Método de subamostragem via Curvas Principais

Jonatha Levi dos Santos Lustosa*, Fernando Elias de Melo Borges*,
Marcin Kamiński†, Danton Diego Ferreira*,
*Departamento de Automática, Universidade Federal de Lavras, Lavras, Minas Gerais
Emails: jonatha.lustosa@estudante.ufla.br, fernando.borges4@estudante.ufla.br, danton@ufla.br
†Departamento de Máquinas Elétricas, Universidade de Ciências e Tecnologia de Breslávia, Breslávia, Polônia
Email: marcin.kaminski@pwr.edu.pl

***Resumo*—A ascendente geração de dados complexos e de alta dimensionalidade implica em desafios significativos para o desenvolvimento de modelos de aprendizado de máquina, principalmente em cenários com classes desbalanceadas. Neste cenário, a subamostragem surge como uma técnica promissora para mitigar esse problema, reduzindo o volume de dados da classe majoritária e equilibrando a distribuição das classes. Porém, métodos tradicionais de subamostragem, como a seleção aleatória, podem resultar em perda de informações cruciais e afetar a representatividade do conjunto de dados. Este trabalho propõe um novo método de subamostragem baseado em Curvas Principais (CPs). O método explora a distribuição dos dados ao longo da CP representativa da classe majoritária para selecionar os eventos de forma a manter a distribuição original dos dados no espaço de características. Para avaliar o método, foram realizados testes com diferentes bases de dados sintéticas bidimensionais e com uma base de dados real. Um modelo de classificação foi treinado para fazer a predição das classes no conjunto subamostrado. Os resultados obtidos demonstram o potencial do método proposto em termos de desempenho na classificação, redução no tempo de processamento, e principalmente na conservação da representatividade do conjunto de dados subamostrado em comparação com o conjunto original.**

***Palavras-chave*—Aprendizado de máquina; Subamostragem de dados; Desbalanceamento; Curvas Principais.**

I. Introdução

A crescente geração de grandes volumes de dados provenientes de diversas fontes e formatos impõe desafios significativos ao campo da ciência de dados. Esses desafios incluem o processamento, análise e extração de informações úteis a partir dos dados para tomadas de decisão e desenvolvimento de modelos de aprendizado de máquina. Dentre os problemas enfrentados, destaca-se o desbalanceamento de classes em bases de dados, no qual a quantidade de amostras da classe majoritária é significativamente maior do que a da classe minoritária. Este problema pode comprometer o desempenho de classificadores, que tendem a ser enviesados em favor da classe majoritária. Para mitigar esse cenário, técnicas de sobreamostragem e subamostragem têm sido amplamente aplicadas [1], [2], [3], [4], [5].

A subamostragem pode ser uma importante etapa no pré-processamento dos dados com duas aplicações importantes: a primeira, no sentido de reduzir o conjunto de dados de uma classe majoritária e a segunda reduzir conjuntos de dados no geral, a fim de reduzir o custo computacional, mantendo a capacidade preditiva dos modelos de classificação.

Métodos tradicionais, como a seleção aleatória de amostras, podem causar perda de representatividade da base original. Para contornar esse problema, métodos mais avançados foram desenvolvidos, incluindo o *One-Sided Selection* (OSS) e estratégias baseadas em clusterização, como o uso do algoritmo *K-means* [6], [7], [8], [9]. O OSS utiliza classificadores para identificar e remover amostras menos representativas da classe majoritária. Ele busca criar um subconjunto consistente, onde as amostras restantes são suficientes para treinar um classificador eficaz. Essa técnica também utiliza o método de remoção de *Tomek Links*, que identifica pares de amostras de classes opostas próximos à fronteira de decisão e remove aquelas consideradas ruidosas ou redundantes [7], [8].

Já o método baseado em *K-means* segmenta a classe majoritária em *clusters* e seleciona amostras representativas de cada *cluster*, preservando informações importantes e reduzindo a perda de dados [9]. Apesar da eficácia dessas técnicas, a aplicação de métodos de subamostragem baseados em algoritmos de *clustering* ou seleção iterativa pode demandar elevado custo computacional. Isso reforça a necessidade de desenvolver abordagens mais eficientes e representativas, como as que utilizam Curvas Principais.

O foco deste trabalho está no desenvolvimento de uma técnica de subamostragem baseada em Curvas Principais (CP), visando reduzir o desbalanceamento das classes de forma eficiente e representativa. As Curvas Principais (CP) são uma generalização não linear da análise de componentes principais (PCA), utilizadas para modelar dados de forma unidimensional em espaços multidimensionais. Esse método é capaz de abstrair estruturas intrínsecas de dados complexos, reduzindo sua dimensionalidade enquanto preserva características essenciais. Diferentemente de abordagens lineares, as CPs capturam estruturas não lineares e são especialmente úteis em aplicações de redução de dimensionalidade, identificação de *clusters* e sumarização de dados [10], [11]. Estas características tornaram as CPs boas candidatas para problemas de reconhecimento de padrões, e algumas aplicações interessantes podem ser encontradas na literatura, dentre as quais utilizam o algoritmo de Curvas Principais para situações não somente de amostragem de dados, como também se aplica esse algoritmo como um classificador [12], [13], [14], [15], [16]. Desse modo, destaca-se por sua flexibilidade em representar a complexidade e a não linearidade dos dados, implicando na melhoria em técnicas de clusterização, e na detecção e diagnóstico de determinados

sistemas.

