Uma Ferramenta de Correção de Rótulos com Base em Curvas Principais

Max Deivid do Nascimento
dept. de Automática (DAT)
Universidade Federal de Lavras (UFLA)
Lavras, Brasil
maxnasc23@gmail.com

Danton Diego Ferreira
dept. de Automática (DAT)
Universidade Federal de Lavras (UFLA)
Lavras, Brasil
danton@ufla.br

Flavio Bezerra Costa
Faculty Directory
Michigan Tech
Michigan, United States
fbcosta@mtu.edu

Erivelton Nepomuceno
Faculty of Science and Engineering
National University of Ireland Maynooth
Maynooth, Ireland
erivelton.nepomuceno@mu.ie

Marcin Kamiński
Department of Electrical Machines, Drives and Measurements
Wrocław University of Science and Technology
Wrocław, Poland
marcin.kaminski@pwr.edu.pl

Henrique Luiz Moreira Monteiro
ICTI - Instituto de Ciência, Tecnologia e Inovação
Universidade Federal de Lavras (UFLA)
São Sebastião do Paraíso, Brasil
henrique.monteiro@ufla.br

***Resumo*—Em aprendizado supervisionado, erros nos rótulos comprometem significativamente o desempenho dos modelos. Este trabalho propõe um novo algoritmo baseado em Curvas Principais para a detecção e correção de rótulos incorretos, com o objetivo de melhorar a qualidade dos dados de treinamento. A abordagem desenvolvida é composta por duas etapas: (i) identificação de amostras potencialmente mal rotuladas por meio de análise utilizando o *Local Outlier Factor* (LOF) (ii) correção baseada na consistência estrutural dos dados utilizando Curvas Principais. O método foi comparado ao *Confident Learning* em múltiplos *datasets* com taxas de erro controladas. Os experimentos demonstraram que, dependendo do conjunto de dados analisado, a técnica desenvolvida reduziu o erro total do conjunto em 9 vezes mais que o *Confident Learning*. Além disso, essa técnica pode auxiliar na detecção do comportamento dos dados, considerando as densidades de cada conjunto ou classe, permitindo sua caracterização, classificação de padrões e agrupamentos. Esses resultados sugerem que Curvas Principais oferecem vantagens relevantes na purificação de dados.**

***Palavras chave*—Curvas principais, outliers, erros em rótulos, análise de dados**

I. Introdução

Em tarefas de aprendizado supervisionado, é essencial que os dados utilizados no treinamento estejam corretamente rotulados, uma vez que o desempenho dos modelos depende diretamente da qualidade da informação fornecida. No entanto, em muitos conjuntos de dados do mundo real, é comum a presença de rótulos incorretos — conhecidos como *erros de rótulo* (*label noise*) — que surgem devido a falhas humanas, ambiguidades no processo de anotação ou limitações em sistemas automáticos de rotulagem [1]. Como demonstrado em [2], técnicas como *bootstrapping* podem ser eficazes para mitigar esses problemas em redes neurais profundas.

Outro aspecto importante na análise de dados é a distinção entre *inliers* e *outliers*. *Inliers* são observações que seguem o padrão predominante dos dados, estando dentro de uma região de alta densidade da distribuição. Já os *outliers* são pontos que se desviam significativamente do comportamento geral do conjunto, podendo indicar ruído, anomalias, ou até mesmo exceções legítimas, conforme mostra a Figura 1. O erro de rótulo, problema foco deste trabalho, fica evidente na figura 1. Observe que o erro de rótulo da Classe A (Azul) está dentro da região de *inliers* da Classe B (Vermelha) e, portanto, é visto como um *outlier* da Classe A (Azul). O mesmo pode ser entendido para o erro de rótulo da Classe B (Vermelha). Neste contexto, a detecção de *outliers* torna-se relevante na tarefa de identificação de erro de rótulo, além de ser uma tarefa fundamental tanto para melhorar a qualidade dos dados quanto para evitar que eles causem interferências indesejadas durante o processo de modelagem. Técnicas como o *Isolation Forest* [3], o *LOF* [4] e o One Class Principal Curves (OCPC) [5] são exemplos de métodos dedicados a essa tarefa.

Quando erros de rótulo ocorrem em instâncias que, do ponto de vista estrutural, seriam consideradas *inliers*, o problema torna-se ainda mais delicado, pois tais erros podem passar despercebidos por técnicas tradicionais de detecção de anomalias. Já os *outliers* rotulados incorretamente também podem induzir o modelo a aprender padrões incorretos, aumentando a variância e reduzindo a capacidade de generalização.

Para mitigar esses problemas, diversas abordagens têm sido propostas na literatura, incluindo algoritmos robustos ao ruído, estratégias de reamostragem como o método *Co-teaching* [6] e técnicas de filtragem de rótulos, como o *Learning to Filter Noisy Labels with Self-Ensembling* (SELF) [7], além

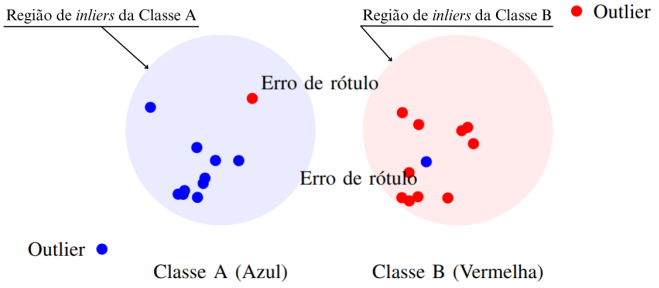

Fig. 1. Representação visual de inliers (pontos internos), outliers (pontos distantes) e erros de rótulo (pontos com cor incompatível com o agrupamento).

de métodos de validação cruzada baseados em múltiplos classificadores. Tais métodos de correção de rótulos possuem limitações relacionadas às características dos dados: exigem que o erro esteja balanceado entre as classes, com cada conjunto de dados de uma classe tendo aproximadamente a mesma quantidade de erros em rótulos, são restritos a modelos básicos como classificadores lineares ou *K-Nearest Neighbors* (KNN) e a condições iniciais específicas das técnicas e, principalmente, apresentam alto custo computacional.

