Transfer Learning for Synthetic Examples Selection in Meta-learning

Regina R. Parente[1], **Ricardo B. C. Prudencio**[1]

[1]Centro de Informática – Universidade Federal de Pernambuco (UFPE)
Caixa Postal 7851 – 50.732-970 – Recife – PE – Brazil

{rrp,rbcp}@cin.ufpe.br

Abstract. *In Meta-learning, training examples are generated from experiments performed with a pool of candidate algorithms in a number of problems (real or synthetic). Generating a good set of examples can be difficult due to the low availability of real datasets in some domains and the high computational cost of labeling. In this paper, we focus on the selection of training meta-examples by combining data manipulation and Transfer Learning via One-class classification. So, the most relevant examples are selected to be labeled. Our experiments revealed that it is possible to reduce the computational cost of generating meta-examples and maintain the meta-learning performance.*

Resumo. *Em Meta-aprendizagem, exemplos de treinamento são gerados a partir de experimentos realizados com um conjunto de algoritmos candidatos em um número de problemas (reais ou sintéticos). A geração de um bom conjunto de exemplos pode ser difícil devido à baixa disponibilidade de dados reais em alguns domínios e ao alto custo de rotulagem. Neste artigo, focamos em seleção de meta-exemplos de treinamento combinando manipulação de dados e Transferência de aprendizagem via Classificação de uma classe. Assim, os exemplos mais relevantes são selecionados para serem rotulados. Nossos experimentos revelaram que é possível reduzir o custo computacional da geração de meta-exemplos e manter o desempenho em meta-aprendizagem.*

1. Introdução

Meta-aprendizado tem sido investigado nos últimos anos com o objetivo de aprender modelos de seleção de algoritmos a partir de experiências na resolução de problemas passados. Em meta-aprendizado, cada exemplo de treinamento (ou meta-exemplo) é associado a um problema de interesse (por exemplo, uma tarefa de classificação) e armazena: (1) as características do problema, ou seja os meta-atributos (por exemplo, número de exemplos, número de instâncias, correlações entre atributos); (2) um atributo-alvo, usualmente indicando o algoritmo avaliado com melhor desempenho empírico para o problema dentre um conjunto de algoritmos candidatos. A partir de um conjunto de meta-exemplos gerados a partir de problemas diversos, um meta-aprendiz é construído (por exemplo, um classificador para seleção do melhor algoritmo) relacionando características dos problemas com desempenho dos algoritmos candidatos.

A qualidade de um meta-aprendiz, assim como em qualquer outro contexto de aprendizagem de máquina, depende da quantidade e qualidade dos exemplos de treinamento. Em meta-aprendizado, cada meta-exemplo é construído a partir de um conjunto de

dados associados a tarefa de interesse. Conjuntos de dados diversos são coletados usualmente em repositórios (como o UCI) para assim gerar uma boa base de meta-exemplos. A dificuldade nesse contexto é que o número de conjuntos de dados disponíveis em repositórios é tipicamente limitado e assim o número de meta-exemplos disponíveis nem sempre é adequado para se aprender um bom meta-aprendiz.

Para minimizar a dificuldade acima, dados sintetizados podem ser utilizados, em combinação com dados reais, para aumentar o número de meta-exemplos. O uso de dados sintetizados no entanto pode gerar outra dificuldade. Apesar de gerar conjuntos de dados potencialmente relevantes e em boa quantidade, muitos dos conjuntos de dados sintetizados podem ser irrelevantes quando se considera as características observadas em problemas reais. Ou seja, a distribuição dos meta-atributos observada nos exemplos sintéticos pode ser bem diferente da distribuição de meta-atributos observada em problemas reais.

Diferenças de distribuições em conjuntos de exemplos é um problema tipicamente tratado na literatura com técnicas de transferência de aprendizagem [Yang and Pan 2010]. Nesse contexto, um classificador deve ser construído para realizar predições em um contexto-alvo onde se têm poucos dados de treinamento disponíveis. Por outro lado, dados de treinamento são disponíveis em grande quantidade em um contexto auxiliar. Técnicas de transferência de aprendizagem baseadas em seleção de dados podem ser usadas para selecionar exemplos de treinamento mais relevantes do contexto auxiliar para aumentar o número de exemplos para construir o classificador no contexto alvo. Isso é feito considerando a distribuição das características dos exemplos do contexto alvo e selecionando os exemplos do contexto auxiliar mais representativos para essa distribuição.

