleatória das amostras para $\mathcal{Z}_1$, serão mais propensos a cair nas regiões de maior densidade no espaço de atributos, em que o centro das classes geralmente se encontram. No problema de atribuição multirótulo, sobreposição entre classe tendem a aumentar quando várias classes são transformados em rótulos individuais. Isso aumenta os erros de propagação de rótulo de OPFSEMI$_{mst}$ e, quando os rótulos individuais são inversamente transformados em várias classes, o desempenho final de OPFSEMI$_{mst}$ tende a ser pior do que o problema de atribuição de único

rótulo.

Por outro lado, os erros de propagação de rótulo tendem a se concentrar nas regiões de menor densidade. Portanto, amostras nos máximos da pdf (centro das classes) são mais confiáveis para re-propagar os rótulos, o que justifica a nossa escolha para OPFSEMI$_{mst+knn}$. Este cenário é ilustrado na Figura 3.7. A Figura 3.7a mostra duas classes sobrepostas e a escolha aleatória de amostras supervisionadas, não supervisionadas e teste. Os resultados de propagação de rótulo e classificação por OPFSEMI$_{mst}$ são apresentados nas Figuras 3.7b e 3.7c, respectivamente. Quando os rótulos são re-propagados a partir dos máximos da pdf, o resultado da classificação por OPFSEMI$_{mst+knn}$ reduz os erros (Figura 3.7d). Com isso, teremos uma maior precisão quando os rótulos individuais serão inversamente transformados em várias classes.

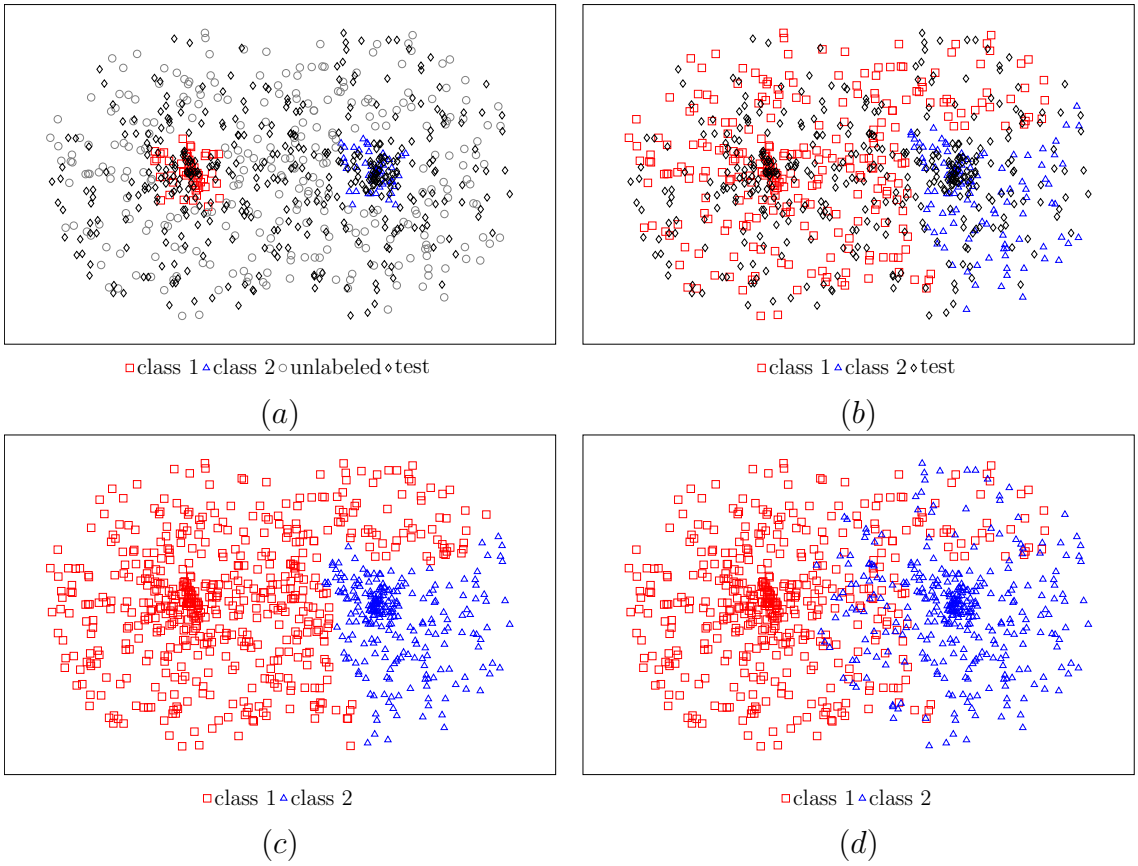


Figura 3.7: (a) Um conjunto de dados com as classes sobrepostas, com amostras de treinamento (supervisionadas e não supervisionadas) e amostras de teste. (b) Propagação de rótulo por OPFSEMI$_{mst}$, (c) classificação para OPFSEMI$_{mst}$ e (d) classificação para OPFSEMI$_{mst+knn}$.

3.4 Aprendizado semisupervisionado ativo usando OPFSEMI e OPFSEMI$_{mst}$

As técnicas apresentadas na Seção 2.9, se concentram em estratégias de aprendizado ativo que classificam e/ou organizam o conjunto de dados inteiro e, posteriormente, realizam a seleção e exibição, em cada iteração, das amostras representativas para serem rotuladas pelo especialista. Para grandes conjuntos de dados, estas fases completas realizadas a cada iteração, se tornam ineficientes e ao mesmo tempo computacionalmente impraticável.

Nesta seção, apresentaremos uma abordagem, chamado *Aprendizado Semisupervisionado Ativo usando Floresta de Caminhos Ótimos* (ASSL-OPF). A proposta difere da abordagem típica de aprendizado semisupervisionado em que os métodos precisam esperar para a coleta de todos os dados supervisionados e não supervisionados antes de iniciar o processo de aprendizagem e ao mesmo tempo ser capaz de fazer previsões semisupervisionadas. Além disso, difere drasticamente de aprendizado ativo padrão no qual todas as amostras no banco de dados têm de ser classificados e/ou reorganizados em cada iteração de aprendizagem. A Figura 3.8 apresenta o pipeline da abordagem de aprendizado semisupervisionado ativo. Nesta nova proposta iremos explorar a combinação de aprendizagem ativo e semisupervisionado usando OPFSEMI e OPFSEMI$_{mst}$, a fim de selecionar as amostras supervisionadas mais representativas, o que terá um considerável impacto na diminuição dos erros de propagação, bem como na construção de classificadores mais robustos.

3.4.1 Estratégia de aprendizado ativo

A estratégia de aprendizagem ativo [74] que usamos reduz a possibilidade de selecionar uma amostra irrelevante de um grande conjunto de aprendizagem. Uma vez aplicado a redução e organização dos dados uma única vez (a priori) será possível reduzir significativamente o conjunto de dados não supervisionados a um conjunto menor contendo amostras mais representativas a serem utilizadas no processo de aprendizado. Importante destacar que tal estratégia difere completamente da abordagem padrão (*baseline*), em que as amostras são selecionadas aleatoriamente da base de dados inteira. A estratégia de redução adotada nesta abordagem é baseada no grafo de agrupamento proposto por Rocha et. al [72] e apresentada na Seção 2.7.

Seja $\mathcal{Z}_1$, inicialmente um conjunto de dados de aprendizagem não supervisionado, de modo que cada amostra $s \in \mathcal{Z}_1$ possui um vetor de atributos $\vec{v}(s)$. Para $s, t \in \mathcal{Z}_1$, seja $d(s,t)$ a distância entre s e t no espaço de atributos (por exemplo, $d(s,t) = ||\vec{v}(t) - \vec{v}(s)||$). O par $(\mathcal{Z}_1, \mathcal{A})$ denota um grafo k-vizinho mais próximo (k-NN). Ou seja, uma relação k-NN sendo $\mathcal{A}$ é definida sobre $\mathcal{Z}_1 \times \mathcal{Z}_1$ tal que uma amostra $t \in \mathcal{Z}_1$ é dita ser adjacente a

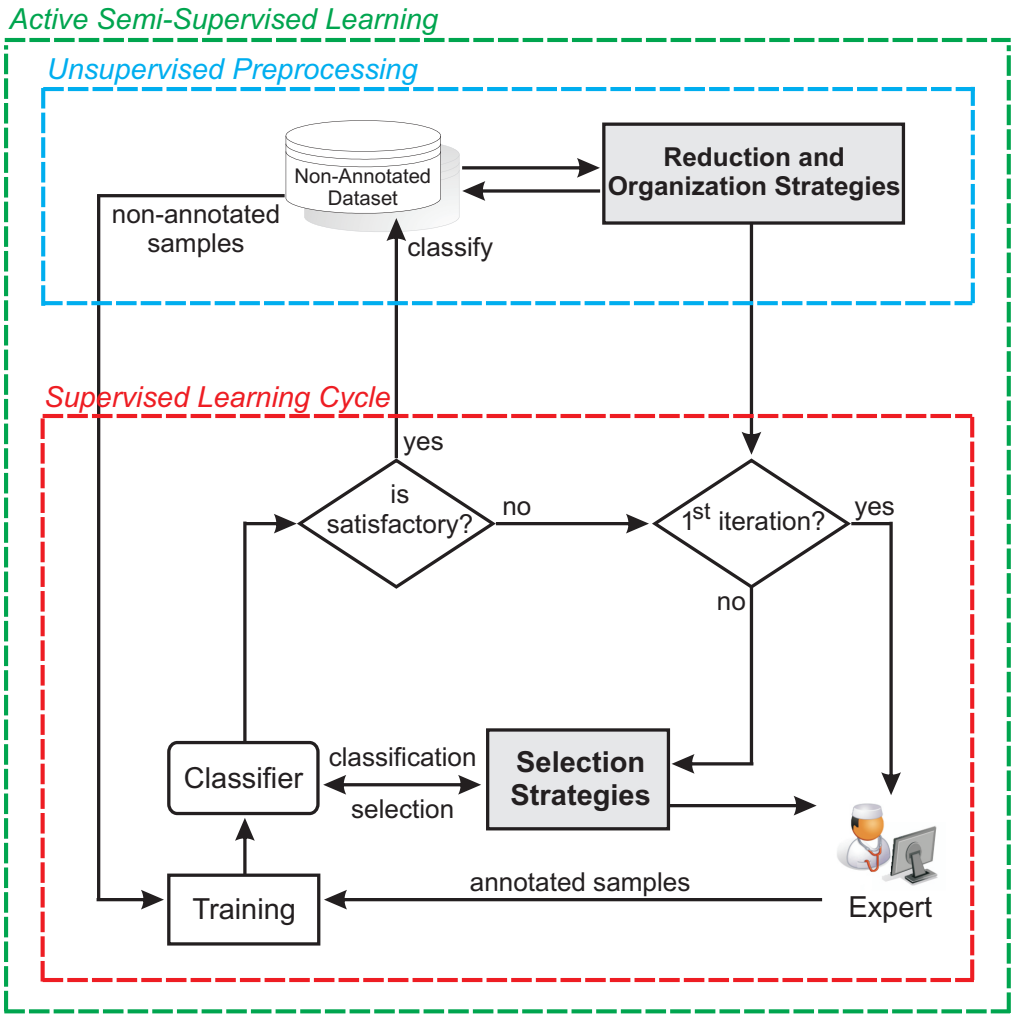


Figura 3.8: Pipeline da proposta de aprendizado semisupervisionado ativo.

uma amostra $s \in \mathcal{Z}_1$, se t é um k-vizinho mais próximo de s de acordo com a função de distância d.

O Algoritmo 5 apresenta a estratégia de redução de dados. A partir do agrupamento de $\mathcal{Z}_1$ usando uma relação k-NN sendo $\mathcal{A}$ (Linha 1), obtemos um conjunto de raízes $\mathcal{R}$ (Linha 2) bem como o limite do conjunto de amostras $\mathcal{Z}'_1$ (Linhas 4–12). O método de agrupamento atribui cluster-ids às amostras, e uma amostra $s \in \mathcal{Z}_1$ é uma amostra de fronteira se existe pelo menos uma amostra adjacente $t \in \mathcal{A}(s)$ cujo agrupamento-id é diferente de s. Assim, (s, t) é uma fronteira entre diferentes grupos, se a amostra t é um dos k-vizinhos mais próximos da amostra s e $s.clusterid \neq t.clusterid$.

Algoritmo 5 – Estratégia de Aprendizado Ativo

Entrada: Um conjunto de dados não supervisionado de aprendizagem $\mathcal{Z}_1$ e uma relação k-NN representado por $\mathcal{A}$.
Saída: Conjunto de fronteira $\mathcal{Z}'_1$ com arestas da MST ordenadas e o conjunto de raízes $\mathcal{R}$

1. *Calcule os grupos de $\mathcal{Z}_1$ usando a relação de adjacência de $\mathcal{A}$;*
2. $\mathcal{R} \leftarrow$ *raízes dos grupos;*
3. $\mathcal{Z}'_1 \leftarrow nil$;
4. **Para Cada** $s \in \mathcal{Z}_1$ **Faça**
5. **Para Cada** $t \in \mathcal{A}(s)$**Faça**
6. **Se** $s.clusterid \neq t.clusterid$**Então**
7. $\mathcal{Z}'_1 \leftarrow \mathcal{Z}'_1 \cup aresta\,(s,t)$;
8. *break;*
9.
10.
11. *Calcule uma MST de $\mathcal{Z}'_1$;*
12. $\mathcal{Z}'_1 \leftarrow$ *arestas da MST com pesos em ordem decrecente.*

As raízes em $\mathcal{R}$ deverão aumentar a possibilidade de selecionar amostras de todas as classes e as amostras de fronteira (amostras mais próximas com grupo distinto) definida por $\mathcal{Z}'_1$, irão conter as amostras mais difíceis para a classificação. Portanto, este resultado irá depender da escolha apropriada de k. Usamos a abordagem de otimização descrita por [72] com uma restrição adicional sendo $k \geq 2c$, onde c é o número de classes, a fim de cobrir todos/ou a maioria das classes. Com isso, consideramos também que uma classe poderá ser representada por mais de um grupo.

Com objetivo de aumentar a probabilidade da coleta de amostras na fronteira entre as classes distintas em $\mathcal{Z}'_1$, a estratégia interpreta esse conjunto como um grafo completo ponderado pela distância $d(s,t)$ entre as amostras no espaço de atributos, calcula um árvore geradora mínima (MST) (Linha 11), e ordena as amostras da fronteira da MST por ordem decrescente de peso (Linha 12). Dado que as arestas da fronteira com pesos menores são mais propensos a estar na mesma classe, esta estratégia permite priorizar amostras conectadas por arestas com pesos maiores e classificadas em classes distintas durante o processo de seleção para o especialista anotar e/ou confirmar.

3.4.2 Aprendizado semisupervisionado e ativo

O processo de seleção consiste em escolher as amostras (conjunto de treinamento) que serão usadas para treinar o classificador ao longo das iterações. Para isso, o conjunto de treinamento $\mathcal{Z}_1$ será composto de amostras supervisionadas e não supervisionadas. As

amostras supervisionadas serão selecionadas a partir do conjunto reduzido, ou seja, as raízes dos grupos $\mathcal{R}$ e suas amostras de fronteira ($\mathcal{Z}_1'$), que serão inicialmente rotuladas pela estratégia de agrupamento. Em seguida, as amostras selecionadas terão seus respectivos rótulos verificados/corrigidos pelo especialista. As amostras não supervisionadas serão selecionados a partir de $\mathcal{Z}_1''$ composta das amostras restantes a partir de $\mathcal{Z}_1 \backslash R \cup Z_1'$.

Na primeira iteração de aprendizagem, as raízes dos grupos calculados durante o processo de redução de dados serão exibidos para o especialista, que informará seus rótulos. Estas amostras rotuladas, constituem no primeiro conjunto supervisionado. O conjunto não supervisionado será selecionado de forma aleatória com o dobro de elementos do conjunto supervisionado. As uniões desses conjuntos constituem o conjunto de treinamento para a primeira instância do classificador semisupervisionado.

Durante o ciclo de aprendizagem, as amostras na lista ordenada de arestas da MST serão rotuladas pelo classificador corrente e as amostras sobre as arestas que recebem diferentes rótulos serão selecionadas. Estas duas fases, classificação e seleção, são realizadas alternadamente até que o número de amostras a ser exibido para o especialista em cada iteração é atingido. Note que esta abordagem não exige a classificação de todas as amostras no conjunto de dados, a cada iteração.

Além disso, uma vez que as amostras selecionadas são automaticamente rotuladas pelo classificador atual, o especialista só precisará anotar aquelas classificadas incorretamente. Portanto, o classificador será otimizado a cada iteração e o número de amostras classificadas incorretamente será cada vez mais reduzido. Em seguida, depois dos rótulos confirmados/corrigidos pelo especialista, as amostras recém-rotuladas, bem como os não rotuladas são escolhidas aleatoriamente para serem incorporadas no conjunto de treinamento e uma nova amostra do classificador é então criada. Ao notar que uma precisão aceitável foi alcançada, o especialista pode direcionar o classificador final para anotar o que resta do conjunto de dados. Em nossos experimentos, considerou-se que um especialista estaria satisfeito sempre que a precisão permanecesse estável ou chegasse a um nível suficientemente elevado para cada aplicação. Desta forma, o tempo e o esforço do especialista será significativamente reduzido.

O Algoritmo 6 apresenta a abordagem de aprendizado semisupervisionado ativo. Após o pré-processamento realizado pelo agrupamento OPF, no processo de redução (Seção 3.4.1), obtemos os conjuntos $\mathcal{R}$ e $\mathcal{Z}_1'$, contendo a raiz de cada grupo e as amostras de fronteira, respectivamente. O conjunto inicial supervisionado $\mathcal{Z}_1^l$ consiste das raízes que formam o conjunto $\mathcal{R}$ (Linha 1). Na Linha 2, o especialista anota as classes das raízes em $\mathcal{Z}_1$. O conjunto inicial não supervisionado $\mathcal{Z}_1^u$ consiste de amostras aleatórias a partir do conjunto remanescente não rotulado $\mathcal{Z}_1''$ Linhas (3–4). Na Linha 5, obtém-se o primeiro conjunto de treinamento $\mathcal{Z}_1$ formado pelos conjuntos supervisionados $\mathcal{Z}_1^l$ e não supervisionados $\mathcal{Z}_1^u$. O laço das Linhas de 6–13 envolve os processos de (re)-treinamento e seleção. A cada

iteração, arestas de $\mathcal{Z}_1'$ serão analisadas. Conforme as arestas são analisadas, suas amostras são rotuladas pelo classificador semisupervisionado e aquelas com classes distintas são selecionadas para serem exibidas para o especialista. Deste modo, o crescimento do conjunto de treinamento é controlado, uma vez que apenas as amostras rotuladas com maior benefício serão mantidas e os seus rótulos serão propagados para as amostras não supervisionadas. O ciclo de aprendizagem é repetido até que o especialista esteja satisfeito com a taxa de sucesso no conjunto selecionado.