Este trabalho se insere nesse cenário, propondo uma abordagem inovadora que explora a boa capacidade das Curvas Principais [8] de representar dados com distribuição não linear, ou seja, distribuições que não seguem padrões simples e bem conhecidos, como a distribuição normal ou gaussiana. Em resumo, a abordagem proposta explora a estrutura geométrica dos dados em alta dimensão para detectar e sugerir correções de possíveis erros de rotulagem.

As Curvas Principais se ajustam aos dados de forma flexível, a partir de parâmetros definidos pelo usuário como o comprimento e número de segmentos. Essa flexibilidade permite que dados em alta dimensão e com distribuições complexas no espaço de *features* possam ser representados em uma única dimensão. As Curvas Principais têm sido aplicadas para diferentes cenários envolvendo reconhecimento de padrões com aprendizado supervisionado e não supervisionado [5], [9], [10].

O *Confident Learning* é uma técnica consolidada utilizada para identificar e corrigir rótulos incorretos em conjuntos de dados, baseando-se na estimativa da incerteza dos rótulos. No contexto do artigo, essa metodologia serve como um ponto de comparação para avaliar a eficácia do novo algoritmo proposto para a detecção e correção de erros de rotulagem. Ambas as técnicas foram aplicadas em diversos conjuntos de dados com taxas de erro controladas (e.g., 10% de erro induzido) para comparar seu desempenho.

II. Curvas Principais

As curvas principais consistem em uma generalização não linear do conceito de Análise de Componentes Principais (PCA). Elas foram definidas inicialmente por Hastie e Stuetzle [8] como curvas unidimensionais parametrizadas com a propriedade de auto-consistência, que atravessam os pontos de dados no espaço de dados original.

Seja $g(t)$ uma curva suave (infinitamente diferenciável) em $\mathbb{R}^d$, parametrizada por $t \in \mathbb{R}$ e, para qualquer $x \in \mathbb{R}^d$, seja $t_g(x)$ o maior valor de parâmetro t para o qual a distância entre x e $g(t)$ é minimizada.

$$g(t) = [g_1(t), g_2(t), \ldots, g_d(t)]^T \tag{1}$$

em que T indica a transposição da matriz. Mais formalmente, o índice de projeção $t_g(x) : \mathbb{R}^d \to \mathbb{R}$ é definido por:

$$t_g(x) = \sup\{t : \|x - g(t)\| = \inf \|x - g(\mu)\|\} \tag{2}$$

em que $\|\cdot\|$ representa a norma Euclidiana em $\mathbb{R}^d$ e μ é uma variável auxiliar definida em $\mathbb{R}$.

De acordo com Hastie e Stuetzle (1989), a curva suave $g(t)$ é uma curva principal se: (i) $g(t)$ não se auto-intercepta, (ii) $g(t)$ tem comprimento finito dentro de qualquer subconjunto limitado de $\mathbb{R}^d$, e (iii) g é auto-consistente, ou seja, $g(t) = \mathbb{E}(X \mid t_g(X) = t)$ para todo $t \in \mathbb{R}$. Intuitivamente, se $g(t)$ é o valor médio (segundo a distribuição de X) das observações com projeção em t, e isso vale para todo t, então g é uma curva principal. Assim, as curvas principais podem ser consideradas como curvas que passam pelo "meio" da nuvem de dados no espaço original de *features*, gerando uma representação não linear e unidimensional dos dados.

Métodos alternativos para estimar curvas principais foram introduzidos após o trabalho pioneiro de Hastie e Stuetzle, incluindo o algoritmo de k-segmentos proposto em [11], que foi utilizado neste trabalho por ser mais robusto e apresentar convergência garantida.

A. Discussão sobre Suavidade das Curvas

Embora as Curvas Principais sejam definidas como curvas suaves $g(t) \in \mathbb{R}^d$, com continuidade infinita e auto-consistência $g(t) = \mathbb{E}(X \mid t_g(X) = t)$, o algoritmo de k-segmentos utilizado neste trabalho constrói inicialmente uma linha poligonal por meio da junção de segmentos lineares. Assim, a suavidade no sentido clássico não é garantida a priori, sendo possível apenas por meio de etapas adicionais de suavização.

A ausência de suavidade pode impactar aplicações que exigem continuidade de primeira ou segunda ordem, como análise de trajetória, modelagem física e estimação de derivadas. No entanto, na tarefa específica de correção de rótulos com base na menor distância euclidiana entre pontos e curvas, a suavidade não é essencial. A aproximação segmentada oferece simplicidade computacional e desempenho satisfatório, especialmente em conjuntos de dados com estrutura bem definida.

B. O Algoritmo de k-Segmentos

O algoritmo de k-segmentos primeiro localiza k segmentos de reta diferentes no conjunto de dados, e depois conecta esses segmentos para formar uma linha poligonal. Esta linha serve como uma primeira aproximação da curva principal, podendo

ser suavizada posteriormente. O processo de extração da curva principal é realizado em três etapas:

1) **Inserção do primeiro segmento:** Considerando todos os eventos do conjunto de dados, define-se um centro correspondente ao valor médio dos dados. A partir desse centro, obtém-se o primeiro segmento na direção da primeira componente principal com comprimento igual ao desvio padrão associado a essa componente.
2) **Inserção de novos segmentos:** Um novo ponto é definido como centro e, com uma versão modificada do algoritmo k-means, definem-se os eventos que formarão o novo agrupamento. Este agrupamento é baseado em regiões de Voronoi[1]. O primeiro segmento é recalculado, pois o conjunto de eventos que o define foi alterado. Os segmentos obtidos são unidos por linhas retas.
3) **Critério de parada:** O algoritmo pára quando atinge o número máximo de segmentos definido pelo usuário ou quando ocorre convergência, que se dá quando o maior agrupamento possível possui menos de três eventos.