A seleção de meta-exemplos sintetizados é tratada no nosso trabalho como um problema de transferência de aprendizagem. Nessa abordagem, o contexto alvo se refere a problemas reais de classificação, enquanto que o contexto auxiliar se refere a conjuntos de dados sintetizados. Assim, o objetivo de transferência de aprendizagem será selecionar meta-exemplos gerados a partir de conjuntos de dados sintetizados para construir meta-aprendizes capazes de selecionar algoritmos para conjuntos de dados reais (ou seja, para o contexto alvo). Técnicas baseadas em seleção de dados serão usadas para selecionar conjuntos de dados sintetizadas com distribuição de meta-atributos similar a distribuição de meta-atributos em problemas reais. Essa estratégia é original na literatura de meta-aprendizado.

Para testar a viabilidade da proposta, realizamos um estudo de caso onde *PU learning* [Liu and Li 2002] foi usado para selecionar conjuntos de dados manipulados a partir de problemas reais. *PU learning* é uma estratégia de classificação de uma classe (classificação unária), adaptada em [Xia et al. 2013] para seleção de instâncias. Um meta-aprendiz foi construído para seleção dentre quatro algoritmos de classificação candidatos e avaliados em uma base de 144 conjuntos de dados reais e 1434 conjuntos de dados sintetizados. Foi observado que o método proposto é capaz de identificar os exemplos que são realmente úteis para o meta-aprendiz.

O restante do artigo é organizado como se segue. A seção 2 apresentará uma breve visão geral sobre meta-aprendizado aplicado à seleção de algoritmos, enquanto a seção 3 examinará o conceito de transferência de aprendizagem e suas aplicações, enfatizando o uso de classificação de uma classe para seleção de instâncias. A seção 4 apresenta o

trabalho desenvolvido na abordagem proposta e detalha um estudo de caso, incluindo o processo experimental realizado, algoritmos de aprendizagem utilizados, as métricas adotadas e as abordagens de aplicação e avaliação. Em seguida, a seção 5 apresentará e analisará os resultados experimentais obtidos. Finalmente, o artigo terminará na seção 6 com uma observação geral e uma nota sobre trabalhos futuros.

2. Meta-Aprendizagem Aplicada à Seleção de Algoritmos

Meta-aprendizagem busca descobrir como os algoritmos de aprendizado de máquina podem aumentar sua eficiência mediante a experiência. A Figura 1 ilustra o processo geral de meta-aprendizagem para seleção de algoritmos.

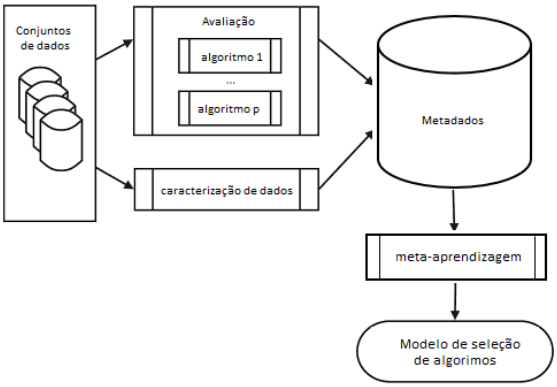

Figura 1. Processo de meta-aprendizagem para seleção de algoritmos. Adaptado de [Brazdil et al. 2009]

Em contexto geral, o processo de meta-aprendizagem para seleção de algoritmos inicia-se com a aquisição de um conjunto de problemas que represente, apropriadamente, os problemas para os quais a recomendação será feita. O próximo passo é, a partir de cada base de dados, avaliar o algoritmo empregado e extrair as características, de acordo com as métricas pré-estabelecidas. Obtendo-se essas duas informações (desempenho do algoritmo e características da base) de cada problema, gera-se um meta-exemplo, o qual é formado pelas meta-características de entrada e pela meta-classe[1], respectivamente. Ao conjunto dos meta-exemplos disponíveis dá-se o nome de meta-data. Com a finalidade de induzir o mapeamento entre as meta-características de entrada e meta-classes, é aplicado um algoritmo de aprendizado de máquina, o qual é chamado de meta-aprendiz. Por meio dele, é possível utilizar o meta-conhecimento obtido do processo de aprendizagem e realizar seleção de algoritmos no contexto de meta-aprendizagem.

Na literatura, diversos trabalhos têm investigado meta-aprendizagem de maneira mais profunda em tarefas de classificação e regressão e tem sido aplicada à recomendação de algoritmos em uma gama de domínios de aplicações, como seleção de modelos de séries temporais [Prudêncio and Ludermir 2004] e medidas de caracterização de comitês [Parente et al. 2013].

