1.03.99 - Ciência da Computação

ANÁLISE DE TÉCNICAS DE PRÉ-PROCESSAMENTO E APRENDIZADO DE MÁQUINA SUPERVISIONADO BASEADO EM UMA ÚNICA CLASSE PARA A CLASSIFICAÇÃO AUTOMÁTICA DE TEXTOS

Marcos Paulo Silva Gôlo[1], Rafael Geraldeli Rossi[2]
1. Estudante de Sistemas de Informação da Universidade Federal de Mato Grosso do Sul (UFMS-CPTL)
2. Professor de Sistemas de Informação da UFMS-CPTL - Orientador

Resumo

Atualmente existe uma quantidade massiva de dados digitais em formato textual, tornando humanamente impossível realizar tarefas como organizar, gerenciar e extrair conhecimento. Normalmente utiliza-se o aprendizado de máquina multi-classe (AMMC) para automatizar tais tarefas [1], na qual o usuário deve conhecer e rotular textos de todas as classes de um problema e, após o treinamento, o algoritmo irá atribuir obrigatoriamente umas das classes informadas no treinamento à um novo texto.

O aprendizado de máquina baseado em uma única classe (AMUC) pode sanar as limitações do AMMC como conhecer e rotular textos de todas as classes [2]. Porém, não há na literatura relatos sobre as melhores técnicas de pré-processamento e de AMUC para classificação automática de textos (CAT). Dado isso, o objetivo é realizar uma avaliação empírica extensa para determinar os algoritmos e técnicas de pré-processamento de textos mais acuradas para a CAT utilizando AMUC supervisionado.

Palavras-chave: mineração de textos; representação estruturada de coleções textuais; sensoriamento de categorias.

Apoio financeiro: CNPq.

Introdução

Grande parte dos dados produzidos nos dias atuais estão em formato textual [3]. Os textos contêm informações valiosas para estudos ou para empresas. Porém, é humanamente impossível extrair informações manualmente desses dados devido aos seus grandes volumes. Para viabilizar tais atividades, pode-se fazer uso da CAT.

A maneira mais viável de realizar a CAT é por meio da utilização de técnicas de aprendizado de máquina (AM). O objetivo do AM é organizar, gerenciar e extrair padrões dos textos [4, 5].

Para que os algoritmos de AM possam processar os textos, é necessário que eles sejam representados em um formato estruturado. A representação no modelo espaço-vetorial (MEV) é a mais utilizada [6]. Neste modelo, cada documento é representado por um vetor, cada dimensão corresponde à uma característica da coleção, e o valor de cada dimensão corresponde ao peso da característica no documento. Vale ressaltar que diferentes esquemas de definição de pesos dos termos bem como das características da coleção textual podem impactar o AM.

Normalmente utiliza-se o AMMC, no qual o usuário deve conhecer e rotular exemplos para todas as classes de um problema na fase de aprendizado, e um novo documento é obrigatoriamente rotulado com uma das classes informadas na fase de classificação. Porém, para diminuir o esforço de rotulação e conhecimento de todas as classes, mesmo que o usuário tenha interesse em uma classe particular, o AMUC pode ser utilizado para viabilizar o uso da classificação automática em situações práticas [7, 8]. No AMUC, o usuário rotula apenas textos de uma classe de seu interesse, e o algoritmo aprende a discriminar tal classe das demais.

Geralmente os trabalhos na literatura envolvem um ou dois algoritmos de AMUC, uma ou duas técnicas de pré-processamento, poucas coleções textuais, sendo geralmente de um domínio específico. Portanto, não há indícios suficientes para determinar qual algoritmo ou técnica é recomendado para um caso geral ou coleções de domínios específicos ou com determinadas características. Dado isso, o objetivo deste trabalho é realizar uma avaliação empírica extensa de forma a determinar os algoritmos e técnicas de pré-processamento de textos mais acuradas para a CAT utilizando AMUC supervisionado.

Metodologia

Foram coletadas 20 coleções textuais de *benchmarking* [9]. As coleções textuais são de diferentes domínios como páginas *web*, artigos de notícias, documentos médicos e científicos, e características, como número de documentos, termos e classes.

