Estudo comparativo entre algoritmos de árvores de classificação e máquinas de vetores suporte, baseados em *ensembles* de classificadores

Melina Brilhadori, Marcelo S. Lauretto

Escola de Artes, Ciências e Humanidades – Universidade de São Paulo (USP)
Av. Arlindo Béttio, Ermelino Matarazzo – 03828-000 – São Paulo – SP – Brasil

melinabrilhadori@gmail.com, marcelolauretto@usp.br

Abstract. *This article presents a comparative analysis of the performance of one algorithm of the classification tree family (J48) and one of the support vector machines (SMO), when combined as ensembles bagging and boosting. The two main questions are: a) For a certain algorithm, which ensemble setting (between Bagging and Boosting) achieves higher accuracy? b) Is there any evidence that a particular classifier performs consistently better than the other under the ensemble setting? Results suggest that J48 tends to achieve a higher accuracy under the Boosting configuration, while SMO seems less sensitive to the ensemble adopted. Nonetheless, both algorithms attained similar numbers of victories among the used datasets.*

Resumo. *Este artigo apresenta uma análise comparativa de desempenho de um algoritmo de árvores de classificação (J48) e um algoritmo de máquinas de vetores suporte (SMO) quando combinados na forma de ensembles bagging e boosting. As duas questões principais são: a) Para um certo algoritmo, que configuração entre Bagging e Boosting resulta em maior acurácia? b) Há evidência de que um algoritmo seja consistentemente superior ao outro sob as configurações ensemble? Os resultados sugerem que o J48 tende a obter maiores acurácias sob a configuração Boosting, enquanto o SMO parece menos sensível à configuração ensemble utilizada. Não obstante, ambos os algoritmos obtiveram números de vitórias similares entre os datasets.*

1. Introdução

Diante de problemas complexos, são comuns as tomadas de decisões através de comitês, em que os membros apresentam suas opiniões – com base em suas experiências e conhecimentos sobre o domínio – e assim contribuem conjuntamente em sua resolução.

O mesmo princípio, tão usualmente aplicado no cotidiano, vem sendo amplamente estudado e aplicado para problemas de aprendizado supervisionado. Segundo este princípio, pode-se obter um resultado para um problema de classificação através de comitês de classificadores, denominadas usualmente metaclassificadores ou *ensembles* [Nascimento, 2009]. *Ensemble* é um paradigma de aprendizado em que um grupo finito de propostas alternativas é utilizado para a solução de um dado problema.

Segundo Coelho (2006), "*o uso da abordagem ensembles tem sido bastante explorado na última década, por se tratar de uma técnica simples e capaz de aumentar*

a capacidade de generalização de soluções baseadas em aprendizado de máquina. No entanto, para que um ensemble seja capaz de promover melhorias de desempenho, os seus componentes devem apresentar bons desempenhos individuais e, ao mesmo tempo, devem ter comportamentos diversos entre si."

Neste trabalho apresentamos uma análise de desempenho de dois algoritmos de classificação baseados em *ensembles* de árvores de classificação e de máquinas de suporte vetorial (Support Vector Machines, SVM). O foco central é comparar os desempenhos de *ensembles* desses algoritmos combinados por meio dos métodos *bagging* e *boosting*. As principais questões a serem respondidas são:

a) Para um certo algoritmo, que configuração de *ensemble* (entre Bagging e Boosting) resulta em maior acurácia?

b) Há evidência de que um algoritmo seja consistentemente superior ao outro sob as configurações *ensemble*?

Este artigo está organizado da seguinte forma. Na Seção 2 são apresentados os algoritmos classificadores e *ensembles*, bem como as metodologias de teste e de análise adotadas. Na Seção 3 são apresentados e discutidos os resultados, e na Seção 4 são apresentadas as conclusões.

2. Metodologia Experimental

Resumidamente, o estudo compreendeu quatro experimentos envolvendo a combinação entre dois classificadores e dois métodos de *ensembles*, realizados sobre 21 conjuntos de dados públicos, conforme descrito a seguir.

Os algoritmos utilizados foram o J48 e SMO como classificadores base, e o *Bagging* e o *AdaBoost.M1* como construtores de *ensemble*, todos implementados dentro do ambiente Weka [Hall et al, 2009; Witten & Frank, 2005]. Os scripts de teste foram implementados no ambiente R [The R Project, 2011].

