Seleção customizada de classificadores e oportunidades para paralelismo

Paulo Henrique da Silva, Wellington Santos Martins, Thierson Couto Rosa

[1]Instituto de Informática – Universidade Federal de Goiás (UFG)
Alameda Palmeiras, Quadra D, Campus Samambaia
CEP 74690-900 – Goiânia – GO – Brasil

{paulohsilva, wellington, thierson}@inf.ufg.br

Abstract. *Automatic document classification (ADC) is considered one of the most relevant and challenging tasks in the context of information retrieval, due to the high dimensionality and sparse data. Some works advocate the use of dynamic selection of the classifier to improve the accuracy of this task. This work proposes the customized selection of the classification method performed in query time (test) as well as the exploitation of parallelism to speed up the ADC task. Experimental results, using standardized databases, show competitive and promising results in applications, and new opportunities for exploiting parallelism.*

Resumo. *A classificação automática de documentos (ADC) é considerada uma das tarefas mais relevantes e desafiadoras no contexto de recuperação de informações, devido a alta dimensionalidade e esparsidade dos dados. Alguns trabalhos defendem o uso da seleção dinâmica do classificador para melhorar a acurácia desta tarefa. Este trabalho propõe a seleção customizada de método de classificação realizada em tempo de consulta (teste), bem como a exploração de paralelismo para acelerar a tarefa de ADC. Resultados experimentais, utilizando bases de dados padronizadas, mostram resultados competitivos e promissores nas aplicações, e novas oportunidades para exploração de paralelismo.*

1. Introdução

O recente aumento nos dados armazenados digitalmente estimulou o desenvolvimento de métodos para organizar e extrair conhecimento relevante desse grande volume de dados. A ADC é um desses métodos, sendo considerada uma das tarefas mais relevantes e desafiadoras no contexto de recuperação de informações [Mendes 2020]. Por exemplo, se considerarmos mensagens de e-mail como documentos, é comum querermos classificar novas mensagens como spam ou não spam. Isso é feito considerando as palavras (termos) presentes na mensagem, geralmente associando um vetor (do tamanho do vocabulário usado) com pesos (ex. frequência do termo) a cada documento. A ADC pode ser adotada em várias outras aplicações reais como sistemas de recomendação, análise de sentimentos, entre outras. A alta dimensionalidade e esparsidade dos conjuntos de dados usados pela ADC (vetores longos e esparsos) tornam essa tarefa desafiadora. Técnicas de aprendizado de máquina são geralmente empregadas para aprender modelos, baseados nas características (termos) dos documentos, que sejam capazes de classificar novos documentos (desconhecidos). Trabalhos recentes defendem o uso da seleção dinâmica do classificador para melhorar a acurácia na ADC [Cruz et al. 2018] e [Britto Jr et al. 2014].

A tarefa de ADC pode ser implementada usando um classificador ou um conjunto de classificadores (*ensemble*) para decidir a classe de um dado documento. Para atingir um melhor desempenho geralmente recorre-se à técnica de conjunto de classificadores, na qual vários algoritmos são combinados de maneira inteligente para obter melhores resultados. Segundo [Cruz et al. 2018] sistemas de múltiplos classificadores (MCS) tem sido amplamente estudados como uma alternativa para aumentar a precisão no reconhecimento de padrões. Uma das abordagens de MCS mais promissoras é a seleção dinâmica (DS), na qual os classificadores são selecionados em tempo de consulta, de acordo com cada novo documento de consulta a ser classificado.

Este trabalho propõe uma nova maneira de se realizar a seleção customizada do método de classificação realizada em tempo de consulta. Além disso, como a ADC requer um alto custo computacional, devido ao tamanho dos conjuntos de dados utilizados, este trabalho discute abordagens paralelas para acelerar o tempo de execução desta tarefa.

2. Seleção Dinâmica do Classificador

A seleção dinâmica (DS) do classificador tornou-se uma pesquisa importante nos últimos anos [Cruz et al. 2018]. As técnicas de DS estimam o nível de competência de cada classificador do conjunto de classificadores. Apenas o mais competente, ou o subconjunto dos classificadores mais competentes, é selecionado para classificar o documento de consulta.

