

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

Hygor Xavier Araújo

Uma busca ordenada *branch-and-bound* para solução do problema de classificação semissupervisionada usando classificadores de larga margem

Juiz de Fora

2019

Hygor Xavier Araújo

Uma busca ordenada *branch-and-bound* para solução do problema de classificação semissupervisionada usando classificadores de larga margem

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

Orientador: Prof. D.Sc. Saulo Moraes Villela

Coorientador: Prof. D.Sc. Raul Fonseca Neto

Juiz de Fora

2019

Ficha catalográfica elaborada através do Modelo Latex do CDC da UFJF
com os dados fornecidos pelo(a) autor(a)

Xavier Araújo, Hygor.

Uma busca ordenada *branch-and-bound* para solução do problema de classificação semissupervisionada usando classificadores de larga margem / Hygor Xavier Araújo. – 2019.

45 f. : il.

Orientador: Saulo Moraes Villela

Coorientador: Raul Fonseca Neto

Dissertação (Mestrado) – Universidade Federal de Juiz de Fora, Instituto de Ciências Exatas. Programa de Pós-Graduação em Ciência da Computação, 2019.

1. Inferência transdutiva. 2. Aprendizado semissupervisionado. 3. Busca ordenada admissível. 4. Máquina de vetores suporte. 5. Separação de baixa densidade. I. Villela, Saulo Moraes, orient. II. Fonseca Neto, Raul, coorient. III. Título.

Hygor Xavier Araújo

Uma busca ordenada *branch-and-bound* para solução do problema de classificação semissupervisionada usando classificadores de larga margem

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

Aprovada em 5 de setembro de 2019

BANCA EXAMINADORA

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

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

Prof. D.Sc. Carlos Cristiano Hasenclever Borges
Universidade Federal de Juiz de Fora

Prof. D.Sc. Saul de Castro Leite
Universidade Federal do ABC

AGRADECIMENTOS

A todos os meus familiares, em especial meus pais Claudyr e Judith, e irmã Laryssa, pelo apoio incondicional e por serem a base que me permite buscar novas conquistas.

A minha namorada Ana Paula, por todo seu incentivo, apoio e companheirismo durante todos os momentos.

Aos meus orientadores, Saulo e Raul, por todo apoio, confiança e conhecimentos transmitidos a mim.

A todos colegas, professores e técnicos do PPGCC, que de alguma forma contribuíram para que fosse possível chegar ao fim deste processo.

RESUMO

Para a solução do problema de classificação através da inferência transdutiva, é necessário encontrar os rótulos de um conjunto previamente definido. No entanto, calcular a melhor rotulação dessas amostras é um problema combinatorial NP-difícil. Neste trabalho, um método que combina os métodos de busca *branch-and-bound* e *best-first* é proposto para resolver o problema de rotulação buscando pela solução ótima. Para orientar a busca, foram usados classificadores baseados em margem, como a Máquina de Vetores Suporte (*Support Vector Machine* – SVM), e uma função de avaliação monótona com base nos valores de margem deste classificador, o que leva à solução globalmente ótima. Para lidar com o alto custo computacional da solução de máxima margem, também foi proposta uma solução heurística que é usada como um limite inferior sendo computado em tempo constante através da solução de um problema de classificação com o SVM. Comparando o método proposto com a Máquina de Vetores Suporte Transdutiva (*Transductive Support Vector Machine* – TSVM), os resultados mostraram melhorias significativas no tempo de execução e valores superiores de margem. Além disso, duas novas heurísticas são apresentadas para reduzir o número de estados explorados e acelerar a exploração do espaço de busca. O método e suas heurísticas são avaliados e comparados ao SVM e ao TSVM, mostrando resultados competitivos.

Palavras-chave: Inferência transdutiva. Aprendizado semissupervisionado. Busca ordenada admissível. Máquina de vetores suporte. Separação de baixa densidade.

ABSTRACT

To solve the classification problem through the transductive inference, it is necessary to find the labels of a previously defined set. However, computing the best labeling of these samples is an NP-hard combinatorial problem. In this work, a method that combines the branch-and-bound and the best-first search methods is proposed to solve the labeling problem by searching for the optimal solution. To guide the search, margin-based classifiers, such as the Support Vector Machine (SVM), and a monotone evaluation function based on the margin values of this classifier were used, leading to the optimal global solution. To deal with the high computational cost of the maximum margin solution, we also propose a heuristic solution that is used as a lower bound, being computed in constant time by solving a classification problem with SVM. Comparing our method with the Transductive Support Vector Machine (TSVM), the results showed significant improvements in the runtime and higher margin values. Furthermore, two new heuristics are presented to reduce the number of explored states and speed up the exploration of the search space. The method and its heuristics are evaluated and compared to SVM and TSVM, showing competitive results.

Keywords: Transductive inference. Semi-supervised learning. Best-first search. Support vector machine. Low density separation.

LISTA DE FIGURAS

Figura 1 – O processo de ramificação.	21
Figura 2 – Processo de computação dos limites superior (linha tracejada) e inferior (linha sólida).	23
Figura 3 – O processo de ramificação com seleção de duas amostras.	25
Figura 4 – Comparação entre uma árvore de estados completa e compacta. Em cada estado é representado o conjunto de treinamento (amostras +/-) e o conjunto de trabalho (pontos pretos). Os estados O1 e C1 são equivalentes, assim como os estados O3 e C2.	26

LISTA DE TABELAS

Tabela 1 – Informações sobre as bases utilizadas nos experimentos de margem.	29
Tabela 2 – Comparação entre os valores de margem do TSVM e do BFBB-TC.	29
Tabela 3 – Comparação dos resultados do BFBB-TC em comparação ao TSVM.	31
Tabela 4 – Informações sobre as bases utilizadas nos experimentos com <i>kernel</i> linear.	32
Tabela 5 – Resultados de inferência dos experimentos com <i>kernel</i> linear. . . .	33
Tabela 6 – Resultados de exploração dos experimentos com <i>kernel</i> linear. . .	35
Tabela 7 – Informações sobre as bases utilizadas nos experimentos com <i>kernel</i> RBF.	36
Tabela 8 – Resultados de inferência dos experimentos com <i>kernel</i> RBF (1). . .	37
Tabela 9 – Resultados de inferência dos experimentos com <i>kernel</i> RBF (2). . .	38
Tabela 10 – Resultados de inferência dos experimentos com <i>kernel</i> RBF (3). . .	39
Tabela 11 – Resultados de exploração dos experimentos com <i>kernel</i> linear (1).	40
Tabela 12 – Resultados de exploração dos experimentos com <i>kernel</i> linear (2).	40
Tabela 13 – Resultados de exploração dos experimentos com <i>kernel</i> linear (3).	41

LISTA DE ABREVIATURAS E SIGLAS

AOS	<i>Admissible Ordered Search</i>
BFBB-TC	<i>Best-First Branch-and-Bound Transductive Classifier</i>
PTSVM	<i>Progressive Transductive Support Vector Machine</i>
RBF	<i>Radial Basis Function</i>
S ³ VMS	<i>Semi-Supervised Support Vector Machine</i>
SVM	<i>Support Vector Machine</i>
TSVM	<i>Transductive Support Vector Machine</i>
WellSVM	<i>Weakly Labeled SVM</i>

SUMÁRIO

1	INTRODUÇÃO	10
1.1	TRABALHOS RELACIONADOS	11
1.2	OBJETIVOS	13
1.3	CONTRIBUIÇÕES	13
1.4	ORGANIZAÇÃO	13
2	FUNDAMENTAÇÃO TEÓRICA	14
2.1	APRENDIZADO SEMISSUPERVISIONADO	14
2.2	CLASSIFICAÇÃO BINÁRIA	14
2.3	MÁQUINA DE VETORES SUPORTE	15
2.3.1	Funções <i>kernel</i>	16
2.4	INFERÊNCIA TRANSDUTIVA E INDUTIVA	17
2.5	<i>BRANCH-AND-BOUND</i>	18
2.6	BUSCA ORDENADA	18
3	CLASSIFICADOR TRANSDUTIVO COM BUSCA ORDE-	
	NADA E BRANCH-AND-BOUND	20
3.1	ESPAÇO DE ESTADOS E BUSCA HEURÍSTICA	20
3.2	RAMIFICAÇÃO	20
3.3	AVALIAÇÃO E PODA	21
3.4	LIMITES INFERIOR E SUPERIOR	22
3.5	MÚLTIPLAS AMOSTRAS E JANELA	24
3.6	PSEUDOCÓDIGO	24
4	ANÁLISE EXPERIMENTAL E RESULTADOS	28
4.1	AVALIAÇÃO DA MARGEM DE MÉTODOS TRANSDUTIVOS	28
4.2	AVALIAÇÃO DA MARGEM E GENERALIZAÇÃO DE MÉTODOS SEMISSUPERVISIONADOS	31
4.2.1	Experimentos com <i>kernel</i> linear	32
4.2.2	Experimentos com <i>kernel</i> RBF	35
5	CONCLUSÕES E TRABALHOS FUTUROS	42
	REFERÊNCIAS	44

1 INTRODUÇÃO

O processo para criação de um conjunto de dados que possa ser utilizado para resolver um problema de classificação é, frequentemente, custoso, demorado e podendo exigir o conhecimento de um especialista para realização do mesmo. Este desafio pode ser um impedimento para obtenção de bons resultados ao se aplicar métodos de aprendizado supervisionado, que dependem de conjuntos de dados em que suas amostras estejam totalmente rotuladas para serem usadas durante o aprendizado. Do contrário, a aquisição de novos dados não rotulados pode ser relativamente fácil, barata e rápida, o que pode ser adequado para métodos de aprendizado não supervisionado, em que os rótulos não são utilizados, porém seu uso na aprendizagem de um modelo para classificação não seria tão direta. Nesses casos, o uso de métodos de aprendizado semissupervisionado podem oferecer uma melhor abordagem ao utilizar amostras rotuladas e não rotuladas, também chamados de conjuntos de treinamento e de trabalho, respectivamente, para treinar um modelo de classificação.

Para obter a solução ótima de um problema de classificação binária utilizando amostras não rotuladas (conjunto de trabalho), é preciso encontrar o melhor esquema de rotulação das mesmas. No entanto, avaliar todas as possíveis combinações de rotulações torna este um problema combinatório NP-difícil, que é computacionalmente proibitivo para conjuntos de dados com grandes tamanhos de amostras não rotuladas. Para um problema de classificação binária com um conjunto de trabalho de tamanho m , existem 2^m possíveis soluções de rotulação.

Para lidar com o aspecto combinatório do problema, utilizou-se o método *branch-and-bound* em conjunto com uma busca ordenada *best-first* que explora eficientemente o espaço de soluções. Essa combinação foi aplicada por Villela *et al.* (2015) com o algoritmo Busca Ordenada Admissível (*Admissible Ordered Search* – AOS), para resolver o problema de seleção de características. Inicialmente, aplicou-se essa mesma combinação no método Classificador Transdutivo com Busca Ordenada e *Branch-and-Bound* (*Best-First Branch-and-Bound Transductive Classifier* – BFBB-TC), para resolver o problema de classificação binária em uma configuração de aprendizado semissupervisionado, porém concentrando-se na realização e avaliação da inferência transdutiva (Araújo *et al.*, 2018, 2019). Neste trabalho, são realizados novos experimentos demonstrando a capacidade do algoritmo transdutor e, além disso, também foi realizada uma avaliação sob a perspectiva da inferência indutiva.

Para definir uma função de decisão para guiar a busca, levou-se em consideração a suposição de agrupamento: se amostras pertencem ao mesmo agrupamento, provavelmente elas pertencem a mesma classe. Esta suposição é frequentemente aplicada em métodos semissupervisionados e também pode ser interpretada em relação ao limite de decisão, que

supostamente deve estar em regiões de baixa densidade (Chapelle *et al.*, 2006a). Dada essa suposição, faz sentido utilizar classificadores baseados em margem, como a Máquina de Vetores Suporte (*Support Vector Machine* – SVM), no método proposto, devido a sua característica de encontrar um limite de decisão em uma região do espaço que aumente a distância entre as classes, ou seja, uma região de baixa densidade. Desta forma, os valores de margem do SVM são utilizados como função de decisão, pois à medida que cada nova amostra do conjunto de trabalho é adicionada ao conjunto de treinamento, o valor da margem só pode diminuir ou permanecer o mesmo, tornando-a uma função monotonicamente decrescente, o que satisfaz a condição de utilização de uma função admissível para a busca.

O algoritmo BFBB-TC é capaz de encontrar a solução globalmente ótima ao usar o SVM como um classificador de margem. Encontrar a solução ótima, no entanto, tem um alto custo computacional e o método só é utilizável para conjuntos de trabalho de tamanho médio. Para superar este problema, é também feita a proposta do uso de soluções heurísticas que obtenham uma solução competitiva e que permitam a aplicação em conjuntos de trabalho maiores devido ao seu menor custo computacional. Estas soluções heurísticas são usadas para lidar com o número de estados explorados durante a busca e também como um limite inferior no processo *branch-and-bound*.

1.1 TRABALHOS RELACIONADOS

Gammerman *et al.* (1998) propuseram o primeiro método para inferência transdutiva em problemas de classificação binária. O método é uma modificação do SVM e atribui a cada nova amostra um valor de previsão combinado com um grau de confiança baseado no pressuposto de que a nova amostra poderia ser ou não um vetor suporte em qualquer uma das classes. Portanto, este não é um método combinatório que encontre a hipótese de margem máxima.

Graepel *et al.* (1999) modelaram o problema de inferência transdutiva por uma perspectiva bayesiana. Neste contexto, a probabilidade do rótulo de uma nova amostra é determinada como a medida posterior do subconjunto correspondente do espaço de hipóteses. Nesse sentido, os autores consideram que a probabilidade dos rótulos é determinada pela razão do volume no espaço de versões, porque uma nova amostra divide o espaço de versões em dois subespaços de acordo com as duas possibilidades de rotulação. No entanto, a principal desvantagem dessa abordagem é a dependência de uma técnica eficiente para calcular o maior volume dos subespaços.

Bennett e Demiriz (1999) apresentam a Máquina de Vetores Suporte Semissupervisionada (*Semi-Supervised Support Vector Machine* – S³VM) com o foco de resolver o problema de transdução, encontrando apenas os rótulos do conjunto de trabalho ao encontrar o hiperplano de margem máxima. Nesse sentido é mostrado que o problema de

otimização do SVM pode ser modificado para incluir o conjunto de trabalho e transformá-lo em um problema de programação inteira mista, que pode ser resolvido por métodos de programação inteira. Para tornar o problema mais fácil de resolver, os autores tentam minimizar a norma L_1 do vetor normal, definindo um modelo de programação linear robusto com variáveis binárias. Este método é prático apenas para resolver problemas de pequeno porte.