Algoritmo 6 – Aprendizado Ativo e Semisupervisionado

Entrada:	Conjunto $\mathcal{Z}_1$, conjunto com amostras de fronteira $\mathcal{Z}_1' \subset \mathcal{Z}_1$ ordenados pelo peso das arestas na MST, conjunto de raízes $\mathcal{R} \subset \mathcal{Z}_1$ e número de classes c
Saída:	Classificador semisupervisionado
Auxiliares:	Conjunto de amostras não supervisionadas $\mathcal{Z}_1''$, conjunto de treinamento $\mathcal{Z}_1$, conjunto supervisionado selecionado $\mathcal{Z}_1^l$, e conjunto não supervisionado selecionado $\mathcal{Z}_1^u$

1. $\mathcal{Z}_1^l \leftarrow \mathcal{R}$;
2. *Especialista rotula as classes das raízes em* $\mathcal{Z}_1^l$;
3. $\mathcal{Z}_1'' \leftarrow \mathcal{Z}_1 \backslash \mathcal{R} \cup \mathcal{Z}_1'$;
4. $\mathcal{Z}_1^u \leftarrow (2 \cdot |\mathcal{Z}_1^l|)$ *amostras aleatórias de* $\mathcal{Z}_1''$;
5. $\mathcal{Z}_1 \leftarrow \mathcal{Z}_1^l \cup \mathcal{Z}_1^u$;
6. ***Enquanto*** *especialista não está satisfeiro* ***Faça***
7. | *(Re-)treina o classificador semisupervisionado com* $\mathcal{Z}_1$;
8. | $\mathcal{Z}_1^l \leftarrow$ *novas amostras classificadas com classes distintas, seguindo a ordem* $\mathcal{Z}_1'$;
9. | *Especialista aceita/corrige as classes de amostras em* $\mathcal{Z}_1^l$;
10. | $\mathcal{Z}_1^u \leftarrow (2 \cdot |\mathcal{Z}_1^l|)$ *amostras aleatórias de* $\mathcal{Z}_1''$;
11. | $\mathcal{Z}_1 \leftarrow \mathcal{Z}_1 \cup \mathcal{Z}_1^l \cup \mathcal{Z}_1^u$.

Capítulo 4

Resultados Experimentais

4.1 Considerações iniciais

Esta seção tem por objetivo apresentar as bases de dados e experimentos realizados, os quais avaliam os classificadores semisupervisionados baseados em OPF para problemas de único rótulo, multirótulos e aprendizado ativo, com relação a medidas acurácia, erro de propagação e eficiência (tempo computacional), usando técnicas do estado da arte como referências. Os classificadores semisupervisionados baseados em floresta de caminhos ótimos adotados nos experimentos foram aqueles descritos nas Seções (3.2, 3.3) e 3.4. Importante destacar que os resultados apresentados a seguir serão uma apresentação da pesquisa realizada durante os últimos anos referentes às principais publicações desta tese de doutorado. De maneira resumida, seguiremos a seguinte ordem de apresentação:

- **Seção 4.4:** Problemas de único rótulo;
- **Seção 4.5:** Problemas multirótulos;
- **Seção 4.6:** Aprendizado semisupervisionado e ativo.

Para atingir o objetivo de avaliação de todas as propostas sobre o contexto do ganho de desempenho na inserção de novas informações no uso de técnicas semisupervisionadas, comparamos cada proposta variando a quantidade de amostras (supervisionadas e não supervisionadas) que são inseridas no conjunto de treinamento. Como cada experimento possui suas particularidades, estaremos apresentando os detalhes dos métodos comparados e parâmetros em sua respectiva seção. Importante destacar que todos os métodos baseados em Floresta de Caminhos Ótimos aqui apresentados, serão disponibilizados na versão mais recente da LibOPF 3.0[1].

[1] www.ic.unicamp.br/afalcao/libopf/

4.2 Base de dados de único rótulo

Os experimentos baseados em único rótulo foram realizados em bases de dados de diversos domínios. Todas as bases de dados utilizadas possuem atributos númericos, de tal forma que se pode usar medidas de dissimilaridade como a distância Euclidiana entre os vetores de atributos. A Tabela 4.1 apresenta as base de dados selecionadas com os respectivos números de amostras, atributos e classes. Os seis primeiros conjuntos de dados são sintéticos e disponíveis ao público. Os dois últimos (Cowhide e Parasites), são bases privadas obtidas de único rótulo de aplicações reais. A seguir, mais detalhes sobre as bases de dados.

Tabela 4.1: Características das bases de dados de único rótulo (ID - Identificador): número de amostras, número de atributos e número total de classses.

ID–**Base de dados**	#amostras	#atributos	#*classes*
l_1– Statlog [30]	2.310	19	7
l_2– Spambase [24]	4.601	57	2
l_3– Faces [1]	1.864	162	54
l_4– Pendigits [4]	10.992	16	10
l_5– KddCup [77]	48.898	41	23
l_6– Letter [31]	20.000	16	26
l_7– Cowhide [7]	1.690	160	5
l_8– Parasites [76]	1.660	262	15

Faces: base de dados obtida a partir de estudos da Universidade de Notre Dame [1], originalmente criada com objetivo de estudar efeitos do tempo sobre reconhecimento facial com diferentes expressões, como: neutro, sorridente, triste, entre outros. A Figura 4.1(g) apresenta alguns exemplos de amostras desta base de dados. **Statlog:** é uma base de dados obtida de imagens (valores multi-espectrais dos pixels) do satélite Landsat [30]. **Pendigits:** é uma base de dados criada para reconhecimento de dígitos manuscritos [4]. Consiste basicamente de uma grande quantidade de amostras com 16 atributos (elaborados por re-amostragem de dígitos manuscritos), distribuídas em 10 classes correspondendo aos dígitos [0...9]. **Spam:** criada pelo laboratório Hewlett-Packard [24] com mensagens de spam e não spam como amostras. **KddCup:** é uma base de dados com indicadores para o problema de detecção de intrusão em redes de computadores. Devido à grande massa

de dados e com o intuito de reduzir o custo computacional, usamos um subconjunto com 10% dos dados de treinamento sobre a base de dados original. **Letter:** base de dados que contém um conjunto de 16 atributos obtidos a partir de um conjunto de imagens com as letras do alfabeto A a Z (letra maiúscula). Seu objetivo é classificar uma imagem (com uma única letra) como uma das 26 letras do alfabeto inglês. Todas essas bases de dados estão disponíveis no repositório da UCI Machine Learning[2].

A base de dados **Cowhide** é composta por cinco tipos de regiões de interesse na fase de processamento Wet-Blue[3]: sarna, carrapatos, marca-ferro, corte e regiões sem defeito (Figure 4.1a-e). A principal razão para a seleção de amostras de defeitos do couro é o desafio do problema, especialmente em áreas com proximidades entre diferentes defeitos. A base de dados **Parasites** contém amostras de 15 espécies de protozoários e helmintos. Esses objetos foram obtidos a partir do processamento de amostras fecais usando imagens de microscopia óptica. O desequilíbrio das classes com relação ao número de amostras é um desafio, uma vez que o número de amostras por classe possui uma grande variância e as impurezas fecais são muito mais frequentes. A Figure 4.1f exibe exemplos de todas as espécies da base de dados.

4.3 Base de dados multirótulos

Oito bases de dados multirótulos foram usadas nos experimentos, como apresentado na Tabela 4.2, e sendo divididas em cinco domínios: multimídia ('Emotions', 'Scene' e 'Mediamill'), biologia ('Yeast'), áudio ('Birds'), música ('Cal500') e texto ('Enron' e 'Medical'). Todas essas bases de dados estão disponíveis no repositório da Mulan[4].

4.4 Problemas de único rótulo

Os experimentos envolveram a comparação de métodos supervisionados e semisupervisionados em várias bases de dados e uma variedade de dimensões nos respectivos espaço de atributos. Entre as abordagens semisupervisionadas, comparamos todas as propostas baseadas em Floresta de Caminhos Ótimos: OPFSEMI [5], OPFSEMI$_{mst}$ [6], OPFSEMI$_{mst+knn}$ e demais técnicas tradicionais na área como: Transductive Support Vector Machines (TSVM) com Concave-Convex Optimization (CCP), implementado em UniverSVM [22], o método baseado em harmonic functions e Gaussian fields (implementado em SemiL [91]), a abordagem manifold regularization [10] com LapSVM[5] e a abordagem

[2]https://archive.ics.uci.edu/ml/datasets.html
[3]Estágio intermediário entre não-curtido e couro acabado.
[4]http://mulan.sourceforge.net/datasets-mlc.html
[5]http://manifold.cs.uchicago.edu/manifold_regularization/

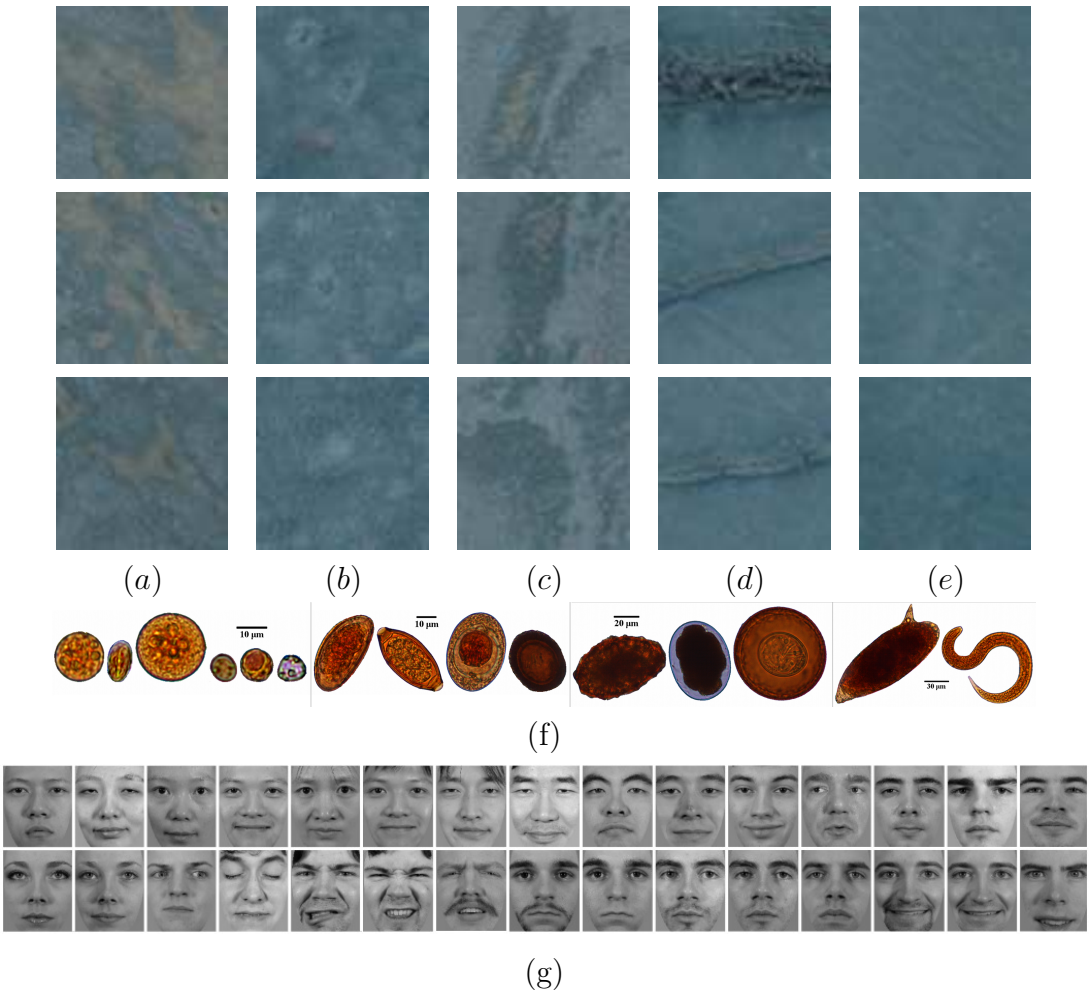


Figura 4.1: (a) Sarna, (b) Carrapato, (c) Marca-ferro, (d) Corte, (e) sem defeito, (f) exemplos de imagens de cada classe das estruturas de parasitas intestinais e (g) exemplo de imagens da base de dados Faces.

semisupervisionada usando Extreme Learning Machine [42] (SSELM[6]). Entre os métodos supervisionados comparamos com a versão mais popular dos classificadores supervisionados baseados em Floresta de Caminhos Ótimos, OPFSUP [66], e SVM com núcleo Radial Basis Function (RBF) [22] (também usado na implementação de UniverSVM).

Os experimentos foram executados sobre as bases de dados l_1 até l_8 (Tabela 4.1). Cada base de dados foi dividida aleatoriamente em duas partes: 70% conjunto de treinamento $\mathcal{Z}_1$ e 30% conjunto de teste $\mathcal{Z}_2$. Avaliamos também os métodos para diferentes proporções de amostras aleatórias selecionadas para o conjunto supervisionado $\mathcal{Z}_1^l$ e o conjunto não

[6]http://www.ntu.edu.sg/home/egbhuang/elm_codes.html

Tabela 4.2: Base de dados experimentais (ID - Identificador). Descrição dos problemas em termos de domínio de aplicação, número de amostras (#amostras), número de atributos (#atributos), número total de rótulos (l_n), cardinalidade do rótulo (l_c) e densidade do rótulo (l_d).

ID–**Base de dados**	**domínio**	#amostras	#atributos	l_n	l_c	l_d
d_1– Scene	Multimídia	2,047	294	6	1,074	0.179
d_2– Yeast	Biologia	2,417	103	14	4,237	0.303
d_3– Emotions	Multimídia	593	72	6	1,869	0.311
d_4– Mediamill	Multimídia	43,907	120	101	4,376	0.043
d_5– Birds	Áudio	645	260	19	1,014	0.053
d_6– Cal500	Música	502	68	174	26,044	0.150
d_7– Enron	Texto	1,702	1001	53	3,378	0.064
d_8– Medical	Texto	978	1,449	45	1,245	0.028

supervisionado $\mathcal{Z}_1^u$, sendo $\mathcal{Z}_1^l \cup \mathcal{Z}_1^u = \mathcal{Z}_1$, repetido 100 vezes, sendo que cada instância do experimento formado pelos conjuntos disjuntos $\mathcal{Z}_1^l \cap \mathcal{Z}_1^u \cap \mathcal{Z}_2 = \emptyset$. Os tamanhos de $\mathcal{Z}_1^l$ e $\mathcal{Z}_1^u$ variam de 1%–99% e 10%–90% para 50%–50% referente ao tamanho de $\mathcal{Z}_1$. As abordagens supervisionadas foram treinadas usando $\mathcal{Z}_1^l$, e os classificadores testados em $\mathcal{Z}_2$, enquanto que os métodos semisupervisionados primeiramente propagam os rótulos de $\mathcal{Z}_1^l$ para $\mathcal{Z}_1^u$, treinam sobre $\mathcal{Z}_1$, e por fim os classificadores são testados em $\mathcal{Z}_2$. O desempenho dos classificadores em acurácia, segue o mesmo formato de medida sugerido por Papa el. al [69].

Para a análise estatística foi aplicado o teste de Friedman [32] sobre os resultados, fortemente recomendado por Demšar [26] quando o número de algoritmos comparados seja > 5. O teste de Friedman é um teste não-paramétrico para testar as diferenças entre os vários classificadores, sendo uma alternativa para a análise de variância, utilizada quando o mesmo parâmetro foi medido sob condições diferentes. Quando a diferença de desempenho é estatisticamente significativa (ou seja, a hipótese nula foi rejeitada), o próximo passo é um teste post-hoc para detectar quais algoritmos apresentam essas diferenças. Para esse caso, adotamos o teste post-hoc Nemenyi [26]. A diferença de desempenho para dois classificadores é considerada estatisticamente significativa quando a média dos seus ranks (melhor desempenho é atribuída a primeira posição, para o segundo melhor a posição dois e assim em diante) diferem em mais de uma distância crítica.

4.4.1 Otimização dos parâmetros

TSVM introduz vários hiper-parâmetros que precisam ser ajustados. Em nossos experimentos, para cada instância dos experimentos aplicamos validação cruzada em 5 partições sobre o conjunto de treinamento, e selecionamos a configuração com melhor desempenho para validação do conjunto de teste, através dos parâmetros $C \in \{10^{-5}, 10^{-3}, 10^{-1}, 10, 10^{3}, 10^{5}\}$ e $C^{*} \in \{10^{-5}, 10^{-3}, 10^{-1}, 0, 10\}$ — ou seja, os parâmetros que regulam a margem de erro de treinamento dos dados supervisionados e não supervisionados, respectivamente. Também utilizamos o núcleo RBF tanto para SVM, TSVM e LapSVM com $\gamma \in \{10^{-5}, 10^{-3}, 10^{-1}, 1, 10\}$. Para SSELM, nós usamos a função *sigmoid* e o número de neurônios escondidos foi fixado em 2000. Os parâmetros para C e γ foram selecionados a partir da sequência exponencial $\{10^{-6}, 10^{-5}, ..., 10^{6}\}$ (conjunto de valores propostos por [42]). Em SemiL, o peso das matrizes W foram calculados com duas funções de distância diferentes, distância euclidiana e cosseno, núcleo RBF e usando abordagem *hard-label* com smoothness maximization (Gaussian Random Field Model - GRFM), parâmetros fortemente recomendados por [44]. Para escolhermos o valor de k referente a OPFSEMI$_{mst+knn}$, seguimos a proposta apresentada na Seção 3.3, selecionando o maior valor de k que mantenha o percentual de discordância de rotulação L_1 e L_2 menor ou igual a $\mathcal{E}$, sendo ($k_{\max}$=$|\mathcal{Z}_1|$ até $k_{\min}$=$|1|$). Os parâmetros restantes utilizados foram o padrão pelas ferramentas.

4.4.2 Resultados e análise estatística

Os resultados disponíveis nas Tabelas 4.3–4.10 são apresentados da seguinte forma: $a \pm b$, onde a e b são, respectivamente, a acurácia média e seu desvio padrão para cada classificador em $\mathcal{Z}_2$. O tempo de treinamento, *tt.*, em segundos e a porcentagem de erro de propagação dos rótulos em $\mathcal{Z}_1^u$, $\mathcal{E}$, também são apresentados para OPFSEMI, OPFSEMI$_{mst}$ e OPFSEMI$_{mst+knn}$. Além disso, iremos apresentar o melhor k obtido para cada base de dados usando OPFSEMI$_{mst+knn}$. Os melhores resultados entre os métodos de ambas as tabelas são exibidas em negrito.