Esses algoritmos foram escolhidos dentre um conjunto inicial de algoritmos candidatos por sua popularidade, cada qual dentro de sua família, a facilidade de adaptação nos experimentos e sua integração com a ferramenta utilizada para desenvolvimento [R Core Team, 2013], além de estarem disponíveis no ambiente Weka.

2.1. Algoritmos de Classificação J48 e SMO

O J48 é um indutor *top-down* de árvores de classificação que reimplementa na suíte Weka o algoritmo C4.5, proposto por Quinlan (1993). A seleção da melhor partição dos nós e o critério de parada são baseados na entropia de Shannon, como é usual em parte da família de indução de árvores de classificação [Mitchell, 1997]. O J48 também possui uma fase de pós-poda da árvore após a expansão, na qual são convertidas para folhas as sub-árvores que não representam ganhos de informação acima de um limiar especificado [Quinlan, 1993; Basgalupp, 2010]. O algoritmo é capaz de lidar com classes binárias, nominais e valores faltantes de classe. Suporta atributos binários, de data, nominais, numéricos e valores faltantes.

O SMO (*Sequential Minimal Optimization*) é um algoritmo para treinamento de máquinas de suporte vetorial, proposto por Platt (1998). O SMO se propõe a resolver o

problema de programação quadrática (PQ) do SVM sem o armazenamento de matrizes extras e sem o uso de métodos de solução numérica de PQ [Platt, 1998]. A solução proposta pelo SMO é a decomposição do problema de PQ global em uma série de sub-problemas e então escolher o menor problema de otimização possível para resolver em cada etapa [Platt, 1998]. Estes pequenos sub-problemas são então solucionados de forma analítica, evitando assim o uso de métodos numéricos de otimização, computacionalmente mais intensivos.

O SMO lida particularmente bem com conjuntos de dados esparsos, bem como com dados de entrada binários ou não-binários [Platt, 1998]. Este algoritmo é capaz de lidar com valores faltantes, através de substituição global, e com atributos nominais, onde os transforma em binários [Witten & Frank 2005]. A quantidade de espaço em memória necessária para a execução do SMO é linear no tamanho do conjunto de treinamento, o que o torna capaz de lidar com conjuntos de treinamento grandes [Platt, 1998]. Utiliza geralmente *kernels* polinomiais ou de Gauss.

2.2. Métodos de Ensembles Bagging e Boosting

O método *Bagging* (*Bootstrap Aggregating*), proposto por Breiman (1996), é um meta-algoritmo para construção de classificadores agregados. O método consiste em gerar subconjuntos de exemplos através de sorteio simples com reposição, sobre o conjunto de dados de treinamento original. Cada subconjunto amostrado é utilizado para a construção de um novo classificador. A classificação final é realizada por um sistema de votação, em usualmente se atribui para uma nova instância a classe com maior número de votos entre os classificadores. Uma vez que a classe prevista resulta da combinação das decisões individuais dos classificadores, demonstra-se que as previsões se tornam mais confiáveis à medida que se têm mais votos. Dessa forma, o método *bagging* aposta na expansão da quantidade de classificadores a fim de se reduzir a variância do conjunto, especialmente na presença de dados ruidosos [Breiman, 1996; Nascimento, 2009].

Combinar múltiplos classificadores ajuda quando estes apresentam diferenças entre si e quando cada um lida bem com uma parte do conjunto de dados, complementando-se, ao invés de duplicar um ao outro. O método *boosting*, proposto por Freund e Schapire (1996), explora esse conceito selecionando explicitamente modelos que se complementem. Embora se baseie em reamostragem dos dados do conjunto de treinamento, não adota a reamostragem uniforme. O processo de criação dos subconjuntos de treinamento os condiciona a serem dependentes dos desempenhos individuais de cada classificador [Magalhães, 2007].

O *AdaBoost.M1* é uma implementação do *boosting* de Freund e Schapire (1996). Foi projetado especificamente para problemas de classificação e pode ser aplicado combinado com qualquer algoritmo de aprendizagem de classificação. Trabalha apenas com problemas de classes nominais. O algoritmo inicia atribuindo pesos iguais a todas as instâncias nos dados de treinamento. Em seguida, executa o algoritmo de aprendizagem para gerar um classificador para estes dados e redistribui os pesos para cada instância de acordo com a saída deste classificador. Os pesos de instâncias corretamente classificadas são diminuídos e os pesos dos casos mal classificados são aumentados. Em todas as iterações subseqüentes, um classificador é construído para os

dados reponderados, concentrando-se, portanto, em classificar corretamente as instâncias erroneamente classificadas pelos classificadores anteriores.