Normalmente, a competência dos classificadores é estimada com base em uma região local do espaço de características (termos) onde o documento de consulta está localizado. Esta região pode ser definida por diferentes métodos, como a aplicação do *k-NN* (k mais próximos), ou agrupamento. Em seguida, estima-se o nível de competência dos classificadores considerando apenas os documentos pertencentes a essa região local e de acordo com algum critério de seleção como a precisão dos classificadores nesta região local ou ranqueamento. O(s) classificador(es) que alcançaram um determinado nível de competência são selecionados.

3. Método Proposto

3.1. Definição do Problema

Considere a tarefa de ADC para M classes $W_1, ..., W_M$. Cada documento é representado por um vetor de caraceterísticas $\mathbf{X}$ e $R(\mathbf{X})$ representa seu rótulo (classe a que pertence). Com L classificadores diferentes $C_i, i = 1, ..., L$ treinados para resolver o problema de ADC, $C_i(\mathbf{X}) \in \{1, ..., M\}$ indica o rótulo (classe) associado ao documento $\mathbf{X}$ pelo classificador C_i. Na seleção dinâmica, a etapa principal é como selecionar o(s) classificador(es) mais competente(s) para um documento de consulta.

3.2. Seleção Customizada do Classificador

Para realizar a tarefa de ADC propomos uma estratégia de seleção customizada do classificador, denominada método *SMART*. Na seleção customizada é feita a escolha de um classificador ou de um conjunto (*ensemble*) de classificadores para cada documento de consulta. Após a seleção, o classificador ou o conjunto de classificadores selecionado será utilizado para classificar o documento de consulta. Considerando os trabalhos [Cruz et al. 2018] e [Britto Jr et al. 2014] e baseados na realização de experimentos iniciais, onde alcançamos melhores resultados com um conjunto de classificadores, nosso método usa a seleção de um conjunto de classificadores. A seguir o método é detalhado.

Considere um conjunto de classificadores C e um conjunto de dados D (dividido em treino, validação e teste). Para cada classificador C_i é feito o treinamento utilizando a base de treino. Em seguida é realizada a etapa de validação com predição dos documentos do conjunto de validação. Para cada documento de validação é definido um vetor $V(\mathbf{X}) = \{C_1(\mathbf{X}), ..., C_L(\mathbf{X})\}$, cada elemento indica se o classificador C_i acertou ou não a predição do documento, $C_i(\mathbf{X}) = 0$, se $C_i(\mathbf{X}) \neq R(\mathbf{X})$; $C_i(\mathbf{X}) = 1$, se $C_i(\mathbf{X}) = R(\mathbf{X})$.

É feito um novo treinamento, usando os conjuntos de treino e validação, para cada um dos classificadores do conjunto. Em seguida, é iniciada a etapa de consulta. Para cada documento de consulta são identificados os k vizinhos mais próximos, no conjunto de validação. Os classificadores, que acertaram a predição de pelo menos um dos k vizinhos, são selecionados e utilizados para fazer a predição do documento de consulta.

Com as predições, nosso método, pode realizar uma das três formas de *ensemble* a seguir: **Votação majoritária** (SMART vm) - escolha da classe com maior votação; **Ponderação por similaridade** (SMART ps) - soma as distâncias entre o documento de consulta e seus vizinhos, para cada classificador, normalizado pelo maior valor acumulado; **Ponderação por acerto** (SMART pa) - calcula o número de predições corretas nos k vizinhos, para cada classificador, normalizado pelo maior valor acumulado.

O que faz o nosso método se diferenciar dos outros métodos que usam seleção dinâmica de classificador(es) é a forma de treinamento. Nosso método faz o treinamento dos classificadores utilizando os conjuntos de dados de treino e validação, antes da etapa de consulta (teste). Os outros métodos fazem o treinamento usando os k vizinhos mais próximos do documento de consulta informado, no momento da consulta.

3.3. Paralelismo

A ADC é uma tarefa que requer um alto custo computacional pois, no caso da abordagem postergada, deve operar no momento da classificação, além disso envolve o uso de grandes conjuntos de dados. Como forma de acelerar o método proposto, foi utilizado o paralelismo em determinadas partes da implementação com o uso do *Scikit-Learn* (biblioteca de aprendizado de máquina para linguagem *Python*). Essa biblioteca possibilita o uso de paralelismo com alteração de parâmetros no momento da chamada de seus métodos.