Para representar os documentos, foi utilizado o MEV. Foram considerados termos simples como características dos documentos: *bag-of-words* (*bow*). Foram consideradas técnicas de redução de dimensionalidade para agrupar características semanticamente relacionadas: *Latent Dirichlet Allocation (LDA)* [11]*, Probabilistic Semantic Analysis (PLSA)* [12] e *Bisecting k-Means (BkM)* [1]. Além disso, como peso das características na *bow*, foram utilizados: *term frequecy (tf)*, *term-frequency - inverse document frequency (tf-idf)*, ou *binary* [10].

Neste trabalho foram considerados os principais algoritmos de aprendizado baseado em uma única classe: *k-Nearest Neighbor Density-based (kNND), kNN Relative Density-based (kNNRD)*, *k-Means-based*

(KME) [1], e *One-Class Support Vector Machines (OCSVM)* [13]. Além desses, também foi proposta a adaptação do algoritmo *Inductive Model Generation Based on Hetterogeous Network (IMBHN)-based* [17], o qual obteve resultados superiores ao estado-da-arte no aprendizado multi-classe para a classificação de textos.

Os algoritmos de AMUC geram um *score* para um novo documento d_i a ser classificado ($f(d_i)$), i.e., um valor que indique o quão próximo esse documento está de pertencer a classe de interesse ou ser um *outlier,* exceto o *OCSVM* que gera o valor 0 ou 1. Um documento pertencerá à classe de interesse se seu *score* estiver acima de um limiar.

Como limiares de classificação, foram considerados valores manualmente definidos, sendo que *threshold* = $0.05 \times x, x \in \mathbb{N} : 1 \leq x \leq 19$. Também foi utilizada a estratégia *6σ* para definição dos limiares [18]. Neste caso, geram-se os *scores* $f(d_i)$ para todo documento d_i utilizado no treinamento da classe de interesse, calcula-se a média *(μ)* e o desvio padrão *(σ)* de todos os *scores*, e se defini os limiares como *threshold* $\in \{\mu - 3\sigma, \mu - 2\sigma, \mu - 1\sigma, \mu, \mu + 1\sigma, \mu + 2\sigma, \mu + 3\sigma\}$.

Para obter a performance dos algoritmos e compará-los, foi utilizado uma adaptação da estratégia *x-Fold Cross-Validation* [1]. Neste caso, iterativamente cada classe da coleção é considerada como classe de interesse. Cada classe de interesse é dividida em *x* pastas, e em cada uma das *x* repetições, *x-1* pastas são utilizadas para treinamento e a pasta restante é utilizada para teste.

Foi utilizada como performance de classificação a medida F_1 [1,8]:

$$F_1 = \frac{2 * \text{Precisao} * \text{Revocacao}}{\text{Precisao} + \text{Revocacao}} \quad (1)$$

na qual a Precisão é dada pela Equação 2 e a Revocação é dada pela Equação 3.

$$\text{Precisao} = \frac{\text{Verdadeiro Positivo}}{\text{Verdadeiro Positivo + Falso Positivo}} \quad (2)$$

$$\text{Revocacao} = \frac{\text{Verdadeiro Positivo}}{\text{Verdadeiro Positivo + Falso Negativo}} \quad (3)$$

O resultado final da medida F_1 corresponde à uma média das F_1 obtidas em cada iteração do procedimento *x-Fold Cross-Validation* para todas as classes de uma coleção. Os resultados das medidas F_1 considerando as diferentes representações e algoritmos foram comparados utilizando o teste de significância estatística de Friedman com pós teste de Nemenyi [14, 15, 16].

Resultados e Discussão

Nesta seção são exibidos os diagramas de diferença crítica do Teste de Friedman com pós-teste de Nemenyi [16], os quais foram utilizados para comparar as técnicas de pré-processamento e os algoritmos[1].

Depois de extraídos os resultados, os mesmos foram analisados para verificar se os resultados dos algoritmos e das técnicas de pré-processamento são diferentes com significância estatística. Para realizar tal análise foi aplicado o teste de Friedman com pós-teste de Nemenyi.

