Uma Busca Ordenada *Branch-and-Bound* para solução do Problema de Inferência Transdutiva usando Máquinas de Vetores Suporte

Hygor Xavier Araújo, Raul Fonseca Neto, Saulo Moraes Villela

Universidade Federal de Juiz de Fora, Brasil
hygor.araujo@ice.ufjf.br, {raulfonseca.neto, saulo.moraes@ufjf.edu.br}

Abstract. Nesse artigo é apresentado um novo método para resolver o problema de inferência transdutiva cujo objetivo é predizer os rótulos binários de um subconjunto de pontos de interesse de uma função de decisão desconhecida. É utilizada a Máquina de Vetores Suporte para tentar encontrar um limite de decisão. Para obter a hipótese de margem máxima sobre as amostras rotuladas e não rotuladas, é empregada uma busca ordenada (*best-first*) admissível com base nos valores de margem. Evidências empíricas sugerem que esta solução globalmente ótima pode obter excelentes resultados no problema de transdução. Devido à estratégia de seleção usada, o algoritmo de busca explora apenas uma pequena fração de amostras não rotuladas, tornando-a eficiente para bases de dados de tamanho médio. Os resultados obtidos foram comparados com os resultados da *Transductive Support Vector Machine*, demonstrando melhores resultados em valores de margem.

Categories and Subject Descriptors: I.2.6 [**Artificial Intelligence**]: Learning

Keywords: inferência transdutiva, aprendizado semissupervisionado, busca ordenada admissível, máquina de vetores suporte, separação de baixa densidade

1. INTRODUÇÃO

Em muitas aplicações, o processo de rotular amostras em um conjunto de dados é muito difícil, caro ou demorado, em alguns casos exigindo a classificação manual por um especialista. Nestes casos, geralmente existe um pequeno conjunto de dados rotulados e um grande número de dados não rotulados. O aprendizado semissupervisionado surge como uma solução para este tipo de situação. Neste tipo de aprendizado, alguns dados rotulados (conjunto de treinamento) são necessários para a construção do modelo e, além disso, também é possível usar dados não rotulados (conjunto de trabalho) na construção do modelo. Com essa configuração, espera-se que a solução encontrada seja melhor do que seria possível apenas com os dados rotulados ou não rotulados.

Portanto, é possível utilizar a aprendizagem semissupervisionada em problemas de classificação como uma tentativa de melhorar a capacidade de generalização, usando dados rotulados e não rotulados simultaneamente. Quase todos os métodos relacionados a aprendizagem semissupervisionada empregam a suposição de agrupamento. Esta hipótese afirma que o limite de decisão deve estar em regiões de baixa densidade [Chapelle et al. 2006]. Assim, faz sentido usar um classificador de larga margem, como a Máquina de Vetores Suporte (*Support Vector Machine* – SVM), para encontrar um hiperplano separador de margem máxima dos conjuntos de treinamento e trabalho. Dessa forma, as máquinas de vetores suporte transdutivas [Vapnik 1995] implementam a suposição de agrupamento diretamente, tentando encontrar uma superfície de decisão que esteja longe das amostras rotuladas e não rotuladas. A indução transdutiva pode ser tratada como um caso especial de aprendizado semissupervisionado se a hipótese transdutiva for usada para inferir os rótulos de novas amostras.

No entanto, encontrar a solução ótima exata do SVM transdutor ou o melhor esquema de rótulos para o conjunto de trabalho é um problema combinatório NP-difícil, tornando-se computacionalmente proibitivo para bases de dados com um grande número de amostras não rotuladas. Dado um problema

de classificação binária e um conjunto de trabalho de tamanho n, existem 2^n possíveis esquemas de rotulação.

Para superar este problema, uma busca ordenada que explore eficientemente o espaço de todos os esquemas de rotulação é proposta, encontrando a hipótese de margem máxima. O algoritmo emprega como função de avaliação os valores de margem. Esta é uma função monótona, pois os valores de margem diminuem monotonamente quando novos pontos são inseridos no espaço do problema e, portanto, o algoritmo de busca é admissível.