Outra abordagem testada para acelerar o desempenho do nosso método foi a utilização da biblioteca $RAPIDS\ cuML$ [Team 2018], que implementa o paralelismo de granularidade fina em aceleradores gráficos (*GPU*). Esta abordagem se encontra sendo avaliada mas tem grande potencial para acelerar as operações do nosso método.

3.4. Experimentos

Para avaliar o método SMART consideramos conjuntos de dados variados (tamanho, assimetria, esparsidade e número de classes), de duas tarefas importantes da classificação de documentos, categorização de tópicos e análise de sentimentos, conforme tabela 1.

Parâmetros utilizados nos experimentos: SGD, Logistic Regression, SVM, NB Multinomial, Random Forest e k-NN (Classificadores); SMART vm, SMART ps e SMART pa (métodos); *GridSearch* para selecionar os melhores hiperparâmetros para os classificadores; Os conjuntos de dados foram separados em treino, validação e teste; Utilizada validação cruzada com 10 folds; k = {5, 10, 100, 200} para encontrar os vizinhos do documento de consulta.

Tabela 1. Informações gerais dos conjuntos de dados.

Dataset	# Documentos	# Termos	# Classes	Balanceado
20NG	18.846	49.025	20	Balanceado
REUT90	13.327	17.029	90	Altamente Desbalanceado
4UNI	8.199	22.581	7	Desbalanceado
ACM	24.897	23.110	11	Desbalanceado
YELP_REVIEWS	5.000	21.940	2	Balanceado

Como métricas de avaliação usamos *macroF1* (média aritmética simples da métrica F1 para cada classe do conjunto de dados) e *microF1* (fração dos documentos classificados corretamente). Os resultados obtidos são apresentados na tabela 2. Comparamos as versões do método proposto SMART com o método *MetaLazy* (*baseline*) e com outros métodos descritos na literatura, em especial incluímos o método BERT [Devlin et al. 2018] (do Google) que vem recebendo bastante atenção recentemente.

Tabela 2. Resultados dos experimentos (melhores resultados em negrito).

		20NG	REUT90	4UNI	ACM	YELP_REVIEWS
SMART vm	macroF1	**90.10**	35.95	72.29	68.29	95.00
	microF1	**90.35**	**68.45**	**83.10**	**79.97**	95.00
SMART ps	macroF1	**90.05**	**36.10**	**73.42**	**70.01**	94.95
	microF1	**90.30**	**69.57**	**83.56**	**79.80**	94.95
SMART pa	macroF1	**90.10**	35.89	**73.08**	**69.61**	94.94
	microF1	**90.36**	**69.45**	**83.35**	**79.92**	94.94
MetaLazy	macroF1	**90.49**	**36.46**	71.61	67.66	92.28
	microF1	**90.75**	67.68	82.60	77.50	92.28
BERT	macroF1	84.05	35.37	69.58	59.25	**97.26**
	microF1	84.39	67.57	**83.64**	76.46	**97.26**
SVM	macroF1	88.86	33.55	71.57	67.13	94.32
	microF1	89.01	68.51	80.71	76.80	94.32

4. Considerações Finais

A avaliação dos resultados mostra ganho de acurácia ou resultados muito competitivos tanto em relação ao método *MetaLazy* (baseline) quanto em relação aos outros métodos. Para melhorar os resultados pode-se adotar técnicas de escolha da região de competência do documento de consulta onde o nível de competência do classificador é avaliado.

Pretendemos continuar a investigação do uso de paralelismo com GPUs, para acelerar a tarefa de ADC, e a avaliação do método proposto em conjuntos de dados maiores, como RCV1 E MEDLINE.

Referências

Britto Jr, A. S., Sabourin, R., and Oliveira, L. E. (2014). Dynamic selection of classifiers—a comprehensive review. *Pattern recognition*, 47(11):3665–3680.

Cruz, R. M., Sabourin, R., and Cavalcanti, G. D. (2018). Dynamic classifier selection: Recent advances and perspectives. *Information Fusion*, 41:195–216.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint:1810.04805*.

Mendes, L. F., e. a. (2020). "keep it simple, lazy"– metalazy: a new metastrategy for lazy text classification. *Conference on Information and Knowledge Management – CIKM*.

Team, R. D. (2018). *RAPIDS: Collection of Libraries for End to End GPU Data Science*.