Felipe Nunes Walmsley

Um Método de Geração de Pools de Classificadores Usando Instance Hardness

Proposta de Trabalho de Gradução

Universidade Federal de Pernambuco – UFPE

Centro de Informática

Graduação em Engenharia da Computação

Orientador: George Darmiton da Cunha Cavalcanti

Recife

Abril, 2017

Resumo

Em Aprendizagem de Máquina, métodos de *ensemble* vêm recebendo grande atenção. Técnicas como *bagging* e *boosting* têm sido aplicadas com sucesso a diversos problemas. Entretanto, essas técnicas ainda são suscetíveis aos efeitos de ruído e de *outliers*, que podem estar presentes no conjunto de treinamento. Tanto *outliers* quanto instâncias ruidosas serão provavelmente classificadas erroneamente. Um indicativo da dificuldade em classificar corretamente uma instância é a chamada *instance hardness*. Nesse trabalho, propõe-se um novo método de geração de pools de classificadores baseado em *bagging*, em que a probabilidade de seleção durante o processo de reamostragem é inversamente proporcional à dificuldade da instância. O objetivo do método é remover instâncias ruidosas e *outliers* e evitar assim seus efeitos indesejáveis sobre o treinamento.

Palavras-chaves: ensemble methods. bagging. dados ruidosos. instance hardness

Abstract

In Machine Learning, ensemble methods have been receiving a great deal of attention. Techniques such as bagging and boosting have been successfully applied to a variety of problems. Nevertheless, such techniques are still susceptible to the effects of noise and outliers, which might be present in the training data. Both outliers and noisy instances are intrinsically likely to be misclassified, regardless of the choice of classifier. We propose a new method for the generation of pools of classifiers, based on bagging, in which the probability of an instance being selected during the resampling process is inversely proportional to its instance hardness. The goal of the proposed method is to remove noisy data without sacrificing the hard instances which are likely to be found on class boundaries.

Key-words: ensemble methods. bagging. noisy data. instance hardness.

Introdução

Em Aprendizagem de Máquina, técnicas de *ensemble learning* (1) são técnicas que combinam múltiplos preditores treinados independentemente no mesmo conjunto de treinamento, utilizando uma combinação das saídas de cada preditor como saída do sistema. Embora seja comum em Aprendizagem de Máquina utilizar um único modelo treinado em todo o conjunto de treinamento, espera-se que ao utilizar técnicas de *ensemble* seja possível obter um *pool* com competências complementares. Assim, os *pools* gerados podem obter ganhos de desempenhos superiores sobre estratégias que utilizam um único classificador, dado que encontrar o modelo ótimo para um problema pode ser extremamente difícil.

Um método de *ensemble learning* que tem sido amplamente aplicado é *bootstrap aggregating* (2), ou simplesmente *bagging*. A técnica consiste em gerar um *ensemble* de N classificadores treinados a partir de N conjuntos de treinamento, conjuntos esses gerados a partir de um conjunto de treinamento original. Os conjuntos são gerados escolhendo aleatoriamente com reposição exemplos do conjunto original, de acordo com uma distribuição uniforme.

O uso de *bagging* é interessante no caso de conjuntos de dados pequenos, ruidosos, ou ambos (3). Em termos gerais, espera-se que com o uso de *bagging* os classificadores treinados possuam competências complementares, de forma que a decisão do conjunto como um todo seja melhor do que a de um único classificador treinado em todo o conjunto de treinamento (2).

Ainda dentro do domínio das técnicas de *ensemble learning*, outra técnica de interesse é o *boosting* (4). *Boosting* é uma técnica em que um conjunto de classificadores é treinado iterativamente, adicionando ao conjunto existente o classificador que minimiza o erro do conjunto como um todo . Dessa forma, a minimização realizada pelo algoritmo de treinamento tende a levar cada novo classificador adicionado a se especializar nos exemplos em que os classificadores anteriores erram.

Embora técnicas de *ensemble learning* possam oferecer ganhos de performance, elas não evitam completamente dois problemas comuns em aprendizagem de máquina: *outliers* e ruído. *outliers* são exemplos que diferem muito de outros exemplos pertencentes à mesma classe. Como *outliers* não se assemelham aos demais membros de sua classe, pode ser difícil para um classificador reconhecê-lo como pertencendo à classe.

Em contrapartida, entende-se por ruído o dado que teve seu valor modificado do valor verdadeiro por algum processo. Os processos mais comuns pelos quais o valor de um dado pode ser modificado são erros de medição ou devido à incerteza inerente na medição.

Embora os conceitos de ruído e *outlier* possam parecer semelhantes, nesse trabalho denomina-se *outliers* as instâncias que, apesar de diferentes de outras instâncias da mesma classe, ainda assim pertencem à classe, e que de fato correspondem à distribuição dos dados. Ruído, por outro lado, é aquilo que não pertence de fato à distribuição.