Este processo é ilustrado na Figura 2, onde se pode observar a inserção e o refinamento dos segmentos que formam a curva principal de maneira iterativa.

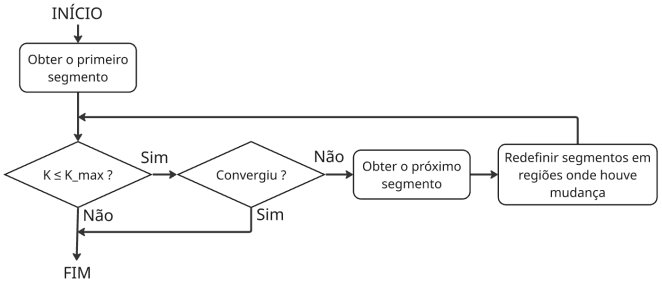

Fig. 2. Diagrama em blocos do algoritmo de k-segmentos para extração de curvas principais.

III. Algoritmo Proposto

O algoritmo foi projetado para ser leve, rápido, mais sustentável e o mais automatizado possível, a fim de facilitar seu uso em diferentes contextos.

Basicamente, o algoritmo proposto se divide em duas etapas: (i) detecção de rótulos errados e (ii) correção dos rótulos errados. O processo completo é ilustrado na Figura 3.

A. Detecção de Rótulos Incorretos

A etapa de detecção de rótulos incorretos fundamenta-se no algoritmo *Local Outlier Factor* (LOF) [4], selecionado por sua eficácia na detecção de *outliers* em conjuntos de dados com variações de densidade. Proposto por Breunig et al. em [4], o LOF identifica uma instância como *outlier* se sua densidade local for significativamente menor que a de seus vizinhos mais próximos.

[1]Dado um conjunto finito de pontos $\{p_1, p_2, \ldots, p_k\}$ em $\mathbb{R}^n$, a região de Voronoi $V(p_i)$ associada a p_i é o conjunto de todos os pontos $x \in \mathbb{R}^n$ tais que $d(x, p_i) \leq d(x, p_j)$ para todo $j \neq i$, onde d é a distância euclidiana [12]

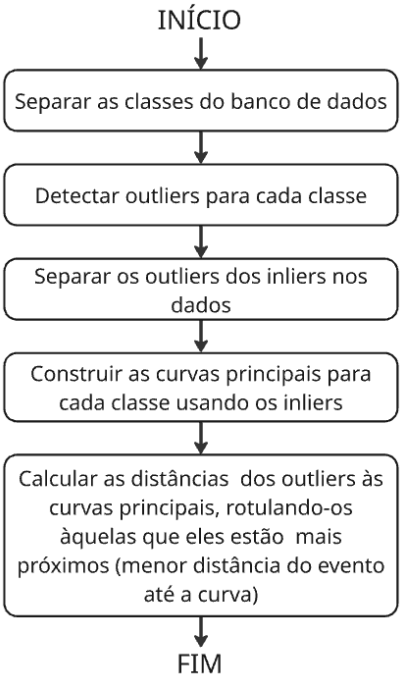

Fig. 3. Fluxo do método proposto

Para formalizar o LOF, definem-se as seguintes métricas-chave para um ponto p e seus k vizinhos:

1) **Distância k-ésimo vizinho (*k-distance*)**: É a menor distância $d(p, o)$ tal que p tenha pelo menos k pontos a essa distância ou menos.
2) **Conjunto de Vizinhos Alcançáveis k-distância ($N_k(p)$)**: Inclui todos os pontos o' cuja distância de p é menor ou igual à k-distância(p).
3) **Distância de Alcançabilidade (*reach-dist*$_k(p, o)$)**: $\max\{k\text{-}distance(o), d(p, o)\}$. Garante que a distância entre p e o seja no mínimo a k-distância de o.
4) **Densidade de Alcançabilidade Local (LRD)**: Inverso da média das distâncias de alcançabilidade de p para seus vizinhos:

$$LRD_k(p) = \frac{|N_k(p)|}{\sum_{o \in N_k(p)} reach\text{-}dist_k(p, o)}$$

Valores altos de LRD indicam alta densidade local.

5) **Fator de Outlier Local (LOF)**: Razão entre a LRD média dos vizinhos e a LRD do ponto p:

$$\text{LOF}_k(p) = \frac{\sum_{o \in N_k(p)} LRD_k(o)}{|N_k(p)| \cdot LRD_k(p)}$$

Valores de $\text{LOF}_k(p) > 1$ indicam que p possui densidade local menor que a de seus vizinhos, caracterizando-o como *outlier*. Esta abordagem é eficaz em dados com variações de densidade, pois realiza uma análise adaptativa ao contexto local.

Após o LOF, as amostras são categorizadas: *outliers* recebem -1 e *inliers* 1. Este resultado é usado na correção de rótulos.

No pipeline proposto (Figura 3), o LOF é crucial por sua capacidade de identificar anomalias relativas à densidade da própria classe, mesmo que o ponto esteja em uma região densa de outra classe. A eficácia da detecção de rótulos incorretos depende da habilidade do estimador em capturar essas nuances locais, impactando a qualidade da correção. Embora outros estimadores de densidade (e.g., Gaussianas, GMMs, KDEs, Normalizing Flows, EBMs) pudessem ser usados, o LOF destaca-se pela sua natureza local, sendo menos dependente de modelagens globais e mais robusto para a detecção de rótulos errados.

B. Correção de Rótulos

Esta etapa usa uma metodologia baseada apenas em Curvas Principais, devido ao seu baixo custo computacional e energético e à sua interpretabilidade. Novas curvas são geradas para cada classe utilizando apenas os *inliers*. Os *outliers* então separados (dados com possíveis erros de rótulo) têm suas distâncias às novas curvas principais de cada classe calculadas, e seus rótulos são vinculados à classe representada pela curva principal mais próxima deles. Nesta etapa, o quadrado da distância Euclidiana foi utilizado como métrica. Em seguida, o vetor de rótulos é reconstruído com os rótulos corrigidos.

