Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Bacharelado em Sistemas de Informação

Seleção de características baseada em classificadores de larga margem aplicada no problema de classificação multiclasse

Jônatas Sousa de Faria André

JUIZ DE FORA
MARÇO, 2021

Seleção de características baseada em classificadores de larga margem aplicada no problema de classificação multiclasse

Jônatas Sousa de Faria André

Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Bacharelado em Sistemas de Informação

Orientador: Saulo Moraes Villela

JUIZ DE FORA
MARÇO, 2021

Seleção de características baseada em classificadores de larga margem aplicada no problema de classificação multiclasse

Jônatas Sousa de Faria André

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTEGRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE BACHAREL EM SISTEMAS DE INFORMAÇÃO.

Aprovada por:

Saulo Moraes Villela
D.Sc. em Engenharia de Sistemas e Computação

Raul Fonseca Neto
D.Sc. em Engenharia de Sistemas e Computação

Carlos Cristiano Hasenclever Borges
D.Sc. em Engenharia Civil

JUIZ DE FORA
15 DE MARÇO, 2021

À Deus, aquele que merece toda honra e toda glória.

À minha esposa pelo apoio e paciência.

Aos pais, pelas orações e torcida.

Resumo

Em problemas que envolvem aprendizado de máquina existe um fator que deve ser levado em conta quando o objetivo é encontrar a melhor solução possível com a melhor acurácia possível. Faz-se necessário aplicar técnicas que, conhecidamente, podem trazer melhora na performance do classificador. Em problemas de classificação multiclasse, uma das soluções possíveis é realizar uma seleção de características de forma a reduzir a dimensionalidade do conjunto de dados utilizado para treinamento e, como consequência, aumentar a acurácia da solução e diminuir a quantidade de características necessárias pra se realizar a classificação. Foi proposto para tal, uma seleção de características baseada em classificadores de larga margem, aplicando a mesma ao problema de classificação multiclasse.

Palavras-chave: Seleção de características. Classificação multiclasse. *Machine learning.* Classificadores de larga margem. Classificação.

Abstract

In problems involving machine learning there is a factor that must be taken into account when the objective is to find the best possible solution with the best possible accuracy. It is necessary to apply techniques that are known to improve the classifier's performance. In multiclass classification problems, one of the possible solutions is to perform a feature selection in order to reduce the dimensionality of the dataset used for training and, as a consequence, increase the accuracy of the solution and decrease the number of characteristics necessary to carry out the classification. For this purpose, a feature selection based on large margin classifiers was proposed, applying the same to the problem of multiclass classification.

Keywords: Feature selection. Multiclass classification. Machine learning. Large margin classifiers. Classification.

Agradecimentos

Primeiramente agradeço à Deus, aquele que me deu o dom da vida e o privilégio de cursar uma faculdade. À Deus toda honra e toda glória. Louvo a Deus "porque ele é bom, porque a sua misericórdia dura para sempre." (Salmos 106.1)

Agradeço o apoio e companheirismo de minha esposa, Monique. Em meio a tantas horas dedicadas à pesquisa, tantas noites mal dormidas, nunca me deixou perder o foco no objetivo. "A mulher sábia edifica a sua casa." (Provérbios 14.1)

Agradeço aos meus pais pelo apoio incondicional aos meus objetivos acadêmicos e profissionais desde o início de minha trajetória.

Agradeço ao meu orientador, Saulo Moraes Villela, pelo tema, pela paciência em ensinar e por me levar por caminhos antes desconhecidos que agora se materializam nesse trabalho.

Agradeço aos amigos de faculdade que, em meio a tantas horas de estudo, listas de exercício, trabalhos, provas e risadas, contribuíram imensamente para minha formação acadêmica, pessoal e profissional.

"O conhecimento é uma arma. Arme-se bem antes de ir para a batalha.".

Meistre Aemon (Crônicas de Gelo e Fogo)

Conteúdo

Lista de Figuras

Lista de Tabelas

Lista de Abreviações

SVM	*Support Vector Machine*
AOS	*Admissible Ordered Search*
IMA_p	*Incremental p-Margin Algorithm*
RFE	*Recursive Feature Elimination*
SMOTE	*Synthetic Minority Over-sampling Technique*

1 Introdução

Problemas de classificação são cada vez mais presentes em nosso cotidiano. Sabendo disso, a computação vem ao longo dos anos desenvolvendo e aprimorando técnicas para fazer com que esse processo seja cada vez menos custoso e mais preciso em seus resultados. Um problema de classificação binária consiste em uma situação onde é necessário realizar uma classificação de "sim-não", i.e., decidir se uma instância pertence a uma classe específica ou não. Para este tipo de problema, existem soluções que têm se mostrado muito eficientes como classificadores de larga margem, como exemplo as Máquinas de Vetores Suporte (*Support Vector Machines* – SVMs).