A Máquina de Vetores Suporte Transdutiva (*Transductive Support Vector Machine* – TSVM), apresentada por Joachims (1999), realiza uma pesquisa local ao rotular todo o conjunto de trabalho e, em seguida, realiza alterações nos rótulos encontrados, invertendo os rótulos a cada duas amostras selecionadas enquanto há uma melhoria na função objetivo. O método foi aplicado inicialmente no contexto de classificação de texto. Como não é um método exato e usa uma forma de busca local, é projetado para lidar com bases de dados de tamanho grande. Uma característica do método é a necessidade de informar a distribuição entre as classes positiva e negativa do conjunto de trabalho, uma informação que não se espera ser conhecida, usando, por padrão, a mesma proporção do conjunto de treinamento.

Na tentativa de estender o TSVM para lidar com diferentes distribuições de classes, a Máquina de Vetores Suporte Transdutiva Progressiva (*Progressive Transductive Support Vector Machine* – PTSVM) é proposta por Chen *et al.* (2003). Neste método, o conjunto de trabalho não é rotulado ao início como no TSVM, ao invés disso, a rotulação é feita gradualmente de acordo com uma estratégia de seleção de amostras. A estratégia de seleção de amostras utilizada consiste em selecionar uma ou duas amostras que violem o limite da margem e sejam as mais próximas ao limite de cada classe. Caso alguma inconsistência seja encontrada com amostras previamente rotuladas, onde a rotulação se torna errada na nova solução encontrada, a rotulação é removida. Com o aumento do tamanho do conjunto de trabalho e pela estratégia de rotulação utilizada, a complexidade do método cresce mais rapidamente que o TSVM.

Chapelle *et al.* (2007) apresentam uma formulação do S^3VM usando a técnica *branch-and-bound* para obter a solução ótima global, tentando aprender a suposição do separador de baixa densidade. O método é muito semelhante ao método proposto aqui, mas difere nos três principais processos: ramificação, poda e exploração, e é apropriado apenas para bases de dados de tamanho pequeno. Como será visto no Capítulo 3, foram implementadas estratégias alternativas para esses processos, tornando o modelo proposto mais eficiente e aplicável em bases de dados de tamanho médio.

Finalmente, Li *et al.* (2013) propõem o algoritmo *Weakly Labeled SVM* (WellSVM). Neste trabalho, o uso de dados fracamente rotulados (*weakly labeled data*) é analisado sob a perspectiva de três diferentes tarefas de aprendizado: aprendizado semissupervisionado, aprendizado multi-instância (*multiple-instance learning*) e agrupamento (*clustering*). O

método convexo apresentado aplica uma estratégia iterativa de geração de vetores de rótulos mais violados, e realiza a combinação dos mesmos através de técnicas de aprendizado de múltiplos *kernels*.

1.2 OBJETIVOS

Este trabalho apresenta os seguintes objetivos:

- Desenvolvimento do método BFBB-TC (Araújo *et al.*, 2018, 2019) para solução de problemas com conjuntos de trabalho maiores ao utilizar uma abordagem heurística;
- Avaliação dos resultados de margem obtidos com diferentes bases de dados;
- Análise da aplicação do método ao utilizar um *kernel* diferente do linear em novas bases de dados;
- Análise da capacidade de generalização dos resultados do BFBB-TC com diferentes configurações.

1.3 CONTRIBUIÇÕES

As principais contribuições deste trabalho foram:

- Apresentação do Classificador Transdutivo com Busca Ordenada e *Branch-and-Bound* (*Best-First Branch-and-Bound Transductive Classifier* – BFBB-TC), focado na solução do problema de transdução;
- Desenvolvimento de uma heurística de limite inferior para o método BFBB-TC que permite obter uma solução viável a qualquer momento durante a exploração;
- Criação de duas novas heurísticas para a geração e eliminação de estados durante a busca, permitindo uma maior e mais rápida exploração do espaço de busca.

1.4 ORGANIZAÇÃO

Após esta introdução, o Capítulo 2 apresenta as bases teóricas utilizadas para o desenvolvimento deste trabalho, como aprendizado semissupervisionado, classificação binária, inferência transdutiva e indutiva, máquina de vetores suporte, *branch-and-bound* e busca ordenada. No Capítulo 3, o método BFBB-TC é apresentado com todas suas características e configurações. No Capítulo 4, são apresentados os resultados dos experimentos realizados e é feita a análise sobre os mesmos. Por fim, no Capítulo 5, são feitas as considerações finais sobre o trabalho e seus resultados, e são apresentadas sugestões de trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

A seguir são apresentadas as bases teóricas utilizadas no desenvolvimento desse trabalho.

2.1 APRENDIZADO SEMISSUPERVISIONADO

Na área de aprendizado de máquina, existem diferentes tipos de configurações de aprendizado. É possível citar entre as mais comumente observadas o aprendizado supervisionado, no qual têm-se como entrada amostras de dados que foram previamente rotuladas, e aprendizado não supervisionado, onde os dados de entrada não possuem qualquer rotulação. O aprendizado semissupervisionado pode ser definido como um método intermediário entre o supervisionado e não supervisionado Chapelle *et al.* (2006a). Esta definição vem das características dos dados de entrada usados para a tarefa de aprendizado semissupervisionado, onde são utilizadas amostras rotuladas e não rotuladas para o treinamento.

Dado um conjunto X_l de amostras $x_i \in \mathbb{R}^d$, um conjunto X_u de amostras $x_j \in \mathbb{R}^d$ e outro conjunto Y_l de valores escalares onde $i = 1, \dots, m$ e $j = m + 1, \dots, m + k$ com $i, j, m, k \in \mathbb{Z}^+$, é possível definir os dados de entrada para a aprendizagem semissupervisionada como dois conjuntos. O primeiro conjunto é chamado de conjunto de treinamento e é definido de forma semelhante ao conjunto de treinamento do aprendizado supervisionado $Z_l = \{z_i = (x_i, y_i) \mid x_i \in X_l; y_i \in Y_l\}$, onde as amostras x_i têm um rótulo y_i associado a elas. O outro conjunto incluído nos dados de entrada é o conjunto de trabalho e as amostras nele não possuem um rótulo, de forma semelhante aos dados de entrada no aprendizado não supervisionado. O conjunto de trabalho pode ser definido como $Z_u = \{z_j = x_j \mid x_j \in X_u\}$. Ambos os conjuntos são utilizados na fase de aprendizagem.

Uma das razões para a utilização de métodos semissupervisionados é obter uma solução melhor do que seria possível com o uso de métodos supervisionados apenas com os dados rotulados ou métodos não supervisionados com apenas dados não rotulados. Outro aspecto importante da aprendizagem semissupervisionada é que os métodos desenvolvidos neste cenário geralmente visam resolver problemas com pequenos conjuntos de treinamento e conjuntos de trabalho maiores, o que diminui o custo para criação de bases de dados para o treinamento já que, desta forma, haverá menos esforço para rotular amostras.

2.2 CLASSIFICAÇÃO BINÁRIA

Ao resolver um problema de classificação, deseja-se aprender uma função que seja capaz de separar corretamente nossas amostras de dados de entrada em classes diferentes e que essa função tenha uma boa generalização para novas amostras de dados. A função

aprendida define um limite de decisão e pode ser vista como um hiperplano, podendo ser definida pelo vetor de pesos $w \in \mathbb{R}^d$ e uma constante $b \in \mathbb{R}$ chamada de viés (*bias*).

Com a função do hiperplano $h(x) = w \cdot x + b$ é possível avaliar-se uma nova amostra $x_{new} \in \mathbb{R}^d$ e obter uma rotulação $y_{new} \in \mathbb{Z}$ para a mesma. Para que o problema seja definido como de classificação binária é preciso que os valores possíveis para o rótulo y_{new} sejam restritos a dois, por exemplo, $y_i \in \{-1, +1\}$.

Para algumas bases de dados não haverá um hiperplano capaz de separar as amostras, porque o conjunto de dados Z utilizado para o treinamento não é linearmente separável, ou seja, não é possível encontrar um hiperplano que separe as amostras em seu espaço original, porém em um espaço projetado de maior dimensão o mesmo se torna possível. Para um conjunto de treinamento linearmente separável, deseja-se encontrar (w, b) sujeito a $y_i(w \cdot x_i + b) \geq 0, \forall (x_i, y_i) \in Z$. Para Z aceitar uma margem $\gamma \geq 0$, deve haver um hiperplano $\mathcal{H} := \{x \in \mathbb{R}^d : w \cdot x + b = 0\}$ sujeito a $y_i(w \cdot x_i + b) \geq \gamma, \forall (x_i, y_i) \in Z$.

Um método para encontrar um hiperplano que resolva um problema de classificação binária é através de um classificador de margem que é descrito na Seção 2.3.

2.3 MÁQUINA DE VETORES SUPORTE

Um classificador que define uma distância entre o limite de decisão e as amostras é chamado de classificador baseado em margem. A distância entre o limite de decisão e as amostras mais próximas do mesmo é chamada de margem.

O SVM é um classificador de máxima margem Boser *et al.* (1992), o que significa que ele encontra um hiperplano que maximiza a distância entre as classes. O SVM é definido como um problema de otimização da seguinte maneira:

$$\max_{(w,b)} \left(\min_i \frac{y_i(w \cdot x_i + b)}{\|w\|} \right)$$

sujeito a $y_i(w \cdot x_i + b) > 0, \forall (x_i, y_i) \in Z$,

onde $\gamma_i = y_i(w \cdot x_i + b)$ é definida como a margem funcional. A margem geométrica γ_g precisa ser definida para se ter uma definição adequada de distância em relação ao hiperplano. A distância perpendicular do hiperplano \mathcal{H} até a origem é $|b|/\|w\|$. Definindo dois hiperplanos paralelos a \mathcal{H} como $\mathcal{H}^+ := \{x \in \mathbb{R}^d \mid w \cdot x + (b - \gamma) = 0\}$ e $\mathcal{H}^- := \{x \in \mathbb{R}^d \mid w \cdot x + (b + \gamma) = 0\}$, com a distância entre eles dada por:

$$dist(\mathcal{H}^-, \mathcal{H}^+) = \frac{-(b - \gamma) + (b + \gamma)}{\|w\|} = \frac{2\gamma}{\|w\|},$$

então $\gamma_g := dist(\mathcal{H}^-, \mathcal{H}^+)/2$ dá a margem geométrica entre os hiperplanos \mathcal{H}^+ e \mathcal{H}^- .

Com isso, o problema de otimização pode ser reescrito como:

$$\begin{aligned} & \max \gamma_g \\ & \text{sujeito a } y_i(w \cdot x_i + b) \geq \|w\|\gamma_g, \forall (x_i, y_i) \in Z. \end{aligned}$$

Fazendo $\gamma = 1 = \gamma_g \|w\|$ o valor mínimo da margem funcional, a formulação primal do SVM que minimiza a norma euclidiana Vapnik (1995) pode ser definida como:

$$\begin{aligned} & \min \frac{1}{2} \|w\|^2 \\ & \text{sujeito a } y_i(w \cdot x_i + b) \geq 1, \forall (x_i, y_i) \in Z. \end{aligned}$$

Um método de simplificar a solução para este problema é relaxar as restrições de desigualdade, incluindo um conjunto de multiplicadores lagrangeanos não negativos α_i , em que $i \in \{1, \dots, m\}$. Ao usar as restrições relaxadas, obtém-se a função lagrangeana:

$$L(w, b, a) = \frac{1}{2} \|w\|^2 - \sum_i \alpha_i y_i (w \cdot x_i + b) + \sum_i \alpha_i.$$

Para obter a formulação dual do problema do SVM, é necessário minimizar essa função em relação a w e b e maximizá-la em relação a α , sujeito a $\alpha_i \geq 0$. Para encontrar essa solução, é preciso maximizar uma função estritamente dual, substituindo os parâmetros w e b :

$$\begin{aligned} \max L(a) &= \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \\ \text{sujeito a } &\begin{cases} \sum_i \alpha_i y_i = 0 \\ \alpha_i \geq 0, \forall i \in 1, \dots, m. \end{cases} \end{aligned}$$

Com os valores de α^* da solução, o vetor de pesos pode ser criado:

$$w^* = \sum_i \alpha_i^* y_i x_i, \forall i \in 1, \dots, m.$$

2.3.1 Funções *kernel*

Como mencionado na Seção 2.2, existem casos em que uma base de dados pode não ser linearmente separável. No entanto, é importante notar que nestes casos a base de dados não é separável em seu espaço dimensional original. É possível elevar a dimensão de um conjunto de dados ao aplicar-se uma transformação sobre o mesmo, fazendo com que este se torne separável por um hiperplano em um espaço dimensional maior.

Aplicar uma transformação sobre toda uma base de dados pode ter um custo computacional muito grande. Contudo, pode-se observar que para a solução da função de

otimização do SVM não é necessário conhecer os valores das amostras, apenas o valor de seu produto interno:

$$\max L(a) = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j (x_i \cdot x_j)$$

Através do uso de funções *kernel* é possível calcular o valor do produto interno em um espaço de mais alta dimensão sem que seja necessário mapear as amostras para este espaço.

Definindo-se uma função *kernel* como $K(x_i, x_j) = x_i \cdot x_j$, o problema de otimização do SVM pode ser reescrito como:

$$\begin{aligned} \max L(a) &= \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j K(x_i, x_j) \\ \text{sujeito a } &\begin{cases} \sum_i \alpha_i y_i = 0 \\ \alpha_i \geq 0, \forall i \in 1, \dots, m. \end{cases} \end{aligned}$$

Existem diversos tipos de funções *kernel* Hofmann *et al.* (2008) e seus resultados podem variar de acordo com a aplicação. Como exemplo de uma função *kernel* simples e amplamente utilizada, tem-se o *kernel* linear, que pode ser definido como:

$$K(x_i, x_j) = x_i \cdot x_j.$$

Outra função muito utilizada é o *kernel* RBF (*Radial Basis Function*), que pode ser definida como:

$$K(x_i, x_j) = \exp(-\gamma_{rbf} \|x_i - x_j\|^2).$$

2.4 INFERÊNCIA TRANSDUTIVA E INDUTIVA

Como resultado de uma tarefa de aprendizado, tem-se uma função $f : \mathcal{X} \rightarrow y$ que é definida para todo o espaço de entrada \mathcal{X} . Com essa função, seria possível inferir uma rotulação para novas amostras $x_{new} \in \mathcal{X}$. Isto é o que é chamado de inferência indutiva, quando a geração do modelo é feito a partir de amostras específicas para obter uma função geral capaz de realizar inferência sobre todo o espaço de entrada.