2.3. Conjuntos de testes

Foram selecionados 21 conjuntos de dados disponíveis no UCI Machine Learning Repository [Frank & Asuncion, 2010], todos caracterizados por classes categóricas, atributos numéricos e sem valores faltantes. Dos 21 problemas, 8 eram do tipo binário e 13 do tipo multiclasses. A Tabela 1 apresenta os sumários dos conjuntos de dados e a Tabela 2 apresenta suas características básicas: número de exemplos, número de atributos, número de classes e distribuição percentual das classes mais frequentes.

Para avaliar a acurácia dos modelos, utilizou-se o procedimento de validação cruzada. Embora Witten & Frank (2005) postulem que o número ideal de subconjuntos da validação cruzada seja aproximadamente 10, neste trabalho foi adotado o seguinte critério: 10 iterações para *datasets* com mais de 300 exemplos e 5 iterações para *datasets* com menos de 300 exemplos. Buscou-se através desse critério evitar que o uso de conjuntos de testes muito pequenos em cada iteração resultasse em variações muito altas nas estimativas das acurácias. Tomou-se também o cuidado adicional de realizar a partição aleatória de cada *dataset* uma única vez, de modo a garantir que os treinamentos e testes dos diferentes classificadores e *ensembles* fossem realizados exatamente sobre os mesmos exemplos. Dessa forma, foi possível considerar as amostras como pareadas, evitando-se que eventuais diferenças de acurácias entre classificadores ou *ensembles* diferentes pudessem ser atribuídas às flutuações decorrentes da amostragem [Mitchell, 1997].

2.4. Análise estatística

Foi realizada uma análise de significância estatística das diferenças entre as acurácias dos classificadores distintos (J48 e SMO) e entre metaclassificadores distintos (Bagging e Boosting). Essa análise consistiu nos seguintes passos:

a) Para cada conjunto de dados, aplicamos a transformação log-odd[1] sobre as acurácias obtidas nas iterações da validação cruzada. Sobre os log-odds das acurácias, foi realizado o teste de normalidade de Kolmogorov-Smirnov [Noether, 1983]. Em todos os testes realizados, os níveis descritivos (p-valores) foram superiores a 0,05, permitindo-nos então assumir normalidade para os resultados. Essa etapa foi realizada para que se pudesse confirmar a condição de normalidade, necessária para o teste t pareado, conforme descrito abaixo.

b) Entre cada par de classificadores (fixado o metaclassificador) e entre cada par de metaclassificadores (fixado o classificador), foi realizado o teste t pareado [Mitchell, 1997; Noether, 1983], para a comparação entre os log-odds das acurácias obtidas nas iterações de cada validação cruzada. A Figura 1 ilustra o esquema de comparações entre os classificadores e entre os metaclassificadores. Foram

[1] Dada um número $p \in (0,1)$, a transformação log-odd (ou logito) de p é definida como $\log(p/(1-p))$. Essa transformação é usualmente recomendada para variáveis aleatórias limitadas no intervalo (0, 1) [Hosmer and Lemeshow, 2000, pg 6].

consideradas significantes as diferenças entre performances quando o nível descritivo (p-valor) era menor do que 0,05 (indicando, portanto, um classificador ou metaclassificador vencedor). Nos casos em que o nível descritivo foi igual ou maior a 0,05, considerou-se empate entre os competidores.

3. Resultados e Discussões

A Tabela 3 apresenta as médias das acurácias obtidas pela validação cruzada sobre os *datasets*, nas seis configurações possíveis: J48 simples (sem combinação com *ensembles*), SMO simples, Bagging com J48 (Bagg-J48), Bagging com SMO (Bagg-SMO), Boosting com J48 (Boos-J48) e Boosting com SMO (Boos-SMO). Cada linha da tabela contém uma célula destacada em cinza, representando a maior acurácia obtida sobre o *dataset* correspondente, dentre as seis configurações.