Nas Figuras 1 e 2 são apresentados os resultados do teste estatístico, por meio de diagramas de diferença crítica[2], respectivamente comparando as técnicas de pré-processamento considerando diferentes algoritmos e os resultados dos algoritmos de AMUC considerando diferentes técnicas de pré-processamento de textos. Observa-se que os resultados dos testes estatísticos apresentados na Figura 1 demonstram que, em geral, a técnica de pré-processamento *LDA* obtém os melhores resultados para os algoritmos de AMUC, uma vez que este obteve o melhor *ranking* médio para a maioria dos algoritmos. Observa-se também que em geral as técnicas de pré-processamento *tf* e *binary* obtêm os piores resultados, independente do tipo de algoritmo utilizado, uma vez que na maioria das comparações obtiveram o pior *ranking* médio.

Nos resultados apresentados na Figura 2, nota-se que, em geral, o algoritmo *kNN Density* obtém os melhores resultados para coleções que utilizaram redução de dimensionalidade, uma vez que este obteve o melhor *ranking* médio para a maioria das técnicas de redução de dimensionalidade. Já o algoritmo *k-Means* obtém os melhores resultados para coleções que não utilizaram redução de dimensionalidade. Observa-se também que, em geral, o algoritmo *OCSVM* obteve os piores resultados independente do tipo de representação utilizado.

Conclusões

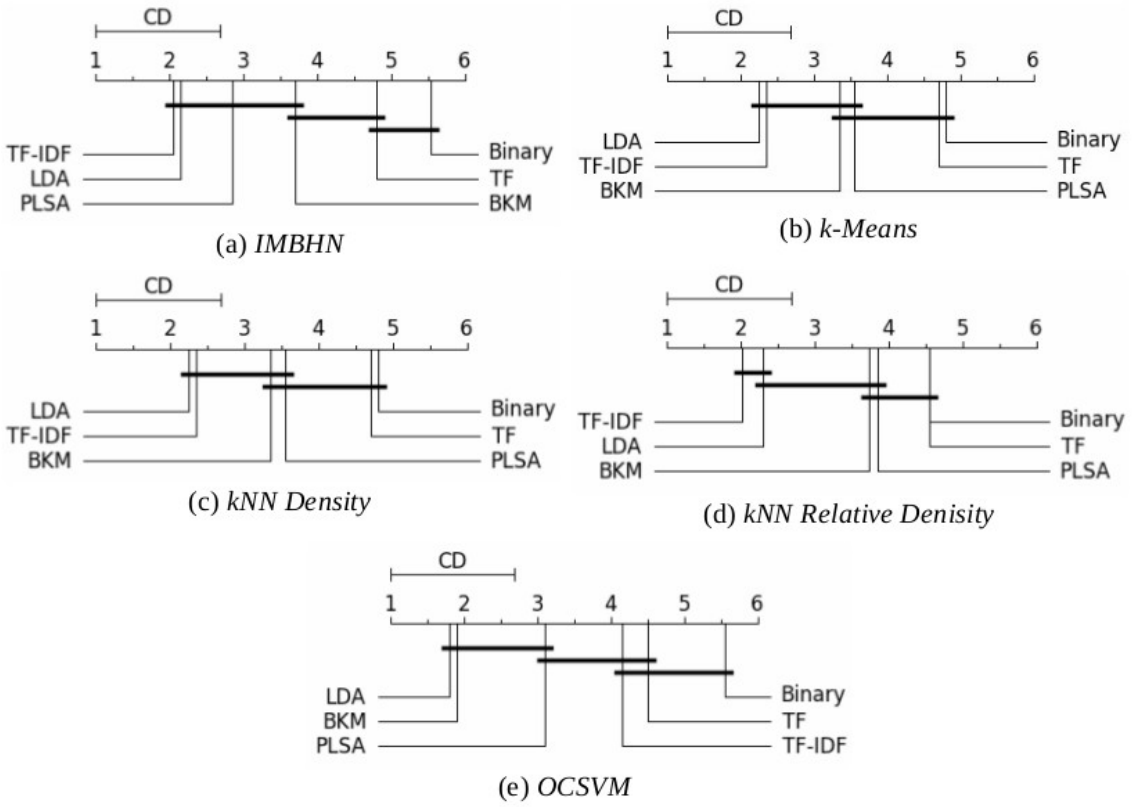
Dado o massivo volume de dados textuais produzidos, dificultando o processo de organizar, gerenciar e extrair conhecimento, a CAT se mostra importante nos dias atuais. Porém, para tornar a aplicação da CAT mais viável na prática, áreas de AM como o aprendizado baseado em uma única classe tem ganhado destaque nos últimos anos.