Uma avaliação extensiva do desempenho do modelo é fornecida através de um conjunto de experimentos de inferência transdutiva. Os resultados obtidos são comparados com os resultados da *Transductive Support Vector Machine* (TSVM), proposta em [Joachims 1999], demonstrando melhores resultados em valores de margem.

Após esta breve introdução, na Seção 2 são apresentados alguns trabalhos relacionados. Na Seção 3, conceitos preliminares como o problema de classificação binária, as máquinas de vetores suporte, o aprendizado semissupervisionado e as diferenças entre inferência indutiva e transdutiva são definidas. A Seção 4 apresenta o algoritmo de transdução proposto e a Seção 5 relata os experimentos e resultados. Finalmente, a Seção 6 apresenta a discussão e as perspectivas de trabalhos futuros.

2. TRABALHOS RELACIONADOS

Em [Gammerman et al. 1998] foi proposto o primeiro método para inferência transdutiva em problemas de classificação binária. O método é uma modificação do SVM e atribui a uma 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.

Em [Graepel et al. 1999] o problema de inferência transdutiva é modelado 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.

Em [Bennett and Demiriz 1999] é apresentada a *Semi-Supervised Support Vector Machine* (S^3VM). Neste método 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 facilitar a resolução do problema, 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 *Transductive Support Vector Machine* é apresentada em [Joachims 1999], que 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 pela primeira vez no contexto da 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.

Finalmente, [Chapelle et al. 2007] apresenta 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 à nossa proposta, mas difere nos três principais processos: ramificação, poda e exploração, e é apropriado apenas para bases de dados de tamanho pequeno. Como será visto na Seção 4, foram implementadas estratégias alternativas para esses processos, tornando o modelo proposto mais eficiente e aplicável em bases de dados de tamanho médio.

3. PRELIMINARES

3.1 Problema de classificação binária

Dado um conjunto de amostras X de tamanho m pertencentes a um espaço de entrada $\mathbb{R}^d$ de dimensão d com cada amostra x_i associada a um escalar $y_i \in Y$, pode-se definir o conjunto de treinamento de um problema de classificação como $Z = \{z_i = (x_i, y_i) \mid i \in \{1, \ldots, m\}, x_i \in X \text{ e } y_i \in Y\}$. Em um problema de classificação binária $y_i = -1$ ou $+1$.

O principal objetivo em um problema de classificação é encontrar uma função que generalize a partir de um conjunto de dados utilizado para treinamento, ou seja, que seja capaz de classificar novas amostras com uma acurácia considerada satisfatória. Esta função pode ser definida como um hiperplano com vetor normal $w \in \mathbb{R}^d$, também chamado de vetor de pesos, e uma constante $b \in \mathbb{R}$, chamado de viés. Este hiperplano deve separar o espaço de modo que as amostras $\{(x_i, y_i) \in Z \mid y_i = +1\}$ fiquem em um subespaço separado por ele e $\{(x_i, y_i) \in Z \mid y_i = -1\}$ no outro.

Para um conjunto de treinamento linearmente separável, é feita a busca por (w, b) sujeito a $y_i(w \cdot x_i + b) \geq 0, \forall (x_i, y_i) \in Z$. Para alguns conjuntos de treinamento, não haverá um hiperplano capaz de separar as amostras, porque Z não é linearmente separável em seu espaço original mas torna-se em um espaço projetado de maior dimensão. 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 $y_i(w \cdot x_i + b) \geq \gamma$, $\forall\, (x_i, y_i) \in Z$.

Uma maneira possível de encontrar esse hiperplano é usar um classificador de larga margem. Essa classe de algoritmo é capaz de definir uma distância entre o limite de decisão e as amostras.