C. Métricas de Avaliação

Para avaliar o método desenvolvido, os resultados foram compilados e expressos por meio de métricas específicas. O objetivo foi quantificar, de forma objetiva, a eficácia dos ajustes realizados pela técnica na correção de rótulos, sem a necessidade de uma comparação manual exaustiva.

As métricas empregadas, apresentadas na Tabela I, refletem o desempenho alcançado durante o processo de correção.

TABLE I
MÉTRICAS CALCULADAS PARA AVALIAÇÃO DA CORREÇÃO DE RÓTULOS.

Métrica	Descrição
Taxa de Erros Detectados Corretamente	Proporção de erros de rótulo reais identificados como tal.
Taxa de Erros Detectados Incorretamente	Proporção de rótulos corretos erroneamente identificados como erros.
Erros de Rótulo Corrigidos	Número absoluto de rótulos incorretos que foram corrigidos.
Taxa de Erros Corrigidos	Proporção dos erros de rótulo originais que foram corrigidos.
Taxa de Erros Não Corrigidos	Proporção dos erros de rótulo originais que permaneceram incorretos.
Novos Erros Gerados	Número de rótulos corretos que foram transformados em incorretos pela técnica.

A seguir, detalha-se o significado de cada métrica:

- **Taxa de Erros Detectados Corretamente**: Representa a porcentagem de rótulos verdadeiramente incorretos que foram corretamente identificados como anômalos pela técnica, em relação ao total de dados. Essa métrica indica a capacidade do método em detectar a contaminação real.
- **Taxa de Erros Detectados Incorretamente**: Indica a porcentagem de rótulos que, embora fossem corretos, foram erroneamente classificados como erros pela técnica, em relação ao total de dados. Essa métrica reflete a taxa de falsos positivos na detecção de erros.
- **Erros de Rótulo Corrigidos**: O número absoluto de rótulos que foram ajustados de uma categoria incorreta para a categoria correta. Esta métrica quantifica diretamente a melhoria na qualidade dos rótulos.
- **Taxa de Erros Corrigidos**: A proporção dos erros de rótulo originais no conjunto de dados que foram efetivamente corrigidos pela técnica. Esta métrica avalia a eficiência do método na mitigação da contaminação inicial.
- **Taxa de Erros Não Corrigidos**: O complemento da Taxa de Erros Corrigidos, indicando a proporção dos erros de rótulo originais que a técnica não conseguiu corrigir.
- **Novos Erros Gerados**: O número de rótulos que eram inicialmente corretos, mas foram incorretamente alterados para uma categoria errada pelo processo de correção. Esta métrica é crucial para avaliar se a técnica introduziu novas contaminações.

D. Avaliação de eficiência energética e pegada de carbono

Outra métrica fundamental para a análise comparativa das técnicas implementadas reside na avaliação da eficiência energética e da pegada de carbono associada a cada execução. Para esta finalidade, empregou-se a biblioteca *CodeCarbon* [13], uma ferramenta robusta que possibilita o registro e o monitoramento detalhado do consumo de energia de programas Python e, opcionalmente, de toda a máquina hospedeira. Uma característica distintiva da *CodeCarbon* é sua capacidade de considerar a localização geográfica do dispositivo de execução, um fator crucial na estimativa precisa das emissões de carbono devido às variações na intensidade de carbono das fontes de energia em diferentes regiões.

A *CodeCarbon* realiza a estimativa do consumo energético monitorando o tempo de execução (t) do código e agregando o consumo médio de energia da Unidade Central de Processamento (CPU, P_{CPU}), da Unidade de Processamento Gráfico (GPU, P_{GPU}), caso utilizada, e da memória de acesso aleatório (RAM, P_{RAM}) do sistema durante o período de processamento. A formulação básica para o cálculo da energia consumida (E) pode ser expressa como:

$$E = t \times (P_{\text{CPU}} + P_{\text{GPU}} + P_{\text{RAM}}) \tag{3}$$

É importante notar que esta equação representa uma simplificação do modelo subjacente da *CodeCarbon*, que também considera fatores como a eficiência da fonte de alimentação e outros componentes do sistema.

Os dados de métricas gerados pela *CodeCarbon* são subsequentemente registrados em arquivos no formato CSV, proporcionando um conjunto abrangente de informações. Os campos tipicamente incluídos nestes arquivos compreendem o *timestamp* do início da coleta de dados, a região geográfica da coleta, o tempo total de processamento, o consumo energético em Watts-hora (Wh) da CPU, GPU e RAM, bem como os valores de potência elétrica instantânea de cada um desses componentes do sistema. Adicionalmente, são registradas informações detalhadas sobre a máquina em que o script foi executado, incluindo os modelos específicos da CPU, GPU e RAM instaladas, a versão do interpretador Python em uso e a versão da própria biblioteca *CodeCarbon*. Dados de localização geográfica, como o Estado, latitude, longitude, país e o código do país (ISO 3166-1 alpha-2), também são capturados para refinar a estimativa das emissões de carbono.

As métricas de consumo energético e emissões de carbono são tipicamente encontradas nas colunas denominadas *energy_consumed* (geralmente em kWh) e *emissions* (geralmente em $kgCO_2e$), respectivamente. A coluna *pue* (Power Usage Effectiveness) também pode estar presente, representando a eficiência energética do data center ou da infraestrutura onde a execução ocorreu, embora sua disponibilidade dependa da configuração e do ambiente de execução.

A análise destes resultados permite uma avaliação comparativa robusta da eficiência energética das diferentes técnicas implementadas, fornecendo *insights* valiosos sobre o impacto ambiental de cada abordagem computacional. A consideração da localização geográfica torna a estimativa da pegada de carbono mais precisa, permitindo uma avaliação mais completa da sustentabilidade de cada solução.

