Adaptação do algoritmo LAX-Reg – *Lean Adaptive XGBoost Regressor* para suportar a regressão de fluxo de dados com mudança de conceito recorrente

Felipe Kenji Nishioka Jojima, Fabiano Baldo

INTRODUÇÃO

O crescimento de dispositivos conectados às redes impulsionou a demanda por algoritmos de aprendizado de máquina capazes de lidar com fluxos de dados ininterruptos e não estacionários (Grando, 2023). Portanto, o algoritmo deve aprender de forma incremental e processar grandes volumes de dados rapidamente (Ikonomovska, Gama e Džeroski, 2015). Além disso, o aprendizado deve refletir as mudanças de conceito na distribuição dos dados, que ocorrem quando as propriedades estatísticas da variável alvo mudam de maneira imprevista (Liao e Wang, 2018). Um algoritmo que se destaca em precisão e velocidade de resposta para esse tipo de tarefa é o LAX-Reg, proposto por Grando (2023). Entretanto, em sua implementação foram utilizadas bibliotecas que tiveram sua atualização descontinuada, como, por exemplo, *scikit-multiflow* (Montiel *et al*., 2018). Portanto, este trabalho deve como objetivo adaptar o LAX-Reg, que é um algoritmo baseado em XGBoost (Chen et al., 2015), para suportar mudanças de conceito recorrentes utilizando novas bibliotecas, como as bibliotecas *River* (Montiel *et al*., 2021) e *Maki Nage* (Picard, 2020).

DESENVOLVIMENTO

Foi feita uma análise e substituição dos métodos que utilizavam a biblioteca *scikit-multiflow* (Montiel *et al*., 2018) do trabalho original pelos métodos da biblioteca *River* (Montiel *et al*., 2021), buscando encontrar métodos equivalentes aos utilizados pela biblioteca antiga. A maior mudança foi a do método "*Evaluate Prequential*", o qual testa o modelo de análise do fluxo de dados e, depois, treina-o com os dados usados no teste. Esse método foi implementado de duas maneiras, sendo a primeira delas utilizando apenas os métodos da biblioteca *River* e a segunda maneira utilizando o framework *Maki Nage* (Picard, 2020), visando melhorar o tempo de execução. Para lidar com a mudança de conceito, foi utilizada o ADWIN – *Adaptive Sliding Window* –, que monitora uma sequência de dados do fluxo usando janelas deslizantes (Barddal, 2019). Essa estratégia foi implementada nos algoritmos LAX-A, que é uma das versões do LAX-Reg (Grando, 2023) e AFXGB (Souza, Grando e Baldo, 2022). A partir da refatoração foram gerados os algoritmos LAX, LAX-T, LAX-A, AFXGB com extensão _R e com extensão _M, representando a implementação *River* e a *Maki Nage*, respectivamente.

RESULTADOS

Para os experimentos, foram considerados os mesmos cenários do trabalho de Grando (2023). Para fins comparativos, como a utilização do framework *Maki Nage* ou da biblioteca *River* não interferem nos resultados de Erro Médio Quadrático (MSE), apenas os resultados do *River* foram considerados. Sobre o MSE, os valores de erro aumentaram em comparação aos obtidos pelo algoritmo original de Grando (2023), com exceção do algoritmo AFXGB_R, como observado na Tabela 1, o que indica um possível gargalo na refatoração dos métodos de avaliação. O teste de Friedman indicou diferenças significativas entre os algoritmos com 95% de confiança, o que permitiu fazer o teste de Nemenyi, apresentado na Figura 1, que mostrou que nenhum algoritmo apresentou um desempenho significativamente diferente dos demais. Sobre o tempo de execução e a memória, houve um aumento de 2 a 3 vezes o valor obtido na implementação original nas duas aplicações, destacando que a implementação com o *Maki*

Nage – que visava um ganho no critério de tempo – apresentou os maiores valores nessa métrica, sendo de 10 a 50 milissegundos mais lento que a adaptação do *River*.