Observa-se então que, na presença de *outliers*, ruído, ou ambos, é possível que o processo de treinamento do classificador torne-se instável ou sofra de sobreajuste (5), independentemente do uso de técnicas de ensemble. Essa preocupação torna-se maior no caso de algoritmos que dão maior peso a instâncias classificadas erroneamente, como o *boosting*, pois é possível que o modelo seja fortemente ajustado de forma a classificar corretamente uma instância que não é representativa da distribuição subjacente dos dados. Nesse cenário, observa-se uma queda na acurácia de generalização do modelo.

Não obstante, há técnicas que buscam remover o ruído de conjuntos de dados, a fim de amenizar os problemas mencionados anteriormente. Um exemplo dessa técnica é a *Edited Nearest Neighbor Rule* (6), que filtra o conjunto de treinamento, removendo as instâncias que não são corretamente classificadas utilizando um classificador do tipo *k-Nearest Neighbors*. Entretanto, técnicas de remoção de ruído podem gerar efeitos indesejáveis sobre o conjunto de treinamento, como a remoção de exemplos que não são ruídos ou remoção exagerada de exemplos na fronteira entre duas classes.

Relacionado ao conceito de ruído há o conceito de *instance hardness*, ou a dificuldade em classificar uma instância. *Instance hardness* pode ser entendida como a probabilidade que uma instância receberá a classificação errada (7). A dificuldade em classificar uma instância pode ser utilizada como um indicador da probabilidade que aquela instância seja um *outlier* ou ruído. Dessa forma, *instance hardness* pode ser utilizado como uma base interessante para ferramentas que buscam remover seletivamente instâncias do conjunto de treinamento.

Uma vez munidos dos conceitos de *outliers*, ruído e *instance harndess*, é natural questionar se não seria possível utilizar alguma medida de dificuldade da instância para remover do conjunto de treinamento as instâncias problemáticas citadas anteriormente. Espera-se que uma vez removidas essas instâncias, o processo de treinamento torne-se mais estável, levando a uma generalização.

Nesse espírito, propõe-se neste trabalho uma técnica baseada em *bagging* que busca remover *outliers* e instâncias ruidosas do treinamento, evitando porém remover instâncias que não sejam realmente ruído ou que sejam importantes para que os classificadores aprendam a separação entre classes.

A ideia central do método é modificar o processo de geração dos conjuntos do *bagging*. Em vez de escolher as amostras de forma equiprovável, a probabilidade de uma amostra ser escolhida para integrar um dos conjuntos passa a ser inversamente proporcional

à sua dificuldade.

Para exemplificar os conceitos discutidos até agora, a Figura 2 mostra uma distribuição ruidosa. Os exemplos em azul são da classe negativa, enquanto os em vermelho são da classe positiva. Os exemplos em verde, por sua vez, são exemplos da classe negativa, que no entanto distribuem-se de uma forma diferente do resto dos exemplos dessa classe, exemplificando o conceito de outlier.

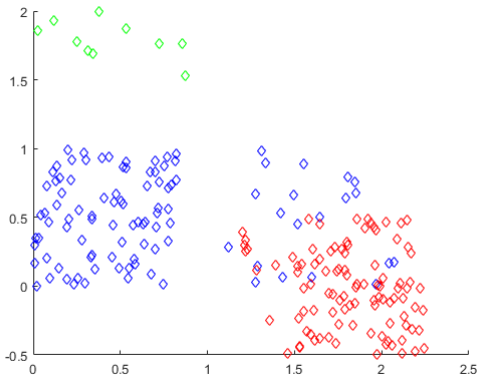

Figura 1 – Distribuição com ruído. Os exemplos em azul pertencem à classe negativa e os vermelhos à classe positiva. Os exemplos em verde são *outliers* da classe negativa.

Os exemplos seguem uma distribuição aleatória uniforme. Os exemplos negativos variam no intervalo (0;1) no eixo x e no intervalo (0;1) no eixo y. Já a classe positiva distribui-se no intervalo (1,5;2,5) no eixo x e no intervalo (-0,5; 0,5) no eixo y. Adicionou-se ruído a 20Já a Figura 2 demonstra o efeito que teria uma remoção ideal do ruído.

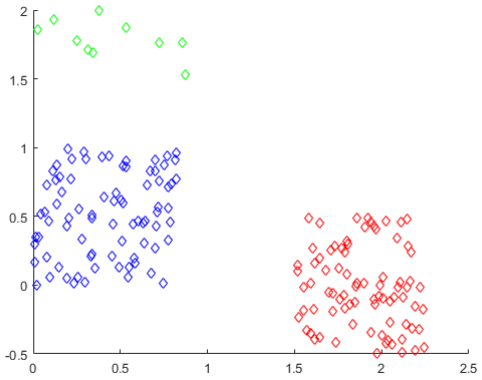

Figura 2 – Remoção ideal do ruído.

