BERT: Melhorando Classificação de Texto com Árvores Extremamente Aleatórias, Bagging e Boosting

Raphael R. Campos[1], **Marcos A. Gonçalves** [1]

[1]Departamento de Ciência da Computação - Universidade Federal de Minas Gerais (UFMG)
Av. Antônio Carlos 6627 - ICEx - 31270-010 Belo Horizonte , Brasil

{rcampos, mgoncalv}@dcc.ufmg.br

Abstract. *One of the most effective methods for text classification is the recently proposed BROOF classifier, a boosted version of Random Forest (RF). In this work, we propose to improve the BROOF strategy by exploiting Extremely Randomized Trees (Extra-Trees) as a "weak learner" in the boosting framework. In this context, we also introduce the Bagging procedure into the Extra-Trees models so that we can estimate a better Out-of-Bag (OOB) error when compared to the original BROOF. Our experiments with several textual datasets, comparing with up to nine state-of-the-art classifiers, show that our proposed method (a.k.a, BERT) is among the top performers classifiers in all tested datasets, outperforming the original BROOF in several cases.*

Resumo. *Um dos métodos mais eficazes para classificação de texto é o recém-proposto BROOF, uma versão Boosting da Floresta Aleatória (FA). Nesse artigo, nós propomos melhorar o método BROOF explorando Árvores Extremamente Aleatórias (AEA) como um "aprendiz fraco" no arcabouço do boosting. Nesse contexto, nós introduzimos o procedimento de Bagging nos modelos de AEA de modo que possamos estimar melhor o erro Out-of-Bag (OOB) se comparado ao BROOF original. Nossos experimentos com vários conjuntos de dados textuais e nove classificadores estado-da-arte, mostram que o método proposto (BERT) está dentre os classificadores com melhores desempenhos em todos os conjuntos de dados testados, saindo-se melhor que o BROOF em vários casos.*

1. Introdução

Classificação de texto é uma das tarefas centrais em recuperação de informação (RI). Desde o advento da *Web* o volume de dados disponível tem crescido de modo impressionante. Classificação textual provê meios de organizar essa informação permitindo melhor compreensão e interpretação dos dados. Dessa maneira, faz-se necessários métodos automáticos eficazes capazes de auxiliar nessa tarefa. Nos últimos anos vários métodos, tais como Florestas Aleatórias (FAs)[Breiman 2001], *Support Vector Machine (SVM)*, k-vizinhos mais próximos, e BROOF[Salles et al. 2015], têm sido propostos para resolver de forma efetiva esse tipo de problema. Esses métodos tentam aprender um mapeamento entre conjunto de documentos e um conjunto de rótulos predefinidos de modo que após aprendido, o mapeamento possa ser aplicado na predição de documentos cujo rótulo é desconhecido. Formalmente, esse tipo de problema é chamado de aprendizado supervisionado e consiste em: Dado um conjunto de N exemplos da forma $\{(X_1, y_1), ..., (X_N, y_N)\}$,

conhecido como conjunto de treino, onde X_i denota a representação vetorial da i-ésima instância e $y_i \in Y$ é um atributo categórico que indica a classe da instância. O objetivo de um algoritmo de aprendizado supervisionado é aprender uma aproximação da distribuição de probabilidade *a posteriori* $P(y_i|X_i)$, que descreve a relação entre os pontos e suas classes associadas, baseado nos dados observados.

Um dos métodos de aprendizado supervisionado mais eficazes conhecido para resolver o problema de classificação de texto é o recém-proposto BROOF[Salles et al. 2015], que combina *boosting* e *bagging* explorando FAs como aprendizes fracos no processo de *boosting*. Nesse trabalho, nós propomos melhorar o método BROOF explorando Árvores Extremamente Aleatórias (AEA) como um "aprendiz fraco" no arcabouço do *boosting*. Nesse contexto, nós introduzimos o procedimento de *bagging*[Breiman 1996] nos modelos de AEA de modo que possamos estimar o erro *Out-of-Bag* (OOB) para o processo do BROOF. Nossos resultados experimentais mostram que nosso método supera (ou pelo menos empata) todos os classificadores *baselines*. Em especial, quando comparado ao BROOF, nós conseguimos superá-lo em metade dos conjuntos de dados testados.

Esse artigo está organizado do seguinte modo. Seção 2 cobre os trabalhos relacionados. Seção 3 descreve o método proposto. Seção 4 apresenta nossa avaliação experimental, resultados e análises. Seção 5 conclui o artigo.