[1]Rótulo que representa o desempenho do algoritmo avaliado

Geração e seleção de exemplos, embora seja uma questão importante, não têm recebido muita atenção em trabalhos anteriores sobre meta-aprendizagem. O processo de meta-aprendizagem é fortemente dependente de instâncias de problemas relevantes e em número suficiente. Intuitivamente, buscar um repositório como UCI [Lichman 2013], por exemplo, pode ser a primeira alternativa a ser considerada. Embora intuitivo, dependendo da classe do problema, não é fácil encontrar um número suficiente de instâncias de problemas para geração de meta-dados significativos para obtenção de um modelo confiável. Na literatura, são encontradas abordagens para contornar a questão citada anteriormente como, por exemplo, geração de problemas sintéticos [Macia et al. 2008].

Neste trabalho, realizamos manipulação de dados para gerar novos conjuntos de dados, a chamada *Datasetoid*[Soares 2009]. Um datasetóide é gerado a partir de conjunto de dados original alterando o atributo alvo com um atributo independente. Dessa forma, o atributo alvo do conjunto de dados original se torna um atributo no datasetóide e o atributo independente selecionado se torna o atributo alvo. Na geração de dados para classificação, o processo é repetido para cada atributo simbólico do conjunto de dados original, criando assim tantos datasetóides quanto o número de atributos simbólicos existente no conjunto de dados. Detalhes de possíveis anomalias em datasetóides e alternativas de solução são encontrados em [Soares 2009]. A Figura 2 ilustra o processo de geração de datasetóides.

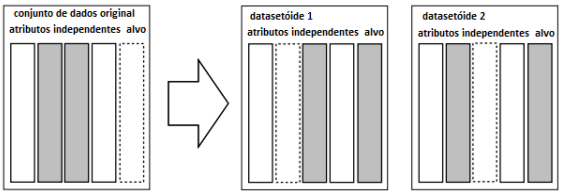

Figura 2. Geração de dois datasetóides de classificação a partir de uma base de dados com dois atributos simbolicos. [Soares 2009]

Neste artigo, realizamos transferência de aprendizagem combinando manipulação de dados e Classificação de uma classe. O conceito de transferência de aprendizagem será examinado a seguir.

3. Transferência de Aprendizagem

Em Aprendizagem de Máquina, comumente, assume-se que os dados de treinamento consistem em exemplos aleatoriamente extraídos da mesma distribuição da qual fazem parte os exemplos de teste sobre os quais o modelo treinado será utilizado para fazer classificação. Em muitas situações práticas, no entanto, esta suposição é violada, pois os modelos podem ser treinados em um ambiente e utilizados em outro ambiente com características diferentes, uma vez que os ambientes não são estacionários.

O processo de coleta de novos dados e reconstrução de modelos de aprendizagem é, muitas vezes, caro ou impossível de serem realizados [Yang and Pan 2010]. Uma alternativa para reduzir a necessidade e o esforço para recolhetar dados de treinamento é o uso de técnicas de Transferência de Aprendizagem (TA). Em TA, os domínios, tarefas e distribuições usadas em treinamento e teste podem ser diferentes. A motivação fundamental em Transferência de Aprendizagem são os métodos de aprendizagem de máquina que

guardem e reutilizem conhecimentos previamente aprendidos [Yang and Pan 2010]. Uma definição de Transferência de Aprendizagem pode ser vista a seguir.

Definição (Transferência de Aprendizagem [Yang and Pan 2010]) Dado um domínio $\mathcal{D}\mathrm{s}$ e uma tarefa de origem $\mathcal{T}\mathrm{s}$, um domínio alvo $\mathcal{D}\mathrm{t}$ e uma tarefa de aprendizagem alvo $\mathcal{T}\mathrm{t}$, transferência de aprendizagem visa melhorar a aprendizagem da função preditiva do alvo $f\mathrm{t}(\cdot)$ em $\mathcal{D}\mathrm{t}$ usando o conhecimento em $\mathcal{D}\mathrm{s}$ e $\mathcal{T}\mathrm{s}$ em que $\mathcal{D}\mathrm{s} \neq \mathcal{D}\mathrm{t}$ ou $\mathcal{T}\mathrm{s} \neq \mathcal{T}\mathrm{t}$. Sabendo que, na definição anterior, um domínio é um par $\mathcal{D}$ = $\{\mathrm{X},\mathrm{P(X)}\}$, a condição $\mathcal{D}\mathrm{s} \neq \mathcal{D}\mathrm{t}$ implica que $\mathrm{Xs} \neq \mathrm{Xt}$ ou $\mathrm{P(X)s} \neq \mathrm{P(X)t}$.

Segundo [Yang and Pan 2010], a Transferência de Aprendizagem pode se configurar quando as tarefas de origem e alvo são as mesmas, porém os domínios de origem e alvo são diferentes, ou seja, $\mathrm{P(X)s} \neq \mathrm{P(X)t}$. Nesse sentido, Transferência de Aprendizagem está relacionada à adaptação de domínios para transferência de conhecimento [Xia et al. 2013], viés de seleção de amostra [Zadrozny 2004] ou mudança de covariável [Shimodaira 2000]. Em nosso trabalho, a adaptação de domínio é realizada com base em seleção de instâncias.