3.2 Máquina de Vetor Suporte

Como mencionado anteriormente, uma maneira de encontrar o hiperplano para um problema de classificação é usando o algoritmo SVM. 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:

$$\max_{(w,b)} \left(\min_i \frac{y_i(w \cdot x_i + b)}{||w||} \right)$$
$$\text{s. a. } y_i(w \cdot x_i + b) > 0, \forall\, (x_i, y_i) \in Z,$$

onde $\gamma_i = y_i(w \cdot x_i + b)$ é a margem funcional. Para se ter uma noção adequada de distância relacionada ao hiperplano, é preciso definir a margem geométrica γ_g. A distância perpendicular a partir do hiperplano $\mathcal{H}$ até a origem é $|b|/||w||$. Defini-se dois hiperplanos paralelos a $\mathcal{H}$ como $\mathcal{H}^+ := \{x \in \mathbb{R}^d | w \cdot x + (b - \gamma) = 0\}$ e $\mathcal{H}^- := \{x \in \mathbb{R}^d | w \cdot x + (b + \gamma) = 0\}$, com a distância entre eles dada por:

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

então $\gamma_g := dist(\mathcal{H}^-, \mathcal{H}^+)/2$ fornece a margem geométrica entre os hiperplanos $\mathcal{H}^+$ e $\mathcal{H}^-$. Com isso, o problema de otimização pode ser reescrito como:

$$\max \gamma_g$$
$$\text{s. a. } y_i(w \cdot x_i + b) \geq ||w||\gamma_g, \forall\, (x_i, y_i) \in Z.$$

Fazendo $\gamma = 1 = \gamma_g||w||$ o valor mínimo da margem funcional, à formulação primal do SVM que minimiza a norma euclidiana, derivada por [Vapnik 1995], é obtida:

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

3.3 Aprendizado semissupervisionado e transdução

A aprendizagem semissupervisionada pode ser considerada estar entre a aprendizagem supervisionada e não supervisionada. A razão para isso é que, para encontrar um classificador, sua fase de aprendizado utiliza não apenas um conjunto de treinamento X_l, com todas as amostras já rotuladas, mas também um conjunto de trabalho X_u de amostras não rotuladas. O objetivo de usar esses dois conjuntos é ter um classificador melhor do que seria possível usando apenas um deles.

Da mesma forma como foi definido na Seção 3.1, o conjunto de treinamento X_l pode ser definido para a aprendizagem semissupervisionada como $X_l = \{(x_i, y_i)|i \in \{1, \ldots, m\}\}$. E o conjunto de trabalho como $X_u = \{x_j|j \in \{1, \ldots, k\}\}$.

Um algoritmo de aprendizagem pode ter como resultado uma função indutiva ou transdutiva. Algoritmos com uma configuração indutiva após sua fase de aprendizado é capaz de produzir uma função $f : \mathcal{X} \to y$ definida em todo espaço $\mathcal{X}$. Pelo contrário, com uma configuração transdutiva, o resultado seria uma função $f : X_u \to y_u$ que só é capaz de rotular as amostras do conjunto de trabalho.

Para novas amostras em uma configuração indutiva, a função resultante pode ser utilizada para fazer previsões sobre os rótulos. Em uma configuração transdutiva, seria necessário retreinar o modelo incluindo as novas amostras no conjunto de trabalho para obter seus rótulos.

A aprendizagem semissupervisionada pode ser vista como uma extensão da inferência indutiva para métodos discriminativos representados pelas hipóteses condicionais $P(y|x)$. Ela também considera o uso de dados não rotulados representados pelo $P(x)$ anterior. É fácil ver que $P(x)$ influencia $P(y|x)$ como no uso da Regra de Bayes para análise de discriminante.

Por outro lado, a ideia principal da aprendizagem transdutiva segue o fato de que, se você está limitado a uma quantidade restrita de informações, não se deve resolver o problema específico resolvendo um problema mais geral [Vapnik 1995].