A situação é diferente quando é necessário resolver um problema de classificação multiclasse, onde é preciso rotular instâncias para uma dentre três ou mais classes. Neste caso, apesar de existir soluções para o problema de classificação multiclasse utilizando SVM, algumas soluções conhecidamente eficientes para a classificação binária, como as SVM, não podem ser estendidas facilmente para o problema de classificação multiclasse devido à alta complexidade computacional (SILVA; VILLELA, 2021).

Para solucionar esse problema, uma das abordagens comumente utilizada é a de separação do problema de classificação multiclasse em várias classificações binárias e aplicar técnicas de classificação binária conhecidas. Para a separação do problema em classificações binárias existem duas abordagens: um contra todos (*one-against-all*) e um contra um (*one-against-one*) (MILGRAM; CHERIET; SABOURIN, 2006).

É possível perceber o quanto o problema de classificação, utilizando-se de técnicas de larga margem, pode se tornar custoso quando o objetivo é obter um resultado com minimização da margem de erro. Um dos motivos desse alto custo é a alta dimensionalidade, que pode ser resolvido, senão amenizado, com boas técnicas de seleção de características (LIU; MOTODA, 1998). Quando se trata de seleção de características, o objetivo é selecionar um subconjunto de características do que se deseja classificar, que pode ter relevância para o algoritmo classificador. Ao reduzir a dimensionalidade, i.e., selecionar características, ou atributos, de uma instância, reduz-se o custo que o classificador terá

ao efetuar a classificação.

Outra grande vantagem da redução da dimensionalidade é aumentar a precisão do classificador (JR., 2004). Quando a intenção é classificar um objeto entre uma classe ou outra, quanto mais atributos relevantes para aquela classificação forem analisados, maior a precisão do classificador. Efetuar uma seleção de características eficiente é importante para diminuir, senão sanar, os efeitos da "maldição da dimensionalidade" (BELLMAN, 1966), que consiste no rápido crescimento na complexidade do problema à medida que o número de características aumenta. Em aprendizado de máquina, a maldição de dimensionalidade é associada ao fenômeno de pico, ou"fenômeno de Hughes" (Hughes, 1968). Esse fenômeno afirma que, a assertividade de um classificador aumenta conforme o número de dimensões aumenta, porém, após um determinado número de dimensões, essa assertividade começa a diminuir. Um exemplo da representação desse fenômeno está na Figura 1.1.

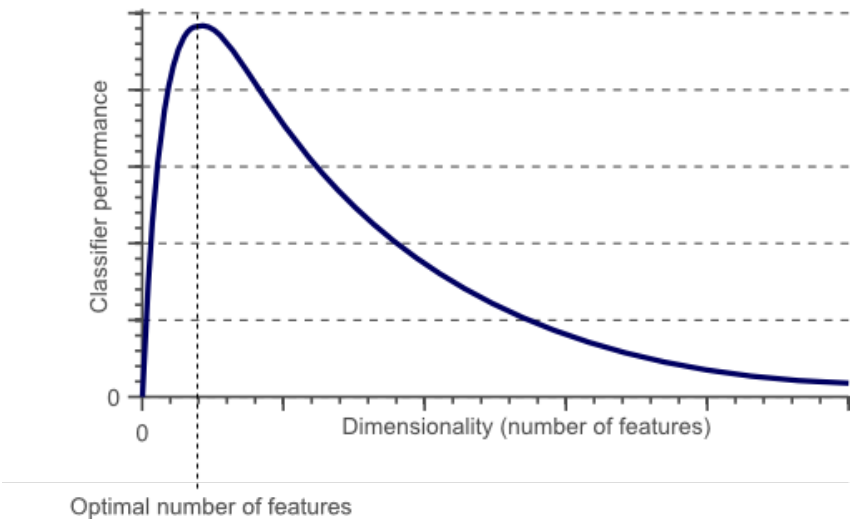


Figura 1.1: Demonstração da maldição da dimensionalidade.