A subamostragem tem como objetivo principal reduzir o número de amostras da classe majoritária enquanto mantém a representatividade da base de dados original, equilibrando a distribuição de classes. Dessa forma, espera-se melhorar o desempenho de classificadores em termos de generalização, reduzindo o custo computacional do treinamento [2] explorando as habilidades de Curvas Principais em mapear espaços multidimensionais e de diferentes distribuições.

II. Curvas Principais

Segundo [11], uma CP é definida como aquela que minimiza a distância quadrada esperada entre os dados e a curva, sujeita a um limite de comprimento. 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 [17] 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 [18], que foi utilizado neste trabalho para extrair as curvas principais por ser mais robusto e ter a convergência garantida.

O algoritmo utilizado neste trabalho para a extração da CP foi o k-Segmentos, descrito por [10]. Neste algoritmo, linhas retas são ajustadas para capturar porções locais dos dados. Essas linhas são então combinadas para formar uma representação global da estrutura dos dados. A suavidade da curva é obtida mediante penalidades aplicadas a ângulos abruptos entre segmentos adjacentes, enquanto a eficiência computacional é assegurada por sua implementação incremental [10].

III. Material e Métodos

A. Método Proposto

O uso de curvas principais para subamostragem fundamenta-se na habilidade dessas curvas de representar os dados de forma compacta e significativa. A ideia central é utilizar as curvas para um mapeamento dos dados no espaço *n*-dimensional, dividindo os dados de acordo com os segmentos. Desta forma, cada grupo de dados corresponderá aos pontos projetados no segmento, baseando-se nas regiões de Voronoi. O algoritmo k-Segmentos é particularmente adequado para essa tarefa, pois permite dividir os dados em segmentos que capturam variações locais e globais de densidade de dados, garantindo que as amostras selecionadas mantenham a representatividade da base original [10].

O processo de subamostragem proposto baseia-se nos seguintes passos:

- Aplicar o algoritmo k-Segmentos à classe majoritária para construir uma curva principal que represente a estrutura dos dados.
- Dividir os dados em grupos com base nas regiões de Voronoi, onde cada grupo será composto pelos dados que projetam o segmento da CP da respectiva região de Voronoi.
- Dividir os grupos em subespaços com base nos percentis das distâncias dos pontos ao segmento. Neste trabalho, foram escolhidos cinco percentis (0-20%; 20-40%; 40-60%; 60-80%; 80-100%).
- Selecionar, por sorteio, os dados inseridos em cada subgrupo com base na densidade de dados por segmento ao longo da CP e o valor de amostras desejado para o conjunto subamostrado.

Essa abordagem apresenta diversas vantagens, como a preservação da estrutura global dos dados, gerando conjuntos de dados mais compactos, mantendo a representatividade dos dados originais. Além disso, ao evitar a seleção totalmente aleatória de amostras, o método reduz a probabilidade de perda de informações importantes, aumentando a qualidade do conjunto subamostrado. De forma a ilustrar a sequência de operações do algoritmo, um fluxograma com as etapas do modelo é ilustrado na Figura 1. Além disso, um exemplo do funcionamento gráfico do método é disposto na Figura 2. Neste exemplo, é mostrada a divisão dos dados agrupados em um segmento em seus respectivos percentis de distância, além da seleção de alguns dados dentro dos subconjuntos.

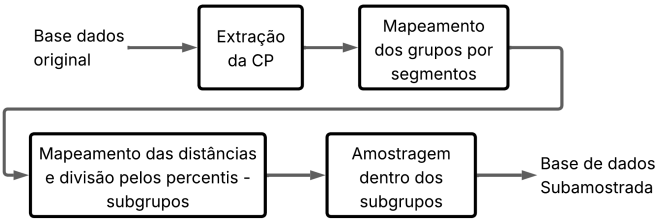


Figura 1: Fluxograma do método proposto.

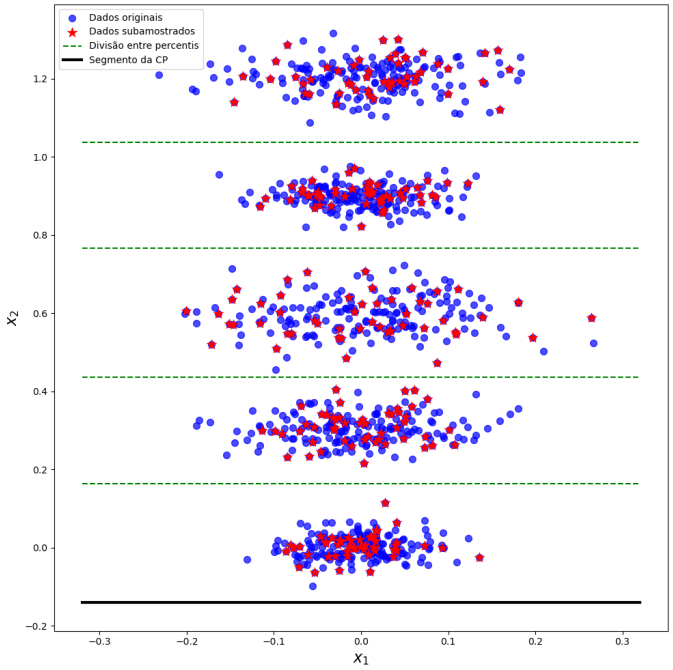


Figura 2: Funcionamento gráfico do método

Como custo computacional, o modelo de subamostragem requer a construção da CP por meio do algoritmo k-segmentos, o qual possui complexidade computacional $O(kn^2)$, onde k é o número de segmentos e n o tamanho do conjunto de dados. Na fase de operacional da subamostragem, a função de maior complexidade é o cálculo das distâncias euclidianas de cada amostra à Curva Principal.