IV. Configuração Experimental

Para facilitar o uso do método desenvolvido, uma biblioteca foi criada em Python. Com o objetivo de verificar o funcionamento e colocar a biblioteca à prova, foi elaborada uma estrutura de testes com quatro diferentes *datasets* numéricos: sendo três deles bem conhecidos e estabelecidos na literatura (breast_cancer [14], load_iris [15] e load_wine [16]) e um dataset 2D sintético construído pelos autores deste artigo. As características de todos os *datasets* estão descritas na tabela II, com 10% de erro induzido, e os resultados foram comparados com uma técnica estabelecida e conhecida, a *Confident Learning* [17], através da biblioteca *cleanlab*.

TABLE II
Datasets utilizados

Dataset	Samples	Dimension	References
breast_cancer	569	30	[14]
load_iris	150	4	[15]
load_wine	178	13	[16]
dataset 2D sintético	404	40	-

A. Erro induzido

Para poder induzir um erro considerado real dentro dos padrões para um *dataset* rotulado por humanos (cerca de 10%) foi necessário criar uma função que pudesse selecionar a parcela de amostras desejada aleatoriamente e errar o rótulo, adicionando o rótulo de alguma das outras classes presentes no *dataset* que não fosse a classe correta. A Figura 4 mostra o *dataset* load_iris antes e depois de ter 10% dos rótulos errados propositalmente. Observe que alguns eventos da classe "setosa" foram erradamente rotulados como pertencentes às classes "versicolor" e "virginica", e estes estão próximos da fronteira de separação entre estas classes, mostrando o quão desafiadora pode ser essa tarefa de correção de rótulos errados.

B. Parametrização da técnica

Com o objetivo de otimizar a configuração dos parâmetros da técnica proposta, empregou-se o algoritmo evolucionário NSGA-II [18]. Ele foi utilizado para explorar diversas combinações de parâmetros, buscando aquelas que resultassem

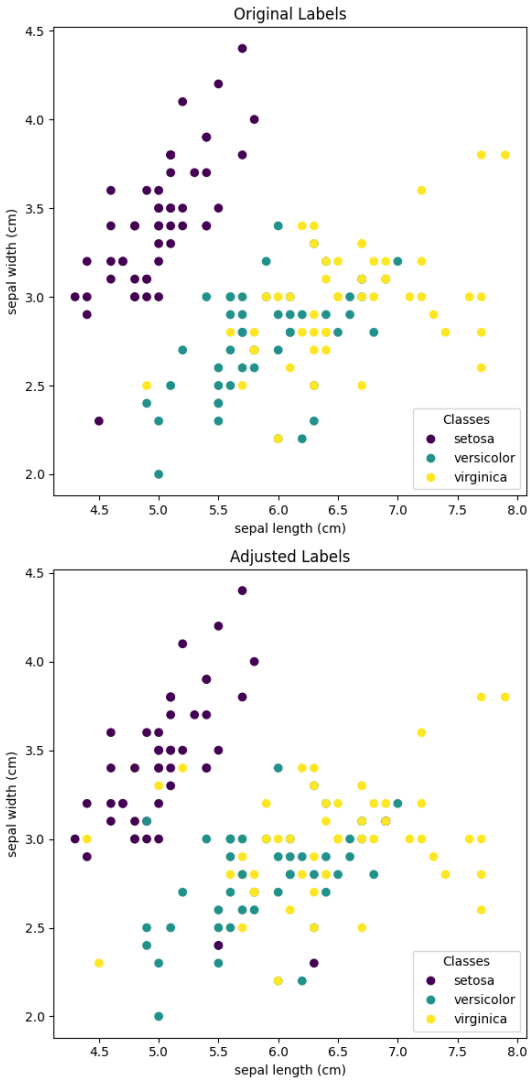

Fig. 4. *dataset* load iris antes e depois do erro induzido

nas melhores saídas para as métricas de desempenho detalhadas na Tabela I.

Os objetivos do NSGA-II foram concentrados nos parâmetros mais influentes do seu funcionamento. Esses parâmetros são listados na Tabela III. Essa abordagem permitiu um ajuste focado em aprimorar esses indicadores, visando, em última instância, melhorar a correção de rótulos incorretos.

Para configurar o NSGA-II, as métricas da Tabela III foram definidas como objetivos, especificando-se o tipo de otimização (minimização ou maximização) para cada uma. Adicionalmente, para ponderar a importância de cada objetivo, foram aplicados os pesos também apresentados na Tabela III. Com uma população de 100 indivíduos e 30 gerações, o NSGA-II pôde realizar uma busca direcionada na superfície dos parâmetros, com o intuito de identificar o conjunto ideal que promovesse uma correção mais precisa e consistente em todos os bancos de dados empregados.

TABLE III
CONFIGURAÇÕES DAS MÉTRICAS DE OTIMIZAÇÃO NO NSGA-II

Métrica	Objetivo	Peso
erros_de_rotulo_ajustados_corretamente	Maximização	1.5
novos_erros_gerados	Minimização	1.2
tempo de processamento	Minimização	1.0

V. RESULTADOS E DISCUSSÕES

Nesta seção, são apresentados os resultados obtidos, sendo estes a obtenção dos parâmetros padrão pelo NSGA II, a construção de uma biblioteca *Python* e os resultados de correção de erros nos rótulos obtidos nos *datasets* mencionados anteriormente, tendo sido estes comparados com os resultados alcançados pela biblioteca *cleanlab*. Por fim, realiza-se a avaliação e comparação dos dois métodos de correção de rótulos em relação ao tempo de processamento, consumo e à eficiência energética.

A. Parametrização da técnica

Como resultado da otimização, observou-se uma melhora significativa no desempenho da técnica. Houve um aprimoramento de cerca de $13{,}43\%$ na métrica de $erros\_de\_rotulo\_ajustados\_corretamente$ (média) e uma redução de $10,00\%$ em $novos\_erros\_gerados$. Além disso, o tempo de processamento foi aumentado em $2,84\%$, o que se traduz em um decremento baixo da eficiência energética da técnica, porém, esse decremento é justificado pelo aumento na eficiência das correções e diminuição dos novos erros gerados. Todos os dados, incluindo as comparações pré e pós-NSGA-II, estão detalhados na Tabela IV, demonstrando a eficácia do uso do NSGA-II na parametrização da técnica desenvolvida. A superfície de Pareto gerada pelo NSGA-II pode ser vista na Figura **??**.