Dependendo da aplicação da tarefa de aprendizado, pode não ser necessário encontrar uma função geral capaz de fazer uma inferência sobre todo o espaço de entrada. Vapnik (1995) apresenta um tipo distinto de inferência chamado de transdução, utilizando como princípio de que um problema mais geral não deveria ser resolvido como um passo para resolver um problema específico. Desta forma, o modelo é construído a partir de amostras específicas para realizar inferência sobre amostras específicas.

Com a transdução, obtém-se uma função $f_u : X_u \rightarrow y_u$ que só é capaz de rotular o conjunto previamente conhecido de amostras não rotuladas X_u . Para uma nova amostra

ser avaliada, o treinamento teria que ocorrer novamente, com a nova amostra adicionada ao conjunto sem rótulo X_u .

Seguindo o modelo primal de SVM descrito na Seção 2.3, o problema de inferência transdutiva pode ser formulado como:

$$\begin{aligned} & \min \frac{1}{2} \|w\|^2 \\ \text{sujeito a } & \begin{cases} y_i(w \cdot x_i + b) \geq 1, \forall (x_i, y_i) \in X_l, \\ y_j(w \cdot x_j + b) \geq 1, \forall x_j \in X_u, y_j \in Y_u. \end{cases} \end{aligned}$$

2.5 BRANCH-AND-BOUND

O método *branch-and-bound* é uma conhecida ferramenta para a solução de problemas de otimização combinatória (Narendra e Fukunaga, 1977). Neste tipo de problema, tem-se uma função objetivo que deve ser minimizada ou maximizada. Este método busca explorar o espaço de soluções através da divisão do mesmo em soluções candidatas. Durante a exploração, as soluções são selecionadas através de uma estratégia específica ao contexto aplicado. A geração de soluções cria uma estrutura de árvore de busca, onde cada nó corresponde a um estado com uma solução candidata. A eficiência do método está fortemente ligada a suas principais operações de ramificação (*branching*) e de poda através dos limites (*bounds*) utilizados.

A operação de ramificação consiste da definição de qual a melhor estratégia para exploração do espaço de soluções, ou seja, a partir de um estado pai da árvore de busca como gerar as melhores soluções filhas para a solução do problema. A definição da estratégia de ramificação é dependente do contexto do problema a ser resolvido, e uma escolha ruim para esta estratégia pode levar a uma exploração desnecessária do espaço de soluções.

Para a realização de uma exploração mais eficiente da árvore de busca, são realizadas podas de subárvores que não levarão a solução ótima. Para determinar se uma solução deve ser podada ou não deve ser possível definir-se um limite superior e inferior em cada nó. O limite superior para um determinado nó pode ser definido pela solução obtida pelo seu estado pai, assim, uma solução que seja maior que o limite superior pode ser descartada pois não levará a solução ótima. O limite inferior deve ser computado de forma que seja sempre igual ou menor a solução do nó em questão, desta forma, qualquer solução posterior que não alcance o limite inferior também pode ser descartada.

2.6 BUSCA ORDENADA

O método de busca ordenada realiza a exploração de uma estrutura de grafo percorrendo os nós de modo a obedecer uma determinada forma de ordenação. Uma forma de aplicar uma busca ordenada é através da busca pelo melhor candidato disponível,

chamada de *best-first search* (Russell e Norvig, 2010). Neste tipo de busca, utiliza-se uma função de avaliação heurística para determinar qual o melhor caminho a ser explorado ao realizar uma busca. A cada iteração a melhor solução existente, definida pela função de avaliação, é escolhida para continuar a exploração.

Para que a otimalidade da busca seja garantida é necessário que a função de avaliação seja admissível (Hart *et al.*, 1968). Portanto, ao utilizar-se a função definida para avaliação do custo em um determinado nó o resultado não deve superestimar o real custo para atingir o objetivo da busca. Uma forma de se garantir a admissibilidade da função é através da utilização de uma função que seja monótona, ou seja, uma função inteiramente não crescente ou não decrescente. Para a solução de um problema de maximização, uma função monótona não crescente deve ser utilizada, e para a minimização uma função monótona não decrescente.

Durante o processo de exploração, os estados gerados são mantidos em uma estrutura auxiliar e ordenados pelo valor definido pela função de avaliação. Para facilitar a manutenção e seleção dos próximos estados a serem explorados, a estrutura utilizada é a de uma lista de prioridades.

3 CLASSIFICADOR TRANSDUTIVO COM BUSCA ORDENADA E BRANCH-AND-BOUND

Neste capítulo, o método proposto chamado de Classificador Transdutivo com Busca Ordenada e Branch-and-Bound (*Best-First Branch-and-Bound Transductive Classifier – BFBB-TC*) é apresentado.

3.1 ESPAÇO DE ESTADOS E BUSCA HEURÍSTICA

Para lidar com o aspecto combinatorial do problema, uma busca heurística *best-first* foi utilizada, na qual cada hipótese do problema é representada por um estado no espaço de estados da busca. A seleção de um estado ocorre através de uma estratégia em que o melhor candidato (aquele com uma maior chance de ser a solução ótima) que se encontra disponível é selecionado em cada iteração. Para determinar o mérito de cada estado, uma função de decisão é necessária e, no caso de problemas de maximização, essa função precisa ser monotonicamente decrescente. Com base no algoritmo AOS (Villela *et al.*, 2015), o algoritmo BFBB-TC foi desenvolvido acoplado a um classificador baseado em margem, neste caso, um SVM de margem rígida.

A admissibilidade da busca é satisfeita empregando uma função monotonicamente decrescente baseada nos valores de margem dados pelo classificador baseado em margem. Assim, em um conjunto de treinamento com m amostras, tem-se $\gamma^m \leq \gamma^{m-1}$, onde γ^{m-1} é a margem real de um estado pai e γ^m é a margem real de um estado filho. A utilização da margem rígida ocorre pela necessidade de garantia da monotonicidade da função de avaliação, do contrário, poderiam ocorrer flexibilizações da margem levando a perda da monotonicidade.

Os estados gerados, ordenados pelos valores de margem, são armazenados em uma fila de prioridades implementada como uma estrutura de *heap* denominada H no Algoritmo 2.

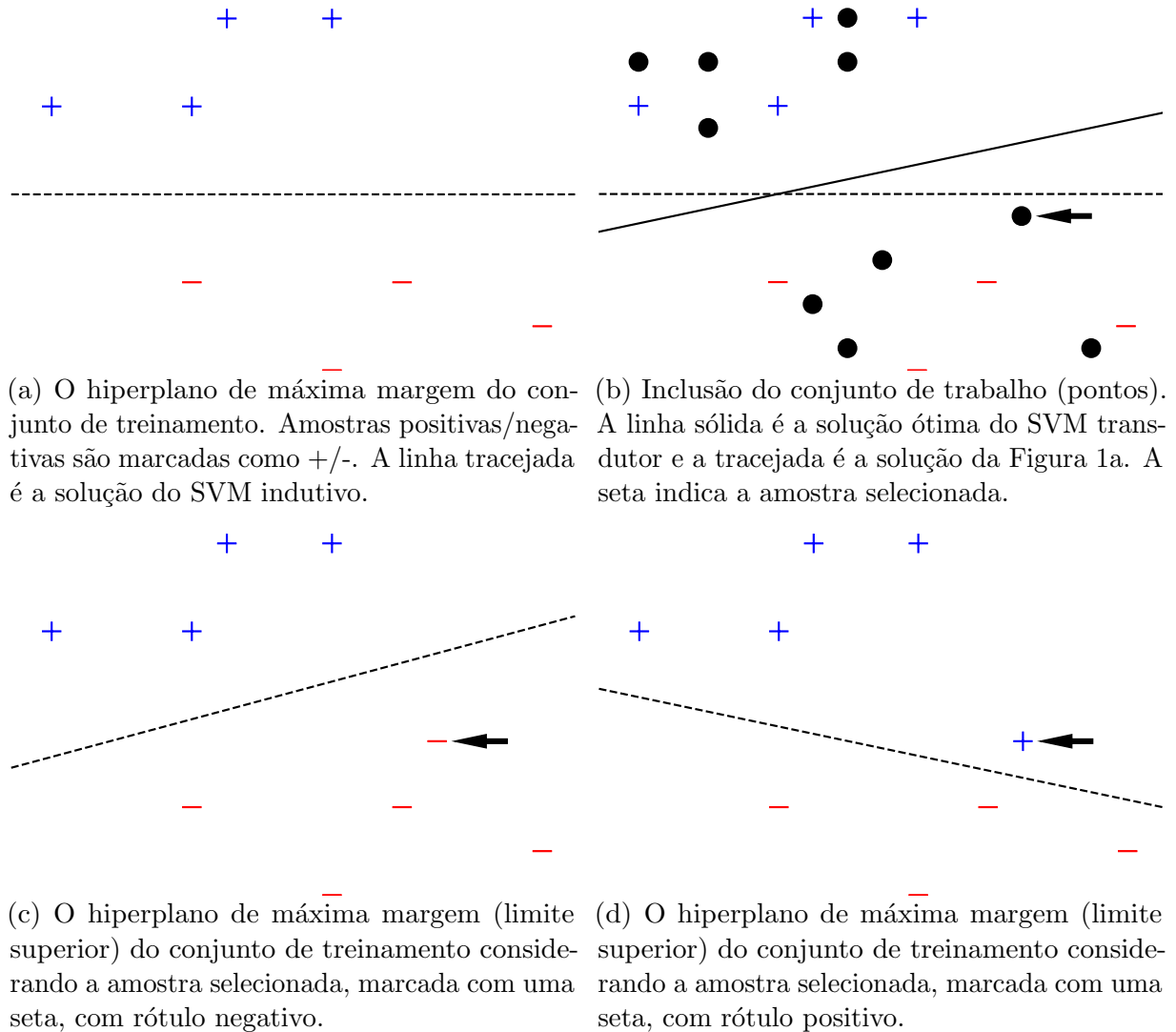
Como o algoritmo permite especificar um número máximo de estados a serem explorados, ao atingir este limite a solução retornada é a que foi encontrada pelo cálculo do limite inferior, explicado em detalhes na Seção 3.4.

3.2 RAMIFICAÇÃO

Observando a Figura 1, o processo de ramificação pode ser explicado da seguinte forma: retire do *heap* H a solução atual com o maior valor de margem (Figura 1a). Em seguida, introduza no espaço de treinamento as amostras não rotuladas do conjunto de trabalho. Se a nova solução for viável e não forçar a margem, a solução ótima foi encontrada. Caso contrário, a solução alcançada é de margem inviável ou margem de erro, e o limite inferior pode ser atualizado através do cálculo de uma nova solução do SVM

(Figura 1b). Em seguida, a amostra mais próximo ao hiperplano pai (marcado com uma seta) é selecionada para ser rotulada e gera dois novos estados S_+ e S_- que devem ser inseridos, após avaliação, no *heap* H (Figuras 1c e 1d).

Figura 1 – O processo de ramificação.



Fonte: Elaborado pelo autor (2019).

3.3 AVALIAÇÃO E PODA

A partir do processo de ramificação, dois novos conjuntos de treinamento são produzidos, X_{l+} e X_{l-} . Cada um desses conjuntos tem todo o conjunto de treinamento anterior mais a amostra mais próxima selecionada com um dos rótulos. A margem do estado pai define um limite superior para as novas soluções. Se o classificador baseado em margem não for capaz de encontrar uma solução viável, o respectivo estado deve ser eliminado e não inserido no *heap*.

Para cada um dos novos conjuntos de treinamento X_{l+} e X_{l-} , novas soluções são computadas com o classificador obtendo-se as margens γ_+ e γ_- . Desta forma, um novo limite superior para essa amostra é construído em cada rotulação. Todos os estados cujo valor de margem é menor que o limite inferior devem ser eliminados.

Com a redução da margem em cada estado e a amostra selecionada sempre sendo a mais próxima do hiperplano separador, essa amostra torna-se um potencial candidato para ser um vetor suporte na solução final.

Com este procedimento, o algoritmo só faz uso de uma pequena porção do conjunto de trabalho. A adição de cada nova amostra ao conjunto de treinamento adiciona uma restrição ao problema de otimização, o que reduz o espaço de hipóteses e também prova a monotonicidade dos valores de margem, considerando o fato de que o novo problema de margem máxima é mais restrito do que o problema dos estados pais. Desta forma, a nova solução só pode ser igual ou menor à solução dos estados pais.

A admissibilidade da busca, garantida pela monotonicidade da função de avaliação, é imperativa para o funcionamento do método. Nesse sentido, o mérito dos estados deve sempre ser uma solução máxima. Qualquer função de avaliação que não seja capaz de manter a admissibilidade possivelmente levará a uma maior exploração dos estados no espaço do estado. Além do aumento do tempo, isso também não garante a solução ótima.

A cada seleção de uma nova solução do *heap*, a margem da solução selecionada é avaliada com o limite inferior corrente e eliminada caso seja inferior ao mesmo. A seleção de uma nova solução ocorre até que um valor de margem superior ao limite seja encontrado ou o *heap* possua uma única solução restante. Assim, é possível evitar que a reconstrução do *heap* ocorra a cada atualização do limite inferior com a eliminação de todos os estados não mais adequados.