B. Base de Dados

Para testar o método desenvolvido, foram utilizados três conjunto de dados artificiais bidimensionais e uma base de dados real do repositório *Kaggle*. As bases sintéticas correspondem a uma única classe e foram geradas com diferentes distribuições de dados. O objetivo dos testes para as bases de dados sintéticas será a avaliação da capacidade de representação dos dados a partir do conjunto subamostrado, enquanto os ensaios com a base real buscam avaliar o método de subamostragem em um problema de classificação. A descrição das bases de dados sintéticas segue abaixo:

- **Base Esférica**: Consiste em um conjunto de dados com distribuição circular uniforme, contendo $n = 1000$ amostras. Os pontos são distribuídos dentro de um disco de raio 1.5 com centro na origem, com adição de ruído Gaussiano com variância de 0.04.
- **Base Parabólica**: Composta por $n = 1000$ amostras distribuídas ao longo de uma parábola com concavidade para cima, vértice em $(0, -1)$, largura 4 e altura 3 com adição de ruído Gaussiano com variância de 0.04.
- **Base Espiral**: Contém $n = 1000$ amostras dispostas em uma espiral com 2 voltas completas. A forma é controlada pelos parâmetros $a = 0.4$ (expansão) e $b = 0.6$ (largura), com ruído Gaussiano de variância igual a 0.01 adicionado.

Cabe salientar que as bases de dados foram geradas com características geométricas distintas, sendo adequadas para a avaliação do desempenho do método de subamostragem. Assim, poderá ser visualizada a capacidade do método em representar dados de diferentes formatos, apontando suas vantagens e possíveis limitações.

Sobre a base de dados real, foi utilizada a base de dados *Telco Customer Churn*, disponibilizada publicamente na plataforma Kaggle[1]. Esta base contém informações de clientes de uma empresa de telecomunicações, com o objetivo de prever o cancelamento de contratos (*churn*). São 7043 registros, compostos por variáveis demográficas, informações sobre serviços contratados, tipo de contrato, forma de pagamento e dados financeiros, como despesas mensais e totais. A variável alvo é denominada *Churn*, que indica se o cliente encerrou (valor "Yes") ou não (valor "No") seu contrato.

Conforme a base de dados real é obtida sem um pré-processamento, esta etapa se fez necessária antes da avaliação dos modelos de subamostragem. Nesta etapa, foram removidos os identificadores dos clientes e codificadas variáveis categóricas em formato numérico. A variável alvo foi transformada em binária (1 para cancelamento e 0 para permanência). Variáveis como gênero, dependentes, serviços contratados e método de pagamento foram devidamente transformadas. A variável *tenure*, que representa o tempo de contrato, foi categorizada em quatro faixas: até 1 ano, entre 1 e 2 anos, entre 2 e 4 anos, e acima de 4 anos. Para os casos de valores ausentes, estes foram imputados pela mediana das suas respectivas variáveis. Ademais, os dados foram divididos em 80% para treino e 20% para teste, por meio de uma amostragem aleatória estratificada.

C. Procedimento Experimental

Para o desenvolvimento deste trabalho, foram realizados 2 experimentos:

- Experimento com bases de dados sintéticas: este experimento busca avaliar a capacidade do método proposto manter a representação dos dados após a subamostragem.
- Experimento com base de dados real: neste experimento, o método proposto é aplicado em um problema de classificação. Desta forma, busca-se avaliar o impacto da subamostragem na classificação dos dados e no tempo de treinamento dos classificadores.

Além do método proposto, outros dois algoritmos de subamostragem foram aplicados para fins de comparação entre os modelos. Os modelos utilizados nos testes comparativos foram a subamostragem aleatória e o *Cluster Centroids* [9].

O ambiente de execução e recursos computacionais para a execução de todos os experimentos foram conduzidos em um ambiente local, utilizando um computador com as seguintes especificações:

- Processador AMD Ryzen 5 3500U, 2.10GHz, 4 núcleos;

[1]https://www.kaggle.com/datasets/blastchar/telco-customer-churn

- Placa de vídeo integrada Radeon Vega 8 Mobile Gfx
- Memória RAM de 8 GB;
- Sistema operacional Windows 11, 64 bits;
- Ambiente de desenvolvimento: IDE Spyder, versão 5.5.1
- Linguagem de programação: Python, versão 3.11.5
- Bibliotecas Python utilizadas: Numpy, Pandas, Matplotlib, Scikit-learn, Ocpc-py, Imbalanced-learn.

De forma a ilustrar o procedimento experimental realizado, um fluxograma de operações é disposto na Figura 3. Nela, estão contidos os passos de execução para cada um dos experimentos realizados.