TABLE IV
COMPARATIVO DOS RESULTADOS: PRÉ E PÓS-OTIMIZAÇÃO COM NSGA-II

Métrica	Pré NSGA-II	Pós NSGA-II
erros_de_rotulo_ajustados_corretamente	16,750	19
novos_erros_gerados	5,000	4,500
tempo de processamento	5,411	5,565

B. Construção da biblioteca Python

Durante o desenvolvimento da técnica, o algoritmo foi construído de forma a possibilitar seu uso em diferentes cenários e conjuntos de dados. Ele foi projetado para ser utilizado como um método integrado ao código do usuário, visando modularidade e facilidade de uso. Por esse motivo, sua estrutura foi concebida para resultar em uma biblioteca *Python*, a ser disponibilizada ao usuário final da mesma forma que a técnica do *Confident Learning* e outras abordagens empregadas no tratamento e processamento de dados em *Python*.

C. Comparação entre as correções dos métodos

A técnica desenvolvida teve seus resultados comparados com os resultados da técnica *Confident Learning* implementada pela biblioteca *cleanlab*. A comparação foi feita com o objetivo de verificar se a técnica desenvolvida poderia ser mais eficiente na correção de rótulos do que o *Confident Learning*.

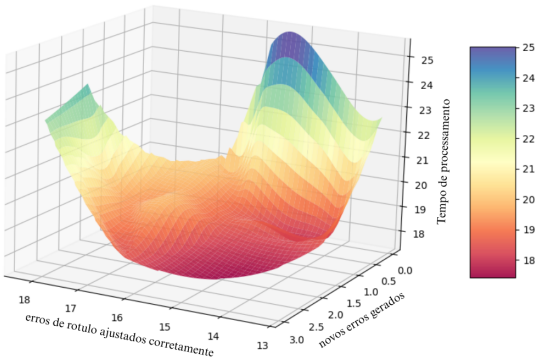

Fig. 5. Superfície de Pareto gerada

Para realizar essa comparação, ambas as técnicas foram utilizadas nos mesmos conjuntos de dados com 10% de erro induzido, e seus resultados foram medidos através das métricas informadas na subseção III-A, com foco maior no número de rótulos ajustados corretamente (*erros_de_rotulo_ajustados_corretamente*) e no número de novos erros gerados (*novos_erros_gerados*), tendo como base o número de erros induzidos no conjunto de dados.

Para a avaliação das métricas de cada *dataset*, foram criados quatro arquivos, um para cada *dataset*, organizados em suas respectivas pastas, contendo arquivos e métricas correspondentes. Cada arquivo individual executava a correção dos rótulos utilizando a técnica desenvolvida e o *Confident Learning*, reunindo, em seguida, as métricas descritas na subseção III-A, para ambas as técnicas. Esses dados eram então salvos para análise posterior.

1) Dados sintéticos: Conforme dito anteriormente, as técnicas foram testadas em diferentes conjuntos de dados. Primeiramente, os testes foram realizados no conjunto de dados sintéticos, devido às suas classes serem mais clusterizadas e os candidatos a erros de rótulo (*outliers*) serem facilmente detectados através da análise gráfica dos dados, o que facilita a validação da detecção e correção.

Quanto aos resultados para esse conjunto, os mesmos foram descritos na Tabela V. Considerando o número de erros induzidos nesse conjunto de 40, conforme a tabela II, nota-se que o *Confident Learning* corrigiu corretamente 3 rótulos a mais que a técnica desenvolvida, porém a técnica desenvolvida não gerou mais nenhum erro no conjunto, totalizando 37 erros corrigidos corretamente de 40, enquanto o *Confident Learning* gerou mais 6 erros no conjunto que não existiam anteriormente, o que pode ser entendido como um saldo de correção de 34, que seria inferior ao resultado da técnica desenvolvida, de 37.

Esse resultado da técnica desenvolvida, deve-se em partes pelo melhor desempenho em conjuntos de dados clusterizados e alongados. Nesses casos, as curvas principais performam melhor, como identificado em [9] e a detecção de *outliers* feita

pelo LOF tem um desempenho superior, devido à densidade dos dados *inliers* ser mais alta que a densidade dos *outliers*.

TABLE V
RESULTADOS DE IDENTIFICAÇÃO E CORREÇÃO DE ERROS EM RÓTULOS PARA O CONJUNTO DE DADOS SINTÉTICOS

Métrica	Técnica desenvolvida	*Confident Learning*
erros_de_rotulo_ajustados_corretamente	37	40
novos_erros_gerados	0	6
Saldo de correção	37	34

2) Datasets reais: Assim como no conjunto de dados sintéticos, a técnica desenvolvida e o *Confident Learning* foram avaliados também nos conjuntos de dados reais, de forma a avaliar seu desempenho com diferentes distribuições de dados.

Através da análise dos resultados das Tabelas VI, VII e VIII foi possível evidenciar as diferenças entre as técnicas de correção.

Nos resultados dos conjuntos de dados *breast_cancer* e *load_wine*, das tabelas VI, VII, o desempenho do *Confident Learning* mostrou-se superior e isso deve-se à distribuição dos dados das classes. Dentro desses conjuntos, as classes de dados se encontram distribuídas de forma esférica, o que impede uma correta detecção de *outliers* devido à homogeneidade da densidade dos dados, e com muita sobreposição das classes, o que diminui a eficiência da correção baseada nas distâncias euclidianas, processos feitos pela técnica desenvolvida.