De acordo com o teste de Friedman [32] (usando a média de acurácia de todos os métodos e proporções de dados supervisionados e não supervisionados em conjunto), os resultados apresentados nas Tabelas 4.3–4.10 rejeitam a hipótese *nula* de que todos os classificadores sejam equivalentes. Portanto, as Figuras 4.2–4.3 apresentam uma representação gráfica do teste Nemenyi, em que o valor 1 representa a melhor técnica, enquanto 8 representa a pior. Grupos de classificadores que são considerados equivalentes (nível de significância $p = 0.05$) são conectados usando uma distância crítica calculada (CD) igual a 4.2863 (Figura 4.2). Apenas no caso da Figura 4.3 (com relação à análise de todas as base de dados em conjunto), obtém uma diferente distância crítica (CD) igual a 1.5154

Tabela 4.3: Resultados $a \pm b$ em $\mathcal{Z}_2$: a - acurácia média (%), b - desvio padrão, tempo de treinamento (*tt.*) em segundos, porcentagem de erro de propagação dos rótulos ($\mathcal{E}$) em $\mathcal{Z}_1^u$ – base de dados Cowhide.

$\mathcal{Z}_1^l$	$\mathcal{Z}_1^u$	**OPFSUP**	**SVM**	**TSVM**	**SemiL**	**LapSVM**	**SSELM**
1%	99%	82.71±0.038	**85.19±0.035**	83.14±0.085	83.81±0.007	83.44±0.007	80.64±0.083
10%	90%	90.88±0.107	85.55±0.033	84.25±0.041	81.61±0.016	92.40±0.009	84.87±0.046
20%	80%	93.19±0.038	85.86±0.085	89.48±0.019	80.14±0.012	92.06±0.050	87.09±0.042
30%	70%	93.59±0.018	86.45±0.050	86.48±0.059	81.58±0.042	94.38±0.043	90.32±0.087
40%	60%	94.29±0.031	86.09±0.083	86.68±0.097	82.63±0.075	95.43±0.087	93.54±0.016
50%	50%	95.77±0.069	87.76±0.040	90.04±0.057	91.36±0.096	96.01±0.064	**96.77±0.076**

$\mathcal{Z}_1^l$	$\mathcal{Z}_1^u$	**OPFSEMI**	$\mathcal{E}$	*tt.*	**OPFSEMI**$_{mst}$	$\mathcal{E}$	*tt.*	**OPFSEMI**$_{mst+knn}$	$\mathcal{E}$	*tt.*
1%	99%	84.54±0.076	24.84	0.282	84.54±0.069	24.84	0.093	84.76±0.083(k^*=6)	23.80	0.403
10%	90%	91.69±0.040	12.01	0.286	92.44±0.015	10.41	0.094	**92.69±0.089**(k^*=8)	9.86	0.441
20%	80%	93.75±0.008	9.50	0.296	94.79±0.053	8.76	0.096	**94.84±0.093**(k^*=9)	8.24	0.439
30%	70%	94.40±0.076	9.52	0.296	95.82±0.053	6.75	0.094	**95.93±0.073**(k^*=9)	6.84	0.469
40%	60%	94.76±0.092	8.59	0.311	**96.16±0.016**	5.49	0.096	95.82±0.013(k^*=5)	6.46	0.488
50%	50%	95.80±0.107	4.56	0.296	96.68±0.069	3.89	0.090	96.59±0.087(k^*=6)	4.31	0.476

Tabela 4.4: Resultados $a \pm b$ em $\mathcal{Z}_2$: a - acurácia média (%), b - desvio padrão, tempo de treinamento (*tt.*) em segundos, porcentagem de erro de propagação dos rótulos ($\mathcal{E}$) em $\mathcal{Z}_1^u$ – base de dados Statlog.

$\mathcal{Z}_1^l$	$\mathcal{Z}_1^u$	**OPFSUP**	**SVM**	**TSVM**	**SemiL**	**LapSVM**	**SSELM**
1%	99%	84.75±0.061	81.45±0.022	73.61±0.013	78.62±0.077	82.53±0.065	82.05±0.024
10%	90%	88.76±0.031	85.41±0.028	88.68±0.081	87.20±0.111	89.70±0.093	85.18±0.032
20%	80%	87.11±0.084	90.33±0.054	85.05±0.042	83.80±0.056	90.04±0.013	89.07±0.068
30%	70%	91.6±0.084	92.33±0.088	89.13±0.048	80.22±0.048	92.21±0.054	90.86±0.010
40%	60%	92.54±0.038	93.20±0.033	89.22±0.015	88.38±0.071	93.56±0.043	91.20±0.012
50%	50%	93.14±0.051	93.25±0.079	89.01±0.067	90.29±0.080	93.99±0.097	94.84±0.053

$\mathcal{Z}_1^l$	$\mathcal{Z}_1^u$	**OPFSEMI**	$\mathcal{E}$	*tt.*	**OPFSEMI**$_{mst}$	$\mathcal{E}$	*tt.*	**OPFSEMI**$_{mst+knn}$	$\mathcal{E}$	*tt.*
1%	99%	85.56±0.013	29.10	0.294	85.72±0.053	29.10	0.093	**85.96±0.076**(k^*=20)	28.97	0.485
10%	90%	91.73±0.107	25.01	0.299	91.76±0.044	18.78	0.094	**91.80±0.088**(k^*=22)	18.19	0.487
20%	80%	91.94±0.099	24.83	0.293	**93.11±0.099**	16.80	0.091	92.83±0.089(k^*=14)	17.02	0.518
30%	70%	91.91±0.046	21.76	0.312	93.85±0.114	15.38	0.095	**93.99±0.047**(k^*=10)	15.13	0.491
40%	60%	93.15±0.081	17.81	0.310	94.47±0.031	15.09	0.095	**94.49±0.096**(k^*=10)	13.43	0.467
50%	50%	93.96±0.014	16.71	0.316	95.21±0.015	14.11	0.091	**95.29±0.054**(k^*=8)	12.95	0.469

Tabela 4.5: Resultados $a \pm b$ em $\mathcal{Z}_2$: a - acurácia média (%), b - desvio padrão, tempo de treinamento (*tt.*) em segundos, porcentagem de erro de propagação dos rótulos ($\mathcal{E}$) em $\mathcal{Z}_1^u$ – base de dados Faces.

$\mathcal{Z}_1^l$	$\mathcal{Z}_1^u$	**OPFSUP**	**SVM**	**TSVM**	**SemiL**	**LapSVM**	**SSELM**
1%	99%	80.77±0.089	75.13±0.019	68.45±0.070	79.25±0.064	80.12±0.014	79.62±0.095
10%	90%	85.47±0.038	71.31±0.051	77.61±0.051	83.81±0.021	91.06±0.005	81.48±0.094
20%	80%	92.95±0.072	80.81±0.002	84.38±0.088	84.37±0.082	92.31±0.005	85.63±0.096
30%	70%	95.43±0.015	82.24±0.017	80.34±0.052	87.49±0.041	95.28±0.084	90.74±0.036
40%	60%	97.15±0.13	90.24±0.013	84.35±0.036	90.38±0.044	96.63±0.017	92.59±0.042
50%	50%	97.48±0.023	97.13±0.008	90.04±0.059	93.61±0.094	98.14±0.060	96.28±0.071

$\mathcal{Z}_1^l$	$\mathcal{Z}_1^u$	**OPFSEMI**	$\mathcal{E}$	*tt.*	**OPFSEMI**$_{mst}$	$\mathcal{E}$	*tt.*	**OPFSEMI**$_{mst+knn}$	$\mathcal{E}$	*tt.*
1%	99%	88.62±0.046	23.21	0.556	89.15±0.053	23.21	0.181	**89.36±0.052**(k^*=18)	23.06	1.027
10%	90%	91.75±0.084	15.22	0.552	93.35±0.031	13.86	0.180	**94.40±0.067**(k^*=16)	12.46	1.015
20%	80%	94.75±0.042	9.57	0.558	**96.12±0.015**	6.96	0.182	95.80±0.092(k^*=11)	7.22	0.957
30%	70%	97.02±0.046	4.26	0.562	**98.14±0.038**	2.40	0.180	97.83±0.046(k^*=9)	4.23	0.976
40%	60%	97.79±0.061	3.16	0.581	98.38±0.137	2.15	0.182	**98.45±0.022**(k^*=7)	1.79	1.089
50%	50%	97.61±0.015	3.11	0.617	98.61±0.053	1.67	0.187	**98.79±0.011**(k^*=6)	1.51	1.030

Tabela 4.6: Resultados $a \pm b$ em $\mathcal{Z}_2$: a - acurácia média (%), b - desvio padrão, tempo de treinamento (*tt.*) em segundos, porcentagem de erro de propagação dos rótulos ($\mathcal{E}$) em $\mathcal{Z}_1^u$ – base de dados Parasites.

$\mathcal{Z}_1^l$	$\mathcal{Z}_1^u$	**OPFSUP**	**SVM**	**TSVM**	**SemiL**	**LapSVM**	**SSELM**
1%	99%	88.09±0.092	78.15±0.035	71.78±0.016	82.52±0.058	82.56±0.051	82.44±0.059
10%	90%	96.11±0.015	94.45±0.029	91.84±0.108	88.25±0.012	92.28±0.026	89.36±0.040
20%	80%	97.26±0.084	98.56±0.047	89.32±0.007	84.33±0.073	94.93±0.041	90.42±0.069
30%	70%	98.00±0.114	98.41±0.068	94.22±0.035	88.11±0.111	94.85±0.064	94.14±0.020
40%	60%	97.93±0.039	97.79±0.013	95.85±0.061	86.66±0.019	95.67±0.034	95.21±0.042
50%	50%	98.42±0.031	98.88±0.040	94.56±0.008	93.42±0.064	96.52±0.009	97.87±0.070

$\mathcal{Z}_1^l$	$\mathcal{Z}_1^u$	**OPFSEMI**	$\mathcal{E}$	*tt.*	**OPFSEMI**$_{mst}$	$\mathcal{E}$	*tt.*	**OPFSEMI**$_{mst+knn}$	$\mathcal{E}$	*tt.*
1%	99%	91.94±0.019	13.03	0.902	92.01±0.053	13.03	0.301	**92.05±0.090**(k^*=25)	11.58	1.447
10%	90%	97.85±0.094	4.56	0.946	97.94±0.038	4.56	0.311	**98.09±0.056**(k^*=22)	4.17	1.465
20%	80%	97.69±0.023	4.40	0.909	97.82±0.046	3.76	0.298	**97.88±0.025**(k^*=20)	3.19	1.480
30%	70%	98.36±0.023	3.14	0.949	98.43±0.094	3.14	0.306	**98.44±0.065**(k^*=20)	2.80	1.508
40%	60%	98.43±0.069	3.31	0.893	98.45±0.094	3.31	0.280	**98.56±0.037**(k^*=16)	2.68	1.580
50%	50%	98.79±0.015	2.64	0.916	98.85±0.084	1.90	0.281	**98.91±0.073**(k^*=15)	1.79	1.559

Tabela 4.7: Resultados $a \pm b$ em $\mathcal{Z}_2$: a - acurácia média (%), b - desvio padrão, tempo de treinamento (*tt.*) em segundos, porcentagem de erro de propagação dos rótulos ($\mathcal{E}$) em $\mathcal{Z}_1^u$ – base de dados Spambase.

$\mathcal{Z}_1^l$	$\mathcal{Z}_1^u$	**OPFSUP**	**SVM**	**TSVM**	**SemiL**	**LapSVM**	**SSELM**
1%	99%	58.76±0.042	60.75±0.026	60.37±0.082	58.59±0.085	65.12±0.028	61.43±0.045
10%	90%	65.89±0.017	64.15±0.066	**67.98±0.069**	62.20±0.102	66.29±0.037	62.75±0.088
20%	80%	66.13±0.038	**74.75±0.031**	70.13±0.082	66.31±0.078	72.87±0.074	66.89±0.096
30%	70%	67.54±0.096	**76.39±0.054**	73.41±0.014	70.23±0.066	76.01±0.028	69.07±0.022
40%	60%	68.99±0.044	**76.95±0.075**	69.84±0.007	71.34±0.014	75.35±0.081	72.11±0.031
50%	50%	70.16±0.099	77.48±0.056	71.03±0.013	71.85±0.099	**78.06±0.016**	73.91±0.030

$\mathcal{Z}_1^l$	$\mathcal{Z}_1^u$	**OPFSEMI**	$\mathcal{E}$	*tt.*	**OPFSEMI**$_{mst}$	$\mathcal{E}$	*tt.*	**OPFSEMI**$_{mst+knn}$	$\mathcal{E}$	*tt.*
1%	99%	64.78±0.053	35.07	1.890	65.22±0.061	33.49	0.607	**65.81±0.063**(k^*=42)	32.84	3.075
10%	90%	65.90±0.027	32.22	1.899	66.12±0.056	30.11	0.605	66.01±0.052(k^*=38)	28.12	3.248
20%	80%	67.25±0.092	30.73	1.950	68.69±0.023	27.32	0.619	68.54±0.091(k^*=19)	26.89	3.148
30%	70%	68.46±0.062	30.42	1.998	71.53±0.021	25.43	0.621	72.15±0.056(k^*=11)	24.32	3.161
40%	60%	69.55±0.058	29.02	2.071	73.42±0.061	24.13	0.690	73.48±0.013(k^*=11)	22.63	3.104
50%	50%	70.81±0.067	27.74	2.133	74.18±0.037	19.25	0.693	74.52±0.038(k^*=9)	18.36	3.266

Tabela 4.8: Resultados $a \pm b$ em $\mathcal{Z}_2$: a - acurácia média (%), b - desvio padrão, tempo de treinamento (*tt.*) em segundos, porcentagem de erro de propagação dos rótulos ($\mathcal{E}$) em $\mathcal{Z}_1^u$ – base de dados Pendigits.

$\mathcal{Z}_1^l$	$\mathcal{Z}_1^u$	**OPFSUP**	**SVM**	**TSVM**	**SemiL**	**LapSVM**	**SSELM**
1%	99%	94.31±0.076	75.05±0.025	74.23±0.088	75.20±0.015	93.05±0.077	94.16±0.042
10%	90%	96.54±0.015	70.27±0.064	70.43±0.083	74.50±0.035	97.20±0.089	96.72±0.083
20%	80%	98.88±0.092	79.07±0.028	76.80±0.094	86.60±0.076	97.49±0.055	97.53±0.094
30%	70%	99.19±0.099	87.68±0.078	88.57±0.078	97.61±0.010	97.93±0.001	98.90±0.020
40%	60%	99.14±0.053	91.22±0.035	82.88±0.023	97.13±0.098	98.05±0.055	99.01±0.024
50%	50%	99.17±0.088	97.87±0.020	85.65±0.047	98.06±0.026	98.29±0.063	99.17±0.089

$\mathcal{Z}_1^l$	$\mathcal{Z}_1^u$	**OPFSEMI**	$\mathcal{E}$	*tt.*	**OPFSEMI**$_{mst}$	$\mathcal{E}$	*tt.*	**OPFSEMI**$_{mst+knn}$	$\mathcal{E}$	*tt.*
1%	99%	95.66±0.015	7.58	6.957	95.78±0.084	7.54	2.141	**95.83±0.079**(k^*=37)	6.55	10.547
10%	90%	98.27±0.053	5.89	7.168	**99.22±0.031**	1.05	2.212	**99.22±0.069**(k^*=37)	0.91	10.749
20%	80%	99.10±0.076	1.33	7.084	99.30±0.069	0.95	2.162	**99.32±0.083**(k^*=26)	0.89	10.950
30%	70%	99.28±0.046	1.07	7.214	**99.44±0.015**	0.85	2.155	99.35±0.027(k^*=13)	0.89	11.745
40%	60%	98.56±0.033	2.62	7.328	**99.44±0.015**	0.73	2.151	99.42±0.033(k^*=10)	0.80	11.231
50%	50%	98.61±0.053	2.67	7.469	99.51±0.071	0.57	2.142	**99.60±0.011**(k^*=8)	0.55	11.821

Tabela 4.9: Resultados $a \pm b$ em $\mathcal{Z}_2$: a - acurácia média (%), b - desvio padrão, tempo de treinamento (*tt.*) em segundos, porcentagem de erro de propagação dos rótulos ($\mathcal{E}$) em $\mathcal{Z}_1^u$ – base de dados KddCup.

$\mathcal{Z}_1^l$	$\mathcal{Z}_1^u$	**OPFSUP**	**SVM**	**TSVM**	**SemiL**	**LapSVM**	**SSELM**
1%	99%	89.23±0.033	80.44±0.029	88.15±0.063	73.92±0.040	85.48±0.052	79.15±0.030
10%	90%	89.67±0.088	87.01±0.038	90.12±0.011	83.57±0.021	89.8±0.065	83.36±0.053
20%	80%	92.99±0.072	86.91±0.071	91.54±0.081	84.35±0.039	**93.16±0.098**	88.14±0.043
30%	70%	93.57±0.092	87.30±0.048	92.69±0.093	87.66±0.042	93.37±0.081	90.74±0.077
40%	60%	93.33±0.048	87.14±0.034	92.99±0.076	90.54±0.028	94.55±0.059	92.32±0.041
50%	50%	94.84±0.086	91.76±0.092	94.76±0.084	92.28±0.060	95.98±0.052	94.47±0.086

$\mathcal{Z}_1^l$	$\mathcal{Z}_1^u$	**OPFSEMI**	$\mathcal{E}$	*tt.*	**OPFSEMI**$_{mst}$	$\mathcal{E}$	*tt.*	**OPFSEMI**$_{mst+knn}$	$\mathcal{E}$	*tt.*
1%	99%	85.57±0.021	0.77	353.842	**90.06±0.021**	0.53	142.629	89.52±0.052(k^*=5)	0.60	402.993
10%	90%	90.53±0.025	0.24	346.997	**90.64±0.057**	0.24	141.964	90.58±0.012(k^*=7)	0.25	421.735
20%	80%	92.78±0.068	0.16	350.471	93.02±0.029	0.14	140.428	93.12±0.099(k^*=7)	0.16	413.255
30%	70%	93.69±0.014	0.13	357.392	93.73±0.026	0.12	140.857	**93.86±0.068**(k^*=6)	0.12	401.142
40%	60%	93.71±0.024	0.15	367.103	93.71±0.083	0.12	141.467	**94.97±0.062**(k^*=4)	0.10	407.327
50%	50%	95.60±0.036	0.15	381.697	96.21±0.092	0.09	141.231	**97.05±0.087**(k^*=4)	0.10	405.503

Tabela 4.10: Resultados $a \pm b$ em $\mathcal{Z}_2$: a - acurácia média (%), b - desvio padrão, tempo de treinamento (*tt.*) em segundos, porcentagem de erro de propagação dos rótulos ($\mathcal{E}$) em $\mathcal{Z}_1^u$ – base de dados Letter.