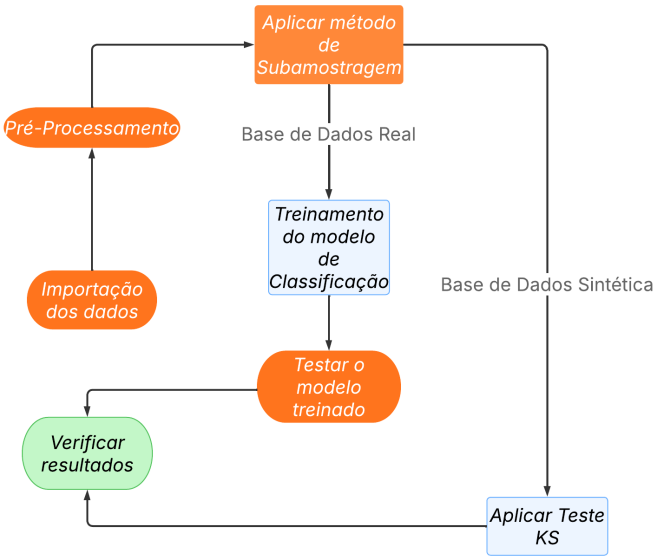


Figura 3: Fluxograma do procedimento experimental.

D. Avaliação dos resultados

Executados os experimentos, os resultados foram avaliados de acordo com as métricas pertinentes à cada uma das aplicações, sendo:

1) Bases de dados sintéticas: De forma a ilustrar o comportamento de cada algoritmo de subamostragem, foram gerados gráficos, tanto da base de dados original, quanto das bases de dados subamostradas para cada um dos modelos. Como resultados quantitativos, foi realizado o teste estatístico Kolmogorov–Smirnov para duas amostras (Teste KS) [19]. Este teste tem por objetivo avaliar se dois conjuntos de dados possuem similaridade entre si. Como hipótese nula, os dados são tidos como iguais e, como hipótese alternativa, os dados são avaliados como diferentes entre si. Os testes foram aplicados com o nível de significância de 5%.

2) Base de dados real: Para os experimentos com a base de dados real, foi realizado o treinamento dos classificadores com o objetivo de avaliar o desempenho preditivo dos modelos mediante as técnicas de subamostragem. Além disso, como forma de se obter um *baseline*, os modelos foram treinados, anteriormente, com o conjunto de dados original.

Foram treinados 2 diferentes modelos de classificação, sendo: *Random Forest* [20], Rede Neural do tipo *Perceptron* Multi-Camadas (MLP, do inglês, *Multi-Layer Perceptron* [21]. Os classificadores foram treinados com ajuste de hiperparâmetros sendo realizado mediante busca aleatória com 30 execuções. Como métrica de decisão do melhor conjunto de hiperparâmetros no algoritmo de busca, foi usada a métrica *f1-score*, uma vez que esta métrica corresponde à média harmônica entre precisão (*precision*) e sensibilidade (*recall*). O ajuste de hiperparâmetros foi realizado na base de dados sem a subamostragem, enquanto os conjuntos de dados subamostrados foram treinados com os hiperparâmetros já selecionados. Além disso, o treinamento foi conduzido utilizando validação cruzada do tipo *k-fold* estratificado com 5 *folds*.

De forma a avaliar os classificadores sob diferentes métricas de desempenho, foram aplicadas as seguintes métricas de avaliação: Acurácia (ACC), Precisão e *Recall* para cada classe, F1-*Score*, índice *Kappa* e área sob à curva ROC (AUC). Além das medidas de classificação, importantes para avaliar o desempenho preditivo, também são apresentados o tempo médio de treinamento, representado pelo tempo médio de treinamento de cada *fold* da validação cruzada. O objetivo de apresentar o tempo de processamento é verificar o impacto da subamostragem no tempo de treinamento dos classificadores.

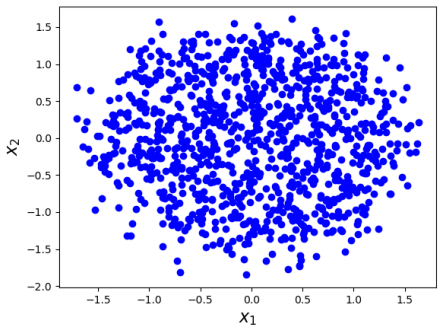
IV. Resultados e Discussão

Inicialmente, os parâmetros para criação da CP para subamostragem foram selecionados de acordo com o trabalho reportado em [22], sendo o número de segmentos igual à 7, a suavização da CP igual à 0.9 e o comprimento inicial do segmento igual à 0.9. Os parâmetros foram avaliados de acordo com seu desempenho e alterados conforme necessário.

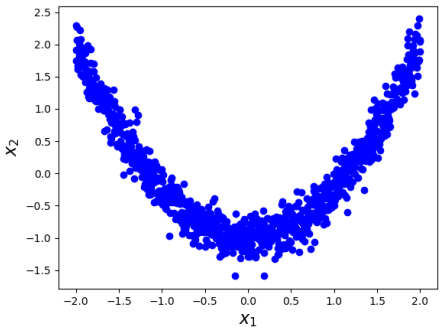
A. Resultados de aplicação nas bases de dados sintéticas

Primeiramente, foram obtidos os conjuntos de dados sintéticos, por meio de sua geração via código Python e, em seguida, aplicada as técnicas de subamostragem e a realização dos testes comparativos. Para ilustrar os conjuntos de dados sintéticos originais e cada um dos conjuntos de dados subamostrados, na Figura 4 estão dispostos os gráficos de dispersão. A fim de ilustrar o comportamento e a representação dos dados pela CP, os conjuntos de dados subamostrados pelas Curvas possuem sua respectiva Curva Principal. Conforme a base de dados espiral possui uma complexidade maior devido à sua geometria, o número de segmentos necessitou de modificação, alterando-se para 17 segmentos.