1.1 Motivação

Como citado, em problemas de classificação multiclasse, é sempre um desafio maximizar a taxa de acerto dos algoritmos de classificação que serão usados para trabalhar em um determinado problema. Como ferramenta para solucionar esses problemas, existe a seleção de características com o objetivo de reduzir a dimensionalidade do problema e se aproximar de um resultado ótimo na relação dimensionalidade X performance do

classificador, como demonstrado na Figura 1.1.

Existem vários métodos de seleção de características, sendo que tais métodos são classificados entre abordagem embutida (*embedded*), abordagem encapsulada (*wrapper*) e abordagem em filtro (KOHAVI; JOHN, 1997). Dentre os métodos de abordagem *embedded*, tem-se como exemplos os algoritmos de aprendizado simbólico CN2 (CLARK; NIBLETT, 1989), C4.5 (QUINLAN, 1993) e ID3 (QUINLAN, 1983). Dentre os métodos de abordagem em filtro, o Golub (GOLUB et al., 1999) pode ser citado como exemplo. Dentre os métodos de abordagem *wrapper*, tem-se como exemplo o RFE (GUYON et al., 2002) e o AOS (VILLELA et al., 2021).

Dentre os algoritmos já citados, vale o destaque para um algoritmo que se utiliza de critérios e medidas oriundas de classificadores de larga margem para efetuar a seleção de características: o algoritmo de Busca Ordenada Admissível (*Admissible Ordered Search* – AOS). Segundo Villela et al. (2021), o AOS apresenta resultados satisfatórios, se mostrando superior a outros métodos testados nos experimentos feitos no desenvolvimento do AOS.

Uma vez que problemas de classificação multiclasse utilizando classificadores de larga margem podem se tornar muito custosos e, conforme a complexidade do conjunto de dados usado para o treinamento do classificador aumenta, a performance do classificador tende a piorar, considerando as diversas abordagens e algoritmos de seleção de características, as dificuldades inerentes à natureza do problema de classificação multiclasse, é de suma importância aplicar técnicas de seleção conhecidamente eficientes nos problemas de classificação multiclasse com o objetivo de reduzir a dimensionalidade sem que haja prejuízo à performance do classificador. É importante ter como foco, a redução da dimensionalidade buscando o pico da curva retratada na Figura 1.1.

O objetivo é que, ao final dos testes, perceba-se redução da dimensionalidade dos conjuntos de dados e aumento na acurácia do classificador.

1.2 Organização

O presente trabalho se organiza da seguinte forma: o Capítulo 2 contém toda a fundamentação teórica necessária para a condução e entendimento da pesquisa. No Capítulo 3,

a abordagem proposta para realizar a pesquisa de seleção de características no problema de classificação multiclasse é apresentada. Já no Capítulo 4, tem-se todos os experimentos realizados com o detalhamento das parametrizações usadas em cada etapa, os dados utilizados, os resultados obtidos e análises de cada abordagem utilizada. Finalmente, o Capítulo 5 apresenta as conclusões obtidas com toda a pesquisa e uma perspectiva futura de novos trabalhos, ampliando as possibilidades encontradas nesta pesquisa.

2 Fundamentação teórica

2.1 Problemas de classificação

O ser humano naturalmente classifica tudo ao seu redor. É a forma que ele tem de discriminar os objetos e pessoas que o cerca e, a partir disso, tomar decisões sobre como lidar com alguma questão. Por exemplo, os carros possuem características como cor, modelo, fabricante, quilometragem rodada, ano de fabricação e assim se consegue chegar numa classificação se o carro é velho ou novo. A partir disso, é possível decidir se vale à pena a compra ou não.

Quando essa necessidade de classificação é trazida para dados computáveis, o cenário não se difere do que foi citado. Necessita-se dizer a qual classe pertence determinada instância de um determinado conjunto de dados. A partir disso, uma análise pode ser feita a respeito daquela instância. O problema de classificação nada mais é que, dado um conjunto de características de uma instância, definir a que classe aquela instância pertence utilizando-se dos padrões identificados ao analisar um conjunto de dados chamado de conjunto de treinamento (FERREIRA, 2005).

O conjunto de treinamento consiste no conjunto de dados com suas características e sua classificação tida como verdade, i.e. deve-se conhecer a que classe cada instância do conjunto de treinamento pertence. A partir desse conjunto de treinamento, um algoritmo de classificação pode gerar um modelo que é utilizado para identificar os padrões da instância analisada e, assim, definir a qual classe aquela instância pertence.