Neste trabalho foram analisadas as performances de classificação providas por diferentes representações de coleções de textos e pelos principais algoritmos da área de aprendizado supervisionado

[1]As tabelas com todos os resultados utilizados nos teste estatístico estão disponíveis no seguinte endereço: http://gepic.ufms.br/one-class/sbpc_2019/.

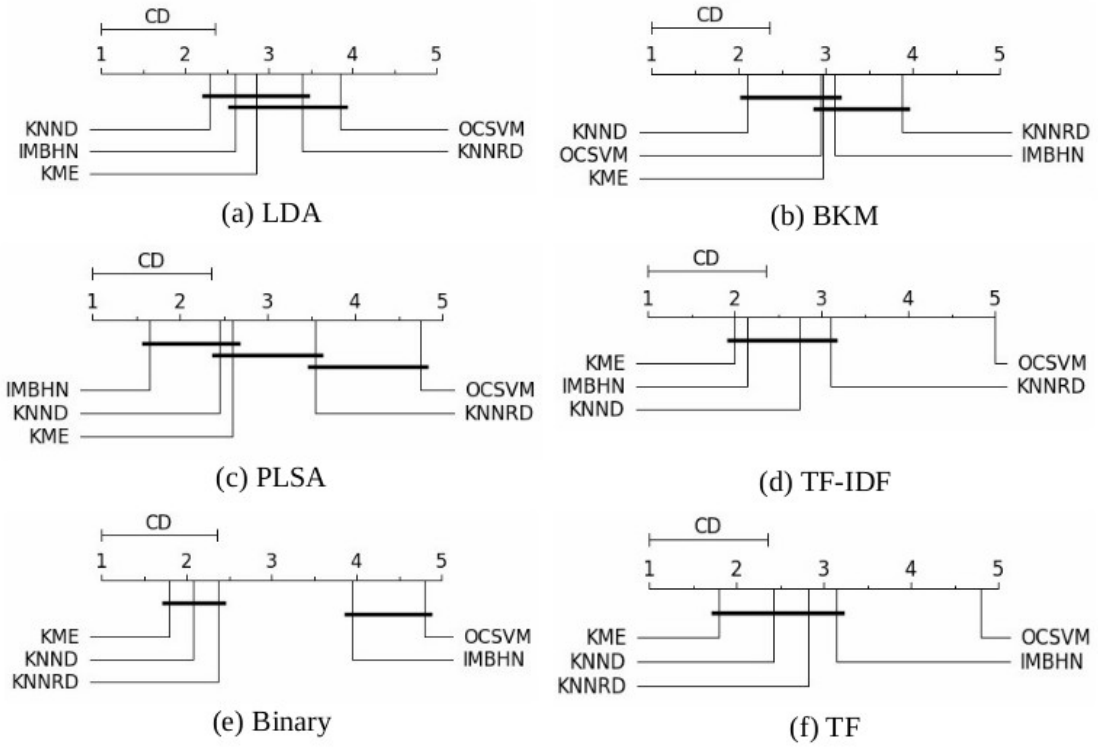
[2]Os diagramas de diferença crítica apresentam os *rankings* médios dos algoritmos e os métodos conectados por uma linha não apresentam diferenças estatisticamente significantes entre si.

Figura 1. Diagramas de diferença crítica do teste de Friedman com pós teste de Nemenyi comparando as técnicas de pré-processamento para cada algoritmo.



Fonte: Minha autoria

Figura 2. Diagramas de diferença crítica do teste de Friedman com pós teste de Nemenyi comparando os algoritmos para cada técnicas de pré-processamento de textos utilizada.