$\mathcal{Z}_1^l$	$\mathcal{Z}_1^u$	**OPFSUP**	**SVM**	**TSVM**	**SemiL**	**LapSVM**	**SSELM**
1%	99%	75.05±0.052	78.68±0.070	74.56±0.017	73.03±0.083	**80.97±0.037**	76.62±0.051
10%	90%	88.76±0.069	80.82±0.016	79.23±0.084	76.41±0.082	82.69±0.031	77.53±0.018
20%	80%	92.94±0.089	84.56±0.021	84.06±0.059	78.68±0.098	84.73±0.010	82.08±0.063
30%	70%	94.17±0.076	89.25±0.035	90.11±0.049	91.95±0.046	92.31±0.060	90.71±0.011
40%	60%	94.33±0.058	92.98±0.055	93.42±0.076	90.82±0.092	94.16±0.055	93.55±0.078
50%	50%	95.43±0.077	93.58±0.024	94.28±0.073	91.56±0.032	96.50±0.088	95.11±0.060

$\mathcal{Z}_1^l$	$\mathcal{Z}_1^u$	**OPFSEMI**	$\mathcal{E}$	*tt.*	**OPFSEMI**$_{mst}$	$\mathcal{E}$	*tt.*	**OPFSEMI**$_{mst+knn}$	$\mathcal{E}$	*tt.*
1%	99%	76.94±0.059	32.79	45.612	77.50±0.084	30.58	15.279	77.60±0.021(k^*=20)	30.31	51.689
10%	90%	88.11±0.083	20.51	47.494	**91.38±0.011**	15.67	15.419	91.16±0.032(k^*=18)	16.88	50.753
20%	80%	89.44±0.091	17.99	47.539	**94.28±0.088**	10.31	15.351	94.01±0.024(k^*=11)	12.21	52.638
30%	70%	91.45±0.046	14.19	48.453	95.21±0.034	8.77	15.296	**95.91±0.032**(k^*=11)	8.22	53.038
40%	60%	90.94±0.012	15.12	51.003	95.87±0.031	7.54	15.743	**96.17±0.080**(k^*=9)	6.32	53.692
50%	50%	93.01±0.053	11.10	51.757	96.17±0.012	7.12	15.658	**96.85±0.022**(k^*=6)	5.78	53.088

(nível de significância $p = 0.05$). Vale a pena notar a importância do teste estatístico, uma vez que os valores médios em alguns casos, não são suficientes para indicar o melhor classificador.

O teste apresentou como os melhores resultados OPFSEMI$_{mst}$ e OPFSEMI$_{mst+knn}$, seguidos por OPFSEMI, LapSVM, OPFSUP, SVM, SSELM e, finalmente, TSVM e SemiL. Devemos destacar o bom desempenho de LapSVM e SVM na base de dados Spambase (Figura 4.2e), sendo que na maioria dos casos LapSVM produziu os melhores resultados de classificação sobre SVM. Outra interessante observação foi que SVM superou sua versão semisupervisionado TSVM, diferente assim de OPFSEMI$_{mst}$ em relação a OPFSUP em qualquer dos casos analisados. Embora o desempenho de SSELM tenha sido inferior ao de OPFSEMI$_{mst}$, SSELM superou TSVM e SemiL na maioria dos casos analisados. Ambos TSVM e SemiL apresentaram desempenho inferior a OPFSEMI$_{mst}$ e OPFSEMI$_{mst+knn}$ nas bases de dados Cowhide e Parasites, mesmo quando usado apenas 1% de amostras supervisionadas.

Realizando uma análise mais aprofundada, podemos comparar estatisticamente os pares de classificadores, OPFSEMI com OPFSEMI$_{mst}$ e OPFSEMI$_{mst}$ com OPFSEMI$_{mst+knn}$, usando Wilcoxon signed rank test [26]. O teste de Wilcoxon é uma análise de grande importância pela capacidade de detectar diferenças mais sensíveis uma vez que não assume distribuição normal. Neste caso, chegamos para o par de classificadores OPFSEMI e OPFSEMI$_{mst}$ no resultado de análise com $p = 3.640^{-9}$, sendo ($p < 0.05$). No caso de OPFSEMI$_{mst}$ com OPFSEMI$_{mst+knn}$, conseguimos um valor de $p = 0.01827$, também sendo ($p < 0.05$), isso confirma que os classificadores são estatísticamente diferentes.

Podemos confirmar também a melhoria de OPFSEMI$_{mst}$ sobre sua versão anterior, em acurácia e eficiência, e superioridade de OPFSEMI$_{mst+knn}$ sobre as demais técnicas, apesar do maior custo computacional. OPFSEMI$_{mst}$ foi em média, três vezes mais rápido do que OPFSEMI para o treinamento e também demonstrado ser robusto na diminuição de erros de propagação de rótulo sobre $\mathcal{Z}_1^u$, variando entre 0.11% até 33.49%. Mas apesar dos melhores resultados por OPFSEMI$_{mst+knn}$, houve uma diferença mínima em OPFSEMI$_{mst}$, e em outras situações com melhores resultados obtidos. Notamos também a variação do valor de k para OPFSEMI$_{mst+knn}$. À medida que o conjunto de dados rotulados aumenta, tende a obter um aumento do valor k. Mas em alguns casos o valor k, diminui nas mesmas condições. Isso pode ser devido ao aumento da base de dados supervisionada, que pode gerar um maior conhecimento sobre os dados, diminuindo assim o erro $\mathcal{E}$, e consequentemente, um menor valor de k irá atingir esse erro de maneira mais rápida. É importante também ressaltar o custo computacional inserido na nova abordagem. Este tipo de critério de avaliação deverá ser aceito quando houver a necessidade da escolha da melhor abordagem semisupervisionada a ser utilizada, e se a perda de eficiência é compensada pelo ganho de precisão ou não.

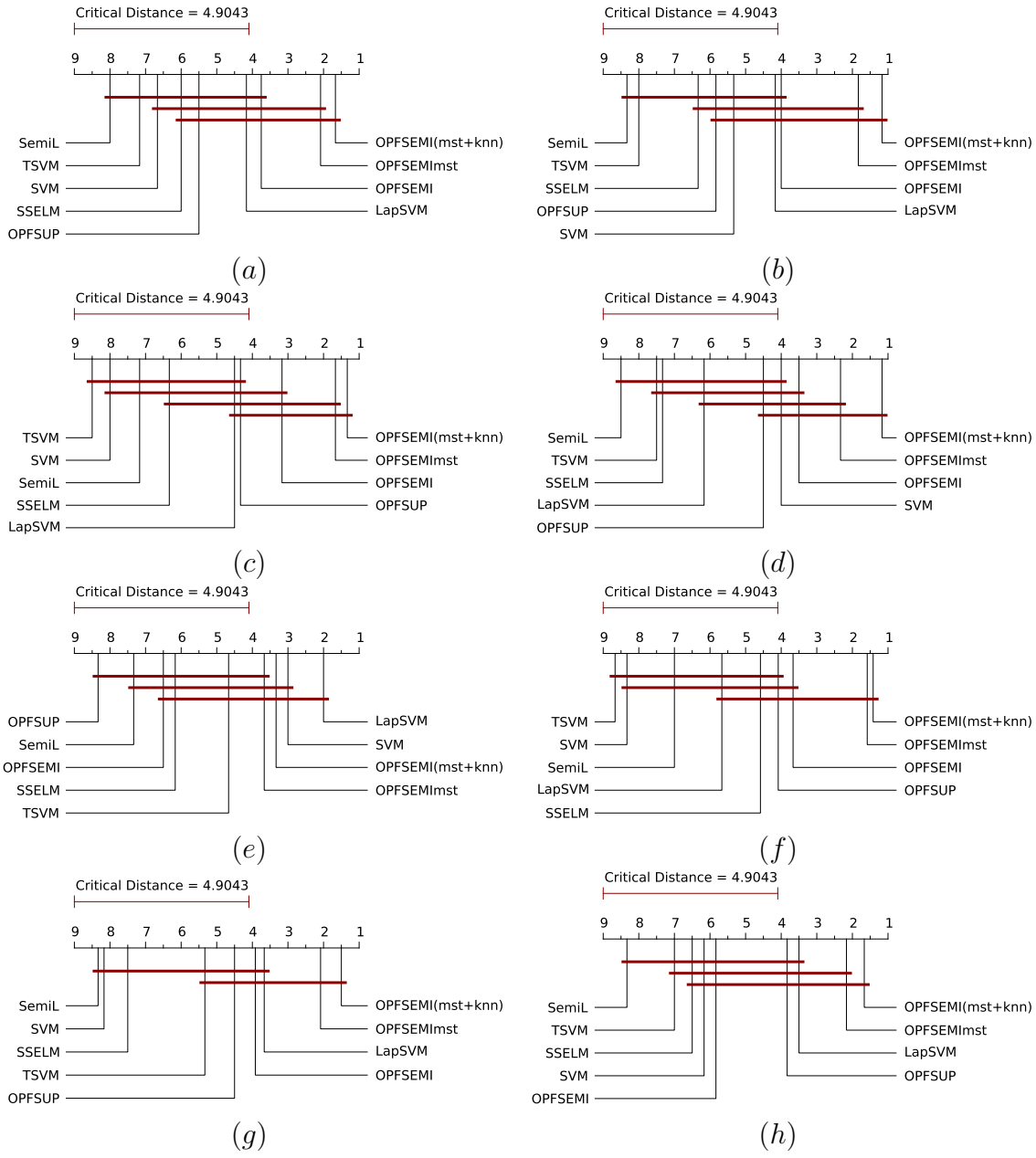


Figura 4.2: Resultados dos testes estatísticos usando Nemenyi para todos os classificadores. Grupos de classificadores equivalentes estão conectados em $p = 0.05$. (a) Cowhide, (b) Statlog, (c) Faces, (d) Parasites, (e) Spambase, (f) Pendigits, (g) KddCup e (h) Letter.

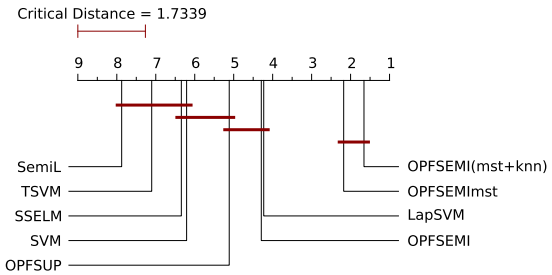


Figura 4.3: Resultados dos testes estatísticos usando Nemenyi para todos os classificadores e sobre todas as bases de dados. Grupos de classificadores equivalentes estão conectados em $p = 0.05$.

4.5 Problemas multirótulos

Nesta seção, apresentamos uma análise experimental com objetivo de avaliar a proposta OPFSEMI$_{mst+knn}$ no contexto multirótulos em comparação com as técnicas OPFSEMI$_{mst}$ [6], TSVM [50] e a abordagem manifold regularization [10] com LapSVM[7]. Para avaliação da tarefa de classificação multirótulo, serão utilizadas quatro metodologias de transformação de dados: Label Powerset (LP), Binary Relevance (BR), Classifier Chains (CC) e Hierarchy of Multi-label Classifiers (HOMER). Também foram avaliados dois métodos de adaptação usando a estratégia de auto formação para propagação dos rótulos para as amostras não supervisionadas, que são: BPMLL e MLkNN, todos implementados e disponíveis na biblioteca Mulan Java[8]. Cada experimento foi repetido 100 vezes com conjuntos gerados de maneira aleatória, divididos em duas partes: 70% para o conjunto de treinamento $\mathcal{Z}_1$ e 30% para o conjunto de teste $\mathcal{Z}_2$. Também avaliamos duas diferentes proporções entre os tamanhos dos conjuntos supervisionados $\mathcal{Z}_1^l$ (10% e 50%) e não supervisionados $\mathcal{Z}_1^u$ (90% e 50%), sendo que cada instância do experimento formado pelos conjuntos disjuntos $\mathcal{Z}_1^l \cap \mathcal{Z}_1^u \cap \mathcal{Z}_2 = \emptyset$. Os resultados foram avaliados por meio das medidas F-measure e Hamming Loss, ambas medidas fortemente sugeridas por trabalhos multirótulos [61]. Oito base de dados multirótulos[9] foram utilizadas nos experimentos, conforme apresentado na Tabela 4.1.

4.5.1 Otimização dos parâmetros

Em relação a TSVM, usamos SVMLight [50]. Em ambas as implementações (TSVM e LapSVM), consideramos o núcleo RBF, sendo seus parâmetros otimizados por va-

[7] http://manifold.cs.uchicago.edu/manifold_regularization/
[8] http://mulan.sourceforge.net/
[9] http://mulan.sourceforge.net/datasets-mlc.html

lidação cruzada em 5 partições sobre o conjunto de treinamento, e selecionamos a configuração com melhor desempenho para validação do conjunto de teste, sendo $C \in \{10^{-5}, 10^{-3}, 10^{-1}, 10, 10^{3}, 10^{5}\}$ e $\gamma \in \{10^{-5}, 10^{-3}, 10^{-1}, 1, 10\}$. Em relação à metodologia de transformação HOMER, avaliamos 5 números diferentes de clusters, $(2-6)$ e selecionamos o melhor valor para construção da hierarquia de classificadores multirótulo. Para escolhermos o valor de k referente a OPFSEMI$_{mst+knn}$, seguimos a proposta apresentada na Seção 3.3, selecionando o maior de valor de k que mantenha o percentual de discordância de rotulação L_1 e L_2 menor ou igual a $\mathcal{E}$, sendo ($k_{\max}$=$|\mathcal{Z}_1|$ até $k_{\min}$=$|1|$). Os parâmetros restantes utilizados foram o padrão pelas ferramentas.

4.5.2 Resultados e análise estatística

As Tabelas (4.11–4.14) e (4.15–4.18) apresentam o desempenho da classificação de acordo com o seguinte formato $a \pm b$, onde a e b denotam, respectivamente, Tabelas (4.11–4.14) a média de F-measure e seu desvio padrão e Tabelas (4.15–4.18) o valor Hamming Loss e desvio padrão. Finalmente, Tabela 4.19 mostra a média F-measure e Hamming Loss e seu desvio padrão para as técnicas MLkNN e BPMLL. Os valores em negrito indicam os melhores resultados por medida de avaliação, considerando os valores (base de dados, porcentagem de $\mathcal{Z}_1^l$ e $\mathcal{Z}_1^u$, método de transformação ou método de adaptação). Por exemplo, OPFSEMI$_{mst+knn}$ (usando F-measure) foi a melhor técnica usando a base de dados Scene (base de dados com identificador - d_1) com HOMER como método de transformação usando 50% de $\mathcal{Z}_1$ para $\mathcal{Z}_1^l$. As Tabelas (4.20–4.21) apresentam a porcentagem de erro de propagação sobre $\mathcal{Z}_1^u$ ($\mathcal{E}$) para OPFSEMI$_{mst+knn}$ e OPFSEMI$_{mst}$, além também do melhor valor de k^* obtido para cada base de dados usando OPFSEMI$_{mst+knn}$.

Em geral, os melhores resultados foram obtidos com os métodos de transformação LP/HOMER usando o classificador OPFSEMI$_{mst+knn}$. A conectividade ótima entre amostras supervisionadas e não supervisionadas permite um desempenho considerável para OPFSEMI$_{mst+knn}$, o que torna sua generalização melhor do que LapSVM e TSVM ao capturar as formas das classes no espaço de atributos. As principais vantagens da nossa abordagem incluem a simplicidade e a utilização eficaz dos dados supervisionados e não supervisionados.

Conceitualmente, uma boa condição para OPFSEMI$_{mst+knn}$ e OPFSEMI$_{mst}$ [6] devem revelar a verdadeira complexidade intrínseca ou dimensionalidade dos pontos de dados (relacionamentos locais lineares), e também capturar certas estruturas globais dos dados como um todo (ou seja, grupos ou subespaços), mesmo após a transformação dos dados para os problemas de único rótulo. Uma possível lacuna refere-se a situações que não asseguram o critério da suavidade entre as classes, ou quando temos dados supervisionados irrelevantes (por exemplo, as amostras rotuladas de forma errada devido a erros

humanos, o que pode ser um problema em estudos de larga escala). Isso pode prejudicar a propagação de rótulo para as amostras não supervisionadas, e essas informações podem não representar a relação real entre as classes, tornando ainda piores os resultados da classificação em relação ao caso que usa apenas dados supervisionados.

Tabela 4.11: F-measure considerando OPFSEMI$_{mst+knn}$.

	$\mathcal{Z}_1^l$	$\mathcal{Z}_1^u$	**OPFSEMI**$_{mst+knn}$			
			LP	BR	CC	HOMER
d_1	10%	90%	0.6100±0.069	0.5838±0.080	0.5928±0.037	0.5765±0.027
	50%	50%	0.6659±0.042	0.6479±0.074	0.6479±0.083	**0.7086±0.049**
d_2	10%	90%	**0.6474±0.051**	0.5850±0.055	0.6168±0.094	0.5837±0.061
	50%	50%	0.6529±0.098	0.5999±0.090	0.6357±0.092	**0.6695±0.057**
d_3	10%	90%	**0.6016±0.047**	0.5977±0.072	0.5714±0.052	0.5331±0.014
	50%	50%	0.6299±0.097	0.5966±0.043	0.6375±0.055	**0.6516±0.043**
d_4	10%	90%	0.4560±0.028	0.4434±0.043	0.4430±0.065	**0.4752±0.043**
	50%	50%	**0.5326±0.059**	0.4859±0.037	0.4870±0.026	0.5158±0.075
d_5	10%	90%	**0.5205±0.032**	0.5131±0.041	0.5198±0.044	0.4900±0.048
	50%	50%	**0.6144±0.027**	0.6088±0.042	0.6070±0.021	0.5622±0.036
d_6	10%	90%	**0.4330±0.026**	0.4193±0.082	0.3926±0.051	0.4125±0.011
	50%	50%	0.6750±0.072	0.6308±0.045	**0.6831±0.032**	0.5936±0.053
d_7	10%	90%	0.4678±0.075	0.4653±0.053	**0.4864±0.031**	0.4787±0.067
	50%	50%	0.5118±0.036	0.5076±0.019	0.5100±0.086	**0.5172±0.032**
d_8	10%	90%	0.3801±0.025	0.3684±0.036	0.4006±0.056	0.4081±0.036
	50%	50%	0.6384±0.094	0.6134±0.057	0.6368±0.050	**0.6475±0.052**

Os resultados apresentaram melhorias geralmente quando o tamanho de $\mathcal{Z}_1^l$ aumentou. Na maioria dos casos, quando se utiliza um conjunto menor de dados supervisionados (ou seja, 10% de $\mathcal{Z}_1$), OPFSEMI$_{mst+knn}$ com LP excedeu os melhores resultados nas mesmas condições para a maioria dos casos analisados, enquanto BPMLL obtém o melhor desempenho entre as estratégias usando método de adaptação. Por outro lado, para os conjuntos supervisionados maiores (ou seja, 50% de $\mathcal{Z}_1$), OPFSEMI$_{mst+knn}$ com HOMER apresentou como a melhor escolha. Acreditamos que LP preserva melhor a relação entre reais rótulos do que HOMER depois da transformação de dados ao usar conjuntos menores de amostras supervisionadas, que normalmente é o caso da classificação multirótulos. Os resultados usando Classifier Chain apresentaram uma melhoria na relação entre as amostras supervisionadas e não supervisionadas. Isso é devido ao fato de existir uma sequência de classificadores binários, fazendo com que cada rótulo seja classificado considerando a predição de rótulos anteriormente analisados. Infelizmente, a técnica BR sem qualquer tratamento não é uma boa solução para este problema, por tratar cada classe individualmente, ignorando assim suas possíveis relações. Uma melhoria possível seria o

Tabela 4.12: F-measure considerando OPFSEMI$_{mst}$.