Inicialmente, observa-se que a configuração Boos-J48 obteve um número maior de vitórias em relação às demais configurações, tendo apresentado a maior acurácia em 9 dos 21 *datasets*. A configuração Bagg-J48 obteve o segundo maior número de vitórias (4). Dessa forma, os *ensembles* do algoritmo J48 obtiveram, conjuntamente, um total de 13 vitórias entre os 21 *datasets*. Por outro lado, em nenhum dos 21 casos o algoritmo J48 simples obteve acurácias simultaneamente superiores às suas duas configurações *ensemble* correspondentes (Bagg-J48 e Boos-J48). Esses resultados sugerem que *ensembles* Bagging ou Boosting do algoritmo J48 tendem a obter acurácia superior à da versão simples.

Entre as três configurações do SMO, houve 3 vitórias na configuração simples, 2 vitórias na configuração Bagging e 3 vitórias na configuração Boosting, não havendo portanto uma prevalência significativa de vitórias para nenhuma configuração. Adicionalmente, observa-se que em 16 casos a versão SMO simples obteve acurácia igual ou superior a de pelo menos uma dentre suas respectivas configurações *ensemble*, não havendo, portanto, evidência de que configurações *ensemble* do SMO tendam a obter acurácia superior à da versão simples.

Comparando-se os classificadores J48 e SMO, observa-se que, na versão simples, o SMO obteve pequena vantagem, com acurácia superior à do J48 em 12 dos 21 casos. Por outro lado, nas configurações *ensemble*, o J48 teve acurácia superior à do SMO em 13 casos, tanto na configuração Bagging como na configuração Boosting. Esse resultado também sugere uma tendência do J48 se beneficiar mais das configurações *ensemble* do que o SMO.

Comparando-se os *ensembles* Bagging e Boosting do J48 (Bagg-J48 e Boos-J48), verifica-se que a versão Boosting leva ligeira vantagem, com acurácia superior em 12 dos 21 casos. Entre as configurações *ensemble* Bagging e Boosting do SMO (Bagg-SMO e Boos-SMO), verifica-se que a versão Bagging obteve acurácia superior em 12 dos 21 casos.

Os resultados apresentados na Tabela 3 e discutidos nos parágrafos anteriores sugerem que o J48 tende a apresentar melhores resultados quando combinado em *ensembles*, em especial os do tipo Boosting. O SMO parece menos sensível às configurações *ensemble*.

Tabela 1. Sumários dos conjuntos de dados utilizados

Dataset	Descrição
breast-w	Diagnóstico de câncer de mama a partir de características dos núcleos das células extraídas a partir de imagens digitalizadas de amostras de tecidos.
diabetes	Diagnóstico de diabetes a partir de medidas fisiológicas e testes clínicos.
haberman	Predição da sobrevivênvia de pacientes após a realização de cirurgias de câncer de mama.
heart-statlog	Diagnóstico de doenças cardíacas a partir de resultados de testes clínicos.
ionosphere	Classificação de sinais oriundos da ionosfera a partir de observações por radar.
liver-disorders	Diagnóstico de distúrbios hepáticos decorrentes do consumo excessivo de álcool em indivíduos do sexo masculino, a partir de exames sanguíneos.
sonar	Classificação de sinais de sonar provenientes de diferentes ângulos, freqüencias, entre outras condições, em materiais metálicos ou rochosos, de acordo com seu nível de intensidade.
spambase	Classificação de e-mails como spam ou não-spam.
balance-scale	Modelagem de resultados psicológicos experimentais, onde cada indivíduo é classificado como tendo escala de equilíbrio tendendo para a direita, tendendo para a esquerda, ou é equilibrado.
glass	Classificação de tipos de vidros a partir das características físicas das amostras.
kdd_Japanese Vowels	Registros de séries temporais de coeficientes *LPC cepstrum* de locutores do sexo masculino – duas vogais em japonês pronunciadas sucessivamente.
Letter	Reconhecimento ótico de caracteres - letras maiúsculas do alfabeto Inglês.
mfeat-factor	Identificação de numerais manuscritos (`0'-` 9 ') extraídos de uma coleção de mapas holandeses de utilidade; vetor de características composto por 216 correlações de perfil.
mfeat-fourier	Identificação de numerais manuscritos (`0'-` 9 ') extraídos de uma coleção de mapas holandeses de utilidade; vetor de características composto por 76 coeficientes de Fourier na forma de caracteres.
Optdigits	Reconhecimento ótico de dígitos manuscritos a partir de um formulário pré-impresso.
page-blocks	Classificação de blocos do layout de páginas de documentos, a partir de processos de segmentação
Pendigits	Reconhecimento ótico de digitos manuscritos, em tela LCD e caneta, a partir de sequências de pontos representando cada dígito.
Segment	Classificar segmentos (tijolo, céu, folhagem, cimento, janela, caminho, capim) em fotografias de paisagens segmentadas manualmente.
Vehicle	Classificação de silhuetas de veículos a partir de características extraídas das silhuetas.
waveform-5000	Classificação de formas de ondas com base em suas características.
wine_quality	Identificação de qualidade de amostras de "vinho verde" português a partir de variáveis físico-químicas e sensoriais.