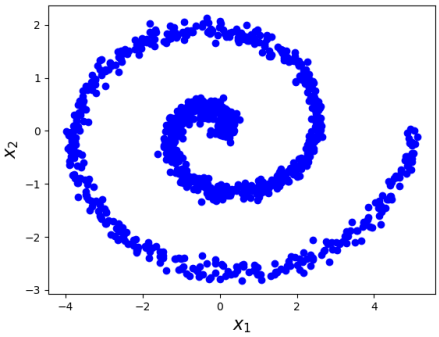
Observando-se os dados subamostrados, verifica-se a inviabilidade em inferir uma comparação visual, sendo necessários métodos quantitativos para a efetiva comparação entre os algoritmos. Para isto, foram realizados os testes KS de duas amostras. Os resultados estatísticos para CP, subamostragem aleatória e por *Cluster Centroids* encontram-se nas Tabelas I, II e III, respectivamente.



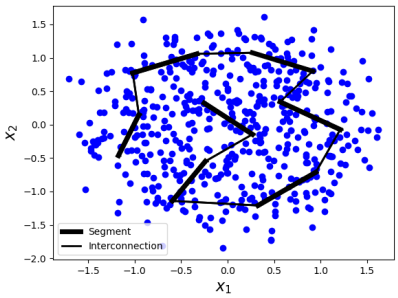
(a) Distribuição esférica



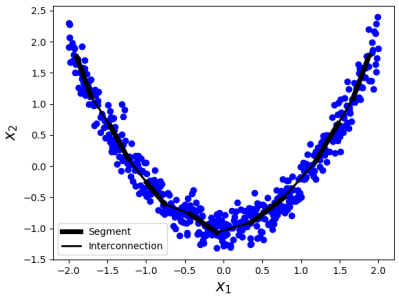
(b) Distribuição em parábola



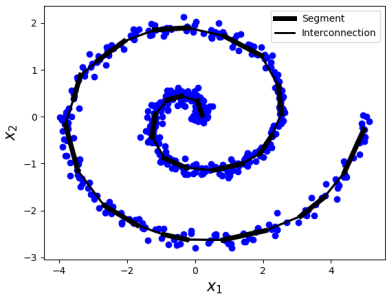
(c) Distribuição em espiral



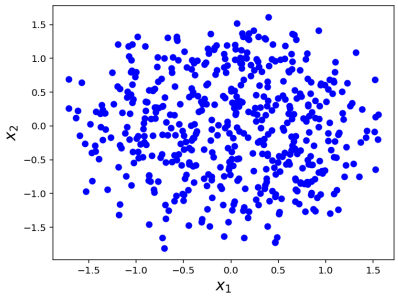
(d) Distribuição esférica.



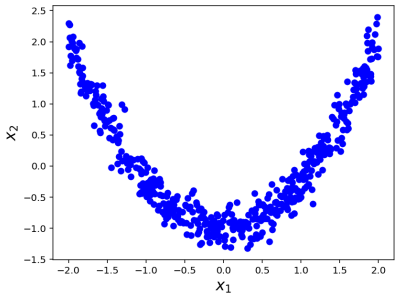
(e) Distribuição em parábola.



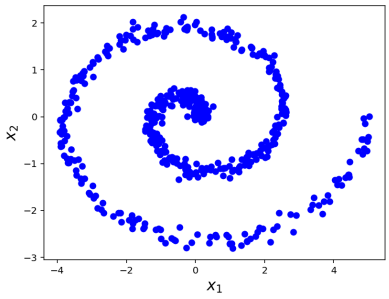
(f) Distribuição em espiral.



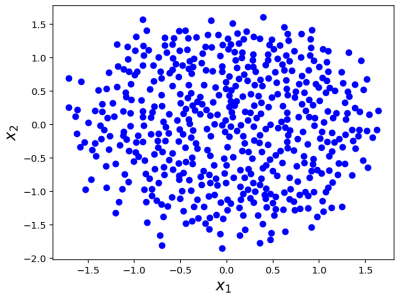
(g) Distribuição esférica.



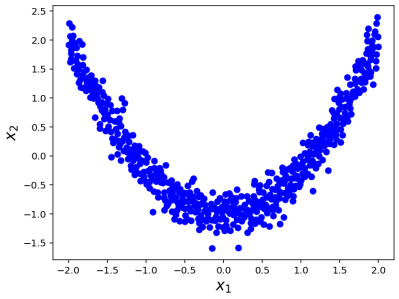
(h) Distribuição em parábola.



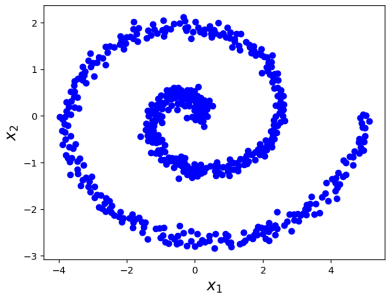
(i) Distribuição em espiral.



(j) Distribuição esférica.



(k) Distribuição em parábola.



(l) Distribuição em espiral.

Figura 4: Conjuntos de dados sintéticos original e suas subamostragens. (a-c) Dados originais; (d-f) Subamostragem por CP; (g-i) Subamostragem aleatória; (j-l) Subamostragem por *Cluster Centroids*.