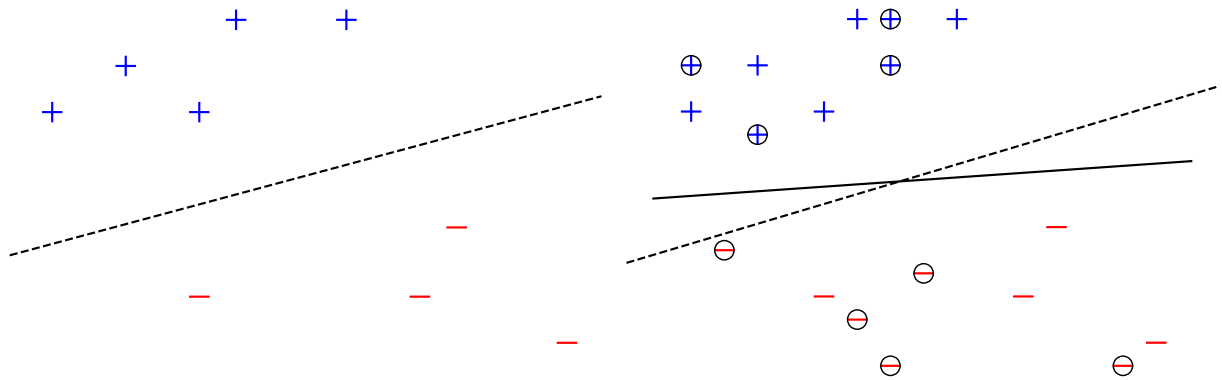
3.4 LIMITES INFERIOR E SUPERIOR

A exploração do espaço de estados de um problema combinatorial e a utilização do SVM como classificador baseado em margem para avaliar a função de decisão exigem um alto custo computacional. Como forma de lidar com esta limitação, uma heurística capaz de fornecer soluções competitivas para problemas com conjuntos de trabalho maiores foi proposta. Com essa abordagem, é preciso executar o SVM apenas duas vezes para calcular os limites superior e inferior e a melhor solução inicial.

Para calcular o limite superior, um SVM indutivo é treinado utilizando-se o conjunto de treinamento. Essa primeira solução também é usada como o estado inicial da busca. Se a solução não for viável, devido a violação da margem por amostras do conjunto de trabalho, um novo problema é gerado ao adicionar as amostras não rotuladas ao conjunto de treinamento. Assim, o hiperplano de separação rotulará essas amostras de acordo com

a posição do mesmo no espaço de entrada. Então, novamente é feito o treinamento de um SVM indutivo com todo este conjunto, e a margem da solução obtida é usada como um limite inferior para a busca. Esse limite inferior pode ser considerado uma solução heurística e esse valor pode ser melhorado durante a busca. A Figura 2 ilustra esse processo juntamente com o Algoritmo 1.

Figura 2 – Processo de computação dos limites superior (linha tracejada) e inferior (linha sólida).



(a) Solução de limite superior e conjunto de treinamento inicial.

(b) Soluções de limite inferior e superior com conjuntos de treinamento e de trabalho (amostras circuladas, com rotulação dada pelo limite superior).

Fonte: Elaborado pelo autor (2019).

Realizar o cálculo do limite inferior a cada estado explorado acrescentaria um custo computacional considerável. Dado que para todo limite inferior lb e solução γ tem-se $lb \leq \gamma$, e que uma solução de mesmo nível necessariamente será menor ou igual a qualquer anterior, considerou-se que o primeiro novo limite inferior encontrado em um nível é uma melhoria satisfatória para o mesmo. Assim, o cálculo do limite é feito em cada nível somente até que a primeira solução capaz de atualizar o limite inferior seja encontrada no nível.

Ao aplicar essa heurística juntamente com o SVM como o classificador de margem máxima, sempre haverá uma solução viável no limite inferior, o que evita que o método fique preso ao procurar a solução ideal em conjuntos de dados maiores. Caso o algoritmo seja limitado a executar um número máximo de estados, o classificador usado como solução é o último computado durante o cálculo do limite inferior.

É importante esclarecer que a rotulação inicial do TSVM é diferente do esquema de rótulos usado para calcular a solução heurística. Eles são distintos porque no TSVM é necessário informar o percentual de cada classe do conjunto de trabalho que será posteriormente rotulado.

3.5 MÚLTIPLAS AMOSTRAS E JANELA

Além da heurística apresentada na Seção 3.4, duas novas abordagens foram propostas com o objetivo de reduzir o número de estados explorados durante a busca.

A primeira abordagem oferece uma nova estratégia para seleção da próxima amostra a ser rotulada. Nesta nova estratégia duas amostras são escolhidas para serem rotuladas simultaneamente, cada uma localizada em um dos subespaços separados pelo hiperplano da solução pai. A partir das amostras selecionadas e suas possíveis rotulações, três novos conjuntos de treinamento são gerados X_{l++} , X_{l--} e X_{l+-} , em que cada um possui o conjunto de treinamento utilizado na solução pai acrescentado das amostras selecionadas com os rótulos $\{+1, +1\}$, $\{-1, -1\}$ e $\{+1, -1\}$, respectivamente. O quarto possível estado X_{l-+} com rótulos $\{-1, +1\}$ é descartado, pois durante testes realizados no desenvolvimento foi verificado que o mesmo conduzia a soluções piores. No caso de não haver uma amostra em um dos subespaços, é feita a seleção e rotulação de uma única amostra como na proposta para solução exata. A Figura 3 ilustra a estratégia utilizada.

Ao utilizar esta abordagem, a geração da árvore de busca se torna mais compacta se comparada a árvore gerada pelo método exato, como demonstrado na Figura 4. Esta diferença na geração dos estados faz com que seja possível chegar a soluções que estariam em níveis mais profundos da árvore completa, podendo levar a uma solução melhor de forma acelerada e com um menor número de estados explorados do que seria necessário. No entanto, ao gerar esta árvore compacta é possível que estados que levariam a solução ótima não sejam gerados, impedindo que a solução encontrada por esta abordagem seja a ótima global em tais casos.

A segunda abordagem proposta visa diminuir o número de estados explorados ao estabelecer um limite entre a diferença do maior nível já explorado com o nível do estado filho selecionado, a este limite dá-se o nome de janela. Considerando-se uma janela w , o nível mais profundo explorado d e um possível estado filho de nível l , é preciso que $d - l \leq w$, do contrário, o estado deve ser descartado e uma nova seleção é feita até que a condição seja satisfeita.

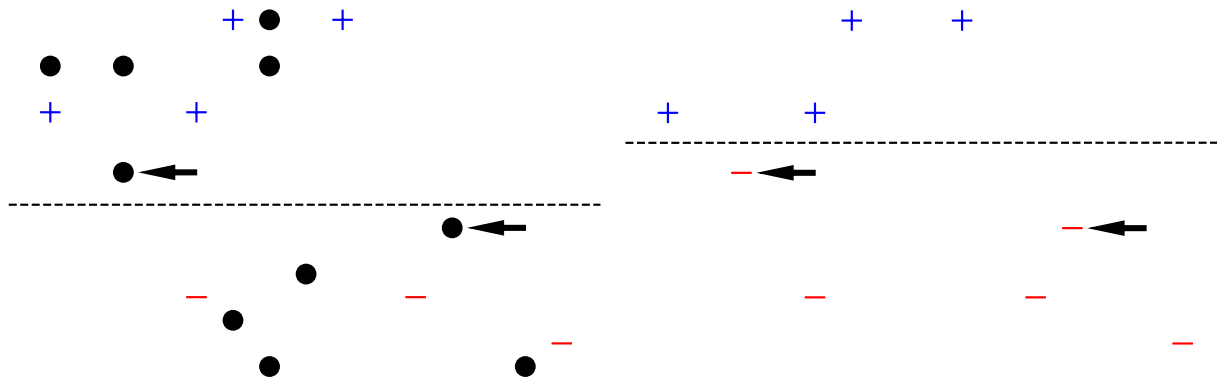
Ao utilizar a técnica da janela, a exploração é feita de forma a evitar estados de níveis menores (mais próximos a raiz) já explorados, de forma que a busca alcance níveis mais profundos mais rapidamente, onde espera-se que as soluções candidatas estejam mais próximas da solução ótima.

3.6 PSEUDOCÓDIGO

O Algoritmo 1 descreve o método para computar o limite inferior.

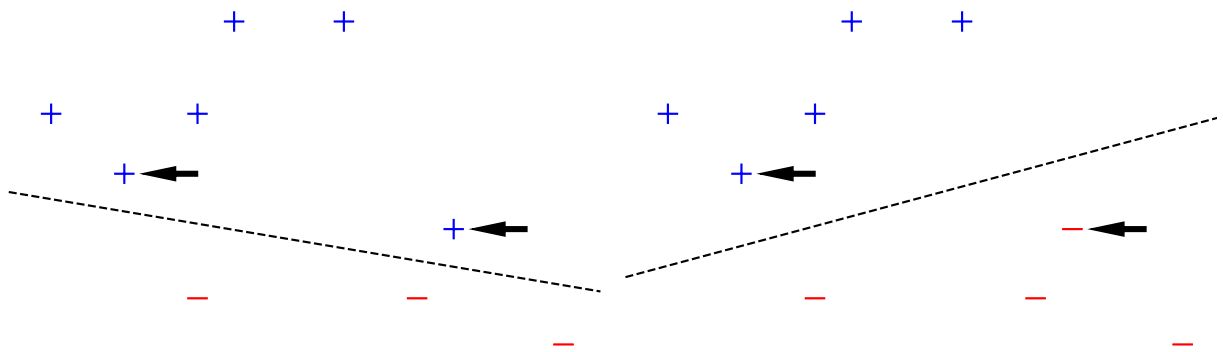
O Algoritmo 2 descreve o *Best-First Branch-and-Bound Transductive Classifier*.

Figura 3 – O processo de ramificação com seleção de duas amostras.



(a) Conjunto de treinamento (amostras +/-) e conjunto de trabalho (pontos). A linha tracejada é a solução do SVM indutivo com o conjunto de treinamento. As setas indicam as amostras selecionadas para rotulação.

(b) O hiperplano de máxima margem (limite superior) do conjunto de treinamento considerando as amostras selecionadas, marcadas com uma seta, com rótulos negativos.



(c) O hiperplano de máxima margem (limite superior) do conjunto de treinamento considerando as amostras selecionadas, marcadas com uma seta, com rótulos positivos.

(d) O hiperplano de máxima margem (limite superior) do conjunto de treinamento considerando as amostras selecionadas, marcadas com uma seta, com rótulo positivo e negativo.

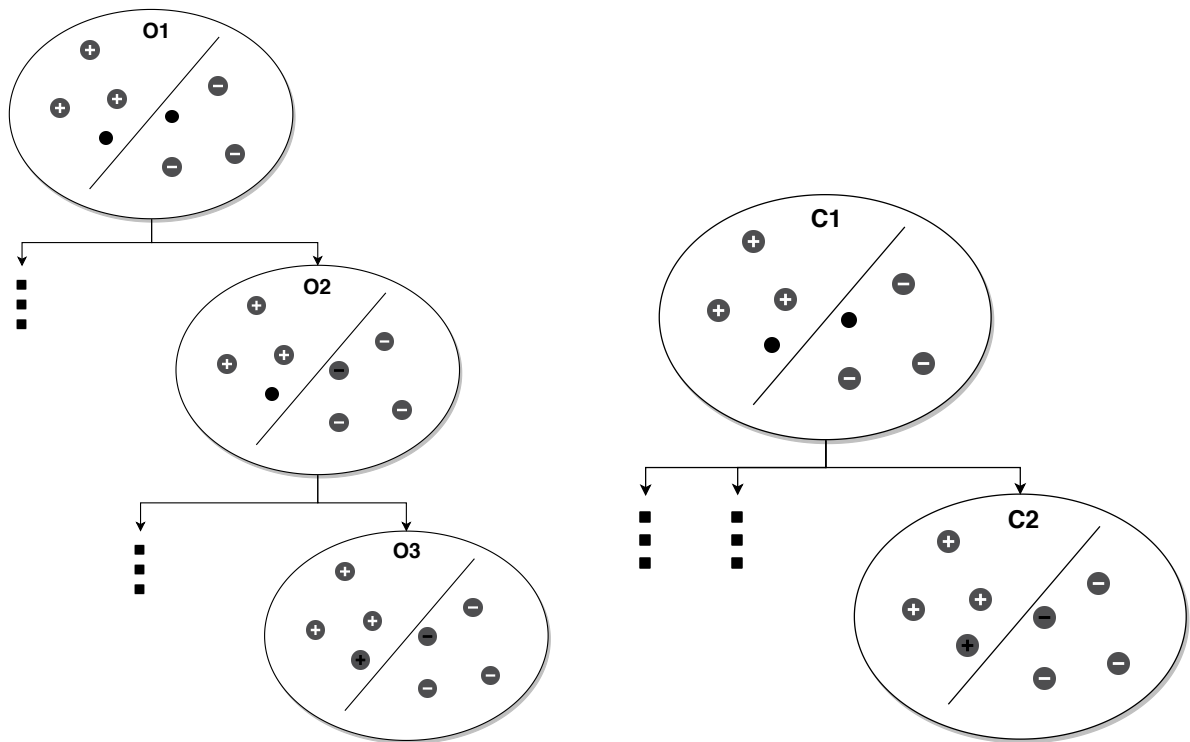
Fonte: Elaborado pelo autor (2019).

Algoritmo 1: Computar limite inferior

Entrada: conjunto de treinamento: X_l ;
conjunto de trabalho: X_u ;
classificador do estado atual: svm ;
Saída: limite inferior;

- 1 **início**
- 2 rotular X_u com svm ;
- 3 treinar novo classificador svm_{lb} com X_l e X_u com os rótulos encontrados;
- 4 **retorna** margem do svm_{lb} ;
- 5 **fim**

Figura 4 – Comparação entre uma árvore de estados completa e compacta. Em cada estado é representado o conjunto de treinamento (amostras +/-) e o conjunto de trabalho (pontos pretos). Os estados O1 e C1 são equivalentes, assim como os estados O3 e C2.



(a) Representação de parte de uma árvore de estados completa.

(b) Representação de parte de uma árvore de estados compacta gerada pela proposta de seleção de duas amostras.

Fonte: Elaborado pelo autor (2019).