Tabela 2. Características básicas dos conjuntos de dados utilizados

	Conjunto de Dados	Nº de Exemplos	Nº de Atributos	Nº de Classes	Distribuição de exemplos nas principais classes (%)			
					1º	2º	3º	Demais
Classes Binárias	breast-w	699	9	2	65,5	34,5		
	diabetes	768	8	2	65,1	34,9		
	haberman	336	3	2	75,9	24,1		
	heart-statlog	270	13	2	55,6	44,4		
	ionosphere	351	34	2	64,1	35,9		
	liver-disorders	345	6	2	58,0	42,0		
	sonar	208	60	2	53,4	46,6		
	spambase	4601	57	2	60,6	39,4		
Multiclasses	balance-scale	625	4	3	46,1	46,1	7,8	
	glass	214	9	7	35,5	32,7	13,6	18,2
	kdd_JapaneseVowels	4274	14	9	14,2	12,7	12,2	60,9
	letter	20000	16	26	4,1	4,0	4,0	87,9
	mfeat-factor	2000	216	10	10,0	10,0	10,0	70,0
	mfeat-fourier	2000	76	10	10,0	10,0	10,0	70,0
	optdigits	5620	64	10	10,2	10,2	10,1	69,6
	page-blocks	5473	10	5	89,8	6,0	2,1	2,1
	pendigits	10992	16	10	10,4	10,4	10,4	68,8
	segment	2310	19	7	14,3	14,3	14,3	57,1
	vehicle	846	18	4	25,8	25,7	25,1	23,5
	waveform-5000	5000	40	3	33,8	33,1	33,1	
	wine_quality	1599	11	10	42,6	39,9	12,4	5,1

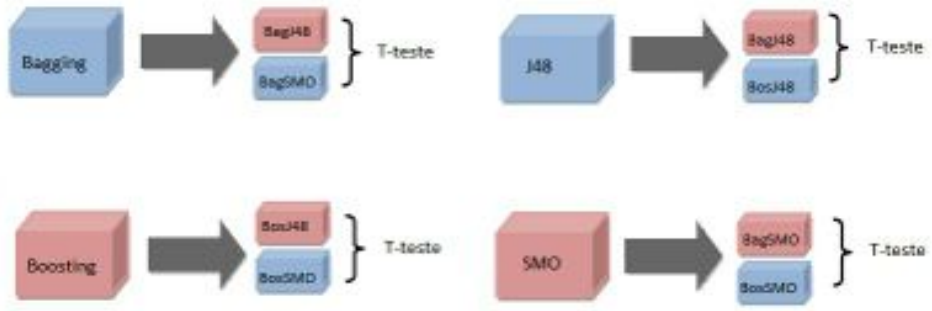

Figura 1. Esquema da realização dos testes comparativos de performance entre os dois classificadores quando fixado o *ensemble* (esquerda) e entre os dois métodos de *ensemble* quando fixado o classificador (direita).

Tabela 3. Acurácias médias dos classificadores nas três configurações possíveis: isoladamente, *ensemble* Bagging e *ensemble* Boosting