2. Trabalhos Relacionados

É sabido que a Floresta Aleatória (FA) [Breiman 2001] e suas derivações produzem resultados estado-da-arte em várias tarefas de recuperação de informação (RI) tais como classificação, regressão e ranqueamento [Fernández-Delgado et al. 2014, Salles et al. 2015]. No entanto, é também sabido que as FAs podem ter a eficácia prejudicada quando são aplicadas a dados com muitos atributos irrelevantes ou ruidosos[Segal 2004], que são características de tarefas como classificação de texto. Há na literatura várias abordagens para melhorar o modelo da FA, nesse trabalho focaremos em duas: Árvores Extremamente Aleatórias(AEA) e BROOF.

As Árvores Extremamente Aleatórias(AEA) [Geurts et al. 2006] diferem das FAs em dois aspectos: (1) elas não aplicam o procedimento de *bagging* ao construir o conjunto de exemplos de treino para cada árvore. Portanto, o mesmo conjunto de treino é usado para treinar todas as árvores; (2) elas escolhem o nodo para divisão de forma extrema (tanto o índice do atributo quanto o valor do limiar são escolhidos aleatoriamente), por outro lado a FA encontra o melhor ponto de corte (i.e., o valor ótimo tanto para o índice quanto para valor de corte) dentre o subconjunto de atributos escolhidos aleatoriamente. Essas mudanças tornam o algoritmo competitivo e em alguns casos superior à FA original.

Em [Salles et al. 2015] os autores combinam *boosting* e *bagging* explorando FAs como aprendizes fracos no processo de *boosting*. *Boosting* é um algoritmo sequencial que treina vários "aprendizes fracos" (isto é, classificadores capazes de predizer melhor que adivinhação) para gerar predições precisas. Para cada iteração i do *boosting*, seja Δ^i uma distribuição de probabilidade sobre o conjunto de treino de tamanho M. Quando $i = 0$, $\Delta^0(j) = \frac{1}{M}$, $\forall j|_{j=1}^{M}$. Na iteração $i > 0$, para cada exemplo de treino x_j, se x_j for classificado errado, seu peso é aumentado de modo que, na próxima iteração, a distribuição atualizada Δ^{i+1} seja considerada, dando-se maior ênfase aos exemplos

classificados incorretamente (mais difíceis de classificar). Em contraste com AdaBoost [Freund and Schapire 1997], a regra de ponderação em [Salles et al. 2015] foi alterada para usar o erro *Out-of-Bag*(OOB), provido pela Floresta Aleatória durante o treino, além de "suavemente" atualizar o peso apenas das instâncias OOB. Essas duas abordagens juntas previnem o *boosting* de convergir rapidamente e mantêm as Florestas Aleatórias focadas nas regiões do espaço de entrada de difícil classificação. Assim, o sistema se torna mais robusto ao problema de sobre-ajuste e a questão do ruído, gerando modelos com maior capacidade de generalização.

3. BERT

Nessa seção, descrevemos o algoritmo de aprendizado proposto: o classificador BERT (*Boosted Extremely Randomized Trees*). BERT combina *boosting* e Árvores Extremamente Aleatórias, explorando a ideia do BROOF de usar o erro *Out-Of-Bag* (OOB) como uma estimativa de erro mais robusta para o processo de ponderação do *boosting* e apenas atualizar os pesos das instâncias OOB.

O erro *Out-of-Bag* (OOB) muitas vezes chamado de erro de generalização, é computado diretamente enquanto a Floresta Aleatória é treinada. Durante o processo de *Bootstrap* usado pelas FAs para gerar subconjuntos de treino para treinar cada árvore do comitê, cada árvore de decisão é treinada com aproximadamente $1 - \frac{1}{\epsilon} \approx 63\%$ do conjunto de treino original[Hastie et al. 2009]. Similarmente a validação cruzada, as instâncias deixadas de fora (Out-of-Bag) do treino das árvores podem ser utilizadas para produzir uma estimativa sem viés da taxa de erro esperado. [Salles et al. 2015] mostra que o uso de uma estimativa de erro menos enviesada, como erro OOB, juntamente com a ponderação apenas das instâncias deixadas de fora (OOB), melhoram drasticamente o poder de generalização do *boosting* e FAs, sobretudo quando aplicados a dados textuais.