Tabela I: Resultados dos testes KS por base de dados com subamostragem por CP

Base de Dados (atributo)	Valor-p	Conclusão
Esférico (x_1)	0.984322	Não significativo
Esférico (x_2)	0.999901	Não significativo
Parábola (x_1)	1.000000	Não significativo
Parábola (x_2)	0.999706	Não significativo
Espiral (x_1)	0.999901	Não significativo
Espiral (x_2)	0.999994	Não significativo

Tabela II: Resultados dos testes KS nas diferentes bases de dados utilizando a subamostragem aleatória.

Base de dados (atributo)	Valor-p	Conclusão
Esférico (x_1)	0.954336	Não significativo
Esférico (x_2)	0.999706	Não significativo
Parábola (x_1)	0.831692	Não significativo
Parábola (x_2)	0.448824	Não significativo
Espiral (x_1)	0.746963	Não significativo
Espiral (x_2)	0.996730	Não significativo

Tabela III: Resultados dos testes KS nas diferentes bases de dados utilizando *Cluster Centroids*.

Base de dados (Atributo)	Valor-p	Conclusão
Esférico (x_1)	0.881026	Não significativo
Esférico (x_2)	0.954336	Não significativo
Parábola (x_1)	0.857248	Não significativo
Parábola (x_2)	0.831692	Não significativo
Espiral (x_1)	0.262487	Não significativo
Espiral (x_2)	0.011196	Significativo

Os resultados quantitativos permitiram uma melhor visualização sobre a capacidade de representação dos dados pelos métodos de subamostragem. Desta forma, podem-se destacar alguns fatores: (i) Nos testes realizados pela subamostragem por CP, os resultados do valor *p* indicaram todos os valores próximos de 1 (o menor valor-p obtido foi 0.98), isso indica uma baixa perda da informação dos dados, uma vez que o teste aponta que as distribuições são altamente similares; (ii) Para os testes realizados com a subamostragem aleatória, os valores-*p* apresentados apontam que as distribuições dos dados são iguais, pelo nível de significância de 5%, todavia, os valores *p* gerados foram menores que os apresentados pelas CP, especialmente para a base em Parábola, onde as diferenças foram maiores. Para as outras bases (exceto na *feature* x_1 da base de dados Espiral), as diferenças foram ínfimas; (iii) Para a subamostragem utilizando o *Cluster Centroids*, observa-se um valor *p* inferior aos demais modelos, especialmente para a base de dados Espiral, a qual apresenta um valor *p* que resultou um teste significativo, ou seja, que aponta que as distribuições possuem diferença estatística ao nível de 5%. Tal situação pode ser apontada pela característica de representação de dados pelo *k-means*, a qual possui dificuldade em conjuntos de dados de formato espiral [15]. (iv) Sumarizando, pode-se apontar que o modelo de CP apresentou resultado similar ou superior aos resultados estatísticos dos métodos já existentes na literatura, apontando um potencial do método em aplicações reais.

B. Resultados de aplicação na base de dados real

Para os testes com a base de dados real, inicialmente foi realizado o pré-processamento e o treinamento da base de dados original, sem subamostragem. Assim, foram obtidos os hiperparâmetros dos classificadores utilizados neste experimento. Os valores de cada hiperparâmetro ajustado para a *Random Forest* e a Rede MLP seguem, respectivamente, nas Tabelas IV e V. Treinados os modelos, os resultados para os conjuntos de validação cruzada e teste estão dispostos nas Tabelas VI e VII, respectivamente. Vale ressaltar que, para o conjunto de validação cruzada, os resultados estão dispostos em média $\pm$ desvio-padrão de cada uma das métricas.

Tabela IV: Hiperparâmetros ajustados para o classificador *Random Forest*

Hiperparâmetro	Valor
Número de árvores (`n_estimators`)	200
Amostras mínimas para dividir um nó (`min_samples_split`)	2
Amostras mínimas em uma folha (`min_samples_leaf`)	2
Número máximo de features (`max_features`)	$\sqrt{\text{features}}$
Profundidade máxima das árvores (`max_depth`)	10
Peso das classes (`class_weight`)	`balanced`
Amostragem com reposição (`bootstrap`)	`True`

Tabela V: Hiperparâmetros ajustados para o classificador Rede MLP

Hiperparâmetro	Valor
Função de ativação (`activation`)	`logistic`
Taxa de aprendizado inicial (`alpha`)	1.0438×10^{-5}
Tamanho do *batch* (`batch_size`)	300
Tamanho da camada oculta (`hidden_layer_sizes`)	15
Algoritmo de otimização (`solver`)	`lbfgs`
Tolerância para otimização (`tol`)	5.7623×10^{-9}

Avaliando o desempenho preditivo para o conjunto de validação cruzada (Tabela VI), nota-se que o método de subamostragem *Cluster Centroids* apresentou resultados superiores na maioria das métricas para os dois classificadores. Tal cenário aponta um conjunto de dados mais representativo para a classificação por meio deste algoritmo. Entretanto, o modelo proposto apresentou valores próximos ao *baseline* e/ou superiores para algumas métricas. Com relação ao conjunto de teste, a situação se altera, com os resultados não apontando uma superioridade explícita entre os métodos de subamostragem e *baseline*. Tal cenário coloca o método proposto como alternativa viável para a subamostragem.

Além das medidas de avaliação preditivas, foi realizado um comparativo relacionado ao tempo de processamento médio do treinamento. Cabe salientar que, todos os conjuntos subamostrados possuem o mesmo tamanho. Os resultados relativos ao tempo de treinamento são apresentados na Tabela VIII. Assim como na Tabela VI, os resultados seguem dispostos nos valores média $\pm$ desvio-padrão.