	Conjunto de dados	Acurácia média (%)					
		J48	SMO	Bagg-J48	Bagg-SMO	Boos-J48	Boos-SMO
Classes Binárias	breast-w	94,2	96,4	95,1	96,7	97,0	96,4
	diabetes	75,1	77,6	77,1	77,0	74,9	77,8
	haberman	73,0	73,3	73,0	73,0	69,3	74,0
	heart-statlog	79,6	83,3	81,5	85,2	80,7	84,1
	ionosphere	90,0	88,3	90,9	88,9	92,3	89,4
	liver-disorders	68,2	57,6	71,8	57,9	67,9	57,9
	sonar	77,1	77,2	80,5	76,6	77,6	77,1
	spambase	92,6	89,1	94,4	89,3	95,5	89,7
Multiclasses	balance-scale	79,5	87,7	81,8	87,4	76,9	87,9
	glass	65,2	51,0	71,9	43,8	74,8	47,6
	kdd-JapaneseVowels	87,1	94,2	93,4	94,2	96,3	94,1
	letter	87,9	80,7	92,7	80,8	95,6	80,6
	mfeat-factors	89,2	97,9	93,7	97,6	95,6	97,2
	mfeat-fourier	76,9	83,4	80,0	82,9	81,8	81,9
	optdigits	90,4	98,3	95,3	98,4	97,4	98,1
	page-blocks	97,0	92,2	97,1	92,3	97,0	92,2
	pendigits	96,4	97,6	98,0	97,8	99,0	98,0
	segment	97,1	92,1	97,6	92,0	98,3	91,9
	vehicle	72,7	71,1	71,9	70,5	78,1	69,9
	waveform-5000	77,0	86,6	81,5	86,5	81,2	86,5
	wine_quality	61,4	57,7	67,7	57,7	66,7	58,3

A fim de corroborar essas hipóteses, a Tabela 4 apresenta os níveis descritivos (p-valores) do teste t pareado, para os quatro pares de configurações comparados: na 1ª e 2ª colunas são apresentados os p-valores da comparação de desempenho entre Bagging e Boosting, fixado o classificador; analogamente, a 3ª e 4ª coluna apresentam os p-valores da comparação entre J48 e SMO, fixado o metaclassificador. Para cada caso em que
p-valor < 0,05, é apresentado o melhor competidor.

Comparando-se os *ensembles* Bagging e Boosting sobre o algoritmo J48, em 11 casos houve diferenças significantes entre acurácias, dentre os quais em nove o Boosting mostrou-se a melhor configuração – contra apenas dois casos favoráveis para o Bagging. Por outro lado, para o algoritmo SMO, em apenas dois casos se observou diferença significante de performance entre os *ensembles*, sendo um caso favorável ao Bagging e outro favorável ao Boosting. Esses resultados confirmam aqueles apresentados na Tabela 3, e sugerem que o J48 tem sua performance influenciada pelo *ensemble* utilizado, tendendo a apresentar melhores resultados na configuração Boosting; e que o SMO apresenta baixa sensibilidade em relação ao metaclassificador.

Tabela 4. Níveis descritivos (p-valores) das diferenças entre o log-odds das acurácias médias entre os pares de configurações: Bagg-J48 × Boos-J48 (1ª coluna); Bagg-SMO × Boos-SMO (2ª coluna); Bagg-J48 × Bagg-SMO (3ª coluna); Boos-J48 × Boos-SMO (4ª coluna); nos casos em que p-valor < 0,05, são apresentadas as configurações com maior acurácia.

	Conjunto de dados	J48		SMO		Bagging		Boosting	
		Melhor ensemble	p-valor	Melhor ensemble	p-valor	Melhor classif.	p-valor	Melhor classif.	p-valor
Classes Binárias	breast-w	Boosting	0,02		0,17		0,10		0,36
	diabetes		0,15		0,08		0,90	SMO	0,04
	haberman		0,09		0,19		1,00	SMO	0,04
	heart-statlog		0,54		0,50	SMO	0,01		0,10
	ionosphere		0,24		0,81		0,28		0,26
	liver-disorders	Bagging	<0,01		>0,99	J48	<0,01	J48	0,01
	sonar		0,37		0,75		0,18		0,75
	spambase	Boosting	<0,01		0,25	J48	<0,01	J48	<0,01
Multiclasses	balance-scale	Bagging	0,01		0,47	SMO	0,01	SMO	<0,01
	glass		0,39		0,33	J48	<0,01	J48	<0,01
	kdd_JapaneseVowels	Boosting	<0,01		0,45		0,18	J48	<0,01
	letter	Boosting	<0,01	Bagging	0,02	J48	<0,01	J48	<0,01
	mfeat-factors	Boosting	0,01		0,07	SMO	<0,01	SMO	<0,01
	mfeat-fourier		0,12		0,17	SMO	0,02		0,96
	optdigits	Boosting	<0,01		0,11	SMO	<0,01	SMO	0,02
	page-blocks		0,42		0,17	J48	<0,01	J48	<0,01
	pendigits	Boosting	<0,01	Boosting	0,03		0,09	J48	<0,01
	segment	Boosting	0,02		0,43	J48	<0,01	J48	<0,01
	vehicle	Boosting	0,01		0,56		0,53	J48	<0,01
	waveform-5000		0,57		1,00	SMO	<0,01	SMO	<0,01
	wine_quality		0,34		0,05	J48	<0,01	J48	<0,01