O nosso classificador BERT combina essa ideia com Árvores Extremamente Aleatórias. *Nossa hipótese é que o uso de um método de aprendizado mais robusto como "aprendiz fraco" produza estimativas melhores do erro OOB, e potencialmente, melhore o poder de generalização do classificador final.*

Algoritmo 1: Treino BERT: Pseudo-Código

Dados: $D_{treino} = \{(X_i, y_i) : \forall i \in \{1, .., N\}\}$, $n_{arvores}$
Resultado: Lista L contendo as AEA treinadas com seus respectivos pesos por iteração

$w_1 \leftarrow \frac{1}{|D_{treino}|}$;
$L \leftarrow \emptyset$;
para *cada* $m \in \{1, .., M\}$ **faça**
  $(h_m^{AEA}, (X, y, \hat{y})_i^{oob}) \leftarrow AEA(D_{treino}, n_{arvores}, w_m)$;
  $OOB_{err}^w \leftarrow \frac{\sum_{i \in O} w_m^i I[y \neq \hat{y}]}{\sum_{i \in O} w_m^i}$, onde $O = (X, y, \hat{y})_i^{oob}$;
  $\alpha_m \leftarrow \frac{1 - OOB_{err}^w}{OOB_{err}^w}$;
  $w_{m+1}^i \leftarrow w_m^i \epsilon^{\alpha_m I[y \neq \hat{y}]}$;
  $L \leftarrow L \cup \{(h_m^{AEA}, \alpha_m)\}$;
fim

Como mencionado anteriormente, AEA não utilizam o procedimento de *Bagging*. Para estimar o erro OOB nós introduzimos esse procedimento ao processo de treino

das AEA. Formalmente, o classificador Árvores Extremamente Aleatórias constrói várias árvores, cada uma das quais é construída do seguinte modo: Primeiro, o procedimento de *bagging* (*bootstrap aggregating*) é executado. Introduzido por [Breiman 1996], *bagging* é um método que almeja controlar a variância criando várias versões do classificador e tirando a média delas. Dado um conjunto de treino D_{treino}, o procedimento de *bagging* gera M conjuntos de treino $D^i_{treino}|^M_{i=1}$ via amostragem por substituição do conjunto original de dados. Segundo, M árvores extremamente aleatórias $h_i|^M_{i=1}$ são treinadas com os recém-criados conjuntos de treino, utilizando o algoritmo descrito em [Geurts et al. 2006]. Finalmente, a predição é dada pela média de cada h_i, $1 \leq i \leq M$.

O algoritmo 1 resume o funcionamento do método proposto, que possui ideia similar a descrita em [Salles et al. 2015] com a diferença de usar Árvores Extremamente Aleatórias com *bagging* (como foi supracitado) no processo (linha 4).

4. Projeto Experimental - Classificação Textual

Um dos grandes desafios quando categorizamos dados textuais em tópicos é que esses dados são geralmente representados por uma grande quantidade de atributos (alta dimensionalidade) e muito deles são irrelevantes ou ruidosos, devido às propriedades inerentes da linguagem humana. Todavia, apesar de ser uma tarefa desafiadora, é de grande valia e importância hoje em dia, devido sua vasta aplicação e demanda. Para avaliação do método proposto para categorização de tópicos, nós consideramos quatro conjuntos de dados textuais reais, conhecidos como: *20 Newsgroups* (Notícias), *Four Universities* (Páginas web), *Reuters* (Notícias) e *ACM Digital Library* (Artigos Científicos em Computação).Para todos os conjuntos de dados, nós executamos pré-processamento padrão: removemos *stopwords* usando a lista padrão SMART e aplicamos uma simples remoção de atributos removendo termos com baixa "frequência nos documentos (FD)"[1]. Em relação à ponderação de termos, nós utilizamos TF para todos os classificadores baseados em Floresta Aleatória e Naïve Bayes, e usamos TDIDF com normalização L2 para os classificadores KNN e SVM. Todos os conjuntos de dados são de rótulo único.

Conjunto	Classes	# atrib	# docs	Densidade	Tamanho
4UNI	7	40,194	8,274	140.325	14MB
20NG	20	61,049	18,766	130.780	30MB
ACM	11	59,990	24,897	38.805	8.5MB
REUT90	90	19,589	13,327	78.164	13MB

Tabela 1. Informações gerais sobre os conjuntos de dados.