2.1.1 Classificação binária

Dado um conjunto de dados A de cardinalidade n, chamado conjunto de treinamento, composto de pontos $x_i \in \mathbb{R}^d$ e as classes de cada ponto $y_i \in \{-1, +1\}$, o problema de classificação binária consiste em definir se as instâncias de um conjunto B, composto pelas instâncias a serem classificadas pelo modelo gerado pelo algoritmo classificador,

pertencem a uma ou outra classe y_i (CASTRO, 2016).

Um algoritmo de classificação binária que pode ser citado como exemplo é o Algoritmo de Margem Incremental com norma p – IMA$_p$, proposto por Villela, Leite e Neto (2016). Esse algoritmo aproxima a solução de margem p utilizando uma estratégia incremental. A solução é obtida resolvendo sucessivas classificações de margem fixa onde essa margem é incrementada a cada iteração. Segundo Villela, Leite e Neto (2016), esse classificador calcula uma aproximação da margem L_p máxima, permitindo maior flexibilidade e evitando o uso de métodos de programação linear e de ordem superior. Classificadores de larga margem com a norma L_∞, na qual minimiza a norma L_1, são muito úteis na seleção de características, uma vez que produzem soluções esparsas (VILLELA et al., 2021).

2.1.2 Classificação multiclasse

Dado um conjunto de dados C de cardinalidade q, chamado conjunto de treinamento composto de pontos $x_i \in \mathbb{R}^d$ e as classes de cada ponto $y_i \in D$, sendo $|D| = n$ e $n \in \mathbb{N}$ e $n > 2$, o problema de classificação multiclasse consiste na análise de uma ou mais instâncias de um conjunto de instâncias a serem classificadas a fim de definir a qual classe de D a instância analisada pertence. Esse problema difere do problema de classificação binária pelo número de classes. Enquanto a classificação binária, em regra, pretende definir se uma instância pertence à uma classe específica $(+1)$ ou não (-1), a classificação multiclasse é mais complexa, pois pretende definir exatamente a qual classe a instância pertence em meio a um conjunto de m classes possíveis. Quando se utiliza métodos de classificação baseados em hiperplano de separação, esse problema pode se tornar mais complexo de se resolver em relação ao problema de classificação binária.

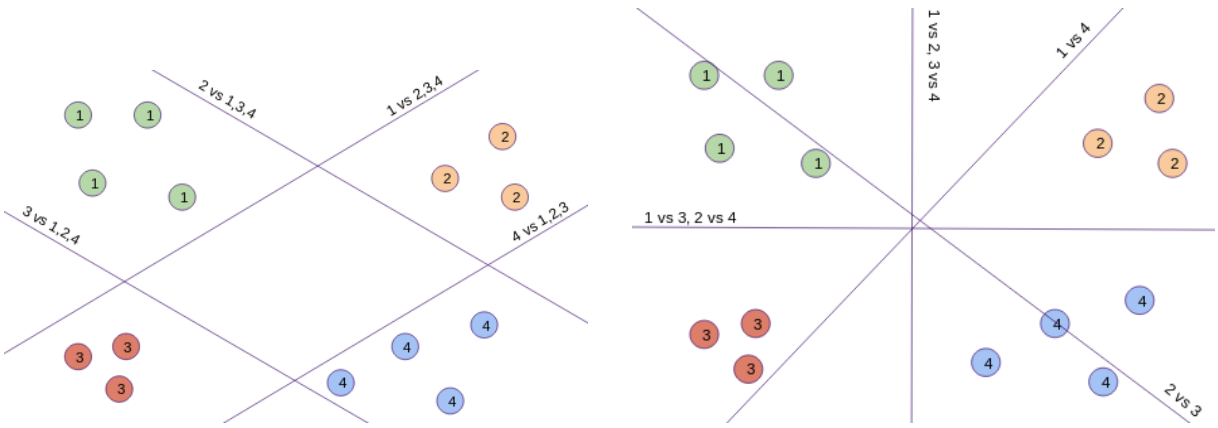
Segundo Milgram, Cheriet e Sabourin (2006), existem duas principais abordagens para se trabalhar com classificação binária em problemas de classificação multiclasse quando se utiliza métodos de classificação de hiperplano separador: um contra todos (*one-against-all*) e um contra um (*one-against-one*).