Algoritmo 2: Best-First Branch-and-Bound Transductive Classifier

Entrada: conjunto de treinamento: $X_l = \{(x_i, y_i) | i \in \{1, \dots, m\}\}$;
 conjunto de trabalho: $X_u = \{x_j | j \in \{m + 1, \dots, k\}\}$;
 número máximo de estados: $max_estados$;
kernel: K ;
 janela: w ;

- 1 **início**
- 2 inicializar max-heap H ;
- 3 computar a solução usando o SVM com *kernel* K e X_l para o estado inicial S ;
- 4 inserir S em H ;
- 5 computar limite inferior LB inicial;
- 6 $num_estados_explorados \leftarrow 0$;
- 7 $profundidade \leftarrow 0$;
- 8 **enquanto** H não está vazio e a solução em S não é factível e $num_estados_explorados < max_estados$ **faça**
- 9 $num_estados_explorados \leftarrow num_estados_explorados + 1$;
- 10 atualizar $profundidade$ se necessário;
- 11 encontrar a(s) amostra(s) mais próxima(s) ao hiperplano $(x_{prox}$ ou x_{prox+} e $x_{prox-})$;
- 12 computar novo limite inferior e atualizar LB se necessário;
- 13 gerar os novos conjuntos de treinamento:
- 14 $X_{l+} = X_l + \{(x_{prox}, +1)\}$ e $X_{l-} = X_l + \{(x_{prox}, -1)\}$ ou
- 15 $X_{l++} = X_l + \{(x_{prox+}, +1), (x_{prox-}, +1)\}$,
- 16 $X_{l--} = X_l + \{(x_{prox+}, -1), (x_{prox-}, -1)\}$ e
- 17 $X_{l+-} = X_l + \{(x_{prox+}, +1), (x_{prox-}, -1)\}$;
- 18 remover a(s) amostra(s) mais próxima(s) selecionada(s) de X_u ;
- 19 computar soluções usando os novos conjuntos de treinamento para os novos estados S_+ e S_- ou S_{++} , S_{--} e S_{+-} ;
- 20 remover S de H ;
- 21 para cada novo estado, se seu respectivo γ for maior que LB inserir o novo estado em H ;
- 22 selecionar novo S de H ;
- 23 **enquanto** $\gamma_s < LB$ **faça**
- 24 | selecionar novo S de H ;
- 25 **fim enquanto**
- 26 **se** janela w informada **então**
- 27 | **enquanto** $profundidade - nivel_s > w$ **faça**
- 28 | | selecionar novo S de H ;
- 29 | **fim enquanto**
- 30 **fim se**
- 31 **fim enquanto**
- 32 **fim**

4 ANÁLISE EXPERIMENTAL E RESULTADOS

Neste capítulo são apresentados os experimentos realizados e a análise dos resultados obtidos. A Seção 4.1 apresenta os experimentos focados na avaliação da margem com o crescimento do conjunto de trabalho e a respectiva diminuição do conjunto de treinamento. Na Seção 4.2 são apresentados os resultados da aplicação do método para a inferência com a análise das acurácias obtidas.

Para comparação dos resultados, também foram realizadas avaliações do SVM com margem rígida sendo treinado apenas com o conjunto de treinamento inicial e do TSVM com o mesmo conjunto de treinamento e trabalho utilizados pelo BFBB-TC. No SVM, tanto o executado para comparação dos resultados como o utilizado pelo BFBB-TC, o hiperparâmetro C foi definido com o valor de 10000 para forçar uma solução de margem rígida.

O BFBB-TC foi implementado em Python utilizando a implementação do SVM presente na biblioteca *Scikit-Learn* (Pedregosa *et al.*, 2011), que utiliza a linguagem Cython para criar uma interface com a biblioteca LIBSVM (Chang e Lin, 2011) implementada em C. O LIBSVM utiliza o algoritmo SMO (Platt, 1999) para resolver o problema de otimização do SVM. De forma similar, o TSVM, que também é implementado em C no programa¹ *SVM^{light}*, é executado através de uma interface implementada em Cython nos experimentos.

4.1 AVALIAÇÃO DA MARGEM DE MÉTODOS TRANSDUTIVOS

A Tabela 1 mostra as quatro bases de dados escolhidas para execução dos experimentos, com informações sobre o número de dimensões, amostras e proporção entre as classes. Estas bases foram selecionadas do *benchmark*² criado por Chapelle *et al.* (2006b). Todas as bases de dados selecionadas têm duas classes para o atributo alvo e são linearmente separáveis.

Para cada um das bases de dados selecionadas, executou-se os algoritmos com conjuntos de trabalho (WS) de tamanhos 50, 100, 200, 300 e 500. Devido ao custo computacional de executar o BFBB-TC com um SVM de margem rígida, só foi possível executar os experimentos no intervalo de 50 a 300, com exceção do conjunto de dados BCI devido ao seu tamanho reduzido. Os conjuntos de trabalho foram criados a partir das bases de dados completas originais, criando-se dez divisões aleatórias para cada um dos tamanhos para gerar um conjunto de treinamento e de trabalho. O único pré-processamento feito para os dados foi normalizar os valores no intervalo $[-1, +1]$.

¹ O programa *SVM^{light}* está disponível em <http://svmlight.joachims.org/>

² As bases do *benchmark* podem ser encontradas em <http://olivier.chapelle.cc/ssl-book/benchmarks.html>

O objetivo dos experimentos foi comparar os valores de margem, a proporção do conjunto de trabalho deixado na solução final do BFBB-TC e também como os métodos comparam-se em tempo de execução. Todos os experimentos foram executados na plataforma Google Cloud em uma máquina do tipo n1-highmem-8 (8 vCPUs, 52 GB de memória).

Tabela 1 – Informações sobre as bases utilizadas nos experimentos de margem.

Base	Atributos	Amostras		
		Pos.	Neg.	Total
BCI	117	200	200	400
COIL ₂	241	750	750	1500
DIGIT1	241	734	766	1500
USPS	241	1200	300	1500

Fonte: Elaborado pelo autor (2019).

A Tabela 2 mostra os valores médios das dez execuções do TSVM e do BFBB-TC, com o SVM e a solução heurística de limite inferior. A coluna “WS” indica o tamanho do conjunto de trabalho. A coluna “SVM” indica o limite superior obtido ao executar o SVM somente com o conjunto de treinamento inicial, que define a margem máxima possível para o problema. A “Limite inferior” indica o limite inferior que representa os valores da solução heurística. O “TSVM” mostra a margem média das soluções do TSVM e a coluna “BFBB-TC” mostra os resultados da execução do algoritmo com um SVM de margem rígida. Os melhores resultados de margem são destacados em negrito.

Tabela 2 – Comparação entre os valores de margem do TSVM e do BFBB-TC.

Base	WS	SVM	TSVM	BFBB-TC ³	Limite inferior
BCI	50	0,00635±0,00093	0,00620±0,00081	0,00633±0,00093	0,00632±0,00094
	100	0,00856±0,00154	0,00818±0,00153	0,00843±0,00150	0,00837±0,00148
	200	0,01833±0,00401	0,01574±0,00356	0,01750±0,00383	0,01605±0,00342
COIL ₂	50	0,00820±0,00048	0,00796±0,00052	0,00817±0,00048	0,00815±0,00049
	100	0,00887±0,00061	0,00845±0,00057	0,00882±0,00060	0,00880±0,00060
	200	0,01034±0,00062	0,00996±0,00056	0,01021±0,00061	0,01017±0,00059
	300	0,01224±0,00082	0,01136±0,00102	0,01184±0,00073	0,01182±0,00074
	500	0,01571±0,00103	0,01365±0,00111	*	0,01396±0,00067
DIGIT1	50	0,05289±0,00050	0,05087±0,00138	0,05271±0,00049	0,05267±0,00050
	100	0,05519±0,00262	0,05283±0,00269	0,05482±0,00270	0,05472±0,00272
	200	0,05821±0,00290	0,05453±0,00366	0,05714±0,00287	0,05692±0,00283
	300	0,06256±0,00330	0,05635±0,00521	0,06051±0,00302	0,05985±0,00325
	500	0,07145±0,00477	0,06137±0,00678	*	0,06385±0,00339
USPS	50	0,01352±0,00089	0,01331±0,00103	0,01351±0,00088	0,01349±0,00089
	100	0,01533±0,00162	0,01503±0,00149	0,01531±0,00161	0,01529±0,00161
	200	0,02027±0,00220	0,01992±0,00206	0,02021±0,00218	0,02018±0,00219
	300	0,02687±0,00374	0,02624±0,00358	0,02663±0,00362	0,02646±0,00355
	500	0,03801±0,00443	0,03529±0,00420	*	0,03630±0,00413

Fonte: Elaborado pelo autor (2019).

Comparando os resultados “BFBB-TC” com a coluna “SVM”, é possível observar como o método se aproxima dos resultados do limite superior comparada as outras soluções. Esses resultados são esperados, pois o uso de um classificador de máxima margem leva à solução globalmente ótima. Também é possível observar que o BFBB-TC com a solução heurística encontrou valores de margem comparáveis às demais soluções, e sempre superior ao TSVM. Uma melhor comparação entre os métodos BFBB-TC e TSVM é feita na Tabela 3.

Observando todos os resultados da Tabela 2, pode-se notar que, a medida que o tamanho do conjunto de trabalho aumenta, também aumenta o tamanho das margens para cada solução. Esse comportamento pode ser explicado pela perspectiva do problema de otimização que está sendo resolvido. Como o problema começa com menos restrições (amostras no conjunto de treinamento), ele permite que os métodos localizem margens maiores inicialmente, que não são drasticamente alteradas a medida que novas restrições são inseridas.

A Tabela 3 mostra uma comparação dos resultados do BFBB-TC em comparação ao TSVM em valores de margem e tempo, em que o TSVM é a referência base em 100%. A coluna “% WS” indica quanto do conjunto de trabalho não foi explorado na solução final encontrada pelo BFBB-TC. Os valores destacados em negrito enfatizam os resultados que foram melhores em comparação com o TSVM, em que os valores de margem são melhores quando maiores que 100% e o tempo quando inferior a 100%.

Na Tabela 3 pode-se examinar como o BFBB-TC se compara ao algoritmo TSVM com as abordagens de máxima margem e limite inferior. Tomando a margem da solução heurística em consideração, pode-se ver como ela foi capaz de encontrar valores melhores que o TSVM em todos os casos. A principal força da solução heurística é seu baixo custo computacional quando comparado aos demais, o que leva a uma solução viável em tempo constante, uma vez que é computada desde o início com a solução inicial.

Observando os valores médios apresentados na coluna “% WS”, pode-se ver uma variação no intervalo de 4,30% a 14,88% do conjunto de trabalho que está sendo usado. Com base nesses dados empíricos, é possível calcular uma aproximação do número de amostras do conjunto de trabalho que está sendo usado na solução final. Tomando m como o tamanho do conjunto de trabalho inicial em um problema de classificação binária e um pior caso de 15% do conjunto de trabalho sendo explorado, o número total aproximado de amostras do conjunto de trabalho seria $2^{0,15*m}$. Sabendo que o número total de possíveis esquemas de rotulação é de 2^m , é preciso apenas de uma pequena fração do conjunto de trabalho para encontrar uma solução viável.

Como mencionado anteriormente, a busca pela solução globalmente ótima usando o SVM de margem rígida tem um alto custo computacional. Observando a coluna “% tempo”

³ O * indica que o tempo máximo de execução permitido foi excedido.

Tabela 3 – Comparação dos resultados do BFBB-TC em comparação ao TSVM.

Base	WS	BFBB-TC			Limite inferior
		% margem	% tempo	% WS	% margem
BCI	50	102,10%	3,70%	89,60%	101,87%
	100	103,05%	39,52%	87,10%	102,28%
	200	111,15%	7387,24%	85,22%	102,00%
COIL ₂	50	102,73%	0,22%	92,20%	102,42%
	100	104,40%	0,52%	93,40%	104,14%
	200	102,51%	8,27%	92,35%	102,06%
	300	104,19%	165,37%	90,85%	104,01%
	500	–	–	–	102,28%
DIGITI	50	103,62%	22,07%	90,40%	103,54%
	100	103,75%	30,89%	92,10%	103,58%
	200	104,78%	286,11%	90,65%	104,38%
	300	107,38%	7099,56%	90,93%	106,22%
	500	–	–	–	104,04%
USPS	50	101,54%	0,95%	96,00%	101,34%
	100	101,87%	1,40%	95,70%	101,71%
	200	101,42%	11,40%	94,80%	101,30%
	300	101,48%	215,13%	93,40%	100,85%
	500	–	–	–	102,86%

Fonte: Elaborado pelo autor (2019).

sob “BFBB-TC”, é possível ver que, para conjuntos de trabalho de tamanho pequeno, o método BFBB-TC é capaz de encontrar a melhor solução em um tempo melhor que o TSVM, no entanto, a medida que o tamanho do conjunto de trabalho aumenta, o custo computacional também cresce em uma escala ainda maior.

4.2 AVALIAÇÃO DA MARGEM E GENERALIZAÇÃO DE MÉTODOS SEMISSUPERVISIONADOS

Para realizar a análise do comportamento do método em suas diferentes configurações, os experimentos foram separados pela função *kernel* utilizada (linear ou RBF) já que algumas das bases de dados escolhidas poderiam não ser linearmente separáveis, o que geraria problemas na tentativa de se encontrar uma solução linear de margem rígida para as mesmas.

Em cada uma destas separações foram utilizadas três diferentes configurações para a execução do BFBB-TC. Primeiramente, o algoritmo foi executado em sua implementação inicial, onde apenas uma amostra é selecionada e rotulada para a geração de novos estados, em que é realizada a busca pela solução exata. Os resultados para essa configuração foram identificados nas tabelas de resultados pela própria sigla do método, BFBB-TC. Na segunda configuração foi utilizada a estratégia heurística de seleção de duas amostras a cada estado explorado, sendo identificado como BFBB-TC DS nos resultados. Para a última configuração, além da estratégia de seleção de duas amostras também foi utilizada

a definição de uma janela para seleção na fila de prioridades dos próximos estados a serem explorados. Esta última configuração foi executada com dois valores para o parâmetro da janela, e podem ser identificados nos resultados por BFBB-TC DS W5 e BFBB-TC DS W2.

Para o *kernel* RBF, o valor do parâmetro gama foi definido como $\gamma_{rbf} = \frac{1}{n}$, onde n é o número de dimensões da base de dados sendo utilizada. O número máximo de estados explorados pelo BFBB-TC foi definido como $max_estados = 10000 + 2^{(perc*m)}$, onde m é o tamanho do conjunto de trabalho utilizado e $perc = 0,015$. Por possuir um conjunto de trabalho maior, o número máximo de estados para a base Krvskp foi definido com $perc = 0,0075$. Os valores de janela utilizados para os experimentos foram $w = \{5, 2\}$.

O objetivo destes experimentos foi de avaliar os métodos em relação a seus valores de margem e resultados de inferência transdutiva e indutiva, através dos seus resultados de acurácia. Além disso, avaliou-se as diferenças entre as configurações do BFBB-TC em relação ao número de estados explorados e eliminados pela poda, e também do percentual do conjunto de trabalho utilizado.

Todos os experimentos foram executados na plataforma Google Cloud em uma máquina do tipo n1-highmem-4, com 4 vCPUs e 26GB de memória principal, usando o processador Intel Xeon(R) de 2.30GHz e arquitetura x86_64. O sistema operacional utilizado foi o Debian GNU/Linux 9.9, chamado de Stretch.