As colunas 3 e 4 da tabela indicam uma grande consistência nos resultados comparativos entre o J48 e o SMO: não houve nenhum caso em que o J48 fosse indicado como melhor classificador sob a configuração Bagging e simultaneamente o SMO fosse indicado como melhor na configuração Boosting, ou vice-versa. Ou seja, se um algoritmo combinado com Bagging é considerado melhor do que o outro, seu desempenho na configuração Boosting será pelo menos igual ao do competidor, e vice-versa. A partir dessa observação podemos ter uma indicação final do melhor algoritmo para cada *dataset*, bastando escolher aquele que teve melhor desempenho na configuração Bagging ou na configuração Boosting.

Nos resultados comparativos entre o J48 e o SMO, observou-se um relativo equilíbrio, já que o J48 teve acurácia superior em 10 *datasets* (em uma das configurações Bagging ou Boosting) enquanto o SMO teve acurácia superior em 8 *datasets*. É interessante ressaltar que, mesmo não havendo uma clara superioridade de um algoritmo

em relação ao outro, nota-se que em 18 dos 21 casos um dos classificadores teve desempenho médio estatisticamente superior ao outro. Esse resultado indica que as performances relativas dos algoritmos variam entre problemas de classificação distintos, e reforça a importância de se considerar e testar diferentes classificadores (e, eventualmente, metaclassificadores) para todo novo problema.

Outros trabalhos correlatos também analisam o desempenho dos *ensembles* bagging e boosting. Bauer & Kohavi (1998) conduziram um estudo empírico onde compararam *ensembles* de indutores de árvore de decisão e indutor Naive Bayes. O objetivo foi compreender que fatores afetam o erro de classificação desses algoritmos. Os autores constataram empiricamente que o Bagging performa melhor são usadas estimativas probabilísticas de erro, em conjunto com a não realização da poda. Para o Adaboost, encontrou-se uma associação positiva entre o aumento no tamanho médio da árvore e seu sucesso na redução do erro. Além disso, mostrou-se que os métodos de votação exibem significantes reduções nos erros médio-quadrado quando comparado com métodos sem votação.

A análise elaborada por Maclin & Opitz (1999) indicou que o desempenho dos métodos de Boosting é dependente das características do conjunto de dados e sugere que a maior parte do ganho no desempenho de um ensemble está nos poucos primeiros classificadores combinados, mas que, embora isto, podem ser vistos consideráveis ganhos combinando-se mais de 25 classificadores quando utilizadas árvores de decisão.

Este trabalho e os demais estudos convergem na reunião de esforços em busca de conhecimento sobre os desempenhos dos algoritmos de classificação e *ensembles* para cenários e condições variados.

4. Conclusões e Trabalhos Futuros

Neste trabalho, realizamos uma análise comparativa de desempenho de um algoritmo da família de árvores de classificação (J48) e da família de máquinas de vetores suporte (SMO), sob as configurações *ensemble* Bagging e Boosting. Essa análise foi baseada em experimentos de validação cruzada sobre 21 *datasets*.

Os resultados sugerem que o J48 tem sua performance influenciada pelo ensemble utilizado, tendendo a apresentar melhores resultados na configuração Boosting. Esses resultados são consistentes com aqueles obtidos por Freund & Schapire (1996) que, em seu experimento, concluíram que o Boosting possui desempenho significativo e uniformemente melhor que o Bagging quando o algoritmo de aprendizado base gera classificadores bastante simples. Além disso, afirmaram que, quando combinado com o C4.5 (J48 no Weka), o Boosting parece ter desempenho ligeiramente superior ao do Bagging.

Não foram encontrados indícios de que o SMO tenha sua performance influenciada pelas configurações *ensemble*, sendo que em diversos casos a configuração simples obteve desempenho superior a pelo menos uma das versões *ensemble*.

Os resultados da análise de significância estatística das diferenças de desempenho entre o J48 e o SMO sob as configurações *ensemble* mostraram-se consistentes, no sentido de que não houve nenhum dataset para o qual o J48 fosse simultaneamente melhor do que o SMO em uma configuração *ensemble* e pior em outra. Ou seja, se na

configuração Bagging um algoritmo era considerado melhor do que o outro (p-valor < 0,05), na configuração Boosting seu desempenho era pelo menos igual ao de seu competidor, e vice-versa.