3.1. Seleção de Instâncias

Na literatura, diversos métodos de seleção de instâncias são propostos [Olvera-López et al. 2010], como em [Xia et al. 2013] que é usada para adaptação entre domínios. Os métodos de seleçao de instâncias podem ser divididos em dois grupos, de acordo com a estratégia usada: (1) o critério de seleção é baseado na precisão obtida por um classificador. Somente instâncias que contribuem com a precisão da classificação são adicionadas ao conjunto de treinamento; (2) usa uma função de seleção que não é baseada em um classificador. A proposta de seleção de instâncias neste trabalho está inserida no primeiro grupo, como poderá ser visto mais adiante.

Em nossa abordagem, propomos Classificação de uma classe para seleção de meta-exemplos de treinamento.

3.2. Classificação de Uma Classe

O foco em Classificação de uma classe é definir um limite de classificação em torno da classe positiva, alvo, com a finalidade de aceitar o máximo possível de instâncias da classe positiva, e, ao mesmo tempo, minimizar a chance de aceitar instâncias não positivas.

A estratégia de Classificação de uma classe conhecida como *PU learning* (*Positive and Unlabeled Learning*) é muito aplicada no campo de classificação de texto [Nagy. et al. 2011]. Em [Liu and Li 2002] é utilizada uma técnica de classificação parcialmente supervisionada para identificar documentos que sejam de uma determinada classe. Pesquisadores têm usado diferentes termos para designar conceitos semelhantes ao de Classificação de uma classe [Khan and Madden 2010], como Detecção de *Outlier* ou Detecção de Novidades, por exemplo. Uma pesquisa mais aprofundada sobre Classificação de uma classe e aprendizagem de dados positivos e não rotulados pode ser encontrada em [Khan and Madden 2010].

No presente trabalho, aplicamos Classificação de uma classe para seleção de conjuntos de dados em contexto de meta-aprendizagem. A seguir, é descrito o trabalho desenvolvido na presente pesquisa.

4. Trabalho Desenvolvido

A geração de dados, sejam manipulados ou sintéticos, favorece o aumento do número de exemplos disponíveis para meta-aprendizagem, causando assim um potencial impacto positivo no desempenho no nível meta. Mesmo resultando em meta-exemplos relevantes, é possível porém que o processo de geração de bases de dados produza também meta-exemplos ruídos ou redundantes [Soares 2009]. Além do mais, como visto anteriormente, em meta-aprendizagem, o processo de rotulagem dos meta-exemplos é realizado via avaliação empírica dos algoritmos postulantes nos conjuntos de dados disponíveis, o que pode ser computacionalmente caro. Sendo assim, poder eliminar meta-exemplos irrelevantes e/ou redundantes antes do processo de rotulagem é uma boa estratégia para produzir meta-exemplos.

Diante da motivação acima, no trabalho atual, realizamos uma investigação da combinação de abordagens de manipulação de dados para geração de conjunto de dados [Soares 2009] e abordagens de Classificação de uma classe aplicadas à seleção de meta-exemplos. A Figura 3 apresenta a proposta desse trabalho.

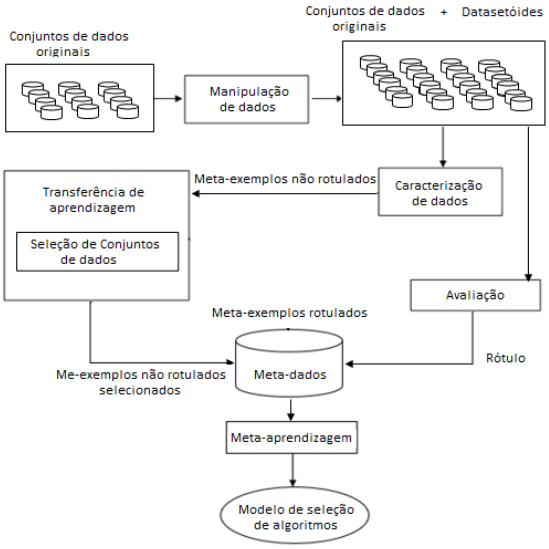

Figura 3. Uma descrição esquemática do processo experimental.