- Um contra todos (*one-against-all*): estratégia que consiste em reduzir o problema a n classificações binárias, onde m é o número de classes. O algoritmo realiza

classificações binárias entre cada classe contra as demais e assim gera n hiperplanos solução e os combina de alguma maneira. É uma estratégia com um custo aceitável, porém tende a gerar resultados ruins, pois, em cada análise, a classe escolhida fica desbalanceada em relação às outras.

- Um contra um (*one-against-one*): estratégia que consiste em reduzir o problema a classificações binárias realizando a combinação de todas as classes entre si. Percebe-se que essa abordagem consegue obter, geralmente, resultados melhores, porém, é custosa, pois gera $n \times (n-1)/2$ hiperplanos solução (e.g., um problema com 15 classes gera 105 hiperplanos).

Em uma base com 4 classes possíveis, as estratégias de transformação em problemas de classificação binária geram diferentes hiperplanos, como está representado na Figura 2.1.



(a) One-against-all (b) One-against-one

Figura 2.1: Representação dos hiperplanos gerados pelas duas estratégias, *one-against-one* e *one-against-all*, em uma base com 4 classes possíveis.

2.2 Técnicas de balanceamento

O desbalanceamento entre classes é um dos maiores problemas que podem influenciar negativamente o treinamento de um classificador eficiente em problemas de classificação binária ou multiclasse. Nessas situações, o classificador precisa aprender com muito poucos exemplares na classe de interesse em relação aos muitos exemplares da outra, ou, outras classes. Sendo assim, o classificador pode acabar não obtendo uma acurácia muito boa ao

predizer novas instâncias para a classe de interesse. Para mitigar esse problema, tem-se técnicas de *oversampling* e *undersampling*, que são técnicas que aumentam a representatividade da classe minoritária ou diminuem a representatividade da classe maioritária, respectivamente (SILVA; VILLELA, 2021).

Em problemas de classificação multiclasse, quando a estratégia de geração de hiperplanos escolhida é a *one-against-all*, pode ocorrer um problema de desbalanceamento entre classes. Isso se dá pois, quando se confronta uma classe com todas as outras, a tendência é que haja mais amostras nas classes confrontadas do que na classe escolhida como base naquela iteração.

Esse desbalanceamento tem sido um grande problema durante o treinamento de um classificador, pois, nesse cenário, o classificador acaba tendo um viés para classificar uma nova instância para a classe mais representativa. Na Figura 2.2 está representado um conjunto de dados com instâncias distribuídas em 4 classes possíveis antes da transformação em um problema de classificação binária e o mesmo conjunto de dados após aplicar a estratégia *one-against-all* para uma das classes. Percebe-se o quanto os subconjuntos podem ficar desbalanceados.

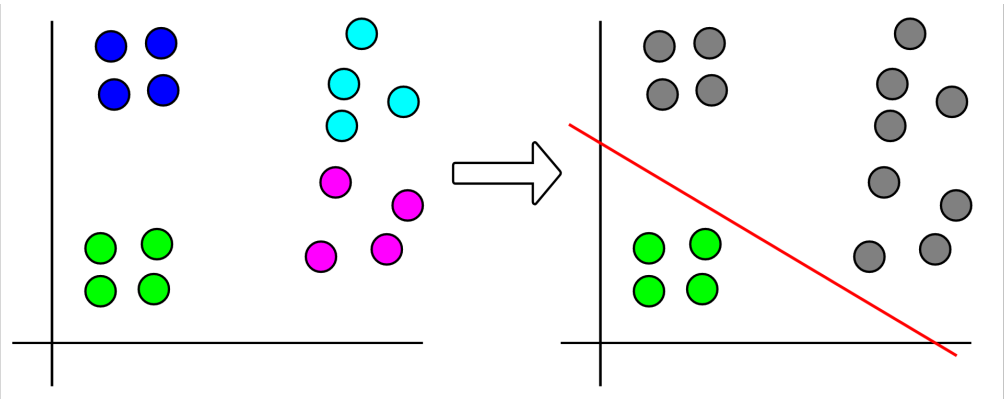


Figura 2.2: Representação do hiperplano gerado a partir da estratégia *one-against-all* e o desbalanceamento entre classes.