CONSIDERAÇÕES FINAIS

As adaptações dos algoritmos para bibliotecas mais recentes, com exceção do AFXGB, apresentaram uma queda no desempenho, enquanto a implementação do ADWIN nos algoritmos LAX mostrou uma melhora em relação aos demais algoritmos. Esses resultados mostram a necessidade de um aprimoramento nos métodos substituídos para torná-los mais competitivos, verificando possíveis gargalos presentes nos métodos de avaliação.

Palavras-chave: refatoração, fluxo de dados, regressão, mudança de conceito, xgboost.

ILUSTRAÇÕES

Tabela 1. *Comparativo entre os valores de MSE dos algoritmos LAX, LAX-T, LAX-A e AFXGB nas execuções pelo Scikit-Multiflow e pelo River.*

Dataset	LAX	LAX-T	LAX-A	AFXGB	LAX_R	LAX-T_R	LAX-A_R	AFXGB_R
metro	1738735$^{(2)}$	***1536842***$^{(1)}$	2885753$^{(7)}$	2009869$^{(5)}$	2600338$^{(6)}$	1750625$^{(3)}$	3984337$^{(8)}$	1813253$^{(4)}$
waze	2,810$^{(4)}$	2,890$^{(5)}$	2,790$^{(3)}$	3,350$^{(8)}$	3,034$^{(6)}$	3,086$^{(7)}$	***2,314***$^{(1)}$	2,393$^{(2)}$
bikes	589$^{(6)}$	732$^{(7)}$	2683$^{(8)}$	360$^{(4)}$	273$^{(3)}$	523$^{(5)}$	***151***$^{(1)}$	243$^{(2)}$
barra	***351***$^{(1)}$	355$^{(2)}$	439$^{(3)}$	594$^{(5)}$	922$^{(8)}$	484$^{(4)}$	625$^{(7)}$	606$^{(6)}$
costa	***163***$^{(1)}$	172$^{(2)}$	207$^{(3)}$	316$^{(6)}$	426$^{(7)}$	260$^{(4)}$	567$^{(8)}$	305$^{(5)}$
fried1(A)	78$^{(5)}$	33$^{(2)}$	175$^{(8)}$	49$^{(4)}$	172$^{(6)}$	34$^{(3)}$	174$^{(7)}$	***32***$^{(1)}$
fried1(G)	82$^{(5)}$	***36***$^{(1)}$	135$^{(6)}$	57$^{(4)}$	182$^{(7)}$	41$^{(2)}$	189$^{(8)}$	44$^{(3)}$
reg(A)	***3987***$^{(1)}$	5022$^{(4)}$	4728$^{(3)}$	5208$^{(6)}$	7316$^{(7)}$	5028$^{(5)}$	8113$^{(8)}$	4322$^{(2)}$
reg(G)	***4173***$^{(1)}$	5194$^{(3)}$	4906$^{(2)}$	5520$^{(5)}$	7824$^{(7)}$	6038$^{(6)}$	8544$^{(8)}$	5280$^{(4)}$

Fonte: O autor.

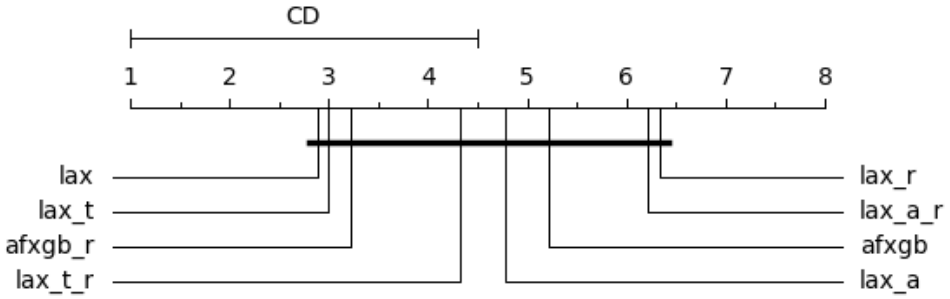

Figura 1. *Teste de Nemenyi das classificações médias dos algoritmos para o MSE.* Fonte: O autor.