Seguindo o modelo primal de SVM descrito na Seção 3.2, o problema de inferência transdutiva pode ser formulado como:

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

4. ALGORITMO TRANSDUTIVO

4.1 Espaço de estados e busca heurística

Um paradigma eficiente para lidar com a natureza combinatória do problema de inferência transdutiva é a busca heurística em que cada hipótese do problema é representada por um estado no espaço de busca. É possível citar, entre os principais métodos de busca, a busca ordenada (*best-first*) que emprega como estratégia de seleção a escolha do melhor entre todos os estados encontrados até o momento. No entanto, este método requer uma função de avaliação para medir o mérito dos estados e a condição de que esta função seja monótona decrescente para resolver problemas de maximização. Seguindo o algoritmo proposto em [Villela et al. 2015] o algoritmo *Best-First Branch-and-Bound Transductive Classifier* (BFBB-TC) foi desenvolvido acoplado a um SVM linear de margem rígida. Enquanto o AOS apresentado em [Villela et al. 2015] tem como objetivo solucionar o problema de seleção de características o BFBB-TC busca solucionar o problema de classificação transdutiva.

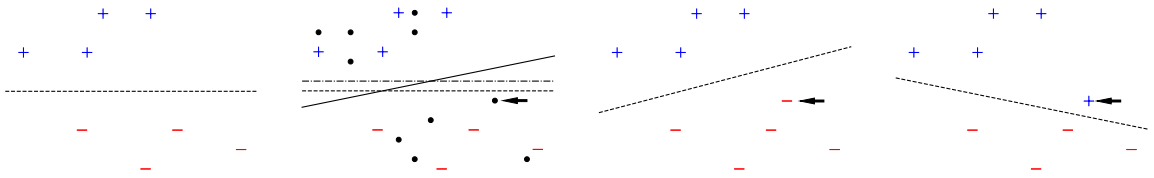
O algoritmo BFBB-TC usa os valores de margem do SVM como uma função de avaliação. Esta função de avaliação é monótona decrescente, satisfazendo a propriedade de admissibilidade e garantindo

a otimização da busca. Então, dado γ^{m+1} como o valor real da margem máxima para uma hipótese de um estado filho e que γ^m seja o valor real da margem máxima para a hipótese de seu pai. Assim, para um conjunto de treinamento com $m+1$ amostras, $\gamma^{m+1} \leq \gamma^m$ para todas transições de estados.

Os estados gerados, classificados pelos valores de margem, são armazenados em uma fila de prioridade implementada como uma estrutura de *heap* denominada H.

4.2 Ramificação

Observando em sequência a Figura 1 o processo de ramificação pode ser explicado da seguinte forma: retire do *heap* H a solução atual de margem de maior valor (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, existe uma margem de erro ou solução de margem inviável e é possível atualizar o limite inferior calculando o valor da margem da amostra que está mais próxima do hiperplano (Figura 1b). Esse limite poderia ser melhorado por uma estratégia de balanceamento. Então, este exemplo é selecionado para ser rotulado e gera dois novos estados S_+ e S_- que devem ser inseridos, após avaliação, no *heap* H (Figuras 1c e 1d).



(a) O hiperplano de máxima margem do conjunto de treinamento. Amostras positivas/negativas são marcadas como +/-. A linha tracejada é a solução do SVM indutivo.

(b) Inclusão do conjunto de trabalho (pontos). A linha sólida é a solução ótima do SVM transdutor e a tracejada pontilhada é a solução balanceada da Figura 1a. A seta indica a amostra selecionada.

(c) O hiperplano de máxima margem (limite superior) do conjunto de treinamento considerando a amostra selecionada, marcada com uma seta, como rótulo negativo.

(d) O hiperplano de máxima margem (limite superior) do conjunto de treinamento considerando a amostra selecionada, marcada com uma seta, como rótulo positivo.

Figura 1: O processo de ramificação.