	$\mathcal{Z}_1^l$	$\mathcal{Z}_1^u$	**OPFSEMI$_{mst}$**			
			LP	BR	CC	HOMER
d_1	10%	90%	0.5512±0.028	0.5371±0.091	0.5649±0.010	0.5534±0.028
	50%	50%	0.6212±0.089	0.6163±0.026	0.6290±0.094	0.6158±0.020
d_2	10%	90%	0.6225±0.041	0.5537±0.094	0.5308±0.051	0.5612±0.031
	50%	50%	0.6400±0.018	0.5942±0.096	0.5706±0.045	0.6495±0.044
d_3	10%	90%	0.5604±0.013	0.4393±0.040	0.5603±0.062	0.5194±0.087
	50%	50%	0.6227±0.079	0.6016±0.033	0.6050±0.090	0.6432±0.028
d_4	10%	90%	0.4408±0.085	0.4281±0.041	0.4282±0.099	0.4560±0.023
	50%	50%	0.5215±0.069	0.4757±0.095	0.4770±0.043	0.5103±0.023
d_5	10%	90%	0.3852±0.080	0.3775±0.011	0.3934±0.015	0.3938±0.020
	50%	50%	0.5931±0.098	0.5895±0.066	0.5917±0.045	0.5617±0.092
d_6	10%	90%	0.3979±0.021	0.3820±0.091	0.3898±0.024	0.3782±0.042
	50%	50%	0.6613±0.013	0.6131±0.099	0.6752±0.059	0.5833±0.017
d_7	10%	90%	0.3272±0.054	0.3062±0.091	0.3100±0.012	0.3514±0.055
	50%	50%	0.3825±0.068	0.3626±0.032	0.3898±0.063	0.3784±0.047
d_8	10%	90%	0.3250±0.011	0.3135±0.047	0.3114±0.070	0.3443±0.044
	50%	50%	0.4104±0.094	0.3932±0.072	0.4234±0.033	0.4410±0.026

Tabela 4.13: F-measure considerando LapSVM.

	$\mathcal{Z}_1^l$	$\mathcal{Z}_1^u$	**LapSVM**			
			LP	BR	CC	HOMER
d_1	10%	90%	**0.6274±0.045**	0.5884±0.077	0.6249±0.090	0.5946±0.031
	50%	50%	0.6944±0.024	0.6255±0.052	0.6847±0.016	0.6868±0.095
d_2	10%	90%	0.5701±0.019	0.5728±0.012	0.5902±0.073	0.5639±0.063
	50%	50%	0.5933±0.014	0.5819±0.040	0.6101±0.095	0.5792±0.053
d_3	10%	90%	0.5873±0.056	0.5731±0.083	0.5769±0.091	0.5770±0.031
	50%	50%	0.6213±0.028	0.5868±0.043	0.6212±0.054	0.6187±0.099
d_4	10%	90%	0.4454±0.071	0.4349±0.065	0.4391±0.082	0.4583±0.086
	50%	50%	0.4841±0.046	0.4516±0.043	0.4862±0.044	0.4932±0.089
d_5	10%	90%	0.4783±0.069	0.4717±0.026	0.4703±0.089	0.4957±0.033
	50%	50%	0.5700±0.045	0.5788±0.039	0.5445±0.019	0.5581±0.095
d_6	10%	90%	0.4250±0.098	0.3502±0.068	0.3474±0.042	0.3641±0.064
	50%	50%	0.5394±0.058	0.4781±0.025	0.4714±0.011	0.4847±0.019
d_7	10%	90%	0.4454±0.065	0.4214±0.073	0.4398±0.069	0.4350±0.053
	50%	50%	0.4710±0.073	0.4606±0.068	0.4893±0.098	0.4816±0.064
d_8	10%	90%	0.4061±0.095	0.3837±0.049	0.3936±0.078	**0.4174±0.016**
	50%	50%	0.6213±0.054	0.5316±0.018	0.5850±0.084	0.6313±0.020

Tabela 4.14: F-measure considerando TSVM.

	$\mathcal{Z}_1^l$	$\mathcal{Z}_1^u$	**TSVM**			
			LP	BR	CC	HOMER
d_1	10%	90%	0.6034±0.023	0.5182±0.010	0.5338±0.024	0.5845±0.085
	50%	50%	0.6842±0.069	0.6145±0.079	0.6291±0.069	0.6551±0.026
d_2	10%	90%	0.5437±0.042	0.5236±0.028	0.5220±0.069	0.5934±0.095
	50%	50%	0.6048±0.090	0.6001±0.008	0.6012±0.026	0.6088±0.043
d_3	10%	90%	0.5116±0.032	0.4382±0.040	0.4567±0.041	0.4892±0.007
	50%	50%	0.5855±0.027	0.5232±0.026	0.5330±0.091	0.5719±0.092
d_4	10%	90%	0.4091±0.045	0.4420±0.014	0.4443±0.002	**0.4752±0.065**
	50%	50%	0.4386±0.060	0.4538±0.071	0.4481±0.024	0.5069±0.004
d_5	10%	90%	0.4778±0.060	0.4867±0.012	0.4893±0.010	0.4771±0.012
	50%	50%	0.5991±0.019	0.5991±0.061	0.5977±0.084	0.5337±0.065
d_6	10%	90%	0.3934±0.066	0.3649±0.073	0.3784±0.029	0.3999±0.030
	50%	50%	0.6543±0.028	0.3791±0.095	0.3581±0.065	0.5421±0.023
d_7	10%	90%	0.3366±0.075	0.3260±0.071	0.3285±0.062	0.3444±0.035
	50%	50%	0.4002±0.028	0.3924±0.054	0.3968±0.010	0.4169±0.038
d_8	10%	90%	0.3298±0.055	0.3140±0.028	0.3487±0.037	0.3612±0.048
	50%	50%	0.4840±0.077	0.4721±0.069	0.4668±0.038	0.4854±0.045

Tabela 4.15: Hamming Loss considerando OPFSEMI$_{mst+knn}$.

	$\mathcal{Z}_1^l$	$\mathcal{Z}_1^u$	**OPFSEMI**$_{mst+knn}$			
			LP	BR	CC	HOMER
d_1	10%	90%	0.1154±0.059	0.1110±0.018	0.1162±0.087	0.1203±0.088
	50%	50%	0.1187±0.076	0.1089±0.040	0.1116±0.072	0.1076±0.059
d_2	10%	90%	**0.2161±0.045**	0.2452±0.054	0.2231±0.044	0.2338±0.046
	50%	50%	0.2091±0.025	0.2385±0.041	0.2143±0.027	**0.2018±0.021**
d_3	10%	90%	**0.2207±0.041**	0.3263±0.077	0.2359±0.028	0.2667±0.063
	50%	50%	0.2066±0.054	0.2630±0.056	0.2266±0.084	0.2125±0.034
d_4	10%	90%	0.0445±0.059	0.0485±0.015	0.0459±0.040	0.0445±0.084
	50%	50%	0.0413±0.078	0.0470±0.060	0.0457±0.072	0.0413±0.081
d_5	10%	90%	0.1388±0.043	0.1441±0.055	0.1320±0.070	0.1218±0.041
	50%	50%	0.0993±0.017	0.1402±0.084	0.1295±0.080	0.0991±0.030
d_6	10%	90%	0.2269±0.083	0.2558±0.049	0.2467±0.031	**0.2123±0.078**
	50%	50%	0.2168±0.059	0.2401±0.065	0.2318±0.031	**0.2070±0.045**
d_7	10%	90%	0.2950±0.055	0.3110±0.070	0.3009±0.061	0.2596±0.058
	50%	50%	0.2168±0.071	0.3043±0.022	0.2816±0.057	0.2140±0.083
d_8	10%	90%	0.0347±0.022	0.0557±0.028	0.0435±0.013	0.0347±0.070
	50%	50%	0.0339±0.017	0.0401±0.027	0.0366±0.084	**0.0311±0.077**

Tabela 4.16: Hamming Loss considerando OPFSEMI$_{mst}$.

	$\mathcal{Z}_1^l$	$\mathcal{Z}_1^u$	**OPFSEMI$_{mst}$**			
			LP	BR	CC	HOMER
d_1	10%	90%	0.1103±0.054	0.1172±0.011	0.1151±0.090	0.1417±0.072
	50%	50%	**0.0817±0.069**	0.0892±0.016	0.0889±0.034	0.1167±0.065
d_2	10%	90%	0.2411±0.072	0.2472±0.052	0.2488±0.053	0.2833±0.084
	50%	50%	0.2057±0.043	0.2053±0.060	0.2027±0.039	0.2359±0.056
d_3	10%	90%	0.2877±0.079	0.2950±0.084	0.3041±0.022	0.3277±0.023
	50%	50%	0.2241±0.079	0.2275±0.058	0.2382±0.041	0.2410±0.036
d_4	10%	90%	0.0360±0.016	0.0343±0.083	**0.0341±0.063**	0.0380±0.028
	50%	50%	0.0348±0.055	0.0339±0.069	**0.0338±0.021**	0.0358±0.066
d_5	10%	90%	0.1418±0.067	0.1456±0.085	0.1356±0.022	0.1253±0.030
	50%	50%	0.1009±0.031	0.1439±0.059	0.1310±0.025	0.1012±0.070
d_6	10%	90%	0.2320±0.049	0.2594±0.010	0.2525±0.029	0.2170±0.025
	50%	50%	0.2191±0.080	0.2461±0.049	0.2360±0.040	0.2123±0.044
d_7	10%	90%	0.2991±0.063	0.3159±0.038	0.3064±0.038	0.2664±0.068
	50%	50%	0.2201±0.030	0.3111±0.071	0.2866±0.012	0.2180±0.059
d_8	10%	90%	0.0356±0.033	0.0569±0.087	0.0439±0.060	0.0350±0.041
	50%	50%	0.0347±0.084	0.0406±0.070	0.0373±0.037	0.0318±0.078

Tabela 4.17: Hamming Loss considerando LapSVM.

	$\mathcal{Z}_1^l$	$\mathcal{Z}_1^u$	**LapSVM**			
			LP	BR	CC	HOMER
d_1	10%	90%	**0.0930±0.065**	0.1252±0.084	0.0981±0.062	0.1236±0.023
	50%	50%	0.0955±0.046	0.1187±0.049	0.0942±0.022	0.1208±0.025
d_2	10%	90%	0.2495±0.077	0.2538±0.059	0.2511±0.085	0.2636±0.064
	50%	50%	0.2530±0.012	0.2560±0.041	0.2412±0.040	0.2563±0.061
d_3	10%	90%	0.2523±0.084	0.2663±0.066	0.2698±0.044	0.2511±0.061
	50%	50%	0.2421±0.033	0.2432±0.077	0.2592±0.060	0.2466±0.058
d_4	10%	90%	0.0459±0.028	0.0493±0.040	0.0462±0.048	0.0451±0.039
	50%	50%	0.0457±0.018	0.0473±0.082	0.0458±0.044	0.0352±0.039
d_5	10%	90%	0.1404±0.019	0.1456±0.058	0.1335±0.041	0.1237±0.088
	50%	50%	0.1011±0.057	0.1427±0.072	0.1316±0.073	0.1008±0.063
d_6	10%	90%	0.2299±0.038	0.2605±0.067	0.2503±0.026	0.2161±0.042
	50%	50%	0.2191±0.031	0.2438±0.022	0.2360±0.044	0.2111±0.023
d_7	10%	90%	0.2983±0.060	0.3172±0.073	0.3050±0.057	0.2631±0.037
	50%	50%	0.2206±0.024	0.3084±0.056	0.2870±0.034	0.2179±0.017
d_8	10%	90%	0.0352±0.029	0.0567±0.071	0.0441±0.072	**0.0342±0.030**
	50%	50%	0.0345±0.063	0.0406±0.076	0.0370±0.029	0.0315±0.056

Tabela 4.18: Hamming Loss considerando TSVM.

	$\mathcal{Z}_1^l$	$\mathcal{Z}_1^u$	**TSVM**			
			LP	BR	CC	HOMER
d_1	10%	90%	0.1534±0.016	0.1546±0.021	0.1543±0.056	0.1541±0.033
	50%	50%	0.1340±0.011	0.1355±0.081	0.1356±0.080	0.1348±0.014
d_2	10%	90%	0.2667±0.062	0.2695±0.062	0.2698±0.074	0.2731±0.061
	50%	50%	0.2502±0.038	0.2550±0.029	0.2548±0.010	0.2584±0.087
d_3	10%	90%	0.3001±0.074	0.3007±0.070	0.3012±0.063	0.3170±0.020
	50%	50%	0.2461±0.079	0.2483±0.031	0.2466±0.057	0.2506±0.022
d_4	10%	90%	0.0509±0.083	0.0511±0.031	0.0511±0.055	0.0489±0.041
	50%	50%	0.0478±0.034	0.0480±0.029	0.0411±0.076	0.0462±0.012
d_5	10%	90%	0.1406±0.026	0.1466±0.087	0.1358±0.014	0.1259±0.061
	50%	50%	0.1028±0.068	0.1424±0.033	0.1308±0.032	0.1017±0.076
d_6	10%	90%	0.2293±0.078	0.2643±0.045	0.2513±0.012	0.2175±0.020
	50%	50%	0.2239±0.077	0.2476±0.049	0.2398±0.035	0.2137±0.038
d_7	10%	90%	0.3028±0.010	0.3169±0.013	0.3078±0.029	0.2666±0.049
	50%	50%	0.2198±0.024	0.3127±0.073	0.2875±0.023	0.2205±0.069
d_8	10%	90%	0.0355±0.073	0.0577±0.022	0.0440±0.027	0.0353±0.081
	50%	50%	0.0344±0.080	0.0414±0.047	0.0374±0.077	0.0321±0.073

Tabela 4.19: F-measure e Hamming Loss considerando MLkNN e BPMLL.

	$\mathcal{Z}_1^l$	$\mathcal{Z}_1^u$	**MLkNN**		**BPMLL**	
			F-measure	Hamming Loss	F-measure	Hamming Loss
d_1	10%	90%	0.5614±0.087	0.1607±0.026	0.5581±0.027	0.2572±0.019
	50%	50%	0.6166±0.016	0.1434±0.033	0.6205±0.057	0.1542±0.012
d_2	10%	90%	0.5942±0.065	0.2918±0.037	0.6083±0.029	0.2321±0.032
	50%	50%	0.5968±0.084	0.3021±0.036	0.6193±0.042	0.2300±0.045
d_3	10%	90%	0.5279±0.023	0.3287±0.071	0.5801±0.018	0.2444±0.035
	50%	50%	0.5948±0.067	0.2879±0.060	0.6440±0.051	**0.2032±0.016**
d_4	10%	90%	0.4747±0.089	0.0489±0.055	0.4637±0.074	0.0696±0.053
	50%	50%	0.5103±0.063	0.0479±0.010	0.4707±0.072	0.0651±0.085
d_5	10%	90%	0.4639±0.054	**0.0830±0.016**	0.4605±0.060	0.1579±0.081
	50%	50%	0.4821±0.062	**0.0822±0.044**	0.5129±0.024	0.0772±0.085
d_6	10%	90%	0.3519±0.012	0.2172±0.046	0.4017±0.041	0.2882±0.070
	50%	50%	0.3801±0.034	0.2075±0.016	0.4417±0.063	0.2694±0.011
d_7	10%	90%	0.3383±0.057	**0.0858±0.013**	0.3717±0.040	0.1367±0.045
	50%	50%	0.4521±0.019	**0.0809±0.069**	0.4106±0.026	0.1425±0.075
d_8	10%	90%	0.3140±0.010	0.0373±0.075	0.3840±0.072	0.0351±0.062
	50%	50%	0.5053±0.011	0.0317±0.018	0.5536±0.011	0.0313±0.049

Tabela 4.20: Porcentagem de erro de propagação ($\mathcal{E}$) sobre $\mathcal{Z}_1^u$ para OPFSEMI$_{mst+knn}$.

	$\mathcal{Z}_1^l$	$\mathcal{Z}_1^u$	**OPFSEMI$_{mst+knn}$**			
			LP	BR	CC	HOMER
d_1	10%	90%	35.37 (k^*=17)	37.50 (k^*=20)	39.12 (k^*=20)	38.30 (k^*=16)
	50%	50%	18.40 (k^*=13)	18.86 (k^*=18)	17.86 (k^*=17)	17.40 (k^*=15)
d_2	10%	90%	39.50 (k^*=9)	37.60 (k^*=15)	37.47 (k^*=13)	37.75 (k^*=11)
	50%	50%	16.63 (k^*=9)	16.09 (k^*=13)	14.77 (k^*=12)	15.94 (k^*=10)
d_3	10%	90%	39.06 (k^*=17)	37.54 (k^*=19)	36.87 (k^*=20)	36.51 (k^*=21)
	50%	50%	17.51 (k^*=13)	16.33 (k^*=16)	15.83 (k^*=18)	14.29 (k^*=16)
d_4	10%	90%	39.32 (k^*=12)	39.91 (k^*=17)	39.98 (k^*=15)	37.11 (k^*=14)
	50%	50%	16.30 (k^*=11)	19.48 (k^*=13)	18.12 (k^*=13)	16.21 (k^*=10)
d_5	10%	90%	25.93 (k^*=5)	27.74 (k^*=8)	26.06 (k^*=9)	20.34 (k^*=12)
	50%	50%	10.25 (k^*=4)	12.91 (k^*=7)	11.80 (k^*=5)	7.46 (k^*=8)
d_6	10%	90%	40.34 (k^*=12)	41.64 (k^*=14)	41.84 (k^*=12)	42.02 (k^*=15)
	50%	50%	16.44 (k^*=5)	18.40 (k^*=7)	19.11 (k^*=12)	18.49 (k^*=10)
d_7	10%	90%	38.29 (k^*=30)	39.05 (k^*=34)	38.71 (k^*=32)	37.11 (k^*=27)
	50%	50%	18.39 (k^*=21)	19.62 (k^*=24)	18.50 (k^*=18)	16.47 (k^*=20)
d_8	10%	90%	32.89 (k^*=32)	34.24 (k^*=40)	33.02 (k^*=36)	30.14 (k^*=25)
	50%	50%	14.87 (k^*=22)	18.56 (k^*=25)	17.43 (k^*=20)	12.08 (k^*=18)

Tabela 4.21: Porcentagem de erro de propagação ($\mathcal{E}$) sobre $\mathcal{Z}_1^u$ para OPFSEMI$_{mst}$.