Fonte: Minha autoria.

baseado em uma única classe. Os resultados obtidos, além de gerar conhecimento sobre o impacto de representações e algoritmos na performance de classificação, guiarão o desenvolvimento de pesquisas futuras por meio da indicação de abordagens e algoritmos promissores.

Por exemplo, foram gerados indícios de que representações geradas por meio das técnicas de redução de dimensionalidade *LDA*, ou *bag-of-words* com o peso *tf-ifd* proveem melhores resultados para a maioria dos algoritmos. Além disso, algoritmos baseados em distância, como o *k-Nearest Neighbors* e o *K-Means* obtiveram melhores performances para a maioria das coleções. Vale ressaltar que esses algoritmos são geralmente descartados em análises experimentais da área.

Trabalhos Futuros

Os resultados obtidos neste trabalho servirão como base comparativa para o aprendizado de máquina semissupervisionado baseado em uma única classe. Porém, para saber se o uso de exemplos não rotulados está auxiliando de fato o aprendizado, é necessário comparar técnicas semissupervisionadas com técnicas supervisionadas e verificar se de fato o uso de exemplos não rotulados está de fato impactando positivamente na performance de classificação.

Referências bibliográficas

[1] Tan, P., Steinbach, M., and Kumar, V. (2013). Introduction to Data Mining: Pearson New International Edition. Pearson Education Limited.
[2] Khan, S. S.; MADDEN, e M. G. One-class classification: taxonomy of study and review of techniques. The Knowledge Engineering Review, v. 29, n. 3, p. 345-374, 2014.
[3] Aggarwal, C. C., Zhao, Y., & Philip, S. Y. (2014). On the use of side information for mining text data. IEEE transactions on knowledge and data engineering, 26(6), 1415-1429.
[4] Witten, I. H., Frank, E., and Hall, M. A. (2011). Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition.
[5] Sebastiani, F. (2002). Machine learning in automated text categorization. ACM computing surveys (CSUR), 34(1):1–47.
[6] Salton, G. (1989). Automatic text processing: The transformation, analysis, and retrieval of. Reading: Addison-Wesley.
[7] Kemmler, M., Rodner, E., Wacker, E.-S., and Denzler, J. (2013). One-class classification with gaussian processes. Pattern Recognition, 46(12):3507–3518.
[8] Tax, D. M. J. (2001). One-class classification.
[9] Rossi, R. G., Marcacini, R. M., and Rezende, S. O. (2013). Benchmarking text collections for classification and clustering tasks. Institute of Mathematics and Computer Sciences, University of Sao Paulo.
[10] Rossi, R. G. (2016). Classificação automática de textos por meio de aprendizado de máquina baseado em redes. PhD thesis, Universidade de São Paulo.
[11] Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. Journal of machine Learning research, 3(Jan):993–1022.
[12] Hofmann, T. (1999). Probabilistic latent semantic analysis. In Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence, pages 289–296. Morgan Kaufmann Publishers Inc.
[13] Müller, K.-R., Mika, S., Rätsch, G., Tsuda, K., and Schölkopf, B. (2001). An introduction to kernel-based learning algorithms. IEEE transactions on neural networks, 12(2).
[14] Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. Journal of Machine learning research, 7(Jan):1–30.
[15] Trawinski, B., Smetek, M., Telec, Z., and Lasota, T. (2012). Nonparametric statistical analysis for multiple comparison of machine learning regression algorithms. Applied Mathematics and Computer Science, 22(4):867–881.
[16] García, S., Fernández, A., Luengo, J., and Herrera, F. (2010). Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. Information Sciences, 180(10):2044–2064.
[17] Junior D. D. S., Rossi R. G. (2017). Classificaçao automática de textos utilizando aprendizado supervisionado baseado em uma unica classe. Trabalho de conclusão de curso de Sistemas de Informação UFMS-CPTL.
[18] Muir, A. (2005). Lean Six Sigma Statistics: Calculating Process Efficiencies in Transactional Project. McGraw Hill professional – Six sigma operational methods series.