IV Workshop de Matemática, Estatística e Computação Aplicadas à Indústria

# Um pipeline de aprendizado robusto a ruído de classe

Lucas de Angelis Oliveira[1], Moacir Antonelli Ponti[2]
Instituto de Ciências Matemáticas e de Computação - USP

## 1 Introdução

Ao se treinar modelos de aprendizado de máquina é essencial utilizar um conjunto de dados que possua um padrão de qualidade [1]. Um desses parâmetros de qualidade se refere ao quão confiáveis são os rótulos de uma base de dados. Em cenários de treinamento de modelos de aprendizado de máquina reais é comum que *datasets* apresentem uma certa porcentagem de casos com rótulo incorreto, o que representa um desafio no treinamento de modelos.

Para lidar com isso os autores em [2] modificaram a função de perda do algoritmo *AdaBoost* para torná-lo menos suscetível ao sobreajuste do ruído de classe. Já em [3] foi utilizada uma combinação de *Unsupervised Learning* com *Semi-Supervised Learning* para determinar um conjunto reduzido de amostras "limpas" e aumentá-lo com um outro conjunto de amostras não rotuladas. Embora ambos os métodos apresentem bons resultados experimentais, o primeiro não ataca o problema de desbalanceamento de classes, enquanto que o segundo foi aplicado somente a conjuntos de dados baseados em imagens.

O objetivo deste projeto de pesquisa é desenvolver um *pipeline* para treinamento de modelos de aprendizado de máquina que seja robusto ao ruído de rótulo e também possa endereçar o problema de desbalanceamento de classes.

## 2 Metodologia

Este *pipeline* será composto de três passos principais: primeiro a subamostragem de casos de fácil aprendizado, seguida por uma etapa de remoção de instâncias ruidosas e, por fim, uma etapa de otimização de hiperparâmetros para definição do melhor modelo para o conjunto de dados especificado.

[1]lucas.angelis.oliveira@usp.br
[2]moacir@icmc.usp.br

## 2.1 Conjuntos de Dados

Para os experimentos foram selecionados três conjuntos de dados públicos de classificação binária: o conjunto *Adult*[3], o conjunto *Breast Cancer Wisconsin (Diagnostic)*[4] e o conjunto *Bank Account Fraud Dataset Suite (NeurIPS 2022)*[5]. Para o caso do conjunto *Bank Fraud* foram selecionadas duas variações: o conjunto base e a variação IV, que introduz um viés temporal na proporção de casos positivos. Para a adequação ao protocolo experimental todos os conjuntos passaram primeiro por uma etapa de separação entre treino, validação e teste. Para os *datasets Adult* e *Breast Cancer* essa separação foi feita de forma aleatória, enquanto que para o *Bank Fraud* se preferiu utilizar uma separação por período de tempo. Em seguida cada base passou por uma etapa de adição artificial de ruído, onde se alterou o rótulo de uma proporção determinada de instâncias segundo a distribuição de classes. Para simular cenários reais e avaliar a performance do *pipeline* em diferentes situações, foram determinadas três proporções a serem aplicadas: 10%, 20% e 30% de ruído. Essa alteração de rótulo foi aplicada somente aos conjuntos de treinamento e validação, deixando o conjunto de teste intacto.

## 2.2 Subamostragem de instâncias

O primeiro passo do *pipeline* consiste em realizar uma subamostragem do conjunto de dados baseado em métricas da dinâmica de treinamento de um modelo. Para isso foi utilizada a técnica de Cartografia desenvolvida em [4] e adaptada para ser utilizada em modelos baseados em *Gradient Boosting* em [5]. É possível, utilizando essa técnica, selecionar quais instâncias do treinamento um determinado modelo consegue aprender com maior facilidade, quais apresenta maior dificuldade e quais ficam em uma região intermediária.

## 2.3 Detecção de ruído de classe

Na etapa de detecção de ruído de rótulo foram utilizadas as técnicas de Cartografia [4, 5] e *SimiFeat* [6], um algoritmo que não utiliza predição de um modelo de aprendizado de máquina, mas uma análise do espaço vetorial das características das bases de dados.

Para se detectar instâncias ruidosas utilizando a técnica de Cartografia é necessário primeiramente calcular uma métrica que consiste na multiplicação de dois valores retornados pelo algoritmo, a *correctness* e a *confidence*. O primeiro valor diz respeito à porcentagem de vezes que o modelo acertou a classificação desta instância durante o treinamento, enquanto que a segunda se refere ao valor médio da predição do modelo durante as iterações. Após a multiplicação desses valores, pode-se comparar cada um a um limiar especificado e assim definir quais registros são possivelmente ruidosos na base, o que configura o método *"Cartography Hard Threshold"* [5]. Também é possível designar um peso inicial para cada instância e iterativamente atualizá-los em relação ao resultado da multiplicação do *correctness* e *confidence*. A esse método dá-se o nome de *"Cartography Weights"* [5].

Utilizando-se a técnica *SimiFeat* é possível detectar observações com ruído por meio de dois algoritmos, um baseado em votação e outro baseado em ranqueamento. No primeiro método é

[3]Disponível em https://archive.ics.uci.edu/dataset/2/adult

[4]Disponível em https://archive.ics.uci.edu/dataset/17/breast+cancer+wisconsin+diagnostic

[5]Disponível em https://www.kaggle.com/datasets/sgpjesus/bank-account-fraud-dataset-neurips-2022

gerada uma vizinhança para cada instância, verifica-se quais são os rótulos para cada elemento e, ao final, é definido se a observação é ruidosa ou não dada a concordância de seu rótulo com o resultado da votação. Para o segundo método é calculado um *score* que quanto maior seu valor, mais "limpa" é a instância e quanto menor, mais ruidosa. Cria-se um ranqueamento com esse valor e é estimada a proporção de casos ruidosos nesse conjunto de dados para se estabelecer um corte e, por fim, determinar quais são os casos com rótulo discrepante.