	$\mathcal{Z}_1^l$	$\mathcal{Z}_1^u$	**OPFSEMI$_{mst}$**			
			LP	BR	CC	HOMER
d_1	10%	90%	36.63	40.79	43.16	40.71
	50%	50%	20.16	23.66	22.56	20.46
d_2	10%	90%	43.87	40.05	40.15	40.74
	50%	50%	17.79	20.51	16.81	17.17
d_3	10%	90%	39.21	37.83	37.35	40.30
	50%	50%	21.16	16.72	19.40	17.62
d_4	10%	90%	39.59	41.34	41.15	38.48
	50%	50%	19.71	21.31	18.96	17.67
d_5	10%	90%	29.97	31.62	30.65	27.18
	50%	50%	14.13	16.32	15.20	9.63
d_6	10%	90%	45.02	46.11	46.58	44.27
	50%	50%	25.02	29.34	30.72	29.56
d_7	10%	90%	43.32	42.01	40.81	42.64
	50%	50%	23.42	23.59	21.77	22.19
d_8	10%	90%	35.29	36.55	35.57	32.54
	50%	50%	16.82	20.34	18.61	17.71

tratamento individual usando *clusters* para ajudar a manter as relações entre amostras. Basicamente, LapSVM e OPFSEMI$_{mst}$ possuem um comportamento muito semelhante, destacando LapSVM pelo número de melhores resultados em diferentes domínios de conjuntos de dados. Uma outra observação interessante é a robustez das técnicas semisupervisionadas em relação ao número de amostras supervisionadas. Isso pode ser explicado pela correção da margem do classificador devido à presença de amostras não supervisionadas em $\mathcal{Z}_1^u$, que são corretamente rotulados a partir de $\mathcal{Z}_1^l$.

A fim de fornecer uma análise estatística dos resultados, foi realizado o teste de Friedman [32]. O objetivo desta avaliação estatística é validar os resultados, e mostrar o comportamento de diferentes modelos de transformação de dados e métodos de adaptação usando aprendizado semisupervisionado. Realizamos nossa avaliação estatística sobre dois cenários. Primeiramente, iremos avaliar todos os algoritmos semisupervisionado usando métodos de transformação, e em um segundo cenário, iremos considerar todas as técnicas juntamente com os métodos de adaptação.

As Figuras (4.4–4.5) (primeiro cenário) e Figuras 4.6 (segundo cenário) ilustram o teste post-hoc Nemenyi, já que foi rejeitada a hipótese nula de que todos os classificadores são equivalentes entre si. Grupos de classificadores que são considerados equivalentes (ou seja, não são significativamente diferentes) estão conectados (com nível de significância $p = 0.05$) através de uma distância crítica (CD), onde o classificador mais a direita é considerado melhor, e a técnica mais próxima do extremo lado esquerdo é a pior. Apenas para os casos das Figuras (4.4e–4.5e) (em relação à análise de todos os métodos de transformação em conjunto), podemos obter uma diferença com uma distância crítica (CD) igual a 0.5863.

Os resultados de ambos os testes (teste post-hoc Nemenyi usando F-measure e Hamming Loss) são nos dois cenários avaliados, em geral, equivalentes. Podemos confirmar os resultados apresentados, mostrando que o melhor resultado foi obtido por OPFSEMI$_{mst+knn}$ seguido por LapSVM, OPFSEMI$_{mst}$ and BPMLL. Outro interessante resultado foi o destaque para OPFSEMI$_{mst+knn}$ como sendo a melhor abordagem usando BR (Figura 4.4(b)), e em geral o melhor classificador (Figuras 4.4e-4.5e), com resultados equivalentes entre LapSVM e OPFSEMI$_{mst}$, e, finalmente, TSVM.

Ao realizar uma análise mais aprofundada, podemos comparar estatisticamente o par de classificadores OPFSEMI$_{mst+knn}$ and OPFSEMI$_{mst}$, usando o teste Wilcoxon Signed Rank [26]. Neste caso, obtivemos um valor de $p = 8.017^{-8}$ usando F-measure e de $p = 0.045$ usando Hamming Loss, ambos ($p < 0.05$), podendo ser considerados estatisticamente diferentes. Isto confirma a melhoria de OPFSEMI$_{mst+knn}$ sobre sua versão OPFSEMI$_{mst}$ em problemas multirótuloss. A técnica OPFSEMI$_{mst+knn}$ mostra o ganho em usar a estrutura proposta com base na MST com OPF e um classificador final com um grafo k-NN, garantindo uma maior relação entre as classes, mesmo depois da

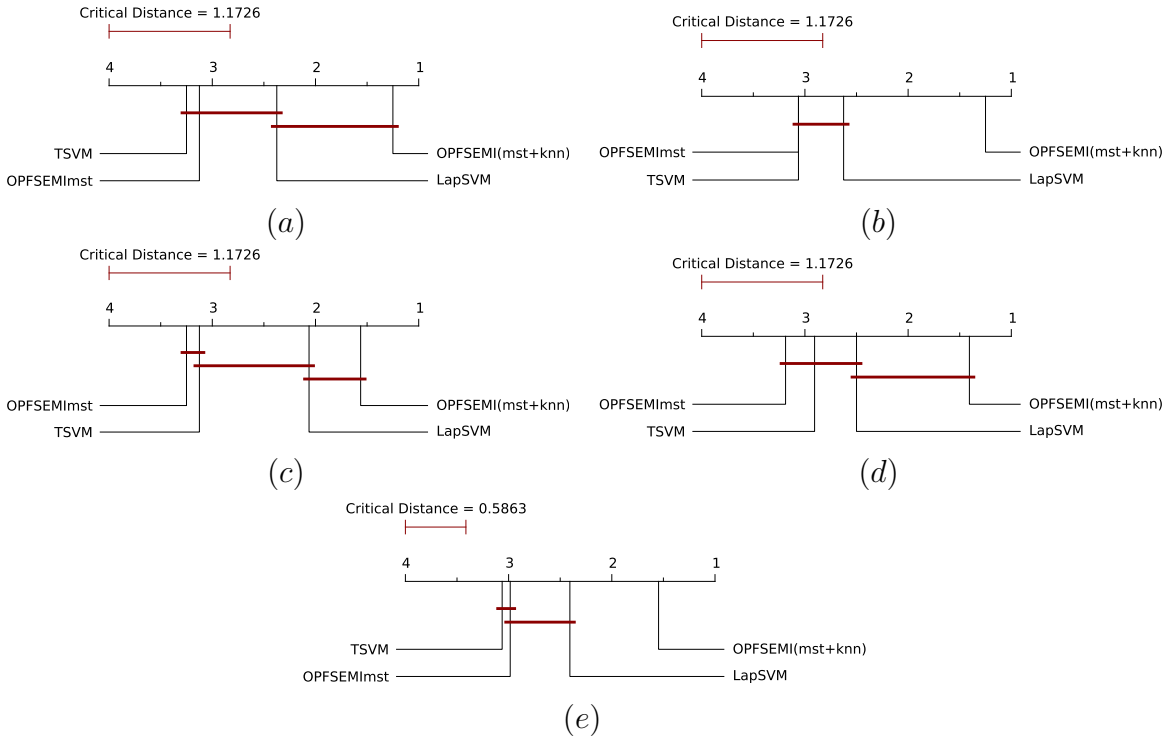


Figura 4.4: Comparação de todos os classificadores um contra ao outro usando o teste post-hoc Nemenyi (usando F-measure). Grupos de classificadores que não são significativamente diferentes (em p = 0.05) estão conectados: (a) Label Powerset, (b) Binary Relevance, (c) Classifier Chain, e (d) Hierarchy of Multi-Label Classifiers, e (e) todos os métodos de trasnformação.

transformação de problemas multirótulos em comparação com outras abordagens semisupervisionadas.

Com base nos resultados apresentados por OPFSEMI$_{mst+knn}$, uma pergunta natural que podemos realizar, é se podemos definir a proposta como a mesma escolha para a aprendizagem semisupervisionada usando Floresta de Caminhos Ótimos. Acreditamos firmemente que para problemas multirótulos, a resposta é sim, até mesmo pelos resultados apresentados e confirmados pelos testes estatísticos. Principalmente também devido aos problemas apresentados, que ocorrem após os processos de transformação de dados, e potencializados quando propagados para o conjunto não supervisionado. Precisamos avaliar todas as características em consideração ao custo adicional relativo ao novo estágio acrescentado sobre OPFSEMI$_{mst}$ para elaboração da nova proposta, isto é, fator que pode impactar diretamente no desempenho e tempo de resposta dependendo do domínio de aplicação.

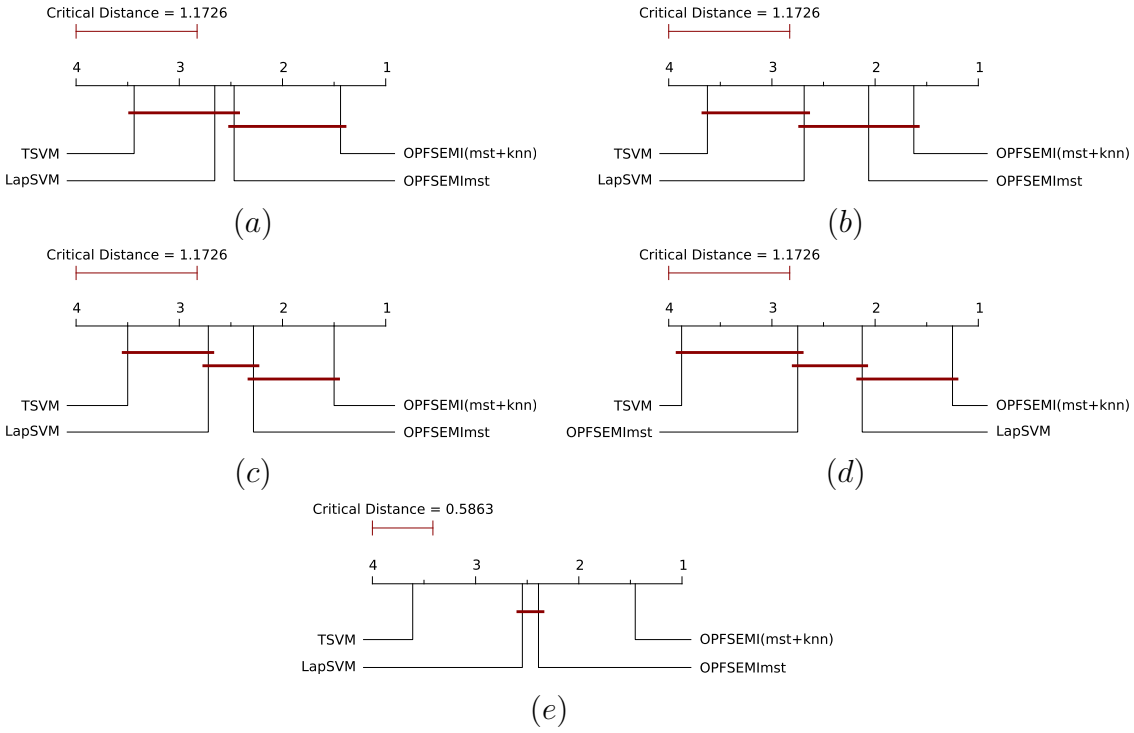


Figura 4.5: Comparação de todos os classificadores um contra ao outro usando o teste post-hoc Nemenyi (usando Hamming Loss). Grupos de classificadores que não são significativamente diferentes (em p = 0.05) estão conectados: (a) Label Powerset, (b) Binary Relevance, (c) Classifier Chain, e (d) Hierarchy of Multi-Label Classifiers, e (e) todos os métodos de trasnformação.

4.6 Aprendizado ativo e semisupervisionado

Nesta seção, avaliamos a abordagem proposta ASSL-OPF [74] usando tanto OPFSEMI (AL-OPFSEMI) quanto OPFSEMI$_{mst}$ (AL-OPFSEMI$_{mst}$), comparados a Rand, uma abordagem de aprendizado semisupervisionado onde são selecionados aleatoriamente amostras supervisionadas e não supervisionadas. Para comparar o desempenho de cada abordagem, ao longo das iterações de aprendizagem, considerou-se a medida de acurácia (em um conjunto de teste não conhecido obtido a partir de cada base de dados), o percentual de erro de propagação no conjunto não supervisionado, bem como o número de classes conhecidas. Os resultados a serem apresentados (medidas de acurácia) foram preparados a partir da média de 10 execuções, com conjuntos de amostras gerados aleatoriamente para o treinamento ($\mathcal{Z}_1$) e teste ($\mathcal{Z}_2$). Para todas as bases de dados usadas, selecionamos 80% de amostras para a aprendizagem e 20% para testes.

As Figuras 4.7-4.11 apresentam os resultados usando AL-OPFSEMI, AL-OPFSEMI$_{mst}$

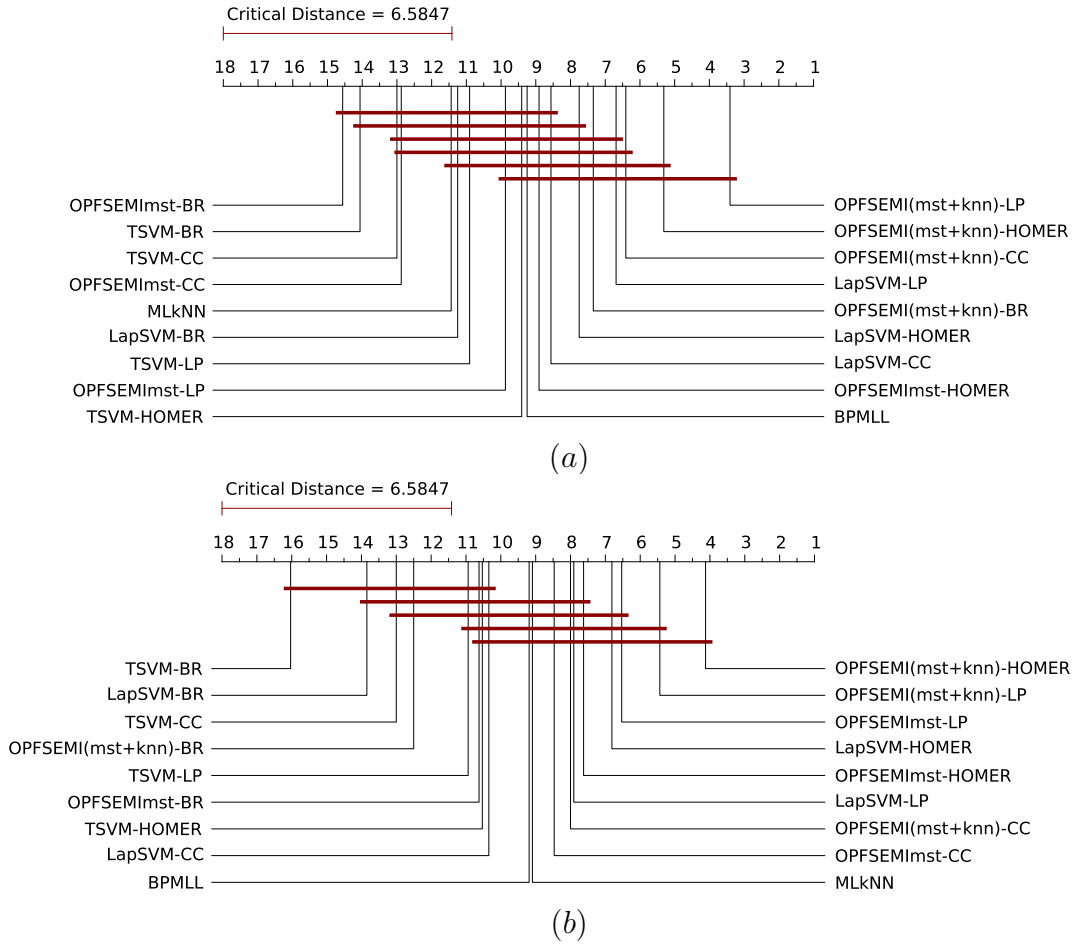
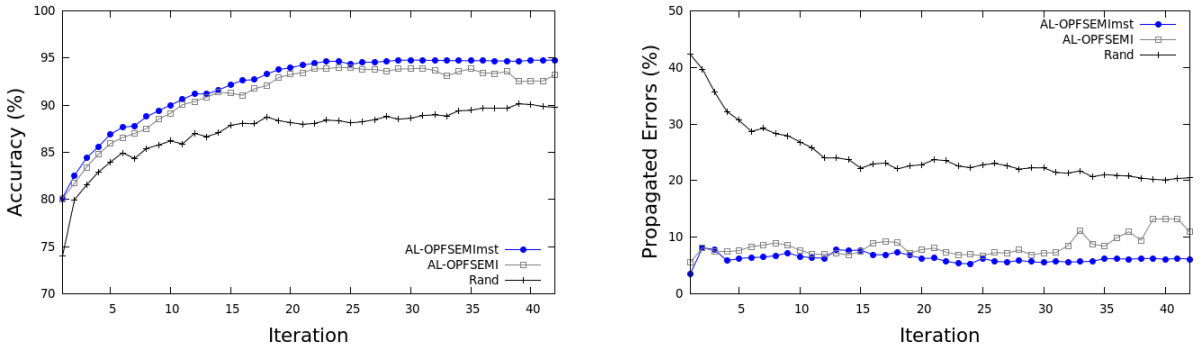


Figura 4.6: Comparação de todos os classificadores um contra ao outro, em conjunto com estratégias de métodos de adaptação, usando o teste post-hoc Nemenyi. Grupos de classificadores que não são significativamente diferentes (em p = 0.05) estão conectados: (a) usando F-measure, (b) usando Hamming Loss.