4.2.1 Experimentos com *kernel* linear

Para a execução dos experimentos com o *kernel* linear foram utilizadas as mesmas bases apresentadas anteriormente na Tabela 1. Diferente dos experimentos anteriores, as divisões dos dados utilizada foram as fornecidas no mesmo *benchmark*. Para cada base de dados foram fornecidas doze divisões dos dados em amostras rotuladas e não rotuladas. Para cada divisão existem cem amostras rotuladas para o conjunto de treinamento e foram selecionadas aleatoriamente 25% do total de amostras para o conjunto de testes, fazendo com que o resto componha o conjunto de trabalho. O tamanho dos conjuntos de treinamento, trabalho e teste podem ser observados na Tabela 4.

Tabela 4 – Informações sobre as bases utilizadas nos experimentos com *kernel* linear.

Base	Tamanho do conjunto		
	Treinamento	Trabalho	Teste
BCI	100	200	100
COIL ₂	100	1025	375
DIGIT1	100	1025	375
USPS	100	1025	375

Fonte: Elaborado pelo autor (2019).

Os resultados obtidos para os experimentos com *kernel* linear podem ser observados

na Tabela 5. Nesta tabela foram sintetizados os valores médios com seus respectivos desvios padrões para os resultados obtidos com as doze divisões de cada base. Foram avaliados os resultados para os valores de margem e acurácia sobre os conjuntos de trabalho e teste. Para cada base, os melhores resultados foram destacados em negrito, exceto para os valores de margem, onde o destaque não considera os resultados do SVM por se tratar do limite superior para a solução.

Tabela 5 – Resultados de inferência dos experimentos com *kernel* linear.

Base	Método	Margem	Acurácia Trab.	Acurácia Teste
BCI	SVM	0,20188±0,04528	0,71083±0,04294	0,71083±0,04100
	TSVM	0,14547±0,02547	0,71541±0,05069	0,70916±0,05160
	BFBB-TC	0,16589±0,02900	0,70208±0,03985	0,71000±0,05325
	BFBB-TC DS	0,16473±0,03128	0,70458±0,05940	0,70750±0,04202
	BFBB-TC DS W5	0,16473±0,03128	0,70458±0,05940	0,70750±0,04202
	BFBB-TC DS W2	0,16583±0,03074	0,70250±0,05821	0,69583±0,04679
COIL ₂	SVM	17,44645±2,56989	0,80341±0,02809	0,80266±0,02649
	TSVM	4,65759±0,90845	0,79317±0,03292	0,78933±0,03267
	BFBB-TC	8,15755±1,28014	0,80715±0,03215	0,79911±0,02998
	BFBB-TC DS	7,93816±1,27330	0,81162±0,02508	0,81155±0,02809
	BFBB-TC DS W5	8,04464±1,24919	0,80650±0,02201	0,80888±0,02648
	BFBB-TC DS W2	8,34866±1,11428	0,79617±0,02518	0,79733±0,03153
DIGIT1	SVM	0,23420±0,02549	0,91845±0,01560	0,92311±0,01545
	TSVM	0,08857±0,00669	0,92121±0,01734	0,91888±0,02370
	BFBB-TC	0,08922±0,00544	0,91325±0,02049	0,91422±0,02080
	BFBB-TC DS	0,08702±0,00520	0,91910±0,01795	0,91711±0,01806
	BFBB-TC DS W5	0,08702±0,00520	0,91910±0,01795	0,91711±0,01806
	BFBB-TC DS W2	0,08716±0,00502	0,91926±0,01778	0,91733±0,01847
USPS	SVM	0,48725±0,07402	0,87162±0,01120	0,87355±0,01347
	TSVM	0,16177±0,01295	0,86796±0,01358	0,86822±0,01849
	BFBB-TC	0,22060±0,02859	0,87487±0,01517	0,87266±0,01616
	BFBB-TC DS	0,22053±0,02795	0,87674±0,01090	0,87088±0,01577
	BFBB-TC DS W5	0,22053±0,02795	0,87674±0,01090	0,87088±0,01577
	BFBB-TC DS W2	0,22087±0,02746	0,87634±0,01110	0,87022±0,01231

Fonte: Elaborado pelo autor (2019).

Como pode ser observado, o BFBB-TC atinge o maior valor de margem em todos os casos. As variações heurísticas do BFBB-TC tendem a encontrar um valor menor de margem quando comparado ao método exato por explorarem uma árvore de busca mais compacta, fazendo com que alcancem estados que estariam em níveis mais profundos na árvore completa. Apesar disso, também é possível observar que em alguns casos (bases COIL₂ e USPS) o método heurístico BFBB-TC DS W2 obteve resultados de margem superiores ao BFBB-TC, o que pode indicar que durante a eliminação de estados a heurística aplicada eliminou ou não gerou um estado intermediário que fazia parte do caminho ótimo.

Exceto pela base COIL₂, todos os resultados obtidos pela configuração BFBB-TC DS W5 foram iguais aos obtidos pela BFBB-TC DS. Esta correspondência de resultados

indica que, para estas bases, para que a técnica da janela produza efeitos durante a exploração é necessário que o valor da janela para eliminação de estados seja, pelo menos, menor do que cinco ($w < 5$).

Em todas as bases, o melhor resultado para o valor de acurácia sobre o conjunto de trabalho foi sempre obtido por um dos métodos semissupervisionados, indicando que a aplicação de métodos que utilizam o próprio conjunto não rotulado para o aprendizado podem levar a melhores resultados quando a inferência de interesse é a transdutiva. No entanto, pode-se verificar que em muitos casos o SVM com apenas o conjunto de treinamento foi capaz de obter resultados melhores que os métodos restantes.

Em seguida, observando-se os resultados de acurácia para o conjunto de teste nota-se que o SVM obteve os melhores resultados em todas as bases, exceto para a COIL₂ onde foi o terceiro melhor. Esta situação sugere que os hiperparâmetros definidos para os métodos semissupervisionados não permitiram o aprendizado de uma boa generalização, e que talvez uma melhor definição da proporção das classes para o TSVM e uma maior exploração de estados para o BFBB-TC permitira atingir resultados melhores.

Pode-se observar que o TSVM obteve os piores resultados na base USPS em todos os três aspectos avaliados. O pior desempenho nesta base pode estar relacionado ao balanceamento das classes, pois, diferente das demais, a base USPS possui muito mais amostras positivas que negativas. Considerando o desbalanceamento desta base e que o método do TSVM se utiliza da informação de proporção entre as classes, adotando por padrão a mesma proporção do conjunto de treinamento para o de trabalho, é possível que nas divisões utilizadas para os experimentos a diferença entre as proporções do conjunto de treinamento e de trabalho tenham sido mais divergentes, tendo um impacto negativo sobre o método.

Na Tabela 6 podem ser novamente observados os valores de margem para todas as configurações do BFBB-TC, acompanhados dos valores do número de estados explorados (“# explorados”), número de estados não inseridos (“# não inseridos”) na lista de prioridade devido a poda pelo limite inferior e, por fim, qual a porcentagem restante do conjunto de trabalho não utilizada pela solução final (“% WS”). Os melhores valores de margem são destacados em negrito.

Pode-se observar pela coluna “% WS” que o método exato utilizou o menor percentual do conjunto de trabalho dentre as configurações, o que é esperado já que as demais selecionam duas amostras para adição no conjunto de treinamento a cada nova ramificação. Pela limitação do número de estados a serem explorados nestes experimentos, quando compara-se o percentual não explorado do conjunto de trabalho com os resultados apresentados na Tabela 3 pode-se observar uma diferença considerável, indicando que ainda seria necessário uma vasta exploração para atingir a solução ótima.

Pelos valores obtidos do número de estados explorados pode-se observar que a

Tabela 6 – Resultados de exploração dos experimentos com *kernel* linear.

Base	Método	Margem	# explorados	# não inseridos	% WS
BCI	BFBB-TC	0,16589±0,02900	10008,0±0,0	0,0±0,0	0,93375±0,00678
	BFBB-TC DS	0,16473±0,03128	10008,0±0,0	884,6±2995,0	0,90583±0,00668
	BFBB-TC DS W5	0,16473±0,03128	10008,0±0,0	885,4±2997,6	0,90583±0,00792
	BFBB-TC DS W2	0,16583±0,03074	9242,0±1852,4	466,2±926,4	0,89250±0,01138
COIL ₂	BFBB-TC	8,15755±1,28014	52495,0±0,0	0,0±0,0	0,98195±0,00267
	BFBB-TC DS	7,93816±1,27330	52495,0±0,0	0,0±0,0	0,97658±0,00332
	BFBB-TC DS W5	8,04464±1,24919	52495,0±0,0	0,0±0,0	0,97333±0,00225
	BFBB-TC DS W2	8,34866±1,11428	49421,5±7202,0	553,7±1318,7	0,90991±0,03873
DIGIT1	BFBB-TC	0,08922±0,00544	52495,0±0,0	0,0±0,0	0,98471±0,00104
	BFBB-TC DS	0,08702±0,00520	52495,0±0,0	0,0±0,0	0,97934±0,00100
	BFBB-TC DS W5	0,08702±0,00520	52495,0±0,0	0,0±0,0	0,97934±0,00100
	BFBB-TC DS W2	0,08716±0,00502	52495,0±0,0	0,0±0,0	0,97951±0,00131
USPS	BFBB-TC	0,22060±0,02859	52495,0±0,0	0,0±0,0	0,98487±0,00121
	BFBB-TC DS	0,22053±0,02795	52495,0±0,0	0,0±0,0	0,98048±0,00166
	BFBB-TC DS W5	0,22053±0,02795	52495,0±0,0	0,0±0,0	0,98048±0,00166
	BFBB-TC DS W2	0,22087±0,02746	52495,0±0,0	0,0±0,0	0,97821±0,00112

Fonte: Elaborado pelo autor (2019).

grande maioria atingiu o limite máximo de estados definido, exceto para as bases BCI e COIL₂ com o BFBB-TC DS W2. Nestes dois casos uma solução final foi obtida antes de se esgotar o número de estados permitidos, e em ambos a solução obtida foi a melhor ou muito próxima da mesma. O alto valor dos desvios padrões sugerem que, apesar da eliminação de estados pela janela poder levar a bons resultados com um menor número de estados explorados, existe uma grande dependência do ponto de partida, ou seja, das amostras selecionadas para treinamento do estado inicial.

Em relação ao número de estados não inseridos na fila para exploração, pode-se ver que os métodos heurísticos foram capazes de realizar podas na base BCI e COIL₂. A inexistência de uma poda na maioria dos outros casos mostra que o método não foi capaz de atualizar o limite inferior o suficiente durante a exploração. Na base BCI, por possuir um tamanho menor, o método atingiu uma maior eficiência na realização das podas. Novamente, os altos valores de desvios padrões mostra a dependência do resultado para cada caso específico.

4.2.2 Experimentos com *kernel* RBF

Para os experimentos com *kernel* RBF foram utilizadas algumas das mesmas bases de dados⁴ apresentadas por Li *et al.* (2013), sendo aproveitadas as mesmas divisões criadas. As bases escolhidas estavam entre as dezesseis que foram utilizadas nos experimentos chamados de *Small-Scale Experiments*, exceto pela base *Sick* que não foi possível executar em tempo hábil. Para cada base foram criadas trinta divisões, selecionando 15% das

⁴ As bases e suas divisões podem ser encontradas em: http://lamda.nju.edu.cn/code_WellSVM.ashx

amostras para o conjunto de treinamento, 25% para o conjunto de teste e o resto para o conjunto de trabalho. As informações sobre quantidade de atributos, tamanho das bases, proporção de cada classe e os tamanhos dos conjuntos de treinamento, trabalho e teste para cada base, podem ser encontradas na Tabela 7.

Tabela 7 – Informações sobre as bases utilizadas nos experimentos com *kernel* RBF.

Base	Atributos	Amostras			Tamanho do conjunto		
		Pos.	Neg.	Total	Treinamento	Trabalho	Teste
Australian	42	383	307	690	77	441	172
Clean1	166	207	269	476	53	304	119
Diabetes	8	268	500	768	86	490	192
Echocardiogram	8	24	108	132	14	85	33
German	59	300	700	1000	112	638	250
Haberman	14	81	225	306	34	196	76
Heart	9	120	150	270	30	173	67
Heart-statlog	13	120	150	270	30	173	67
House	16	108	124	232	26	148	58
House-votes	16	267	168	435	49	278	108
Ionosphere	33	225	126	351	39	225	87
Isolet	51	300	300	600	67	383	150
Krvskp	36	1527	1669	3196	359	2038	799
LiverDisorders	6	200	145	345	38	221	86
Spectf	44	254	95	349	39	223	87

Fonte: Elaborado pelo autor (2019).

Nas Tabelas 8, 9 e 10 são resumidos os resultados dos experimentos com *kernel* RBF. Nestas tabelas são apresentados os valores médios obtidos para margem e acurácia dos conjuntos de trabalho e de teste, com seus respectivos desvios padrões. Novamente, os melhores resultados foram destacados em negrito para cada base, exceto para os valores de margem, onde o valor do SVM não foi considerado por se tratar do limite superior para a solução.

Assim como nos resultados para o *kernel* linear, o BFBB-TC obteve os maiores valores de margem em todas as bases dos experimentos com *kernel* RBF com alguma de suas configurações. Mais uma vez, ocorreram situações em que uma ou mais das configurações heurísticas do método obtiveram um valor de margem superior ao do método exato, reforçando que estas heurísticas podem desviar do caminho ótimo devido a suas estratégias de geração e eliminação de estados.

Pode-se observar que nas bases *Diabetes*, *Haberman* e *LiverDisorders* o BFBB-TC obteve valores de margem superiores ao SVM. Estes resultados não deveriam ser possíveis, já que o resultado do SVM corresponde ao limite superior para a solução. Uma possível explicação para este cenário é que por algum detalhe de implementação do SVM utilizado pelo BFBB-TC houve uma flexibilização da margem, permitindo que fosse possível encontrar uma solução maior que o limite superior. Uma possível solução para este caso seria realizar a poda de estados que violem o limite superior, o que não é feito atualmente.

Tabela 8 – Resultados de inferência dos experimentos com *kernel* RBF (1).