Inicialmente, uma coleção de conjuntos de dados é fornecida como entrada para um módulo de *Manipulação de dados*. Esse módulo usa então a abordagem de [Soares 2009] para geração de conjuntos de dados sintéticos (os datasetóides) a partir dos conjuntos de dados originais. Em seguida, todos os conjuntos de dados (originais + datasetóides) são fornecidos como entrada para o módulo *Caracterização de dados*, para que sejam calculadas as meta-características de cada conjunto de dados, dando origem a um grande conjunto de meta-exemplos não rotulados. No próximo passo, o módulo *Transferênica de aprendizagem* é acionado para seleção dos meta-exemplos que sejam mais relevantes para treinamento. Neste módulo, um processo de seleção de instâncias via classificação de uma classe é adotado para selecionar os dados mais relevantes para o treinamento. O conjunto de meta-exemplos selecionado é então rotulado (módulo *Avaliação*),

formando assim a meta-base de exemplos (*Meta-dados*). Finalmente, a meta-base serve de entrada para o módulo *Meta-aprendizagem*, que tem como saída um modelo de seleção de algoritmos.

A seguir, são apresentados os detalhes de implementação de cada módulo em um estudo de caso realizado para testar a viabilidade da proposta. Nesse estudo de caso, o problema de seleção de algoritmos consiste na seleção de quatro algoritmos candidatos a tarefas de classificação.

4.1. *Manipulação de Dados*

Neste trabalho, que tem ênfase na seleção de algoritmos para tarefas de classificação, adotamos um método para manipulação de conjuntos de dados chamado datasetóides [Soares 2009], apresentado anteriormente (ilustrado na Figura 2 na seção 2), com a finalidade de obter um grande número de dados de treinamento. Essa técnica realiza manipulação de conjuntos de dados existentes formando novas bases, aumentando a coleção de conjuntos de dados e colaborando com a qualidade da predição dos modelos de meta-aprendizado. Uma nova base de dados (datasetóide) é gerada a partir de um conjunto original através da alteração do atributo alvo com um atributo simbólico. Os conjuntos originais de problemas que usamos nesta pesquisa são conjuntos de classificação obtidos do repositório UCI [Lichman 2013] e do repositório KEEL [Alcalá. et al. 2010]. A partir de 144 problemas originais coletados nesses repositórios, foram produzidos 1434 novos conjuntos de dados manipulados. O processo de caracterização dos dados é apresentado a seguir.

4.2. *Caracterização de Dados*

Nesse módulo, as bases de dados originais e sintetizadas são descritas por um conjunto de 14 medidas de caracterização calculadas usando a *Data Complexity library*[2]. Tais medidas são descritas na Tabela 1. A caracterização resultará então em um meta-exemplo não-rotulado para cada conjunto de dados considerado no estudo de caso (original ou datasetóide).

4.3. *Transferência de Aprendizagem*

Em nosso trabalho, a transferência de aprendizagem é efetivada via *PU learning*. Segundo [Liu and Li 2002] *PU Learning* recebe como entrada um conjunto $\mathcal{P}$ de exemplos da classe positiva (classe de interesse) e um conjunto $\mathcal{U}$ de exemplos não rotulados (ou seja, de classe desconhecida). O conjunto $\mathcal{U}$ pode conter tanto exemplos pertencentes a classe positiva como exemplos de outras classes. Um classificador é construído para selecionar as instâncias desconhecidas com maior probabilidade de pertencerem a classe positiva. No contexto de transferência de aprendizagem, $\mathcal{P}$ contém exemplos pertencentes a um domínio-alvo, enquanto $\mathcal{U}$ contém exemplos de um outro domínio relacionado. *PU learning* é usado então para selecionar instâncias de $\mathcal{U}$ representativos para o domínio-alvo.

Em nossa proposta, os dados positivos (classe positiva) representam os conjuntos de dados originais $\mathcal{P}$ e os dados não rotulados são os datasetóides $\mathcal{U}$. Classificação de uma classe usando dados positivos e não rotolaos é então usada para classificar os datasetóides

[2]Disponível em (DCol-http://dcol.sourceforge.net/)

Tabela 1. Medidas de Caracterização

Medidas	Descrição
F1	Razão discriminante máxima de Fisher
F1v	Razão discriminante máxima de Fisher vetor direcional
F2	Sobreposição das distribuições dos valores das características dentro das classes
F3	Máxima eficiência de características individuais
F4	Eficiência coletiva da características (soma da eficiência de cada característica)
L1	Soma minimizada da distância do erro de um classificador linear (SMO linear)
L2	Erro de treinamento de um classificador linear (SMO linear)
L3	Não linearidade de um classificador linear (SMO linear)
N1	Fração de pontos no limite da classe
N2	Razão da média intra/inter classe da distancia do vizinho mais próximo
N3	Taxa de erro de leave-one-out do classificador de um vizinho mais próximo
N4	Não linearidade do classificador de um vizinho mais próximo
T1	Fração de esferas de cobertura máxima
T2	Número médio de pontos por dimensão

mais representativos em relação aos dados originais. O algoritmo de seleção de instâncias implementado em nossa proposta é dividido em dois passos. A Figura 4 ilustra o primeiro passo do método proposto.