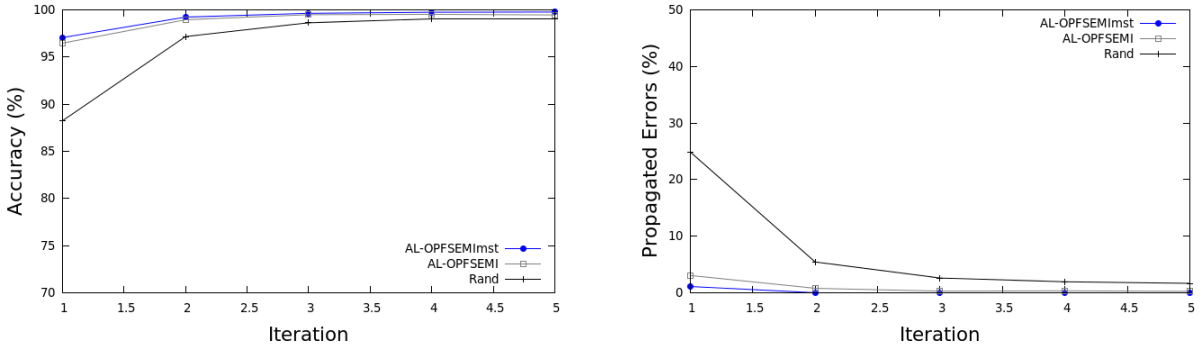
e Rand para as bases de dados Statlog, Faces, Pendigits, Cowhide e Parasites, respectivamente. Como podemos observar, ambos os métodos de aprendizado semisupervisionado ativo AL-OPFSEMI e AL-OPFSEMI$_{mst}$ conseguiram um desempenho superior comparado com a abordagem Rand, iniciando e mantendo a acurácia mais elevada (Figuras 4.7a, 4.8a, 4.9a, 4.10a and 4.11a) e apresentando também os menores erros de propagação em amostras não supervisionadas (Figuras 4.7b, 4.8b, 4.9b, 4.10b e 4.11b). Além disso, Rand requer mais iterações para identificar amostras de todas as classes, como apresentado na Tabela 4.22).

Considerando a base de dados Statlog, na terceira iteração, as abordagens AL-OPFSEMI, AL-OPFSEMI$_{mst}$ atingiram uma acurácia acima de 85%, enquanto que a abordagem ran-



(a) (b)

Figura 4.7: Base de dados Statlog. (a) Acurácia média dos métodos sobre o conjunto de teste. (b) Percentual de erro de propagação.



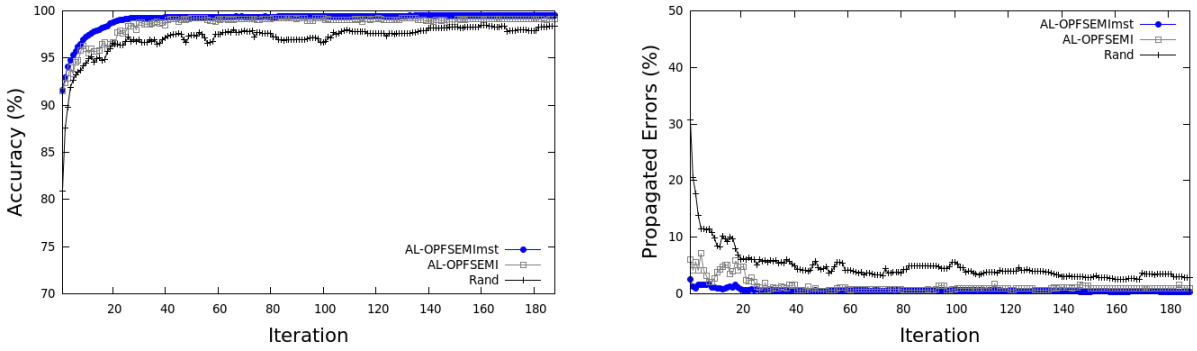
(a) (b)

Figura 4.8: Base de dados Faces. (a) Acurácia média dos métodos sobre o conjunto de teste. (b) Percentual de erro de propagação.

Tabela 4.22: Número total de classes conhecidas na primeira iteração para as abordagens AL-OPFSEMI, AL-OPFSEMI$_{mst}$ e Rand.

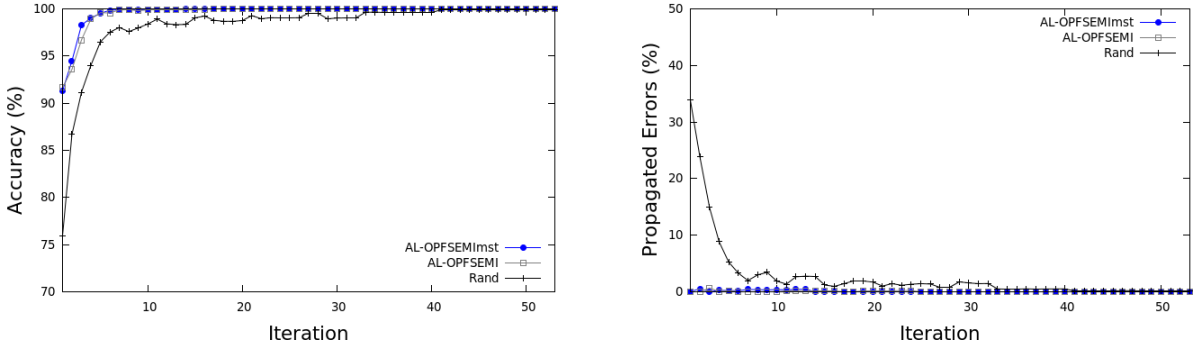
	Faces	**Statlog**	**Pendigits**	**Cowhide**	**Parasites**
AL-OPFSEMI$_{mst}$	54.00	7.00	10.00	5.00	14.80
AL-OPFSEMI	54.00	7.00	10.00	5.00	14.80
Rand	47.90	5.00	8.70	4.20	12.00

domizado alcançou resultados próximos somente na oitava iteração (Figura 4.7a). Além disso, a abordagem Rand propaga muito mais erros de rotulação sobre o conjunto não supervisionado, o que para esse caso atinge 40% (Figura 4.7b). A razão para este comportamento é a ausência de qualquer estratégia para a redução ou definição de prioridades para seleção de amostras e dependência do processo aleatório, sobre uma grande base de dados.



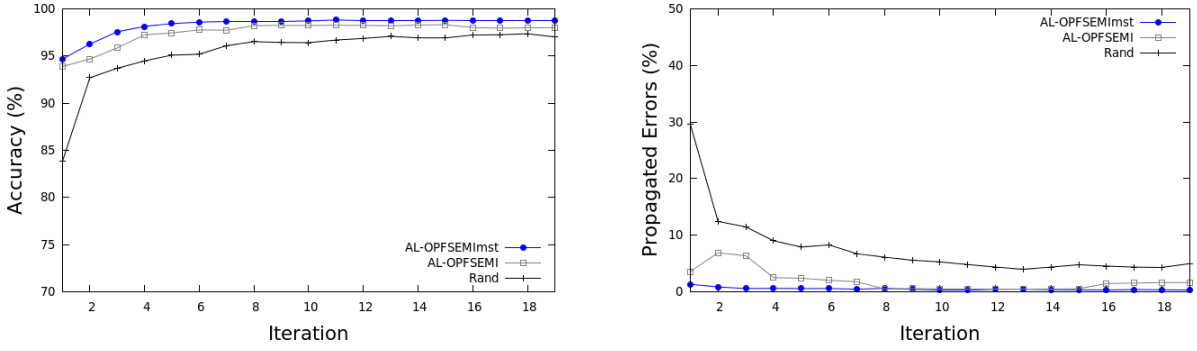
(a) (b)

Figura 4.9: Base de dados Pendigits. (a) Acurácia média dos métodos sobre o conjunto de teste. (b) Percentual de erro de propagação.



(a) (b)

Figura 4.10: Base de dados Cowhide. (a) Acurácia média dos métodos sobre o conjunto de teste. (b) Percentual de erro de propagação.



(a) (b)

Figura 4.11: Base de dados Parasites. (a) Acurácia média dos métodos sobre o conjunto de teste. (b) Percentual de erro de propagação.

Em geral, estes resultados são semelhantes aos observados nas bases de dados Faces, Pendigits, Cowhide e Parasites (Figuras 4.8, 4.9, 4.10 e 4.11). Embora as abordagens AL-OPFSEMI e AL-OPFSEMI$_{mst}$ se comportam com um desempenho amplamente superior, para a abordagem Rand o processo de aprendizagem é bastante lento para alcançar uma alta acurácia e requer mais iterações para identificar amostras de todas as classes (Tabela 4.22). Isso mostra que a abordagem proposta possui um desempenho superior quando comparada à Rand.

Base de dados	$\|\mathcal{Z}_1\|$	**anotada**	**acurácia ± desvio padrão**	**tempo**
Statlog	1,761	8.74%	90.79% ± 0.91	0.14
Faces	1,469	9.12%	99.26% ± 0.25	0.13
Pendigits	8,791	4.72%	98.76% ± 0.48	25.22
Cowhide	1,351	2.27%	99.66% ± 0.10	0.56
Parasites	1,455	6.45%	97.82% ± 1.69	0.85

Tabela 4.23: Tamanho total do conjunto de aprendizagem $\mathcal{Z}_1$, número total de amostras anotadas, acurácia média ± desvio padrão e tempo computacional para seleção (em minutos) para AL-OPFSEMI.

Base de dados	$\|\mathcal{Z}_1\|$	**anotada**	**acurácia ± desvio padrão**	**tempo**
Statlog	1,761	14.03%	94.79% ± 0.58	0.03
Faces	1,469	9.48%	99.78% ± 0.21	0.01
Pendigits	8,791	3.38%	99.54% ± 0.13	3.46
Cowhide	1,351	1.60%	99.98% ± 0.05	0.02
Parasites	1,455	5.60%	98.77% ± 0.33	0.04

Tabela 4.24: Tamanho total do conjunto de aprendizagem $\mathcal{Z}_1$, número total de amostras anotadas, acurácia média ± desvio padrão e tempo computacional para seleção (em minutos) para AL-OPFSEMI$_{mst}$.

Com o objetivo de apreciar a qualidade dos resultados obtidos em cada base de dados por AL-OPFSEMI e AL-OPFSEMI$_{mst}$, apresentamos também o número total de amostras anotadas/corrigidas pelo especialista, acurácia média, desvio padrão e tempo computacional para a seleção das amostras mais representativas no final do ciclo de aprendizagem (Tabelas 4.23 e 4.24). Isso destaca as vantagens em ambos os métodos, principalmente para AL-OPFSEMI$_{mst}$ em aplicações práticas.

A nossa abordagem aumenta a probabilidade de amostras selecionadas serem mais informativas e permite um treinamento mais eficiente e eficaz dos classificadores, levando a uma redução considerável no erro de classificação após algumas iterações. Isto corresponde a uma grande vantagem da nossa abordagem, uma vez que requer poucas iterações

na fase de aprendizagem para alcançar uma acurácia elevada e ainda reduzindo os erros de propagação no conjunto não supervisionado. Além disso, ao contrário das abordagens tradicionais de aprendizagem ativo, uma vez que o conjunto de aprendizagem foi organizado, nossa proposta (aprendizado ativo e semisupervisionado) não requer classificação e reorganização de todas as amostras na base de dados em cada iteração. Por este motivo, o processo de seleção acaba sendo muito rápido, mesmo para grandes bases de dados.

Capítulo 5

Conclusão e trabalhos futuros

5.1 Principais contribuições

O objetivo deste trabalho foi o desenvolvimento de novas abordagens na área aprendizado semisupervisionado baseados em classificadores por Floresta de Caminhos Ótimos. Esta metodologia foi inicialmente proposta para o projeto de operadores de processamento de imagem [29] e, posteriormente estendida para agrupamento [72], classificação supervisionada [69] e agora o aprendizado semisupervisionado. Basicamente, os classificadores computam uma Floresta de Caminhos Ótimos sobre um conjunto de treinamento e buscam classificar novas amostras com o rótulo da raiz da árvore mais fortemente conexa. As amostras são modeladas como sendo nós de um grafo e conectadas por uma determinada relação de adjacência. A partir das amostras mais representativas (protótipos), caminhos de custo ótimo (computados através de uma dada função de conectividade) são oferecidos para as amostras restantes no grafo, iniciando um processo de competição entre protótipos que oferecem caminhos de menor custo. Assim, novos classificadores podem ser obtidos pela execução do algoritmo OPF uma ou várias vezes sobre diferentes grafos de entrada e funções de conectividade.

Esta tese de doutorado, entra em um campo novo no estudo de Floresta de Caminhos Ótimos e mostra que o uso de dados não supervisionados pode melhorar significativamente o desempenho dos classificadores em comparação com as técnicas até então propostas. Neste contexto, três classificadores semisupervisionados foram apresentados: OPFSEMI, foi a primeira proposta semisupervisionada apresentada para a comunidade. Os protótipos são encontrados em cada classe entre as amostras de treinamento supervisionadas, usando a mesma estratégia da abordagem OPF supervisionado [69]. Posteriormente, todas as amostras de treinamento são interpretadas como um grafo completo e cada amostra é atribuída à árvore de caminho ótimo de seu protótipo mais fortemente conexo. Portanto, a classe do protótipo é propagada para todas as amostras de treinamento (supervisionadas

e não supervisionadas) em sua árvore. Uma vez que o conjunto de treinamento é agora inteiramente rotulado, o algoritmo OPF supervisionado é executado, a fim de selecionar mais e/ou melhores protótipos.

Em seguida apresentamos OPFSEMI$_{mst}$, que também explora a conectividade entre amostras supervisionadas e não supervisionadas, mas resultando um classificador final a partir de uma única iteração para propagação dos rótulos. A proposta usa como entrada uma Árvore Geradora Mínina calculada sobre todo conjunto de treinamento. Como todos os caminhos sobre a estrutura MST são ótimos (para a função de conectividade $f_{\max}$), a partição do algoritmo OPF sobre a MST gera uma Floresta de Caminhos Ótimos enraizadas sobre as amostras supervisionadas. Em termos de complexidade, OPFSEMI exige essencialmente duas execuções com tempo de $O(n^2) + O(n \log n)$ cada, enquanto que OPFSEMI$_{mst}$ exige uma execução de $O(n^2)$ para geração da MST e $O(n \log n)$ para geração do classificador final OPF, isto devido ao uso da estrutura MST, e não uso do grafo completo em todas as fases de treinamento como ocorre no OPFSEMI.

A terceira técnica proposta, chamada de OPFSEMI$_{mst+knn}$, tem a finalidade de melhorar a propagação dos rótulos em um cenário multirótulo, basicamente adicionando um último passo no processo de treinamento por OPFSEMI$_{mst}$. Essa estratégia foi adotada, por que em experimentos anteriores, verificou-se que a transformação de dados multirótulos para único rótulo forçava as bases de dados a perder em informações do relacionamento entre as classes. Com essa informação comprometida, as técnicas propagavam para as amostras não supervisionadas informações sobre as classes que não condiziam com o rótulo real, erro assim potencializado no momento de reverter a classificação para multirótulo. O novo classificador semisupervisionado é uma floresta de caminhos ótimos enraizada nos valores máximos de uma função de densidade de probabilidade (pdf), calculada a partir de um grafo k-vizinho mais próximo, cujos nós são as amostras de treinamento. Durante a classificação, as amostras de treinamento mais próximas de suas raízes têm maior prioridade para atribuir rótulos para novas amostras.

Uma vez que amostras classificadas de maneira errada no conjunto de treinamento tendem a estar na maioria dos casos na fronteira entre as classes, onde os valores da pdf são mais baixos, OPFSEMI$_{mst+knn}$ pode melhorar significativamente o desempenho de classificação no problema de atribuição multirótulo. O último passo atribuído sobre OPFSEMI$_{mst}$ é semelhante ao algoritmo de aprendizagem supervisionada [65]. No entanto, a versão supervisionada estima o melhor k para o grafo k-NN, maximizando a acurácia na propagação dos rótulos a partir das raízes da floresta. Este critério resulta em baixos valores de k (isto é, muitos *clusters*), implicando também em uma pobre estimativa da pdf. Dessa forma, propomos encontrar o valor de k avaliando o grau de discordância de rotulação abaixo de um limiar, obtido no conjunto de treinamento usando OPFSEMI$_{mst}$ e a nova rotulação dada pelo classificador final com a proposta de aprendizado supervisi-

onado [65]. Argumentamos que este processo é sempre possível e diminui as chances de erros de propagação serem potencializados após a aplicação de métodos de transformação.

Por fim, nossa última contribuição se insere no campo de aprendizado ativo, em que uma estratégia, chamada de (Aprendizagem Semisupervisionado Ativo) (ASSL), é introduzida. ASSL é uma proposta de integração de aprendizado semisupervisionado, e critérios de redução a-priori e organização [74]. A proposta difere da aprendizagem ativa padrão em que todas as amostras no banco de dados são classificadas e/ou reorganizadas em cada iteração de aprendizagem, produzindo um conjunto de aprendizagem substancialmente reduzido.

Todas as propostas apresentadas foram avaliadas e comparadas com abordagens tradicionais e recentes em cada área. No caso de problemas de único rótulo, nós apresentamos os prós e contras de cada proposta semisuperivisionada por Floresta de Caminhos Ótimos em comparação com duas abordagens de aprendizado supervisionado e quatro métodos semisuperivionados, sobre 8 conjunto de dados com uma diversidade de dimensões no espaço de atributos. OPFSEMI$_{mst+knn}$ e OPFSEMI$_{mst}$ superaram todas as abordagens na maioria dos casos analisados e estatisticamente confirmados pelos testes estatísticos Friedman com teste pos-hoc Nemenyi.

Em problemas multirótulo, experimentos envolvendo oito conjuntos de dados demonstraram que OPFSEMI$_{mst+knn}$ pode superar LapSVM, TSVM e a versão anterior OPFSEMI$_{mst}$, usando quatro dos principais métodos de transformação de dados e dois métodos de adapatação (MLkNN e BPMLL), considerando várias proporções de amostras supervisionadas e não supervisionadas. Na avaliação, usando aprendizado ativo, experimentalmente mostramos que a nova proposta ASSL melhora significativamente a seleção de amostras para supervisão de rótulos e diminui os erros de propagação de rótulos no conjunto não supervisionado ao longo das iterações de aprendizagem semisupervisionada e ativa.

5.2 Trabalhos futuros

A partir de todas as propostas apresentadas e publicações que consolidaram os experimentos, acreditamos que fortalecemos a área de aprendizagem semisupervisionada. Todos as técnicas implementadas estarão disponíveis na nova versão da LibOPF 3.0. As implementações estão sendo organizadas para oferecer da melhor forma e simplicidade o uso de cada técnica, disponibilizando assim para a comunidade em geral uma nova metodologia de aprendizado de máquina usando Floresta de Caminhos Ótimos. Todos os códigos gerados nesse novo pacote de implementação foram usados para a realização dos experimentos descritos anteriormente.

A grande diversidade de possibilidades que cerca o problema de aprendizado semisu-

pervisionado o torna desafiador. No decorrer de cada proposta elaborada nos deparamos com diversos desafios e extensões de trabalhos que estaremos futuramente trabalhando, entre elas podemos citar: avaliação de situações problemáticas no uso da Floresta de Caminhos Ótimos na propagação do rótulo sobre pressupostos não atendidos (suavidade, formação de grupos, separação de baixa densidade e geração de coleções); investigação de agrupamento e regressão semisupervisionado; aplicação de técnicas de pré processamento de dados para redução de dimensionalidade e uso posterior de aprendizado semisupervisionado; criação de uma técnica de aprendizado semisupervisionado para problemas multirótulo baseada em modelos adaptativos; avaliação mais profunda sobre aprendizado semisupervisionado para problemas com grande volume de dados; e por fim a criação de novos métodos de aprendizado ativo com OPFSEMI$_{mst+knn}$.