Em relação aos tempos de processamento, para o *Random Forest*, todos os modelos de subamostragem impactaram em um aumento no tempo de treinamento do classificador, entretanto para o método de subamostragem por CP, nota-se um maior tempo de processamento do que os demais,

Tabela VI: Resultados de classificação para a base de dados reais no conjunto de validação cruzada

Métrica	*Random Forest*				MLP			
	Baseline	Aleatória	*Centroids*	CP	*Baseline*	Aleatória	*Centroids*	CP
Precisão (1)	56.7 ± 1.5	73.7 ± 1.2	**82.3 ± 1.7**	75.6 ± 1.5	64.3 ± 3.3	74.2 ± 1.7	**81.5 ± 1.4**	75.1 ± 2.6
Precisão (0)	88.4 ± 1.3	76.6 ± 1.6	**83.9 ± 2.2**	78.2 ± 1.0	**84.3 ± 0.7**	76.9 ± 1.8	83.4 ± 2.2	78.0 ± 1.4
Recall (1)	70.8 ± 3.6	78.0 ± 1.9	**84.3 ± 2.6**	79.2 ± 1.7	54.2 ± 1.8	78.0 ± 2.8	**83.7 ± 2.7**	79.2 ± 1.9
Recall (0)	80.5 ± 0.4	72.1 ± 1.7	**81.8 ± 2.2**	74.3 ± 2.3	**89.1 ± 1.3**	72.8 ± 3.0	81.0 ± 1.9	73.6 ± 4.0
AUC	75.6 ± 1.9	75.1 ± 1.2	**83.0 ± 1.5**	76.8 ± 0.7	71.6 ± 1.4	75.4 ± 1.1	**82.4 ± 1.5**	76.4 ± 1.7
Kappa	47.5 ± 3.1	50.1 ± 2.4	**66.1 ± 2.9**	53.5 ± 1.4	45.6 ± 3.1	50.8 ± 2.2	**64.7 ± 2.9**	52.8 ± 3.4
F1-Score	63.0 ± 2.3	75.8 ± 1.2	**83.2 ± 1.5**	77.3 ± 0.6	58.8 ± 2.3	76.0 ± 1.2	**82.6 ± 1.6**	77.1 ± 1.3
ACC	77.9 ± 1.1	75.1 ± 1.2	**83.0 ± 1.5**	76.8 ± 0.7	79.8 ± 1.3	75.4 ± 1.1	**82.4 ± 1.5**	76.4 ± 1.7

Tabela VII: Resultados de classificação para a base de dados reais no conjunto de teste

Métrica	*Random Forest*				MLP			
	Baseline	Aleatória	*Centroids*	CP	*Baseline*	Aleatória	*Centroids*	CP
Precisão (1)	**54.4**	51.5	41.2	51.6	**61.4**	51.6	40.8	49.8
Precisão (0)	89.1	90.6	**91.6**	90.9	83.9	89.3	**92.3**	90.0
Recall (1)	73.8	79.1	**85.8**	79.7	53.2	75.4	**87.4**	78.1
Recall (0)	**77.7**	73.0	55.7	73.0	**87.9**	74.5	54.2	71.6
AUC	75.7	76.1	70.7	**76.4**	70.6	**74.9**	70.8	74.8
Kappa	**46.2**	44.5	30.8	44.9	43.0	**43.5**	30.5	42.1
F1-Score	**62.7**	62.4	55.6	**62.7**	57.0	**61.3**	55.7	60.8
ACC	**76.7**	74.7	63.7	74.8	**78.7**	74.7	63.0	73.3

Tabela VIII: Comparativo entre os tempos de treinamento para os classificadores.

Métrica	*Random Forest*				MLP			
	Baseline	Aleatória	*Centroids*	CP	*Baseline*	Aleatória	*Centroids*	CP
Tempo de treinamento [ms]	**474.86 ± 21.86**	779.3 ± 41.8	797.21 ± 92.17	989.48 ± 266.49	1022.83 ± 387.69	897.92 ± 258.91	1090.39 ± 68.85	**792.21 ± 55.28**

devido a seleção que foi realizada mediante a lógica do método, por retirar alguns dos eventos distantes da fronteira de decisão. Nesse sentido, os elementos contidos na base de dados subamostrada por esse método exibem alta densidade local e se acumulam ao longo da fronteira de classe, implicando em uma dificuldade de classificação do modelo *Random Forest* e, consequentemente, um maior tempo de processamento. Portanto, não é possível apontar uma vantagem neste caso. Entretanto, para a Rede MLP, a subamostragem aleatória e a subamostragem por CP indicaram melhoria no tempo de processamento, com destaque para o método proposto. Tal cenário pode apontar que diferentes classificadores possuem diferentes impactos referentes à subamostragem. Outro ponto de destaque, cabe ao desempenho significativamente superior do método proposto no treinamento das Redes Neurais, o que mostra que o método possui capacidade de auxiliar no treinamento de modelos, em relação à redução do tempo de processamento.

Desta forma, analisando os resultados obtidos, alguns apontamentos podem ser realizados para o aperfeiçoamento do método proposto em trabalhos futuros. Primeiramente, a avaliação em diferentes bases de dados reais, verificando como o método se comporta sob dados assimétricos, contínuos e discretos. Tal análise poderá apontar, de forma mais ampla e assertiva, as limitações da representatividade dos dados subamostrados pelo método proposto em problemas de classificação de dados. Outro apontamento é a aplicação do modelo de subamostragem sobre diferentes modelos de classificação supervisionada, a fim de indicar sensibilidade ao classificador e a efetividade do método proposto na obtenção de conjuntos que possam ser representativos com o objetivo de otimizar o treinamento de classificadores.