Nr = NULL;
Ur = NULL
$\mathcal{S}$=amostrar($\mathcal{P}$,10%);
Atribuir a cada exemplo em $\mathcal{P} - \mathcal{S}$ o rótulo de classe c=1;
Atribuir a cada exemplo em $\mathcal{U} \cup \mathcal{S}$ o rótulo de classe c=0;
Construir um classificador NB g usando $\mathcal{P} - \mathcal{S}$ e $\mathcal{U} \cup \mathcal{S}$;
Classificar cada exemplo em $U \cup S$ usando g;
Determine b;
para cada exemplo $e \in \mathcal{U}$
    se a probabilidade Pr(1|e) < b então
        Nr = Nr $\cup \{e\}$;
Ur = $\mathcal{U}$ −Nr;

Figura 4. Pseudo-código do primeiro passo do algoritmo.

No primeiro passo, foi aplicada a técnica conhecida como espiões, em que um pequeno conjunto $\mathcal{S}$ de exemplos da classe positiva $\mathcal{P}$ é misturado aos dados não rotulados $\mathcal{U}$. Os exemplos espiões se comportam de forma semelhante aos exemplos desconhecidos da classe positiva em $\mathcal{U}$. Assim, eles permitem que o algoritmo possa inferir o comportamento destes exemplos em $\mathcal{U}$. Utilizamos 10% dos dados originais de treinamento $\mathcal{P}$, selecionados aleatoriamnte, como espiões. Os conjuntos de dados $\mathcal{P} - \mathcal{S}$ e $\mathcal{U} \cup \mathcal{S}$ são rotulados temporariamente como positivo (c=1) e negativo(c=0), respectivamente. Um

classificador *Naive Bayes* (NB) é aplicado para prever cada exemplo $e \in \mathcal{U} \cup \mathcal{S}$. A probabilidade de cada exemplo em $\mathcal{S}$ pertencer a classe c=1 é usada para determinar b. Primeiro, ordenamos as probabilidades em $\mathcal{S}$. Em seguida, selecionamos b, tal que x% dos exemplos em $\mathcal{U}$ tenham probabilidade de pertencer a classe c=1 menor que b. Experimentamos alguns valores de x e encontrados x=60 como melhor nível de ruído para os dados. Mais detalhes podem ser vistos em [Liu and Li 2002]. Finalmente, usando b como limiar, os exemplos em $\mathcal{U}$ que são considerados verdadeiros negativos são separados em (Nr) e os demais em (Ur). Tanto Nr quanto Ur são entradas para o segundo passo do método proposto nesse trabalho. A ilustração do segundo passo do método pode ser vista na Figura 5.

```
Up = NULL;
enquanto verdade
  use P e Nr para treinar um classificador SVM;
  classifique Ur usando um modelo SVM;
  seja Vn o conjunto de exemlos que um modelo atual classificou como negativo;
  seja Vp o conjunto de exemplos que um modelo atual classificou como positivo;
       se Vn = φ então
            Up = Up ∪ Vp;
            saia;
       senão
            Ur = Ur −Vn  − Vp;
            Nr = Nr ∪ Vn;
            P = P ∪ Vp;
            Up = Up ∪ Vp;
```

Figura 5. Pseudo-código do segundo passo do algoritmo.

No segundo passo, uma máquina de vetor suporte (*Support vector machine* - SVM) iterativa proposta em [Nagy. et al. 2011] é construída. Em cada iteração, $\mathcal{P}$ e Nr são usados na construção do classificador. O SVM é aplicado para classificar os exemplos em Ur. Os exemplos classificados como positivos são adicionados a $\mathcal{P}$ e os negativos a Nr. A ideia principal do SVM é selecionar o hiperplano que separe o espaço (entre as classes positivas e negativas) enquanto maximiza a menor margem. O objetivo por trás dessa abordagem é que o SVM consiga extrair o maior número possível de exemplos relevantes para o conjunto de treinamento. Se o modelo atual não pode identificar mais nenhum exemplo como irrelevante para ser removido do conjunto de treinamento, então o algoritmo é encerrado. Após as iterações, o classificador final é aplicado. Ao término do algoritmo, o conjunto Up contém os exemplos selecionados como mais relevantes para dados de treinamento. Estes exemplos fazem parte da entrada do módulo *Avaliação*.