Também se observou que, na grande maioria dos casos, um dos algoritmos apresentou acurácia estatisticamente superior ao outro, sem no entanto haver uma clara prevalência de um algoritmo sobre o outro em número de vitórias. Esse resultado reforça a necessidade de se considerar e testar diversos algoritmos para todo novo problema de classificação.

Este estudo foi realizado sobre uma coleção moderada de *datasets* (21), em virtude da restrição de se analisar apenas *datasets* sem dados faltantes. Estudos comparativos adicionais poderiam ser realizados com um número maior de datasets (incluindo aqueles com dados faltanes) e incluindo mais algoritmos representantes das duas famílias analisadas (árvores de classificação e máquinas de suporte vetorial). Análises de associação entre o desempenho relativo dos algoritmos e as características dos datasets (número de exemplos, atributos e classes e indicadores associados; presenças de ruídos e outliers; etc) poderão também ser conduzidas futuramente.

Os autores são gratos pelo apoio e financiamento recebidos da EACH-USP, da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e da Fundação de Apoio à Pesquisa do Estado de São Paulo (FAPESP).

Referências

Basgalupp, M.P. (2010). LEGAL-Tree: Um algoritmo genetico multi-objetivo lexicográfico para indução de árvores de decisão. Tese de Doutorado. Instituto de Ciências Matemáticas e de Computação. Universidade de São Paulo, São Carlos.

Bauer,E.; Kohavi R. (1998) An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants. Kluwer Academic Publishers, Boston: Machine Learning, vv, 1-38Bouckaert, R.R., Frank, E., Hall, M., Kirkby, R., Reutemann, P., Seewald, A., Scuse, D. (2012) WEKA Manual for Version 3-7-7. Hamilton: The University of Waikato.

Breiman, L. (1996) Bagging predictors. Machine Learning, 24, 123–140.

Coelho, G.P. (2006). Geração, Seleção e Combinação de Componentes para Ensembles de Redes Neurais Aplicadas a Problemas de Classificação. Dissertação de Mestrado. Faculdade de Engenharia Elétrica e de Computação. Universidade Estadual de Campinas.

Frank, A. and Asuncion, A. (2010). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

Freund, Y.; Schapire, E.R. (1996); Experiments with a new boosting algorithm. In Proceedings of the International Conference on Machine Learning, p.148–156. Morgan Kaufmann, San Francisco.

Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P.; Witten, I. H. (2009); The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1.

Hosmer, D.W., Lemeshow, S. (2000). Applied Logistic Regression. 2nd ed. New York: Wiley.

Lehmann,C.; Koenig, T.; Jelic,V.; Prichep,L.; John,R.E.; Wahlund,L.; Dodgee,Y.; Dierks,T. (2007) Application and comparison of classification algorithms for recognition of Alzheimer's disease in electrical brain activity (EEG). Journal of Neuroscience Methods, 161, 342–350

Loh,W.; Shih,Y. (2000) A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-three Old and New Classification Algorithms. Kluwer Academic Publishers, Boston: Machine Learning, 40, 203-229.

Maclin, R.; Opitz, D. (1999) Popular Ensemble Methods: An Empirical Study. Journal Of Artificial Intelligence Research, Vol. 11, pag. 169-198.Magalhaes, R. M. (2007); Uma investigação sobre a utilização de processamento paralelo no projeto de máquinas de comitê centro de tecnologia. Dissertação de Mestrado. Universidade Federal do Rio Grande do Norte, Natal, RN.

Mitchell, T.M. (1997). Machine Learning. McGraw-Hill.

Nascimento, D.S.C. (2009) Configuração Heterogênea de Ensembles de Classificadores: Investigação em Bagging, Boosting e MultiBoosting. Dissertação de Mestrado, Universidade de Fortaleza, Fortaleza, CE.

Noether, G.E. (1983). Introdução à Estatística: Uma Abordagem Não Paramétrica. 2ed. Rio de Janeiro: Guanabara Dois.

Platt, J. C. (1998) Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. Microsoft Research, Technical Report MSR-TR-98-14, USA.

Quinlan, R. (1993) C4.5: Programs for Machine Learning Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

R Core Team (2011). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. http://www.R-project.org

Witten, I. H.; Frank, E. (2005) Data Mining: Practical Machine Learning Tools and Techniques. Segunda Edição. Ed. Elsevier.