Com a remoção do ruído, fica claro que o problema é linearmente separável, e pode ser facilmente aprendido.

O método proposto foi concebido para obter resultados que aproximem-se do objetivo mencionado anteriormente: gerar conjuntos de treinamento relativamente livres de ruído. Por outro lado, dado que a seleção de instâncias é realizada de forma probabilística, espera-se que mesmo as instâncias difíceis sejam selecionadas, embora em menor frequência. É desejável que instâncias difíceis sejam selecionadas, a fim de evitar problemas como a remoção de instâncias na fronteira entre classes, o que poderia causar perda de informação.

Objetivo

O objetivo deste trabalho é propor um método de geração de *pools* de classificadores baseado em *bagging*, que utilize *instance hardness* para definir as probabilidades de seleção das instâncias. Espera-se que o método proposto seja particularmente apropriado para bases de dados pequenas ou ruidosas. Assim, esses tipos de base serão o foco da avaliação do método, e serão usados nas comparações entre o método proposto e as técnicas presentes na literatura.

Metodologia

Como mencionando anteriormente, será gerado um programa que implemente o método proposto. Esse programa será então utilizado para realizar experimentos comparando o desempenho do método proposto com os outros métodos citados. As variáveis relevantes para os experimentos são:

1. **Base de dados:** Serão utilizadas bases de dados artificiais que facilitem a visualização do problema, além de bases de dados publicamente disponíveis e frequentemente utilizadas na literatura.

2. **Ruído:** O efeito da presença de ruído nos exemplos de treinamento será avaliado, levando em conta diferentes níveis de ruído, e analisado tomando como referência o caso em que não há ruído.

3. **Técnicas de geração do ensemble:** Serão utilizadas como técnicas de geração de ensemble: bagging, boosting e o método proposto. Ademais, o desempenho de um único classificador do tipo do classificador base do ensemble será avaliado.

4. **Tamanho do ensemble:** Diferentes tamanhos de ensemble serão avaliados.

5. **Filtragem de ruído:** Será utilizada a regra ENN para filtragem de ruído, e seu efeito será analisado tomando como referência o caso em que não houve filtragem de ruído.

6. **Métrica e metodologia de comparação:** Será tomada como métrica de desempenho a acurácia global do ensemble, e o Wilcoxon Signed-Rank Test será usado para comparar o desempenho das diferentes técnicas.

Cronograma

Tabela 1 – Cronograma de Atividades

	Período																			
Atividade	Março		Abril				Maio				Junho				Julho					
Revisão bibliográfica																				
Implementação																				
Experimentos																				
Avaliação dos resultados																				
Escrita do TG																				
Preparação da apresentação																				

Referências

1 ZHOU, Z.-H. *Ensemble methods: foundations and algorithms*. [S.l.]: CRC press, 2012. Citado na página 4.

2 BREIMAN, L. Bagging predictors. *Machine Learning*, Springer, v. 24, n. 2, p. 123–140, 1996. Citado na página 4.

3 KHOSHGOFTAAR, T. M.; HULSE, J. V.; NAPOLITANO, A. Comparing boosting and bagging techniques with noisy and imbalanced data. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, IEEE, v. 41, n. 3, p. 552–568, 2011. Citado na página 4.

4 FREUND, Y.; SCHAPIRE, R.; ABE, N. A short introduction to boosting. *Journal of the Japanese Society For Artificial Intelligence*, Japanese Society for Artificial Intelligence, v. 14, n. 771-780, p. 1612, 1999. Citado na página 4.

5 LONG, P. M.; SERVEDIO, R. A. Random classification noise defeats all convex potential boosters. *Machine Learning*, Springer, v. 78, n. 3, p. 287–304, 2010. Citado na página 5.

6 WILSON, D. L. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetics*, IEEE, v. 2, n. 3, p. 408–421, 1972. Citado na página 5.

7 SMITH, M. R.; MARTINEZ, T.; GIRAUD-CARRIER, C. An instance level analysis of data complexity. *Machine learning*, Springer, v. 95, n. 2, p. 225–256, 2014. Citado na página 5.

Possíveis Avaliadores

- Prof. Paulo Salgado Gomes de Mattos Neto - Centro de Informática, UFPE

Assinaturas

Recife, 10 de Abril de 2017

Felipe Nunes Walmsley
Aluno

George Darmiton da Cunha Cavalcanti
Orientador