Base	Método	Margem	Acurácia Trab.	Acurácia Teste
Australian	SVM	0,01545±0,00729	0,77097±0,03896	0,77616±0,04925
	TSVM	0,00789±0,00375	0,77475±0,03612	0,77422±0,05330
	BFBB-TC	0,01102±0,00519	0,77944±0,03567	0,78527±0,05483
	BFBB-TC DS	0,01101±0,00519	0,77815±0,04228	0,77926±0,06027
	BFBB-TC DS W5	0,01097±0,00521	0,77626±0,04740	0,78062±0,05930
	BFBB-TC DS W2	0,01092±0,00519	0,7712±0,042820	0,77403±0,06208
Clean1	SVM	0,01909±0,00275	0,71403±0,03749	0,71624±0,04367
	TSVM	0,00880±0,001010	0,71447±0,04235	0,71652±0,05165
	BFBB-TC	0,01028±0,00120	0,70438±0,04509	0,69971±0,05716
	BFBB-TC DS	0,01018±0,00127	0,70614±0,03773	0,71232±0,05121
	BFBB-TC DS W5	0,01018±0,00127	0,70614±0,03773	0,71232±0,05121
	BFBB-TC DS W2	0,01027±0,00121	0,69802±0,03851	0,70392±0,05112
Diabetes	SVM	0,00321±0,00031	0,70326±0,02975	0,70468±0,03601
	TSVM	0,00171±0,00015	0,70510±0,02831	0,70086±0,03843
	BFBB-TC	0,00363±0,00038	0,71857±0,03090	0,71666±0,03794
	BFBB-TC DS	0,00349±0,00038	0,71952±0,03276	0,72013±0,03577
	BFBB-TC DS W5	0,00318±0,00030	0,71605±0,03807	0,71354±0,03885
	BFBB-TC DS W2	0,00302±0,00027	0,71061±0,03584	0,71215±0,04186
Echocardiogram	SVM	0,04225±0,01744	0,74274±0,07204	0,73636±0,08082
	TSVM	0,01634±0,00623	0,74784±0,04692	0,74545±0,07319
	BFBB-TC	0,02666±0,00958	0,73411±0,09021	0,72020±0,08559
	BFBB-TC DS	0,02704±0,00987	0,75725±0,07934	0,74343±0,10057
	BFBB-TC DS W5	0,02704±0,00987	0,75725±0,07934	0,74343±0,10057
	BFBB-TC DS W2	0,02693±0,01001	0,75333±0,07982	0,74545±0,09765
German	SVM	0,02123±0,00329	0,68479±0,01908	0,67840±0,03131
	TSVM	0,01212±0,00153	0,68239±0,02611	0,67760±0,03294
	BFBB-TC	0,01270±0,00163	0,68531±0,02329	0,67906±0,02553
	BFBB-TC DS	0,01265±0,00160	0,68526±0,02155	0,67986±0,02835
	BFBB-TC DS W5	0,01265±0,00160	0,68526±0,02155	0,67986±0,02835
	BFBB-TC DS W2	0,01271±0,00158	0,68772±0,02109	0,68066±0,02896

Fonte: Elaborado pelo autor (2019).

Diferente dos resultados obtidos no experimento anterior, onde pelo menos um dos métodos semissupervisionados sempre teve o melhor resultado para a acurácia do conjunto de trabalho, o SVM foi capaz de obter melhores resultados de acurácia nas bases *Haberman*, *Ionosphere* e *LiverDisorders*. Este resultado não é tão surpreendente, pois mesmo o SVM não obtendo o melhor resultado sobre o conjunto de trabalho, ele conseguiu superar alguns dos métodos em algumas ocasiões. Ainda assim, a grande maioria dos melhores resultados foi obtida por um dos métodos semissupervisionados, mais uma vez reforçando a capacidade desta abordagem em encontrar melhores resultados em um aprendizado transdutivo.

Avaliando também quais métodos foram capazes de obter os melhores resultados para a acurácia do conjunto de teste, pode-se ver que o SVM só foi capaz de obter os melhores resultados nas bases *Ionosphere* e *LiverDisorders*, sendo o TSVM e o BFBB-TC responsáveis pela maior parte dos melhores resultados. Isto indica que métodos

Tabela 9 – Resultados de inferência dos experimentos com *kernel* RBF (2).

Base	Método	Margem	Acurácia Trab.	Acurácia Teste
Haberman	SVM	0,00682±0,00261	0,65697±0,04931	0,67368±0,05565
	TSVM	0,00369±0,00089	0,65272±0,04382	0,67017±0,06112
	BFBB-TC	0,00686±0,00222	0,64693±0,05763	0,67412±0,06368
	BFBB-TC DS	0,00640±0,00218	0,64217±0,05692	0,65921±0,07184
	BFBB-TC DS W5	0,00635±0,00201	0,63707±0,05759	0,66271±0,06069
	BFBB-TC DS W2	0,00597±0,00152	0,63231±0,06338	0,64824±0,05883
Heart	SVM	0,02096±0,01054	0,75934±0,03616	0,75621±0,07325
	TSVM	0,00920±0,00384	0,76897±0,03823	0,77611±0,07022
	BFBB-TC	0,01318±0,00489	0,75433±0,04145	0,75273±0,05454
	BFBB-TC DS	0,01293±0,00478	0,75394±0,04479	0,76268±0,07871
	BFBB-TC DS W5	0,01293±0,00478	0,75394±0,04479	0,76268±0,07871
	BFBB-TC DS W2	0,01289±0,00482	0,75664±0,04470	0,76169±0,08037
Heart-statlog	SVM	0,04678±0,02102	0,74142±0,05691	0,74577±0,05726
	TSVM	0,02268±0,00843	0,75664±0,04642	0,74776±0,04869
	BFBB-TC	0,02962±0,00962	0,74682±0,05298	0,73482±0,06017
	BFBB-TC DS	0,02923±0,00959	0,73872±0,04957	0,75223±0,05674
	BFBB-TC DS W5	0,02924±0,00959	0,73853±0,04939	0,75223±0,05674
	BFBB-TC DS W2	0,02895±0,00968	0,74258±0,05321	0,74527±0,06708
House	SVM	0,18027±0,04820	0,92522±0,03000	0,92643±0,04123
	TSVM	0,08859±0,02149	0,92162±0,02918	0,93275±0,03660
	BFBB-TC	0,12101±0,02585	0,92680±0,04357	0,93218±0,05469
	BFBB-TC DS	0,11537±0,02348	0,91576±0,05111	0,92298±0,05465
	BFBB-TC DS W5	0,11537±0,02348	0,91576±0,05111	0,92298±0,05465
	BFBB-TC DS W2	0,11537±0,02348	0,91576±0,05111	0,92298±0,05465
House-votes	SVM	0,06290±0,01349	0,91450±0,01939	0,90956±0,03286
	TSVM	0,02650±0,00446	0,91810±0,01811	0,91543±0,02975
	BFBB-TC	0,03982±0,01000	0,89820±0,03605	0,89382±0,04174
	BFBB-TC DS	0,03866±0,01044	0,89424±0,03701	0,89104±0,03772
	BFBB-TC DS W5	0,03866±0,01044	0,89424±0,03701	0,89104±0,03772
	BFBB-TC DS W2	0,03847±0,01065	0,89568±0,03771	0,89259±0,03879

Fonte: Elaborado pelo autor (2019).

transdutivos e semissupervisionados também podem levar a bons resultados de indução.

Observando-se os melhores resultados para a acurácia de trabalho e de teste, é possível ver que há uma maior consistência entre ambos, sendo que em nove das quinze bases utilizadas o método que obteve o melhor resultado para a acurácia do conjunto de trabalho foi o mesmo para o conjunto de teste. Enquanto isso, nos resultados apresentados na Tabela 5 essa consistência ocorreu em apenas uma das quatro bases.

Nas bases *Clean1*, *Echocardiogram*, *German*, *Heart*, *House* e *House-Votes*, os resultados obtidos pelo BFBB-TC DS e pelo BFBB-TC DS W5 foram os mesmos, mostrando mais uma vez que, para o nível de exploração permitido nos experimentos, um valor pequeno de janela deve ser utilizado. Na base *House*, pode-se notar que até mesmo o BFBB-TC DS W2 obtém os mesmos resultados. Como um valor de janela menor ($w = 1$) faria com que a busca só visitasse estados vizinhos ao mais profundo, o mais indicado seria manter a janela como pelo menos dois e aumentar o número total de estados a serem

Tabela 10 – Resultados de inferência dos experimentos com *kernel* RBF (3).

Base	Método	Margem	Acurácia Trab.	Acurácia Teste
Ionosphere	SVM	0,04030±0,01415	0,83733±0,03135	0,83793±0,04042
	TSVM	0,01161±0,00308	0,83288±0,05476	0,82107±0,05513
	BFBB-TC	0,02363±0,00683	0,82829±0,03739	0,82528±0,04241
	BFBB-TC DS	0,02362±0,00655	0,82488±0,03353	0,82107±0,03802
	BFBB-TC DS W5	0,02331±0,00661	0,82192±0,03812	0,81839±0,04660
	BFBB-TC DS W2	0,02244±0,00669	0,80592±0,04323	0,81149±0,04910
Isolet	SVM	0,20804±0,01048	0,98859±0,00519	0,98800±0,00933
	TSVM	0,12105±0,01009	0,98503±0,00947	0,99088±0,00811
	BFBB-TC	0,13692±0,00220	0,99416±0,00507	0,99466±0,00790
	BFBB-TC DS	0,13685±0,00222	0,99103±0,01057	0,99066±0,01233
	BFBB-TC DS W5	0,13686±0,00222	0,99103±0,00753	0,99155±0,00957
	BFBB-TC DS W2	0,13686±0,00222	0,99338±0,00516	0,99444±0,00723
Krvskp	SVM	0,01777±0,00141	0,95618±0,00855	0,95698±0,01112
	TSVM	0,00962±0,00097	0,96169±0,00771	0,95757±0,01016
	BFBB-TC	0,00999±0,00113	0,95996±0,00745	0,95990±0,01003
	BFBB-TC DS	0,00956±0,00120	0,95765±0,00877	0,95594±0,01093
	BFBB-TC DS W5	0,00961±0,00119	0,95742±0,00844	0,95598±0,01113
	BFBB-TC DS W2	0,01012±0,00115	0,95657±0,00975	0,95561±0,01305
LiverDisorders	SVM	0,00426±0,00066	0,64992±0,04864	0,63178±0,06938
	TSVM	0,00254±0,00038	0,64856±0,05083	0,62558±0,06824
	BFBB-TC	0,00587±0,00089	0,60618±0,06399	0,60503±0,06986
	BFBB-TC DS	0,00546±0,00090	0,59954±0,04852	0,60077±0,06789
	BFBB-TC DS W5	0,00533±0,00074	0,60769±0,05045	0,60658±0,08044
	BFBB-TC DS W2	0,00517±0,00067	0,60769±0,04655	0,61007±0,07558
Specif	SVM	0,01441±0,00302	0,74125±0,03979	0,73256±0,05175
	TSVM	0,00723±0,00108	0,74529±0,04653	0,74176±0,06147
	BFBB-TC	0,00961±0,00120	0,74843±0,03892	0,73333±0,04870
	BFBB-TC DS	0,00968±0,00136	0,74813±0,04489	0,74827±0,04689
	BFBB-TC DS W5	0,00969±0,00136	0,74738±0,04529	0,74636±0,04616
	BFBB-TC DS W2	0,01014±0,00137	0,72825±0,05771	0,73601±0,05310

Fonte: Elaborado pelo autor (2019).

explorados para permitir que a heurística tenha efeito.

Nas Tabelas 11, 12 e 13, assim como no experimento anterior, podem ser observados os valores de margem para as várias configurações do BFBB-TC com o melhor resultado destacado em negrito, o número de estados explorados (“# explorados”), o número de estados não inseridos (“# não inseridos”) na lista de prioridade, e a porcentagem restante do conjunto de trabalho não utilizada pela solução final (“% WS”). Todos os valores representam a média dos resultados obtidos acompanhados de seus desvios padrões.

Comparando-se somente os valores de margem do BFBB-TC em suas configurações, a maior margem foi obtida por um dos métodos heurísticos em apenas quatro das quinze bases, o que indica que apesar da possibilidade de afastamento do caminho ótimo por este tipo de abordagem o método foi capaz de se manter consistente na maior parte dos casos.

É interessante de se observar que nas bases *Heart*, *House* e *Isolet* o BFBB-TC DS W2 foi capaz de atingir o mesmo valor de margem (na média) que o BFBB-TC DS e o

Tabela 11 – Resultados de exploração dos experimentos com *kernel* linear (1).

Base	Método	Margem	# explorados	# não inseridos	% WS
Australian	BFBB-TC	0,01102±0,00519	10045,1±294,8	351,0±1546,0	0,95457±0,03198
	BFBB-TC DS	0,01101±0,00519	9018,7±2817,1	881,0±1485,6	0,94588±0,02989
	BFBB-TC DS W5	0,01097±0,00521	8928,8±3044,9	712,1±1014,5	0,93847±0,03518
	BFBB-TC DS W2	0,01092±0,00519	7955,5±3762,3	2086,7±2549,4	0,91836±0,04130
Clean1	BFBB-TC	0,01028±0,00120	10024,0±0,0	0,0±0,0	0,95427±0,00459
	BFBB-TC DS	0,01018±0,00127	10024,0±0,0	0,8±2,6	0,93991±0,00706
	BFBB-TC DS W5	0,01018±0,00127	10024,0±0,0	0,8±2,6	0,94057±0,00503
	BFBB-TC DS W2	0,01027±0,00121	10024,0±0,0	0,9±2,4	0,92653±0,00774
Diabetes	BFBB-TC	0,00363±0,00038	2803,7±2653,7	6,9±10,5	0,73571±0,05643
	BFBB-TC DS	0,00349±0,00038	2168,5±2840,4	26,2±56,0	0,73224±0,06637
	BFBB-TC DS W5	0,00318±0,00030	305,8±114,1	1,4±3,5	0,71102±0,05705
	BFBB-TC DS W2	0,00302±0,00027	161,2±34,9	1,0±2,7	0,70312±0,05309
Echocardiog.	BFBB-TC	0,02666±0,00958	175,7±286,8	162,2±272,2	0,91058±0,04536
	BFBB-TC DS	0,02704±0,00987	55,6±76,6	97,9±138,5	0,87215±0,07070
	BFBB-TC DS W5	0,02704±0,00987	55,1±75,0	96,5±134,4	0,87215±0,07070
	BFBB-TC DS W2	0,02693±0,01001	45,3±52,9	74,0±85,2	0,88156±0,06232
German	BFBB-TC	0,01270±0,00163	10761,0±0,0	0,0±0,0	0,97873±0,00238
	BFBB-TC DS	0,01265±0,00160	10761,0±0,0	0,8±2,7	0,97115±0,00371
	BFBB-TC DS W5	0,01265±0,00160	10761,0±0,0	0,8±2,7	0,97126±0,00286
	BFBB-TC DS W2	0,01271±0,00158	10761,0±0,0	0,9±2,9	0,96739±0,00292

Fonte: Elaborado pelo autor (2019).