4.1. Experimentação

Nós conduzimos experimentos controlados para analisar e comparar a efetividade do método proposto, BERT, e os métodos *baselines* estado-da-arte. Os métodos foram comparados utilizando duas medidas padrões de recuperação de informação: *micro averaged* F_1 (MicroF_1) e *macro averaged* F_1 (MacroF_1). Para compararmos a média dos resultados usando nosso experimento de validação cruzada 5-*folds*, nós avaliamos a significância estatística dos nossos resultados com um teste t pareado, com 95% de confiança e correção de Bonferroni. Este teste garante que os melhores resultados, marcados em **negrito**, são estatisticamente superiores aos outros. Foram avaliados nove algoritmos de aprendizado: (1) **SVM com *kernel* linear**; (2) **k-vizinhos mais próximos (KNN)**; (3) **Naïve Bayes Multinomial (NB)**; (4) **Árvore de Decisão (AD)**; (5)**Floresta**

[1]Nós removemos todos os termos que ocorrem em menos de seis documentos (i.e., FD<6).

Aleatória (FA) ;(6)**Árvores Extremamente Aleatórias (AEA)**, (7) **ADA.FA**, AdaBoost [Freund and Schapire 1997] com FA como "aprendiz fraco"; (8) **ADA.M2**, AdaBoost multi-classe com *decision stumps* como "aprendizes fracos"; todos esses algoritmos estão disponíveis em *Python* pela biblioteca **Scikit-learn** [2]; (9) **BROOF** [Salles et al. 2015]; e (10) **Boosted Extremely Randomized Trees (BERT)**, esses dois últimos foram todos implementados por nós em *Python*. Os hiperparâmetros foram escolhidos usando validação cruzada *5-fold* no conjunto de treino para todos os classificadores, exceto BROOF e BERT. Para esses fixamos o número de árvores e iterações em 8 e 200, respectivamente, e utilizamos os melhores hiperparâmetros encontrados para FA e AEA para os respectivos conjuntos de dados. Gostaríamos de salientar que alguns dos resultados obtidos podem diferir dos reportados em outros trabalhos para os mesmos conjuntos de dados. Tais discrepâncias podem ser devido a vários fatores tais como diferença na preparação dos conjuntos de dados[3], o uso de diferentes divisões do conjunto de dados (e.g., alguns conjuntos de dados têm "divisões padrões" tal como REUT90[4]). Além disso, nós rodamos todas as alternativas sob as mesmas condições em todos os conjuntos de dados, usando o melhor esquema de ponderação, um padronizado e bem aceito procedimento de validação cruzada para otimizar os parâmetros para cada uma das alternativas e aplicamos o apropriado ferramental estatístico para análise dos resultados. Todos os conjuntos de dados utilizados nesse trabalho estão disponíveis mediante requisição.

4.2. Resultados e Discussões

Nossos resultados mostram que BERT supera (ou ao menos empata com) todos os classificadores estados-da-arte (SVM, AD, KNN, NB), bem como FA tradicional e AEA.

		20NG	4UNI	ACM	REUT90
BERT	microF1	**89.45 ± 0.46**	**84.61 ± 0.98**	**74.8 ± 0.59**	**67.33 ± 0.72**
	macroF1	**89.13 ± 0.58**	**73.61 ± 1.85**	**62.1 ± 0.99**	**29.24 ± 1.4**
SVM	microF1	**90.06 ± 0.43**	**83.48 ± 1.08**	**75.4 ± 0.66**	**68.19 ± 1.15**
	macroF1	**89.93 ± 0.43**	**73.39 ± 2.17**	**63.84 ± 0.55**	**31.95 ± 2.59**
BROOF	microF1	87.96 ± 0.24	**84.41 ± 1.07**	73.35 ± 0.79	**66.79 ± 0.97**
	macroF1	87.44 ± 0.28	**73.23 ± 1.1**	60.76 ± 0.8	**28.48 ± 2.17**
NB	microF1	**88.99 ± 0.54**	62.63 ± 1.7	73.54 ± 0.71	65.32 ± 1.13
	macroF1	**88.68 ± 0.55**	51.38 ± 3.19	58.03 ± 0.85	**27.86 ± 0.79**
KNN	microF1	87.53 ± 0.69	75.63 ± 0.94	70.99 ± 0.96	**68.07 ± 1.07**
	macroF1	87.22 ± 0.66	60.34 ± 1.36	55.85 ± 0.97	**29.93 ± 2.48**
AEA	microF1	87.03 ± 0.41	**82.87 ± 1**	73.08 ± 0.55	64.87 ± 0.81
	macroF1	86.65 ± 0.56	68.54 ± 2.6	58.71 ± 0.89	26.18 ± 2.55
FA	microF1	83.64 ± 0.29	81.52 ± 1	71.05 ± 0.31	63.92 ± 0.81
	macroF1	83.08 ± 0.35	65.44 ± 1.91	56.56 ± 0.45	24.36 ± 1.98
ADA.FA	microF1	81.57 ± 0.22	77.95 ± 2.29	62.73 ± 0.98	62.6 ± 0.6
	macroF1	81.04 ± 0.32	61.19 ± 4.18	50.49 ± 0.87	21.35 ± 1.42
AD	microF1	58.94 ± 1.3	71.77 ± 0.67	60.61 ± 0.84	57.35 ± 1.22
	macroF1	58.42 ± 1.12	55.47 ± 4.08	47.89 ± 1.21	20.48 ± 1.33
ADA.M2	microF1	57.28 ± 0.41	72.71 ± 1.75	57.98 ± 1.77	57.43 ± 0.98
	macroF1	58.12 ± 0.35	59.77 ± 2.31	48.44 ± 1.33	25.02 ± 1.41