TABLE VI
RESULTADOS DE IDENTIFICAÇÃO E CORREÇÃO DE ERROS EM RÓTULOS PARA O CONJUNTO DE DADOS BREAST_CANCER

Métrica	Técnica desenvolvida	*Confident Learning*
erros_de_rotulo_ajustados_corretamente	17	54
novos_erros_gerados	11	19
Saldo de correção	6	35

TABLE VII
RESULTADOS DE IDENTIFICAÇÃO E CORREÇÃO DE ERROS EM RÓTULOS PARA O CONJUNTO DE DADOS LOAD_WINE

Métrica	Técnica desenvolvida	*Confident Learning*
erros_de_rotulo_ajustados_corretamente	8	15
novos_erros_gerados	6	4
Saldo de correção	2	11

Nos resultados do conjunto de dados *load_iris*, da tabela VIII, a situação foi diferente devido também à distribuição dos dados e das classes dentro do conjunto. O conjunto *load_iris* possui classes menos sobrepostas que os outros conjuntos reais e os dados de cada classe possuem uma distribuição mais diagonal ou alongada. Essas características favoreceram a detecção dos *outliers* e, principalmente, a construção das curvas para a correção dos rótulos, o que resultou na superioridade da técnica desenvolvida em detrimento do *Confident Learning*.

D. Comparação de eficiência energética

Quanto ao consumo de energia e impacto ambiental da técnica desenvolvida e do *Confident Learning*, as análises foram feitas a partir das execuções dos códigos pelos arquivos de cada *dataset*, durante a avaliação do resultado.

TABLE VIII
RESULTADOS DE IDENTIFICAÇÃO E CORREÇÃO DE ERROS EM RÓTULOS PARA O CONJUNTO DE DADOS LOAD_IRIS

Métrica	Técnica desenvolvida	*Confident Learning*
erros_de_rotulo_ajustados_corretamente	14	15
novos_erros_gerados	1	4
Saldo de correção	13	11

Para a comparação entre as duas técnicas, foram avaliados os parâmetros de tempo de execução (em segundos), quantidade de energia consumida por cada técnica (em kWh) e quantidade de emissão de CO_2 (em kg), dado que todos os testes foram feitos na mesma máquina. Os parâmetros foram medidos para cada *dataset* e cada técnica de correção, conforme mostram as Tabelas IX e X.

TABLE IX
MÉTRICAS DE TEMPO DE EXECUÇÃO, EMISSÃO E QUANTIDADE DE ENERGIA GASTA PARA A TÉCNICA DESENVOLVIDA

Dataset	Tempo de execução (s)	Energia gasta (kWh)	Emissão de CO_2 (kg)
2D sintético	1,359	$4{,}078 \times 10^{-7}$	$9{,}692 \times 10^{-9}$
breast_cancer	1,769	$6{,}655 \times 10^{-6}$	$1{,}582 \times 10^{-8}$
load_wine	1,321	$4{,}021 \times 10^{-6}$	$9{,}558 \times 10^{-9}$
load_iris	1,308	$3{,}844 \times 10^{-6}$	$3{,}701 \times 10^{-7}$

TABLE X
MÉTRICAS DE TEMPO DE EXECUÇÃO, EMISSÃO E QUANTIDADE DE ENERGIA GASTA PARA O *Confident Learning*

Dataset	Tempo de execução (s)	Energia gasta (kWh)	Emissão de CO_2 (kg)
2D sintético	1,150	$2{,}990 \times 10^{-6}$	$2{,}941 \times 10^{-7}$
breast_cancer	12,545	$1{,}811 \times 10^{-4}$	$1{,}781 \times 10^{-5}$
load_wine	1,636	$5{,}426 \times 10^{-6}$	$5{,}337 \times 10^{-7}$
load_iris	1,474	$5{,}079 \times 10^{-6}$	$4{,}995 \times 10^{-7}$

A análise dos resultados evidencia que a eficiência da técnica desenvolvida varia de acordo com o conjunto de dados. Nos datasets *breast_cancer*, *load_wine* e *load_iris*, o método proposto demonstrou superioridade em todas as métricas, conforme mostra a Tabela XI, apresentando reduções de até $85{,}9\%$ no tempo de execução, $96{,}33\%$ no consumo de energia e $99{,}91\%$ na emissão de CO_2, quando comparado ao *Confident Learning*. Já no *dataset* sintético 2D, a técnica apresentou um tempo de execução $18{,}17\%$ maior, no entanto, ainda assim consumiu menos energia e emitiu menos carbono, sugerindo eficiência energética mesmo em situações com leve aumento no tempo.

TABLE XI
COMPARATIVO ENTRE A TÉCNICA DESENVOLVIDA E O *Confident Learning* QUANTO AO TEMPO DE EXECUÇÃO, ENERGIA GASTA E EMISSÃO DE CO_2

Dataset	Δ Tempo (%)	Δ Energia (%)	Δ CO_2 (%)
2D sintético	$+18{,}17$	$+86{,}36$	$+96{,}70$
breast_cancer	$-85{,}90$	$-96{,}33$	$-99{,}91$
load_wine	$-19{,}25$	$-25{,}89$	$-98{,}21$
load_iris	$-11{,}26$	$-24{,}32$	$-25{,}91$

Do ponto de vista energético, a analogia com uma lâmpada LED de 10 Watts, demonstrada na Tabela XII, reforça o impacto das diferenças observadas: enquanto o consumo do *Confident Learning* no *dataset breast_cancer* equivale a mantê-la

acesa por aproximadamente 65,2 segundos, a técnica proposta consome energia suficiente para mantê-la ligada por apenas 2,4 segundos. Mesmo nos demais conjuntos, como *load_iris*, a técnica continua mais eficiente, reduzindo o consumo para 1,38 segundos contra 1,83 do CL.

TABLE XII
CONSUMO EQUIVALENTE EM UMA LÂMPADA LED DE 10 W PARA CADA TÉCNICA

Dataset	LED Técnica (s)	LED CL (s)
2D sintético	0,15	1,08
breast_cancer	2,40	65,20
load_wine	1,45	1,95
load_iris	1,38	1,83

Embora as diferenças no consumo do processamento de cada função possam ser modestas, é fundamental reconhecer que em aplicações de larga escala, com execuções frequentes e conjuntos de dados maiores, essas economias se acumulam significativamente, tornando a técnica desenvolvida uma alternativa mais eficiente e sustentável na maioria dos contextos. A adoção de um critério adaptativo, que escolha dinamicamente entre os métodos com base nas características do conjunto de dados, pode maximizar ainda mais essa eficiência.