### 2.4 Protocolo experimental

Inicialmente foi realizado um ranqueamento dos métodos de detecção de ruído para determinar os que possuem melhor performance nessa tarefa. Além de cada um dos métodos citados anteriormente, também foram experimentadas combinações utilizando operações lógicas *AND* e *OR*. Para cada base de dados e cada nível de ruído foi tomado o conjunto de treinamento e aplicado cada um dos métodos citados para detectar as observações ruidosas, aferindo-se as seguintes métricas para comparação: acurácia, erro de falso ruído detectado e proporção de ruído não detectado.

Definidos os melhores métodos para identificação de ruído de classe, foram executados os seguintes passos para cada conjunto de dados:

- Carrega-se a base com uma determinada proporção de ruído adicionado (10%, 20% ou 30%)
- Gera-se métricas de referência com um modelo com hiperparâmetros otimizados
- Ajusta-se a subamostragem de casos fáceis usando o conjunto de validação
- Com o melhor resultado do item anterior aplica-se a detecção de ruído
- Realiza-se uma otimização de hiperparâmetros e determina-se um modelo vencedor
- Otimiza-se do *threshold* de classificação maximizando o F1-Score no conjunto de validação
- Avalia-se as métricas no conjunto de teste com o modelo gerado

## 3 Resultados

Na primeira etapa dos experimentos foram determinados os melhores métodos para detectar instâncias ruidosas: *"SimiFeat Voting AND Cartography Hard Threshold"*, *"SimiFeat Voting AND Cartography Weights"* e *"SimiFeat Ranking AND Cartography Weights"*.

Utilizando-se esses métodos na configuração experimental foi possível executar o *pipeline* completo para todas as bases de dados, com todos níveis de ruídos estipulados e obter métricas de classificação. Na tabela 1 é possível ver os resultados para um experimento onde foi utilizado o conjunto *Bank Fraud Base*. Pode-se notar que, com exceção da métrica de revocação, os melhores valores foram obtidos por meio da execução da sequência de processos para treinamento robusto a ruído de rótulo. Deve-se destacar que os valores baixos para as métricas de precisão, F1-Score e PR-AUC (*Precision-Recall Area Under Curve*) em todos os experimentos se devem ao caráter de extremo desbalanceamento presente neste *dataset*, bastante comum em problemas de detecção de fraude.

Tabela 1: Resultados experimento *dataset Bank Fraud Base* 10% ruído

| Pipeline | Precision@Test | Recall@Test | F1-Score@Test | PR-AUC@Test |
|---|---|---|---|---|
| Baseline | 0.014 | **0.999** | 0.029 | 0.200 |
| SimiFeat V AND Cart H | 0.090 | 0.673 | 0.160 | **0.202** |
| SimiFeat V AND Cart W | 0.099 | 0.665 | 0.173 | **0.202** |
| SimiFeat R AND Cart W | **0.171** | 0.447 | **0.254** | 0.196 |

# 4 Conclusões e próximos passos

A partir dos resultados apresentados na seção anterior, pode-se dizer que essa metodologia para treinar modelos de aprendizado de máquina na presença de observações com ruído de rótulo proporciona um aumento em algumas métricas de classificação binária. Essa melhora foi observada principalmente nas bases que apresentam grande desbalanceamento de classes, o que motivou a dedicação deste projeto para esse tipo de *dataset* nos seguintes passos. Também se mostra necessária uma melhor calibração dos métodos *SimiFeat*. Essa calibração possibilitaria detectar observações ruidosas com maior precisão, o que resulta em melhor desempenho do *pipeline* como um todo. Por fim, se mostra necessário também adicionar um cenário de 5% de ruído para simular um cenário de pouco ruído.

# Referências


[1] Mello, R.F. and Ponti, M.A. Machine Learning: A Practical Approach on the Statistical Learning Theory, Springer, 2018.

[2] Q. Miao, Y. Cao, G. Xia, M. Gong, J. Liu and J. Song. RBoost: Label Noise-Robust Boosting Algorithm Based on a Nonconvex Loss Function and the Numerically Stable Base Learners. in IEEE Transactions on Neural Networks and Learning Systems, vol. 27, no. 11, pp. 2216-2228, Nov. 2016, doi: 10.1109/TNNLS.2015.2475750

[3] F. R. Cordeiro, R. Sachdeva, V. Belagiannis, I. Reid, and G. Carneiro. LongReMix: Robust Learning with High Confidence Samples in a Noisy Label Environment. Pattern Recognition, vol. 133, p. 109013, 2023. doi: 10.1016/j.patcog.2022.109013

[4] S. Swayamdipta, R. Schwartz, N. Lourie, Y. Wang, H. Hajishirzi, N. A. Smith, and Y. Choi. Dataset Cartography: Mapping and Diagnosing Datasets with Training Dynamics. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). DOI 10.18653/v1/2020.emnlp-main.746.

[5] M. A. Ponti, L. de A. Oliveira, M. Esteban, V. Garcia, J. M. Román, and L. Argerich. Improving Data Quality with Training Dynamics of Gradient Boosting Decision Trees. arXiv preprint arXiv:2210.11327, 2024.

[6] Z. Zhu, Z. Dong, and Y. Liu. Detecting Corrupted Labels without Training a Model to Predict. In: Proceedings of the International Conference on Machine Learning (ICML), pp. 27412–27427, 2022. PMLR.