4.3 Avaliação e poda

O processo de ramificação produz dois novos conjuntos de treinamento X_{l+} e X_{l-}, cada um com o conjunto de treinamento anterior e a amostra selecionada com um dos rótulos. O SVM é executado com X_{l+} e X_{l-} para obter as novas soluções com as margens γ_+ e γ_-. Em seguida, um novo limite superior para essa amostra é construído. Se não houver solução, o valor da margem será negativo e o respectivo estado deve ser eliminado e não inserido na estrutura da pilha. Além disso, todos os estados cujo valor de margem é menor que o limite inferior devem ser eliminados. Como sempre é feita a seleção da amostra que está mais próxima do hiperplano separador, essa amostra é um candidato em potencial para ser um vetor suporte na solução final quando o valor da margem é reduzido. Toda vez que o limite inferior é atualizado, os estados na pilha H com um valor de margem menor do que o limite são removidos da pesquisa.

Nesse sentido, o algoritmo seleciona apenas uma pequena fração das amostras não rotuladas. A propriedade de monotonicidade dos valores de margem é provada considerando o fato de que o novo problema de margem máxima é mais restrito do que o problema do pai, observando o fato de que a adição de uma nova restrição reduz o espaço de hipóteses. Portanto, a nova solução deve ser igual ou inferior à solução do pai.

4.4 Pseudocódigo

O Algoritmo 1 descreve o *Best-First Branch-and-Bound Transductive Classifier*.

Algoritmo 1: *Best-First Branch-and-Bound Transductive Classifier*

Entrada: conjunto de treinamento $X_l = \{(x_i, y_i) \mid i \in \{1, \ldots, m\}\}$;
conjunto de trabalho $X_u = \{x_j \mid j \in \{m+1, \ldots, k\}\}$;
Saída: rótulos do conjunto de trabalho Y_u;

início
  inicializar max-heap H;
  computar a solução usando o SVM com X_l para o estado inicial S;
  inserir S em H;
  enquanto H *não está vazio* **e** *a solução em* S *não é factível* **faça**
    encontrar a amostra mais próxima ($x_{nearest}$) ao hiperplano;
    computar novo limite inferior e atualizar se necessário;
    gerar novos conj de treinamento: $X_{l+} = X_l + \{x_{nearest}, +1\}$ e $X_{l-} = X_l + \{x_{nearest}, -1\}$;
    remover $x_{nearest}$ de X_u;
    computar soluções usando X_{l+} e X_{l-} para os novos estados S_+ e S_-;
    remover S de H;
    remover de H todos os estados onde $\gamma <$ limite inferior;
    se $\gamma_+ >$ *limite inferior* **então**
      inserir S_+ em H;
    fim se
    se $\gamma_- >$ *limite inferior* **então**
      inserir S_- em H;
    fim se
    selecionar novo S de H;
  fim enquanto
  Resultado: rótulos do conjunto de trabalho Y_u;
fim

5. EXPERIMENTOS E RESULTADOS

Nos experimentos, foi feita uma comparação entre o algoritmo BFBB-TC e o TSVM proposto em [Joachims 1999] usando o programa SVM *Light*[1]. A escolha pelo TSVM para comparação foi feita por se tratar de um método heurístico computacionalmente mais eficiente mas que não encontra a solução ótima como o método de busca admissível proposto. O BFBB-TC foi implementado em Python e usa como classificador o algoritmo SMO [Platt 1999] implementado na biblioteca Scikit-Learn [Pedregosa et al. 2011]. Para ambas as implementações, o conjunto de hiper-parâmetros foi o parâmetro de regularização C, com um valor de 10000, e o *kernel* escolhido foi linear. O valor escolhido para o parâmetro C faz com que a solução encontrada pelo SVM seja de margem rígida, o que se faz necessário para a monotonicidade da função objetivo e admissibilidade do método.

Para cada uma das bases, foram feitos experimentos com conjuntos de trabalho (WS) de tamanhos 50, 100, 200 e 300. A base BCI, devido ao seu tamanho reduzido, teve um conjunto de trabalho de tamanho máximo de 200. Esses conjuntos de trabalho foram criados a partir dos dados originais, fazendo 10 divisões aleatórias para selecionar as amostras para o conjunto de treinamento e conjunto de trabalho. O objetivo foi analisar o tamanho da margem e quanto do conjunto de trabalho foi explorado na solução. A escolha da margem como métrica está relacionada a obtenção da solução ótima (de maior margem) e com uma possível melhor generalização da mesma. Para executar os experimentos, o único pré-processamento feito foi normalizar os valores dos dados no intervalo [-1, 1].