REFERÊNCIAS BIBLIOGRÁFICAS

BARDDAL, J. P. Vertical and horizontal partitioning in data stream regression ensem bles. In: IEEE. **2019 International Joint Conference on Neural Networks (IJCNN)**. Curitiba, BR, 2019. p. 1–8.

CHEN, T. et al. Xgboost: extreme gradient boosting. **R package version 0.4-2**, v. 1, n. 4, p. 1–4, 2015.

GRANDO, Julia. **Adaptação ao XGBoost para suporte à regressão de fluxo de dados não-estacionários**. 60 p. Joinville, 2023. Mestrado em computação aplicada – Universidade do Estado de Santa Catarina, Centro de Ciências Tecnológicas. 04 jul. 2023.

HALFORD, M.; BOLMIER, G.; SOURTY, R.; VAYSSE, R.; ZOUITINE, A. **Creme, a Python library for online machine learning**. 2019. Disponível em: <https://github.com/MaxHalford/creme>. Acesso em: 16 ago. 2025.

IKONOMOVSKA, E.; GAMA, J.; DžEROSKI, S. Online tree-based ensembles and op tion trees for regression on evolving data streams. **Neurocomputing**, v. 150, p. 458 470, 2015. ISSN 0925-2312. Special Issue on Information Processing and Machine Learning for Applications of Engineering Solving Complex Machine Learning Problems with Ensemble Methods Visual Analytics using Multidimensional Projections. Disponível em: < https://www.sciencedirect.com/science/article/abs/pii/S0925231214012338>.

LIAO, Z.; WANG, Y. Rival learner algorithm with drift adaptation for online data stream regression. In: **Proceedings of the 2018 International Conference on Algorithms, Computing and Artificial Intelligence**. New York, NY, USA: Association for Com puting Machinery, 2018. (ACAI 2018). ISBN 9781450366250. Disponível em: < https://dl.acm.org/doi/10.1145/3302425.3302475 >.

MONTIEL, J.; READ, J.; BIFET, A.; ABDESSALEM, T. Scikit-multiflow: A multi-output streaming framework. **The Journal of Machine Learning Research**, 19, 72, p. 1-5, 2018. Disponível em: <http://jmlr.org/papers/v19/18-251.html>. Acesso em: 16 ago. 2025.

MONTIEL, J.; HALFORD, M.; MASTELINI, S. M.; BOLMIER, G.; SOUTRY, R.; VAYSSE, R.; ZOUITINE, A.; GOMES, H. M.; READ, J.; ABDESSALEM, T.; BIFET, A. River: machine learning for streaming data in Python. **The Journal of Machine Learning Research**, 22, 110, p. 1-8, 2021. Disponível em: <http://jmlr.org/papers/v22/20-1380.html>. Acesso em: 16 ago. 2025.

PICARD, R. **Maki Nage:** Stream Processing made Easy. 2021. Disponível em: <https://www.makinage.org/>. Acesso em: 16 ago. 2025.

SOUZA, F. M. de; GRANDO, J.; BALDO, F. Adaptive fast XGBoost for regression. In: XAVIER-JUNIOR, J. C.; RIOS, R. A. (Ed.). Intelligent Systems. Cham: Springer International Publishing, 2022. p. 92–106. ISBN 978-3-031-21686-2

DADOS CADASTRAIS

BOLSISTA: Felipe Kenji Nishioka Jojima
MODALIDADE DE BOLSA: PROBIC/UDESC (IC)
VIGÊNCIA: 09/2024 a 08/2024 – Total: 12 meses
ORIENTADOR(A): Fabiano Baldo
CENTRO DE ENSINO: CCT
DEPARTAMENTO: DEPARTAMENTO DE CIENCIA DA COMPUTACAO CCT
ÁREAS DE CONHECIMENTO: Ciências Exatas e da Terra/ Ciência da Computação
TÍTULO DO PROJETO DE PESQUISA: StreamMining v.3.0 Novas Abordagens para Algoritmos de Aprendizagem em Fluxos de Dados Não Estacionários
Nº PROTOCOLO DO PROJETO DE PESQUISA: NPP3216-2023