4.4. *Avaliação*

Nesta investigação, no nível base, escolhemos métodos de classificação com diferentes características. Neste sentido, foram escolhidos quatro métodos de classificação: *k-nearest neighbors* (KNN), *JRip* (JR), *Naive Bayes* (NB) and *Decision tree* (J48). Todos eles são métodos de classificação bem conhecidos [I.H.Witten and Frank 2005]. Cada método de classificação é candidato a rótulo de classe para os meta-exemplos selecionados. O problema de seleção de algoritmo neste trabalho consiste em prever a priori qual o melhor método classificador dado um determinado problema. A medida de desempenho utilizada no nível base foi a precisão da classificação, estimada usando validação cruzada de 10 *folds*. O algoritmo com melhor acurácia é usado como rótulo de classe. A tabela

2 apresenta a distribuição de classes para os metadados, tanto dados originais quanto datasetóides. Os rótulos de classe C1, C2, C3 e C4 representam, respectivamente os algoritmos JR, J48, NB e KNN.

Tabela 2. Distribuição de Classe (%) dos metadados

meta-data	C1	C2	C3	C4
dados originais	18.06%	28.47%	23.61%	29.86%
datasetóides	41.49%	29.15%	15.27%	14.09%

4.5. *Meta-Aprendizagem*

O resultado da etapa anterior é um conjunto de meta-exemplos rotulados. Cada meta-exemplo é associado a um conjunto de dados e armazena as características descritivas e o valor de classe indicando o melhor algoritmo candidato para o conjunto de dados em questão. No trabalho atual, adotamos como meta-aprendiz dois algoritmos de classificação: *Random Forest* (RF) e (KNN). Esses algoritmos são utilizados como meta-classificadores que indicarão o melhor modelo para um determinado conjunto de dados de teste. Em nosso protótipo usamos as implementações disponíveis no ambiente WEKA [I.H.Witten and Frank 2005].

5. Experimentos

Nesta seção, são apresentados os experimentos para avaliar a qualidade dos meta-aprendizes treinados com dados selecionados usando *PU learning*.

5.1. Metodologia de Avaliação

Em nossa abordagem, escolhemos uma meta-aprendizagem simples aplicada ao problema de seleção de algoritmos, como dito anteriormente.

No nível meta, adotamos a metodologia de avaliação de *holdout* com 500 repetições. Em cada iteração, o conjunto de dados original é particionado, aleatoriamente, em dois conjuntos de dados mutuamente exclusivos (teste e treinamento), sendo 10% dos exemplos destinados ao conjunto de teste. Como conjunto de treinamento utilizamos os 90% dos conjuntos originais restantes e todos os datasetóides gerados a partir desses conjuntos. Com isso, nos certificamos de que os datasetóides gerados a partir dos conjuntos de dados de teste não estivessem incluídos no conjunto de treinamento.

Em nossa proposta, a partir dos conjuntos de dados selecionados via *PU learning*, o meta-aprendiz é construído e avaliado para seleção de algoritmos para os conjuntos de teste. O método proposto é comparado com outras duas abordagens: (1) usando como dados de treinamento somente os 90% dos conjuntos originais restantes (DO); (2) os 90% dos conjuntos originais juntamente com todos os datasetóides gerados a partir desses conjuntos sem seleção (DO+datasetóides). A medida de desempenho da meta-aprendizagem é a acurácia média das 500 execuções.

5.2. Resultados

Neste trabalho, os resultados são apresentados de acordo com três perspectivas sobre os dados de treinamento, que são: apenas conjuntos de dados originais (DO); conjuntos originais juntamente com todos os seus respectivos datasetóides (DO+datasetóides)

e dados originais acrescidos da seleção de datasetóides realizada pelo método proposto (DO+método proposto). A tabela 3 apresenta os resultados observados em nossos experimentos.

Tabela 3. Taxa de acurácia média e desvio padrão dos meta-aprendizes investigados e o número de exemplos nos conjuntos de treinamento

algoritmos/dados de treinamento	**DO**	**DO+datasetóides**	**DO+método proposto**
RF	38.23(12.10)	41.87(12.46)	**42.41(12.45)**
kNN	35.07(12.17)	39.11(11.61)	**39.91(12.87)**
Exemplos de Treinamento	129	1445.40	**266.22**

Como visto anteriormente, o método de seleção de meta-exemplos proposto é dividido em dois passos. No primeiro passo, uma média de 444.34 meta-exemplos foram selecionados, como candidatos a exemplos relevantes. O restante foi considerado verdadeiros negativos, ou seja, irrelevantes para o domínio alvo. Os dois conjuntos de meta-exemplos, verdadeiros negativos e candidatos a relevantes, são oferecidos como entrada para o segundo passo do método proposto. O segundo passo, por sua vez, seleciona, desses 444.34, 137.22 meta-exemplos para o conjunto de treinamento. Sendo assim, 69.01% são considerados verdadeiros negativos pelo primeiro passo do método proposto. Após a seleção pelo segundo passo, um SVM iterativo, apenas 9.57% são considerados realmente relevantes para o treinamento. O conjunto de treinamento no método proposto é formado pelos 137.22 datasetóides selecionados adicionados aos 129 conjuntos de dados originais, resultando em um conjunto de treinamento com uma média de 266.22 meta-exemplos.