[1]Disponível em http://svmlight.joachims.org/

5.1 Bases de dados

Para experimentação e análise do método, foram selecionadas quatro bases de dados do *benchmark*[2] criado em [Chapelle et al. 2006]. A escolha das bases se deve por todas serem linearmente separáveis, o que é necessário para encontrar um hiperplano separador considerando a utilização do SVM com margem rígida. Na Tabela I são apresentadas informações sobre as bases.

Tabela I: Informações das bases de dados.

Base	Atributos	Amostras		
		Pos.	Neg.	Total
Digit1	241	734	766	1500
USPS	241	1200	300	1500
$COIL_2$	241	750	750	1500
BCI	117	200	200	400

5.2 Resultados

Na Tabela II são mostrados os valores médios para a margem obtida, com seu respectivo desvio padrão, para as dez execuções do TSVM e do BFBB-TC. A coluna "WS" indica o tamanho do conjunto de trabalho. A coluna "Não exp." indica qual porcentagem do conjunto de trabalho não foi explorada na solução final do algoritmo BFBB-TC com o hiperplano que separa corretamente as amostras dos conjuntos de treinamento e de trabalho. A coluna "%" indica em porcentagem o quanto a margem do BFBB-TC encontrada foi maior que a do TSVM. Os melhores resultados são destacados em negrito.

Tabela II: Comparação entre BFBB-TC e TSVM.

Base	WS	TSVM	BFBB-TC		
		Margem	Margem	%	Não exp.
Digit1	50	0,05249 ± 0,00259	**0,05391 ± 0,00155**	2,71%	91,60%
	100	0,05265 ± 0,00345	**0,05486 ± 0,00127**	4,20%	92,90%
	200	0,05559 ± 0,00224	**0,05794 ± 0,00211**	4,23%	91,55%
	300	0,05723 ± 0,00497	**0,06044 ± 0,00259**	5,61%	91,27%
USPS	50	0,01272 ± 0,00047	**0,01289 ± 0,00046**	1,30%	98,20%
	100	0,01502 ± 0,00113	**0,01529 ± 0,00107**	1,74%	96,70%
	200	0,01946 ± 0,00133	**0,01996 ± 0,00161**	2,53%	95,10%
	300	0,02452 ± 0,00237	**0,02513 ± 0,00247**	2,51%	93,23%
$COIL_2$	50	0,00798 ± 0,00052	**0,00828 ± 0,00048**	3,75%	95,00%
	100	0,00832 ± 0,00057	**0,00864 ± 0,00040**	3,85%	93,20%
	200	0,00980 ± 0,00051	**0,01015 ± 0,00048**	3,56%	91,85%
	300	0,01092 ± 0,00089	**0,01135 ± 0,00076**	3,91%	91,37%
BCI	50	0,00627 ± 0,00076	**0,00646 ± 0,00087**	3,04%	90,80%
	100	0,00837 ± 0,00134	**0,00870 ± 0,00139**	3,90%	87,60%
	200	0,01645 ± 0,00387	**0,01853 ± 0,00477**	12,66%	84,95%

Conforme mostrado na Tabela II o algoritmo BFBB-TC alcançou uma margem maior em todos os casos, como esperado. Dado que uma margem maior é alcançada, espera-se que o classificador também tenha uma melhor generalização. Embora o TSVM seja capaz de encontrar uma solução, mesmo para bases de dados maiores, ela não é a solução ótima.