Tabela 2. Comparação entres os métodos

Pode-se observar na Tabela 2 que o método proposto, juntamente com SVM, são os únicos classificadores que obtêm melhor desempenho considerando todas as métricas

[2]Disponível em `http://scikit-learn.org/`

[3]Por exemplo, alguns trabalhos exploram complexos e sofisticados esquemas de ponderação para os atributos ou mecanismos de seleção de atributos que favorecem alguns algoritmos em detrimento de outros.

[4]Nós acreditamos que rodar experimentos somente nas divisões padrões não é um bom procedimento experimental já que não nos permite um tratamento estatístico apropriado dos resultados.

em todos conjuntos de dados, porém o BERT tem a vantagem de ser mais interpretável (através da visualização das regras contidas nas árvores ou por meio das medidas de importância dos atributos propostas por [Breiman 2001]). Ademais, quando nós comparamos BERT com a FA e AEA, as "bases" da ideia, obtemos ganhos de até 6,94% (20NG) e 3,79% (REUT90) em MicroF_1, respectivamente. De fato, BERT é superior à FA e AEA em todos os conjuntos de dados, apenas empatando com AEA na métrica microF_1 no 4UNI. Além disso, BERT superou o BROOF original em dois conjuntos de dados tanto em termos de MicroF_1 quanto MacroF_1, alcançando ganhos de até 1,69% de MicroF_1 (20NG) e 2,21% de MacroF_1 (ACM). Esses resultados corroboram nossa hipótese: o uso de um algoritmo mais robusto (AEA com *bagging*) combinado às ideias do BROOF produz um algoritmo de aprendizado com maior poder de generalização, melhorando tanto o BROOF quanto a própria AEA.

5. Conclusão e Trabalhos Futuros

Nesse trabalho propusemos melhorar o recém-proposto e altamente eficaz método BROOF, explorando Árvores Extremamente Aleatórias (AEA), e as ideias, introduzidas em [Salles et al. 2015], de usar o erro *Out-Of-Bag* (OOB) como uma estimativa de erro menos enviesada para a ponderação do *boosting* e, suavemente, ponderar apenas as instâncias OOB. Além disso, introduzimos o procedimento de *bagging* nos modelos de AEA de modo que pudemos usá-lo no processo do BROOF. Nossos experimentos no contexto de classificação de texto mostram que o BERT superou (ou no mínimo empatou), com significância estatística, todos os *baselines* em termos de MicroF_1 e Macro$F1$ em todos os conjuntos de dados testados, o que é um resultado salutar. Como trabalho futuro pretendemos fazer um estudo empírico mais extenso abrangendo coleções de outros domínios tais como análise de sentimentos e micro-arranjos.

Referências

Breiman, L. (1996). Bagging predictors. *Mach. Learn.*, 24(2):123–140.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Fernández-Delgado, M., Cernadas, E., Barro, S., and Amorim, D. (2014). Do we need hundreds of classifiers to solve real world classification problems? *J. Mach. Learn. Res.*, 15(1):3133–3181.

Freund, Y. and Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci.*, 55(1):119–139.

Geurts, P., Ernst, D., and Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, 63(1):3–42.

Hastie, T., Tibshirani, R., and Friedman, J. H. (2009). *The Elements of Statistical Learning*. Springer.

Salles, T., Gonçalves, M., Rodrigues, V., and Rocha, L. (2015). Broof: Exploiting out-of-bag errors, boosting and random forests for effective automated classification. In *Proc. of the 38th International ACM SIGIR Conference on Inf. Retrieval*, pages 353–362.

Segal, M. R. (2004). Machine learning benchmarks and random forest regression. Technical report, University of California.