Podemos concluir que o método proposto para seleção de instâncias é capaz de identificar os exemplos que são realmente úteis para o meta-aprendizado, sendo capaz de atingir um nível de precisão até mesmo maior (42.41%) que o obtido por meta-aprendizagem com todos os conjuntos de dados de treinamento (datasetóides) (41.87%), usando, em média, apenas 266.22 dos meta-exemplos.

6. Conclusão

Propusemos a combinação de Classificação de uma classe[Khan and Madden 2010] usando estratégia de *PU learning* [Liu and Li 2002],[Nagy. et al. 2011] para realizar uma seleção de instâncias em dados de treinamento, datasetóides [Soares 2009] em meta-aprendizagem. Duas questões importantes de meta-aprendizagem para seleção de algoritmos foram abordadas: (1) aumento do número de conjuntos de dados para produção de meta-exemplos e (2) redução do custo computacional da coleta de metadados.

Os resultados obtidos mostram que é viável fazer uso dos benefícios do grande número de meta-exemplos fornecidos por datasetóides [Soares 2009] sem que haja custos computacionais adicionais significativos e, além disso, alcançar melhorias no desempenho, devido à eliminação de meta-exemplos irrelevantes.

A presente abordagem ainda poderá ser avaliada em outros problemas de meta-aprendizagem, como regressão por exemplo, com outros grupos de algoritmos de nível base, bem como outras meta-características.

Referências

Alcalá., J., Fernández., A., Luengo, J., Derrac., J., García., S., Sánchez., L., and Herrera., F. (2010). Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic and Soft Computing 17(2-3). 255-287*.

Brazdil, P., Giraud-Carrier, C., Soares, C., and Vilalta, R. (2009). *(Eds.), Metalearning - Applications to Data Mining*. Springer, Berlin, Heidelberg.

I.H.Witten and Frank, E. (2005). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

Khan, S. and Madden, M. (2010). A survey of recent trends in one class classification. In Coyle, L. and Freyne, J., editors, *(eds) Artificial Intelligence and Cognitive Science.AICS 2009. Lecture Notes in Computer Science, vol 6206. Springer, Berlin, Heidelberg*, pages 188–197. Publishing Press.

Lichman, M. (2013). Uci machine learning repository. url http://archive.ics.uci. edu/ml.

Liu, B.and Lee, W. S. Y. P. and Li, X. (2002). Partially supervised classification of text documents. *ICML-02*.

Macia, N., Orriols-Puig, A., and Bernadó-Mansilla, E. (2008). Genetic-based synthetic data sets for the analysis of classifiers behavior. *Proceedings of 15th International Conference on Hybrid Intelligent Systems, pp. 507-512*.

Nagy., T., Farkas., R., and Csirik., J. (2011). On positive and unlabeled learning for text classification. *Habernal I., Matoušek V. (eds) Text, Speech and Dialogue. TSD 2011. Lecture Notes in Computer Science, vol 6836. Springer, Berlin, Heidelberg*.

Olvera-López, J., Carrasco-Ochoa, J., and Martínez-Trinidad, J. (2010). A review of instance selection methods. *Artif Intell Rev (2010) 34: 133. https://doi.org/10.1007/s10462-010-9165-y*.

Parente, R. R., Canuto, A. M. P., and Xavier-Junior, J. C. (2013). Characterization measures of ensemble systems using a meta-learning approach. *Proceedings of International Joint Conference on Neural Networks, Dallas, Texas, USA, August 4-9*.

Prudêncio, R. and Ludermir, T. (2004). Meta-learning approaches to selecting time series models. *Neurocomputing 61, 121–137*.

Shimodaira, H. (2000). Improving predictive inference under covariate shift by weighting the log-likelihood function. *J. Statistical Planning and Inference, vol. 90, pp. 227-244*.

Soares, C. (2009). Uci++, improved support for algorithm selection using datasetoids. pages 499–506. Lecture Notes in Computer Science 5476.

Xia, R., Hu, X., Lu, J., Yang, J., and Zong, C. (2013). Instance selection and instance weighting for cross-domain sentiment classification via pu learning. *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*.

Yang, Q. and Pan, S. J. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering. 1345-1359.*

Zadrozny, B. (2004). Learning and evaluating classifiers under sample selection bias. *Proc. 21st Int'l Conf. Machine Learning, July*.