[2]As bases de dados podem ser encontrados em http://olivier.chapelle.cc/ssl-book/benchmarks.html

Uma questão muito importante relacionada a esse método é que apenas um pequeno percentual, variando de 1,80 a 15,05%, do conjunto de trabalho foi realmente necessário para encontrar a solução final nos experimentos. Com um conjunto de trabalho de tamanho n existem 2^n possíveis esquemas de rotulação para expandir no total, mas caso seja preciso expandir utilizando no máximo 10% das amostras existiriam apenas 2^k estados, onde $k = 0,1 \cdot n$. Levando isso em consideração, se torna possível resolver problemas com conjuntos de trabalho maiores. Outro detalhe interessante é que não seria necessário saber previamente quais amostras do seu conjunto de trabalho são as mais importantes, o algoritmo determinará isso de acordo com o seu treinamento e a distribuição dos dados não rotulados.

6. CONSIDERAÇÕES FINAIS

Nesse trabalho, foi proposto o algoritmo BFBB-TC, que combina uma estratégia de busca ordenada com a técnica *branch-and-bound* e o SVM para encontrar o esquema ideal de rotulação que resolve o problema de transdução. Os resultados, como mostrado na Tabela II, foram muito promissores, incentivando a continuidade dos estudos.

Considerando o fato de que a propriedade de monotonicidade da função de avaliação é preservada no espaço de características, como trabalho futuro, pretende-se desenvolver a implementação dual do modelo permitindo a possibilidade de inferência transdutiva não linear com o uso de funções *kernel* e a consequente solução de problemas não linearmente separáveis.

Também é considerada a possibilidade de alterar o SVM por um classificador de larga margem implementado em uma configuração iterativa. Nesse caso, o problema de otimização pode ser resolvido de forma mais eficiente tomando como solução inicial a solução do problema pai [Villela et al. 2016], o que poderia melhorar a eficiência do método.

REFERÊNCIAS

Bennett, K. P. and Demiriz, A. Semi-supervised support vector machines. In *Proceedings of the 1998 Conference on Advances in Neural Information Processing Systems II*. MIT Press, Cambridge, MA, USA, pp. 368–374, 1999.

Boser, B. E., Guyon, I. M., and Vapnik, V. N. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. ACM, New York, NY, USA, pp. 144–152, 1992.

Chapelle, O., Schölkopf, B., and Zien, A., editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.

Chapelle, O., Sindhwani, V., and Keerthi, S. S. Branch and bound for semi-supervised support vector machines. In *Advances in Neural Information Processing Systems 19*, B. Schölkopf, J. C. Platt, and T. Hoffman (Eds.). MIT Press, Hyatt Regency Vancouver, in Vancouver, B.C., Canada, pp. 217–224, 2007.

Gammerman, A., Vovk, V., and Vapnik, V. Learning by transduction. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 148–155, 1998.

Graepel, T., Herbrich, R., and Obermayer, K. Bayesian transduction. In *Proceedings of the 12th International Conference on Neural Information Processing Systems*. MIT Press, Cambridge, MA, USA, pp. 456–462, 1999.

Joachims, T. Transductive inference for text classification using support vector machines. In *Proceedings of the Sixteenth International Conference on Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 200–209, 1999.

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., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* vol. 12, pp. 2825–2830, 2011.

Platt, J. C. Fast training of support vector machines using sequential minimal optimization. In *Advances in Kernel Methods*, B. Schölkopf, C. J. C. Burges, and A. J. Smola (Eds.). MIT Press, Cambridge, MA, USA, pp. 185–208, 1999.

Vapnik, V. N. *The Nature of Statistical Learning Theory*. Springer-Verlag, Berlin, Heidelberg, 1995.

Villela, S. M., Leite, S. C., and Fonseca Neto, R. Feature selection from microarray data via an ordered search with projected margin. In *Proceedings of the 24th International Conference on Artificial Intelligence*. AAAI Press, Buenos Aires, Argentina, pp. 3874–3881, 2015.

Villela, S. M., Leite, S. C., and Fonseca Neto, R. Incremental p-margin algorithm for classification with arbitrary norm. *Pattern Recognition* vol. 55, pp. 261–272, 2016.