Tabela 12 – Resultados de exploração dos experimentos com *kernel* linear (2).

Base	Método	Margem	# explorados	# não inseridos	% WS
Haberman	BFBB-TC	0,00686±0,00222	1299,2±1834,8	454,8±968,5	0,83265±0,09107
	BFBB-TC DS	0,00640±0,00218	172,1±182,5	132,7±248,4	0,81734±0,10873
	BFBB-TC DS W5	0,00635±0,00201	137,6±111,1	80,8±125,9	0,82006±0,11348
	BFBB-TC DS W2	0,00597±0,00152	68,7±41,9	35,2±45,3	0,82653±0,12957
Heart	BFBB-TC	0,01318±0,00489	4009,1±3409,3	2786,7±2486,3	0,91695±0,03361
	BFBB-TC DS	0,01293±0,00478	1338,5±2462,0	1530,4±1754,9	0,89903±0,03803
	BFBB-TC DS W5	0,01293±0,00478	1315,6±2459,9	1498,5±1794,2	0,89942±0,03793
	BFBB-TC DS W2	0,01289±0,00482	590,9±900,5	690,2±710,3	0,89672±0,04613
Heart-statlog	BFBB-TC	0,02962±0,00962	8293,8±3290,0	1779,2±2373,4	0,91599±0,02101
	BFBB-TC DS	0,02923±0,00959	5650,8±4072,4	6870,6±6057,6	0,88131±0,04409
	BFBB-TC DS W5	0,02924±0,00959	5080,2±3869,2	5617,7±4737,4	0,87321±0,05375
	BFBB-TC DS W2	0,02895±0,00968	2198,8±2279,9	2039,5±1688,6	0,85857±0,06380
House	BFBB-TC	0,12101±0,02585	745,8±1822,5	554,9±873,5	0,82432±0,08568
	BFBB-TC DS	0,11537±0,02348	160,3±306,2	307,4±605,3	0,84279±0,11868
	BFBB-TC DS W5	0,11537±0,02348	141,5±227,1	252,2±375,2	0,84279±0,11868
	BFBB-TC DS W2	0,11537±0,02348	102,9±99,0	157,0±117,1	0,84279±0,11868
House-votes	BFBB-TC	0,03982±0,01000	6058,2±3974,5	3681,9±3044,7	0,94808±0,02976
	BFBB-TC DS	0,03866±0,01044	2637,4±3287,4	4401,5±5482,6	0,90575±0,05827
	BFBB-TC DS W5	0,03866±0,01044	2261,3±2947,0	3347,1±4315,6	0,90575±0,05830
	BFBB-TC DS W2	0,03847±0,01065	960,9±1026,9	980,3±1103,0	0,90527±0,06345

Fonte: Elaborado pelo autor (2019).

Tabela 13 – Resultados de exploração dos experimentos com *kernel* linear (3).

Base	Método	Margem	# explorados	# não inseridos	% WS
Ionosphere	BFBB-TC	0,02363±0,00683	10011,0±0,0	1120,4±1613,2	0,91733±0,01244
	BFBB-TC DS	0,02362±0,00655	9129,6±2248,7	6874,7±3326,3	0,90311±0,02909
	BFBB-TC DS W5	0,02331±0,00661	8737,0±2661,5	8311,6±3649,9	0,88503±0,03570
	BFBB-TC DS W2	0,02244±0,00669	1452,8±1306,4	1238,5±810,4	0,89125±0,07227
Isolet	BFBB-TC	0,13692±0,00220	10054,0±0,0	31,7±115,8	0,96518±0,00309
	BFBB-TC DS	0,13685±0,00222	9901,9±433,8	8162,7±7591,9	0,94447±0,03741
	BFBB-TC DS W5	0,13686±0,00222	5818,2±3197,8	1062,0±821,6	0,84891±0,10098
	BFBB-TC DS W2	0,13686±0,00222	1878,9±1650,1	181,7±75,2	0,75561±0,03713
Krvskp	BFBB-TC	0,00999±0,00113	49925,0±0,0	0,0±0,0	0,99198±0,00134
	BFBB-TC DS	0,00956±0,00120	49925,0±0,0	0,0±0,0	0,98904±0,00100
	BFBB-TC DS W5	0,00961±0,00119	49925,0±0,0	0,0±0,0	0,98881±0,00119
	BFBB-TC DS W2	0,01012±0,00115	49925,0±0,0	0,0±0,0	0,97919±0,00355
LiverDisord.	BFBB-TC	0,00587±0,00089	434,0±328,6	9,0±22,5	0,62337±0,08276
	BFBB-TC DS	0,00546±0,00090	124,7±66,0	12,8±22,6	0,63725±0,12717
	BFBB-TC DS W5	0,00533±0,00074	106,6±49,1	6,2±13,0	0,61719±0,13142
	BFBB-TC DS W2	0,00517±0,00067	72,1±19,1	4,0±6,8	0,62835±0,10277
Spectf	BFBB-TC	0,00961±0,00120	10011,0±0,0	8,7±30,3	0,94020±0,00679
	BFBB-TC DS	0,00968±0,00136	9723,4±1016,7	4634,5±5155,3	0,90403±0,05277
	BFBB-TC DS W5	0,00969±0,00136	9691,2±1063,6	4671,3±4961,8	0,90134±0,05197
	BFBB-TC DS W2	0,01014±0,00137	6362,7±2615,6	3830,9±2502,3	0,78221±0,08746

Fonte: Elaborado pelo autor (2019).

BFBB-TC DS W5 explorando consideravelmente menos estados. O que mostra que, para determinadas bases, uma estratégia mais agressiva de poda e de geração de estados pode tornar o processo mais eficiente.

Dois comportamentos podem ser observados em relação ao número de estados não inseridos na fila quando compara-se as heurísticas com a solução exata. No primeiro caso, há um incremento no número de estados não inseridos, o que sugere uma melhor atualização do limite inferior permitindo uma maior poda. No segundo caso, o número de estados não inseridos diminui, o que pode significar que o caminho percorrido atinge uma solução sem atingir os estados que levaram a uma melhora do limite. Para a base *Krvskp* o método não foi capaz de realizar podas através do limite inferior, o que está relacionado ao maior tamanho do conjunto de trabalho.

Os percentuais do conjunto de trabalho não explorado em “% WS” se mantiveram dentro da expectativa, em que os métodos heurísticos se utilizam de um volume maior das amostras não rotuladas devido as estratégias empregadas. Ainda assim, alguns destes resultados chamam a atenção na Tabela 13. Nas bases *Isolet* e *Spectf* o percentual de amostras remanescentes é consideravelmente menor para o BFBB-TC DS W2, sendo que o resultado de margem na *Isolet* foi igual a outras configurações e na *Spectf* foi maior do que no BFBB-TC, o que indica um desvio do caminho ótimo.

5 CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho foi apresentado o algoritmo *Best-First Branch-and-Bound Transductive Classifier* de classificação semissupervisionada baseado em margem que permite a busca pela solução globalmente ótima. O método combina os processos de busca *branch-and-bound* e *best-first* para a solução do problema de otimização combinatória da rotulação de um conjunto de dados. Para lidar com as limitações de usar o método com um classificador de máxima margem e com a característica combinatorial do espaço de busca, foram propostas soluções heurísticas para resolver problemas com conjuntos de trabalho maiores. Os resultados experimentais mostraram que a heurística de limite inferior pode lidar com conjuntos de trabalho maiores e também encontrar uma solução viável competitiva, com a vantagem de fazê-la em tempo constante. As heurísticas de seleção de amostras e eliminação de estados por janela da árvore podem levar a uma exploração mais eficiente do espaço de busca.

Em relação a heurística de limite inferior, o algoritmo BFBB-TC foi comparado ao TSVM utilizando o *kernel* linear, sempre se mostrando capaz de encontrar uma margem superior nas bases testadas. Com essa estratégia para cálculo do limite inferior, é possível encontrar uma solução de margem viável a qualquer momento da busca, visto que o limite é computado desde o início da exploração.

Para lidar com a característica combinatorial do problema, foram propostas duas novas heurísticas relacionadas a geração e eliminação de estados da árvore de busca. A primeira heurística modifica a estratégia de seleção de amostras do conjunto de trabalho para geração de novos estados, selecionando uma amostra mais próxima a superfície de decisão em cada um dos subespaços. Esta estratégia faz com que a árvore gerada se torne mais compacta, evitando a geração de estados intermediários. A segunda heurística define um limite, chamado de janela, para a seleção do próximo estado a ser explorado, evitando que estados distantes do nível mais profundo atingido sejam expandidos e eliminando os mesmos. Seguindo esta estratégia de eliminação faz com que a busca seja concentrada cada vez mais em níveis mais profundos da árvore. Os experimentos realizados indicam que as heurísticas propostas são capazes de reduzir significativamente o número de estados explorados ao mesmo tempo acelerar a melhoria do limite inferior, o que permite que mais estados sejam podados durante a exploração.

Também foram realizados os primeiros experimentos comparando a capacidade de generalização das soluções encontradas pelo BFBB-TC. Os resultados encontrados mostram que a generalização das soluções encontradas pelo método são comparáveis as de outros métodos. Levando-se em consideração que houve uma limitação da exploração realizada pelo BFBB-TC, é possível que o método seja capaz de encontrar soluções ainda melhores.

Uma avaliação mais profunda sobre os efeitos da aplicação da heurística de janela pode ser realizada. Uma avaliação pode ser feita comparando-se qual a porção de estados são descartados, quais os níveis atingidos e qual o impacto sobre o tempo para a solução, avaliando com uma maior variação dos valores de janela. Além disso, a avaliação pode ser feita diretamente sobre o método exato, desassociada da heurística de seleção de duas amostras como foi feito neste trabalho.

Outra possibilidade para lidar com os limites computacionais de usar o SVM de margem rígida seria usar um classificador baseado em margem implementado em uma configuração iterativa. Em outras palavras, um classificador que pode tirar proveito de uma solução anterior e que também fornece uma função de avaliação monótona com base em seus valores de margem.

Por fim, uma outra possibilidade de trabalho futuro seria modificar a função de avaliação utilizada pela busca ordenada para, ao invés de buscar a maximização da margem, executar a otimização da distribuição da margem (Zhang e Zhou, 2018), caso esta se mantenha monótona.

REFERÊNCIAS

- Araújo, H. X.; Fonseca Neto, R.; Villela, S. M. (2018). Uma busca ordenada branch-and-bound para solução do problema de inferência transdutiva usando máquinas de vetores suporte. In *6th Symposium on Knowledge Discovery, Mining and Learning, KDMiLe 2018, São Paulo, Brazil, October 22-25, 2018*.
- Araújo, H. X.; Fonseca Neto, R.; Villela, S. M. (2019). A best-first branch-and-bound search for solving the transductive inference problem using support vector machines. In *27th European Symposium on Artificial Neural Networks, ESANN 2019, Bruges, Belgium, April 24-26, 2019*.
- Bennett, K. P.; Demiriz, A. (1999). Semi-supervised support vector machines. In *Proceedings of the 1998 Conference on Advances in Neural Information Processing Systems II*, pages 368–374, Cambridge, MA, USA. MIT Press.
- Boser, B. E.; Guyon, I. M.; Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory, COLT '92*, pages 144–152, New York, NY, USA. ACM.
- Chang, C.; Lin, C. (2011). LIBSVM: A library for support vector machines. *ACM TIST*, **2**(3), 27:1–27:27.
- Chapelle, O.; Schölkopf, B.; Zien, A. (2006a). Introduction to semi-supervised learning. In Chapelle *et al.* (2006b), pages 1–12.
- Chapelle, O.; Schölkopf, B.; Zien, A., editors (2006b). *Semi-Supervised Learning*. The MIT Press.
- Chapelle, O.; Sindhwani, V.; Keerthi, S. S. (2007). Branch and bound for semi-supervised support vector machines. In B. Schölkopf; J. C. Platt; T. Hoffman; T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 217–224. MIT Press, Hyatt Regency Vancouver, in Vancouver, B.C., Canada.
- Chen, Y.; Wang, G.; Dong, S. (2003). Learning with progressive transductive support vector machine. *Pattern Recognition Letters*, **24**(12), 1845–1855.
- Gamerman, A.; Vovk, V.; Vapnik, V. (1998). Learning by transduction. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, UAI'98*, pages 148–155, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Graepel, T.; Herbrich, R.; Obermayer, K. (1999). Bayesian transduction. In *Proceedings of the 12th International Conference on Neural Information Processing Systems, NIPS'99*, pages 456–462, Cambridge, MA, USA. MIT Press.
- Hart, P. E.; Nilsson, N. J.; Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, **4**(2), 100–107.
- Hofmann, T.; Schölkopf, B.; Smola, A. J. (2008). Kernel methods in machine learning. *The Annals of Statistics*, **36**(3), 1171–1220.

- Joachims, T. (1999). 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. Morgan Kaufmann Publishers Inc.
- Li, Y.; Tsang, I. W.; Kwok, J. T.; Zhou, Z. (2013). Convex and scalable weakly labeled svms. *Journal of Machine Learning Research*, 14(1), 2151–2188.
- Narendra, P. M.; Fukunaga, K. (1977). A branch and bound algorithm for feature subset selection. *IEEE Trans. Computers*, 26(9), 917–922.
- Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; VanderPlas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, E. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12, 2825–2830.
- Platt, J. C. (1999). Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf; C. J. C. Burges; A. J. Smola; A. J. Smola, editors, *Advances in Kernel Methods*, pages 185–208. MIT Press, Cambridge, MA, USA.
- Russell, S. J.; Norvig, P. (2010). *Artificial Intelligence - A Modern Approach, Third International Edition*. Pearson Education.
- Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag, Berlin, Heidelberg.
- Villela, S. M.; Leite, S. C.; Fonseca Neto, R. (2015). Feature selection from microarray data via an ordered search with projected margin. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pages 3874–3881. AAAI Press.
- Zhang, T.; Zhou, Z. (2018). Semi-supervised optimal margin distribution machines. In J. Lang, editor, *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden.*, pages 3104–3110. ijcai.org.