V. Conclusão

Este trabalho tem por objetivo a proposta de um algoritmo de subamostragem baseado em Curvas Principais. A ideia central do trabalho é explorar a capacidade de representação dos dados pelas CP e obter conjuntos de dados reduzidos com o mínimo possível de perda de informação. Foram realizados testes relativos à capacidade de subamostrar dados representativos e o impacto em problemas de classificação de dados. Os resultados mostraram o potencial do método proposto como uma alternativa viável aos métodos de subamostragem já existentes na literatura. Outro ponto importante, foi a capacidade da subamostragem por CP frente à redução significativa no treinamento da rede MLP, apresentando um resultado superior aos demais, com diferença relevante.

Como trabalhos futuros, tem-se por objetivo, verificar possíveis melhorias e/ou otimizações no modelo proposto, além de amplificar o escopo de avaliação frente à modelos de classificação, com testes em bases de dados diferentes, com diferentes atributos e amostras maiores. Desta forma, poderá ser possível estipular o algoritmo proposto com seus pontos positivos e suas limitações frente aos demais métodos de subamostragem existentes.

Agradecimentos

Os autores agradecem o generoso apoio financeiro e institucional da CAPES, da FAPEMIG e do CNPq, cujo suporte foi fundamental para a realização desta pesquisa.

REFERÊNCIAS

[1] F. E. M. Borges, V. D. Reis, and D. D. Ferreira, "Synthetic data generator based on principal curves," in *Proceedings of the 5th International Conference on Advances in Signal Processing and Artificial Intelligence (ASPAI' 2023)*. Tenerife, Espanha: International Frequency Sensor Association (IFSA), 2023, pp. 254–257.

[2] C. L. d. Castro and A. P. Braga, "Aprendizado supervisionado com conjuntos de dados desbalanceados," *Sba: Controle & Automação Sociedade Brasileira de Automatica*, vol. 22, pp. 441–466, 2011.

[3] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[4] P. J. Huang, *Classification of imbalanced data using synthetic over-sampling techniques*. University of California, Los Angeles, 2015.

[5] H. He, Y. Bai, E. A. Garcia, and S. Li, "Adasyn: Adaptive synthetic sampling approach for imbalanced learning," in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, 2008, pp. 1322–1328.

[6] J. Van Hulse, T. M. Khoshgoftaar, and A. Napolitano, "An empirical comparison of repetitive undersampling techniques," in *2009 IEEE international conference on information reuse & integration*. IEEE, 2009, pp. 29–34.

[7] M. Kubat, S. Matwin *et al.*, "Addressing the curse of imbalanced training sets: one-sided selection," in *Icml*, vol. 97, no. 1. Citeseer, 1997, p. 179.

[8] I. Tomek, "A generalization of the k-nn rule," *IEEE Transactions on Systems, Man, and Cybernetics*, no. 2, pp. 121–126, 1976.

[9] W.-C. Lin, C.-F. Tsai, Y.-H. Hu, and J.-S. Jhang, "Clustering-based undersampling in class-imbalanced data," *Information Sciences*, vol. 409, pp. 17–26, 2017.

[10] J. J. Verbeek, N. Vlassis, and B. Kröse, "A k-segments algorithm for finding principal curves," *Pattern Recognition Letters*, vol. 23, no. 8, pp. 1009–1017, 2002.

[11] B. Kégl, A. Krzyzak, T. Linder, and K. Zeger, "Learning and design of principal curves," *IEEE transactions on pattern analysis and machine intelligence*, vol. 22, no. 3, pp. 281–297, 2000.

[12] D. D. Ferreira, J. M. de Seixas, C. A. Duque, A. S. Cerqueira, and P. F. Ribeiro, "A direct approach for disturbance detection based on principal curves," in *2014 16th International Conference on Harmonics and Quality of Power (ICHQP)*. IEEE, 2014, pp. 747–751.

[13] F. E. d. M. Borges, D. A. Ribeiro, O. F. Mota, D. D. Ferreira, and B. N. Huallpa, "Classificador nao supervisionado baseado em curvas principais para detecçao de falhas em motor de induçao," in *Congresso Brasileiro de Automática-CBA*, vol. 1, no. 1, 2019.

[14] E. C. C. Moraes and D. D. Ferreira, "A principal curve-based method for data clustering," in *2016 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2016, pp. 3966–3971.

[15] 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.

[16] F. E. de Melo Borges, O. F. Mota, D. D. Ferreira, and B. H. G. Barbosa, "One-class classifier based on principal curves," *Neural Computing and Applications*, vol. 35, no. 26, pp. 19 015–19 024, 2023.

[17] T. Hastie and W. Stuetzle, "Principal curves," *Journal of the American Statistical Association*, vol. 84, no. 406, pp. 502–516, Jun. 1989.

[18] 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, no. 3, pp. 281–297, Mar. 2000.

[19] W. Conover, *Practical Nonparametric Statistics*. John Wiley & Sons, 1999.

[20] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[21] S. Haykin, *Redes neurais: princípios e prática*. Porto Alegre: Bookman Editora, 2007.

[22] F. E. de Melo Borges, D. D. Ferreira, and J. M. de Seixas, "Curvas principais para a seleção de dados de treinamento neural com grandes volumes de dados."