Esses resultados adquirem particular relevância ao considerar o menor custo computacional e a maior velocidade de processamento de grandes volumes de dados rotulados proporcionados pela técnica desenvolvida, cenários nos quais o tempo de execução se torna um fator crítico.

Adicionalmente, a menor pegada de carbono associada à técnica desenvolvida representa uma contribuição positiva para a redução dos impactos ambientais decorrentes das emissões de gases de efeito estufa.

VI. CONCLUSÃO

A técnica proposta na biblioteca desenvolvida foi testada utilizando diferentes *datasets* amplamente conhecidos e utilizados na literatura, com o objetivo de validar seu funcionamento e compará-la a uma técnica consolidada e amplamente adotada: o *Confident Learning*. A partir desses testes e comparações, foi possível verificar a eficiência e a velocidade do novo código na identificação e correção de erros em rótulos de *datasets* numéricos.

Entre as principais limitações observadas, destaca-se a dependência do algoritmo em relação à estrutura numérica dos dados, o que restringe sua aplicação a cenários em que os atributos são contínuos ou discretos representáveis numericamente.

Do ponto de vista prático, a nova técnica apresenta potencial para ser integrada a pipelines de aprendizado supervisionado, atuando como uma etapa de pré-processamento capaz de melhorar a qualidade dos dados antes do treinamento dos modelos. Tal aplicação pode beneficiar áreas sensíveis a erros de anotação, como saúde, finanças e diagnóstico industrial, onde a confiabilidade dos rótulos é essencial.

Como trabalhos futuros, espera-se implementar lógica de detecção de *outliers* usando curvas principais, a fim de obter detecções e, consequentemente, correções de erros mais precisas, mantendo a eficiência energética possibilitada pelas curvas principais. Outra melhoria à vista é o aumento da versatilidade do algoritmo, tornando-o capaz de processar diferentes tipos de dados, como imagens e textos, por meio de técnicas de extração de características ou integração com modelos de representação pré-treinados. Também se planeja explorar estratégias de otimização para aprimorar ainda mais a precisão na detecção e correção de rótulos.

AGRADECIMENTOS

Os autores agradecem às entidades apoiadoras mencionadas anteriormente: ao CNPq (Projeto nº 444154/2024-8), à FAPEMIG (Projeto APQ-02323-24) e à CAPES.

REFERÊNCIAS

[1] B. Frénay and M. Verleysen, "Classification in the presence of label noise: a survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 5, pp. 845–869, 2014.
[2] S. Reed, H. Lee, D. Anguelov, C. Szegedy, D. Erhan, and A. Rabinovich, "Training deep neural networks on noisy labels with bootstrapping," in *International Conference on Learning Representations (ICLR)*, 2015.
[3] F. T. Liu, K. M. Ting, and Z. H. Zhou, "Isolation forest," in *2008 Eighth IEEE International Conference on Data Mining (ICDM)*, (Pisa, Italy), pp. 413–422, 2008.
[4] M. M. Breunig, H. P. Kriegel, R. T. Ng, and J. Sander, "Lof: Identifying density-based local outliers," in *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data (SIGMOD '00)*, (Dallas, TX, USA), pp. 93–104, 2000.
[5] D. D. F. B. H. G. B. Fernando Elias de Melo Borges, Otávio Fidelis Mota, "One-class classifier based on principal curves," *Neural Computing and Applications*, vol. 35, no. 26, pp. 19015–19024, 2023.
[6] B. Han and et al., "Co-teaching: Robust training of deep neural networks with extremely noisy labels," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 31, 2018.
[7] J. Li, R. Socher, and S. C. H. Hoi, "Self: Learning to filter noisy labels with self-ensembling," in *International Conference on Learning Representations (ICLR)*, 2020.
[8] T. Hastie and W. Stuetzle, "Principal curves," *Journal of the American Statistical Association*, vol. 84, pp. 502–516, Jun. 1989.
[9] E. C. C. Moraes, D. D. Ferreira, G. B. Vitor, and B. H. G. Barbosa, "Data clustering based on principal curves," *Advances in Data Analysis and Classification*, vol. 14, no. 1, pp. 77–96, 2020.
[10] V. P. S. M. H. F. D. S. C. J. E. d. O. D. D. F. Luiz Paulo Oliveira Sousa, Katia Lumi Fukushima, "A principal curves-based method for electronic tongue data analysis," *Miscellaneous*, vol. 21, no. 4, pp. 4957–4965, 2020.
[11] B. Kégl, A. Krzyżak, T. Linder, and K. Zeger, "Learning and design of principal curves," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 281–297, Mar. 2000.
[12] G. Voronoï, "Nouvelles applications des paramètres continus à la théorie des formes quadratiques. deuxième mémoire. recherches sur les parallélloèdres primitifs.," 1908.
[13] B. Courty and et al., "mlco2/codecarbon: v2.4.1," May 2024.
[14] M. O. S. N. Wolberg, William and W. Street, "Breast Cancer Wisconsin (Diagnostic)." UCI Machine Learning Repository, 1993. DOI: https://doi.org/10.24432/C5DW2B.
[15] R. A. Fisher, "Iris." UCI Machine Learning Repository, 1936. DOI: https://doi.org/10.24432/C56C76.
[16] S. Aeberhard and M. Forina, "Wine." UCI Machine Learning Repository, 1992. DOI: https://doi.org/10.24432/C5PC7J.
[17] C. Northcutt, L. Jiang, and I. Chuang, "Confident learning: Estimating uncertainty in dataset labels," *Journal of Artificial Intelligence Research (JAIR)*, vol. 70, pp. 1373–1411, 2021.
[18] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.