Referências Bibliográficas

[1] Faces. biometrics database distribution. In *The Computer Vision Laboratory, University of Notre Dame*, 2011.

[2] M.M. Adankon and M. Cheriet. Help-training for semi-supervised support vector machines. *Pattern Recognition*, 44(9):2220 – 2230, 2011. Computer Analysis of Images and Patterns.

[3] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In *PROCEEDINGS OF THE 1993 ACM SIGMOD INTERNATIONAL CONFERENCE ON MANAGEMENT OF DATA, WASHINGTON DC (USA*, pages 207–216, 1993.

[4] F. Alimoglu and E. Alpaydin. Methods of combining multiple classifiers based on different representations for pen-based handwriting recognition. In *Proceedings of the Fifth Turkish Artificial Intelligence and Artificial Neural Networks Symposium (TAINN 96)*, Jun 1996.

[5] W.P. Amorim, A.X. Falcão, and M.H. Carvalho. Semi-supervised pattern classification using optimum-path forest. *Conference on Graphics, Patterns and Images (SIBGRAPI), 2014 27th SIBGRAPI*, pages 111–118, Aug 2014.

[6] W.P. Amorim, A.X. Falcão, J.P. Papa, and M.H Carvalho. Improving semi-supervised learning through optimum connectivity. *Pattern Recognition*, 60:72 – 85, 2016.

[7] W.P. Amorim, H. Pistori, M.C. Pereira, and M.A.C. Jacinto. Attributes reduction applied to leather defects classification. *In: 23rd SIBGRAPI - Conference on Graphics, Patterns and Images - Gramado, Rio Grande do Sul*, 2010.

[8] T.W. Anderson. Asymptotic theory for principal component analysis. *Ann. Math. Statist.*, 34:122–148, 1963.

[9] S. Basu, A. Banerjee, and R.J. Mooney. Semi-Supervised clustering by seeding. In *Proceedings of the Nineteenth International Conference on Machine Learning, ICML '02*, pages 27–34, San Francisco, CA, 2002. Morgan Kaufmann.

[10] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *J. Mach. Learn. Res.*, 7:2399–2434, December 2006.

[11] J.O. Berger. *Statistical decision theory and Bayesian analysis*. Springer series in statistics. Springer, New York, NY [u.a.], 2. ed edition, 1985.

[12] P.J. Bickel and B. Li. *Local polynomial regression on unknown manifolds*, volume Volume 54 of *Lecture Notes–Monograph Series*, pages 177–186. Institute of Mathematical Statistics, Beachwood, Ohio, USA, 2007.

[13] C.M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Inc., New York, NY, USA, 1995.

[14] A. Blum, J.D. Lafferty, M.R. Rwebangira, and R. Reddy. Semi-supervised learning using randomized mincuts. In Carla E. Brodley, editor, *ICML*, volume 69 of *ACM International Conference Proceeding Series*. ACM, 2004.

[15] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory (COLT '98)*, pages 92–100, New York, 1998. ACM.

[16] B.E. Boser, I.M. Guyon, and V.N. Vapnik. A training algorithm for optimal margin classifiers. In David Haussler, editor, *Proceedings of the 5th Annual Workshop on Computational Learning Theory (COLT'92)*, pages 144–152, Pittsburgh, PA, USA, July 1992. ACM Press.

[17] N. Bridle and X. Zhu. p-voltages: Laplacian regularization for semi-supervised learning on high-dimensional data. *Workshop on Mining and Learning with Graphs (MLG2013)*, 2013.

[18] F.A.M. Cappabianco, A.X. Falcão, C.L. Yasuda, and J.K. Udupa. Brain tissue mr-image segmentation via optimum-path forest clustering. *Computer Vision and Image Understanding*, 116(10):1047–1059, July 2012.

[19] O. Chapelle, B. Schölkopf, and A. Zien. *Semi-supervised learning*. MIT Press, 2006.

[20] G. Chiachia, A.N. Marana, J.P. Papa, and A.X. Falcão. Infrared face recognition by optimum-path forest. In *Systems, Signals and Image Processing, 2009. IWSSIP 2009. 16th International Conference on*, pages 1–4, 2009.

[21] A. Clare and R.D. King. Knowledge discovery in multi-label phenotype data. In *Lecture Notes in Computer Science*, pages 42–53. Springer, 2001.

[22] R. Collobert, F. Sinz, J. Weston, and L. Bottou. Large scale transductive svms. *J. Mach. Learn. Res.*, 7:1687–1712, December 2006.

[23] C. Cortes and V. Vapnik. Support-vector networks. *Mach. Learn.*, 20(3):273–297, September 1995.

[24] L.F. Cranor and B.A. Lamacchia. Spam! *Commun. ACM*, 41(8):74–83, August 1998.

[25] A.T. da Silva, A.X. Falcão, and L. Magalhães. Active learning paradigms for cbir systems based on optimum-path forest classification. *Pattern Recognition*, 44:2971–2978, December 2011.

[26] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.*, 7:1–30, December 2006.

[27] K. Driessens, P. Reutemann, B. Pfahringer, and C. Leschi. Using weighted nearest neighbor to benefit from unlabeled data. In Wee-Keong Ng, Masaru Kitsuregawa, Jianzhong Li, and Kuiyu Chang, editors, *Advances in Knowledge Discovery and Data Mining*, volume 3918 of *Lecture Notes in Computer Science*, pages 60–69. Springer Berlin Heidelberg, 2006.

[28] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. Wiley, New York, 2 edition, 2001.

[29] A.X. Falcão, J. Stolfi, and R.A. Lotufo. The image foresting transform: Theory, algorithms, and applications. *IEEE Transactions on Patterns Analysis and Machine Intelligence*, 26(1):19–29, 2004.

[30] C. Feng, A. Sutherland, R. King, S. Muggleton, and R. Henery. Comparison of machine learning classifiers to statistics and neural networks. In *Proceedings of the Third International Workshop in Artificial Intelligence and Statistics*, pages 41–52, 1993.

[31] P.W. Frey and D.J. Slate. Letter recognition using holland-style adaptive classifiers. *Machine Learning*, 6(2):161–182, March 1991.

[32] M. Friedman. A Comparison of Alternative Tests of Significance for the Problem of m Rankings. *The Annals of Mathematical Statistics*, 11(1):86–92, 1940.

[33] H. Gan, N. Sang, R. Huang, X. Tong, and Z. Dan. Using clustering analysis to improve semi-supervised classification. *Neurocomputing*, 101:290 – 298, 2013.

[34] J.F. Gantz, C. Chute, A. Manfrediz, S. Minton, D. Reinsel, W. Schlichting, and A. Toncheva. The Diverse and Exploding Digital Universe: An Updated Forecast of Worldwide Information Growth Through 2011. An idc white paper - sponsored by emc, IDC, March 2008.

[35] R. Ghani. Combining labeled and unlabeled data for multiclass text categorization. In *Proceedings of the Nineteenth International Conference on Machine Learning, ICML '02*, pages 187–194, San Francisco, CA, 2002. Morgan Kaufmann.

[36] David F. Gleich and Michael W. Mahoney. Using local spectral methods to robustify graph-based learning algorithms. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, pages 359–368, New York, NY, USA, 2015. ACM.

[37] A. Goldberg, X. Zhu, A. Furger, and J. Xu. Oasis: Online active semi-supervised learning. In *AAAI*, 2011.

[38] A.B. Goldberg, X. Zhu, A. Singh, Z. Xu, and R.D. Nowak. Multi-manifold semi-supervised learning. In D.A.V. Dyk and M. Welling, editors, *AISTATS*, volume 5 of *JMLR Proceedings*, pages 169–176. JMLR.org, 2009.

[39] C. Gong, T. Liu, D. Tao, K. Fu, E. Tu, and J. Yang. Deformed graph laplacian for semisupervised learning. *Neural Networks and Learning Systems, IEEE Transactions on*, PP(99):1–1, 2015.

[40] I.R. Guilherme, A.N. Marana, J.P. Papa, G. Chiachia, L.C.S. Afonso, K. Miura, M.V.D. Ferreira, and F. Torres. Petroleum well drilling monitoring through cutting image analysis and artificial intelligence techniques. *Engineering Applications of Artificial Intelligence*, 24(1):201 – 207, 2011.

[41] X. He, M. Ji, and H. Bao. A unified active and semi-supervised learning framework for image compression. In *CVPR*, pages 65–72, 2009.

[42] G. Huang, S. Song, J.N.D. Gupta, and C. Wu. Semi-supervised and unsupervised extreme learning machines. *Cybernetics, IEEE Transactions on*, 44(12):2405–2417, Dec 2014.

[43] G. Huang, Q. Zhu, and C. Siew. Extreme learning machine: a new learning scheme of feedforward neural networks. In *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, volume 2, pages 985–990 vol.2, July 2004.

[44] Te Ming Huang, Vojislav Kecman, and Ivica Kopriva. *Kernel Based Algorithms for Mining Huge Data Sets: Supervised, Semi-supervised, and Unsupervised Learning*, volume 17 of *Studies in Computational Intelligence*. Springer, 2006.

[45] L.J. Hubert. Some applications of graph theory to clustering. *Psychometrika*, 39(3):283–309.

[46] A.I. Iliev, M.S. Scordilis, J.P. Papa, and A.X. Falcão. Spoken emotion recognition through optimum-path forest classification using glottal features. *Comput. Speech Lang.*, 24(3):445–460, July 2010.

[47] A. Iosifidis, A. Tefas, and I. Pitas. Regularized extreme learning machine for multi-view semi-supervised action recognition. *Neurocomputing*, 145:250–262, 2014.

[48] A. S. Iwashita, J. P. Papa, A. N. Souza, A. X. Falcão, R. A. Lotufo, V. M. Oliveira, Victor H. de Albuquerque, and João M. Tavares. A Path- and Label-cost Propagation Approach to speedup the Training of the Optimum-Path Forest Classifier. *Pattern Recognition Letters*, January 2014.

[49] A.K. Jain and R.C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.

[50] T. Joachims. Transductive inference for text classification using support vector machines. In *Proceedings of the Sixteenth International Conference on Machine Learning*, ICML '99, pages 200–209, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[51] F. Johannes. Separate-and-conquer rule learning. *Artif. Intell. Rev.*, 13(1):3–54, February 1999.

[52] F. Kang, R. Jin, and R. Sukthankar. Correlated label propagation with application to multi-label learning. In *2006 IEEE Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1719–1726, 2006.

[53] L. Kaufman and P.J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis.* John Wiley, 1990.

[54] John D. Lafferty and Larry A. Wasserman. Statistical analysis of semi-supervised regression. In John C. Platt, Daphne Koller, Yoram Singer, and Sam T. Roweis, editors, *NIPS*, pages 801–808. Curran Associates, Inc., 2007.

[55] C.H. Li and P.C. Yuen. Semi-supervised learning in medical image database. In *Proceedings of the 5th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, PAKDD '01, pages 154–160, London, UK, UK, 2001. Springer-Verlag.

[56] K. Li, X. Luo, and M. Jin. Semi-supervised learning for svm-knn. *JCP*, 5(5):671–678, 2010.

[57] W. Li, L. Duan, D. Xu, and I.W. Tsang. Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(6):1134–1148, June 2014.

[58] Y.-F. Li and Z.-H. Zhou. Towards making unlabeled data never hurt. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(1):175–188, Jan 2015.

[59] C.-L. Liu, W.-H. Hsaio, C.-H. Lee, T.-H. Chang, and T.-H. Kuo. Semi-supervised text classification with universum learning. *Cybernetics, IEEE Transactions on*, PP(99):1–1, 2015.

[60] X. Liu, T. Guo, L. He, and X. Yang. A low-rank approximation-based transductive support tensor machine for semisupervised classification. *Image Processing, IEEE Transactions on*, 24(6):1825–1838, June 2015.

[61] G. Madjarov, D. Kocev, D. Gjorgjevikj, and S. Deroski. An extensive experimental comparison of methods for multi-label learning. *Pattern Recognition*, 45(9):3084–3104, September 2012.

[62] R. Minetto, J.P. Papa, T.V. Spina, A.X. Falcão, N.J. Leite, and J. Stolfi. Fast and robust object tracking using image foresting transform. In *Systems, Signals and Image Processing, 2009. IWSSIP 2009. 16th International Conference on*, pages 1–4, 2009.

[63] I. Muslea, S. Minton, and C.A. Knoblock. Active + semi-supervised learning = robust multi-view learning. In *ICML*, pages 435–442, 2002.

[64] P. Niyogi. Manifold regularization and semi-supervised learning: Some theoretical analyses. *Journal of Machine Learning Research*, 14:1229–1250, 2013.

[65] J.P. Papa and A.X. Falcão. A new variant of the optimum-path forest classifier. *In 4th Internation Symposium on Visual Computing*, 2008.

[66] J.P. Papa, A.X. Falcão, V.H.C. Albuquerque, and J.M.R.S Tavares. Efficient supervised optimum-path forest classification for large datasets. *Pattern Recogn.*, 45(1):512–520, January 2012.

[67] J.P. Papa, A.X. Falcão, G.M. Freitas, and A.M.H. Ávila. Robust pruning of training patterns for optimum-path forest classification applied to satellite-based rainfall occurrence estimation. *IEEE Geoscience and Remote Sensing Letters*, v7 i2:396–400, 2010.

[68] J.P. Papa, A.X. Falcão, A.L.M. Levada, D.C. Corrêa, D.H.P. Salvadeo, and N.D.A. Mascarenhas. Fast and accurate holistic face recognition using optimum-path forest. In *Proceedings of the 16th international conference on Digital Signal Processing*, DSP'09, pages 781–786, Piscataway, NJ, USA, 2009. IEEE Press.

[69] J.P. Papa, A.X. Falcão, and C.T.M. Suzuki. Supervised pattern classification based on optimum-path forest. *International Journal of Imaging Systems and Technology*, pages 120–131, 2009.

[70] J.P. Papa, A.N. Marana, A.A. Spadotto, R.C. Guido, and A.X. Falcão. Robust and fast vowel recognition using optimum-path forest. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 2190–2193, 2010.

[71] J.R. Quinlan. Induction of decision trees. *Mach. Learn.*, 1(1):81–106, March 1986.

[72] L.M. Rocha, F. A.M. Cappabianco, and A.X. Falcão. Data clustering as an optimum-path forest problem with applications in image analysis. *International Journal of Imaging Systems and Technology*, 5(19):50–68, 2009.

[73] C. Rosenberg, M. Hebert, and H. Schneiderman. Semi-supervised self-training of object detection models. In *Application of Computer Vision, 2005. WACV/MOTIONS '05 Volume 1. Seventh IEEE Workshops on*, volume 1, pages 29–36, 2005.

[74] P.T.M. Saito, de P.J. Rezende, A.X. Falcão, C.T.N. Suzuki, and J.F. Gomes. An active learning paradigm based on a priori data reduction and organization. In *Expert Systems with Applications*, pages 1–23, 2014.

[75] P.T.M. Saito, R.Y.M. Nakamura, W.P. Amorim, J.P. Papa, P.J. Rezende, and A.X. Falcão. Choosing the most effective pattern classification model under learning-time constraint. *PLoS ONE*, 10(6):e0129947, 06 2015.

[76] C.T.N. Suzuki, J.F. Gomes, A.X. Falcão, J.P. Papa, and S.H. Shimizu. Automatic segmentation and classification of human intestinal parasites from microscopy images. *IEEE Trans. on Biomedical Engineering*, 60(3):803–812, March 2013.

[77] M. Tavallaee, E. Bagheri, L. Wei, and A. A. Ghorbani. A detailed analysis of the kdd cup 99 data set. In *Computational Intelligence for Security and Defense Applications, 2009. CISDA 2009. IEEE Symposium on*, pages 1–6, july 2009.

[78] V.N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.

[79] C. Wang, W. Chen, P. Yin, and J. Wang. Semi-supervised clustering using incomplete prior knowledge. In *Proceedings of the 7th international conference on Computational Science, Part I: ICCS 2007*, ICCS '07, pages 192–195, Berlin, Heidelberg, 2007. Springer-Verlag.

[80] H. Yang, K. Huang, I. King, and M.R. Lyu. Maximum margin semi-supervised learning with irrelevant data. *Neural Networks*, 70:90 – 102, 2015.

[81] L. Yang, L. Wang, Y. Gao, Q. Sun, and T. Zhao. A convex relaxation framework for a class of semi-supervised learning methods and its application in pattern recognition. *Engineering Applications of Artificial Intelligence*, 35:335 – 344, 2014.

[82] D. Yu, B. Varadarajan, L. Deng, and A. Acero. Active learning and semi-supervised learning for speech recognition: A unified framework using the global entropy reduction maximization criterion. *Comput. Speech Lang.*, 24(3):433–444, 2010.

[83] C.T. Zahn. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Trans. Comput.*, 20(1):68–86, January 1971.

[84] L. Zhang, W. Wu, T. Chen, N. Strobel, and D. Comaniciu. Robust object tracking using semi-supervised appearance dictionary learning. *Pattern Recognition Letters*, 62:17 – 23, 2015.

[85] M. Zhang and Z. Zhou. Multilabel neural networks with applications to functional genomics and text categorization. *IEEE Transactions on Knowledge and Data Engineering*, 18(10):1338–1351, 2006.

[86] M. Zhang and Z. Zhou. Ml-knn: A lazy learning approach to multi-label learning. *Pattern Recognition*, 40(7):2038–2048, July 2007.

[87] M. Zhao, T.W.S. Chow, Z. Zhang, and B. Li. Automatic image annotation via compact graph based semi-supervised learning. *Knowledge-Based Systems*, 76:148 – 165, 2015.

[88] S. Zhu, X. Ji, W. Xu, and Y. Gong. Multi-labelled classification using maximum entropy method. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, pages 274–281, New York, NY, USA, 2005. ACM.

[89] X. Zhu. Semi-supervised learning literature survey. *Technical Report 1530, Computer Sciences, University of Wisconsin-Madison*, 2008.

[90] X. Zhu and Z. Ghahramani. Learning from labeled and unlabeled data with label propagation, 2002.

[91] X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *IN ICML*, pages 912–919, 2003.

[92] X. Zhu, J. Lafferty, and Z. Ghahramani. Combining active learning and semi-supervised learning using gaussian fields and harmonic functions. In *Proc. of the ICML-2003 Workshop on the Continuum from Labeled to Unlabeled Data*, pages 58–65, Washington, DC, 2003.

[93] X. Zhu, J. Lafferty, and Z. Ghahramani. Combining active learning and semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, pages 58–65, 2003.