Para contornar este problema, utilizam-se técnicas de *oversampling*, onde se ajusta a distribuição entre classes, aumentando a representatividade da classe minoritária. Dessa forma, a cada hiperplano gerado, o classificador é treinado sem que haja um viés desfavorecendo a classe escolhida como base para aquele hiperplano (SILVA; VILLELA, 2021).

Dentre as várias técnicas de *oversampling*, algumas se destacam:

- *Synthetic Minority Over-sampling Technique* (SMOTE): técnica de *oversampling*, proposta por Chawla et al. (2002), que tem como objetivo gerar amostras artificiais da classe minoritária combinando amostras já existentes dessa mesma classe;

- Borderline-SMOTE: método baseado no SMOTE, proposto por Han, Wang e Mao (2005). Esse algoritmo também gera amostras artificiais da classe minoritária combinando amostras reais da mesma. A diferença é que ele tende a utilizar amostras mais próximas à borda entre classes. Existem duas versões desse algoritmo (Borderline-SMOTE 1 e Borderline-SMOTE 2). Ambas as versões constroem um subconjunto de pontos próximos à borda e aplicam o SMOTE nesse conjunto gerando novos pontos. A diferença entre as versões é que a primeira gera pontos se baseando apenas nos pontos vizinhos ao subconjunto construído presentes na classe minoritária, enquanto a segunda considera também pontos vizinhos presentes na classe majoritária.

2.3 Seleção de características

O problema de seleção de características ou, como também pode ser chamado, seleção de atributos consiste na escolha de um subconjunto do conjunto de atributos originais do problema. Essa escolha não deve ser arbitrária mas, sim, o algoritmo deve estabelecer um critério a ser usado na construção da saída do problema. O problema de seleção está intimamente ligado à diminuição de dimensionalidade do problema (VILLELA; LEITE; NETO, 2015; VILLELA et al., 2021). Cada nova característica do problema acrescenta uma dimensão ao mesmo, tornando-o mais complexo a cada nova dimensão acrescentada. A seleção de características tem como objetivo diminuir a dimensionalidade para que o aprendizado de máquina tenha alguns ganhos como:

- Diminuição do custo computacional (recurso + tempo) ao aplicar o aprendizado de máquina e consequentemente a classificação (VILLELA et al., 2021);

- Aumento da precisão do modelo de classificação (JR., 2004);

- Necessidade de um conjunto menor de treinamento para obtenção de um modelo válido (PERVLOVSKY, 1998).

Dentre as diversas soluções para a seleção de características, existem muitas abordagens. Neste trabalho destacam-se duas principais: abordagem em filtro e abordagem *wrapper*. Métodos em filtro introduzem um processo separado responsável por filtrar características irrelevantes antes de ocorrer o aprendizado. Esse é um passo de pré-processamento, que considera características gerais do conjunto de dados para remover ou manter algumas características. Considera-se que métodos em filtro são independentes do algoritmo de classificação. Já os métodos *wrapper*, apesar de também poderem ser métodos de pré-processamento, utilizam-se de algoritmos de classificação como um avaliador da eficiência da seleção realizada (LEE, 2005).

Para esse trabalho 3 algoritmos de seleção se destacam:

- Golub (GOLUB et al., 1999): método em filtro que faz um ranqueamento das características mais relevantes do *dataset* baseado na diferença da média ponderada das classes dividida pela soma do desvio padrão.

- *Recursive Feature Elimination* (RFE): método introduzido por Guyon et al. (2002), cuja ideia básica é eliminar recursivamente um número fixo de características baseado no menor componente do vetor w, considerando que, nesse caso, o mesmo tem pouca influência sobre a posição do hiperplano. A cada remoção, o classificador é retreinado. Como utiliza-se de um classificador como avaliador da solução encontrada a cada iteração, considera-se um método *wrapper*.

- *Admissible Ordered Search* (AOS): método *wrapper*, proposto por Villela et al. (2021), que utiliza um classificador de larga margem para avaliar subconjuntos de características e realiza uma busca ordenada por esses subconjuntos com o objetivo de encontrar o subconjunto com a menor dimensionalidade e a maior margem.

2.4 Considerações

Todos os conceitos apresentados são utilizados para atingir o objetivo da presente pesquisa, no qual consiste na redução da dimensionalidade dos conjuntos de dados idealmente aumentando a acurácia do classificador.

Problemas de classificações multiclasse com métodos baseados em hiperplanos separadores são comumente resolvidos com a separação do problema em classificações binárias. A técnica de separação *one-against-all* normalmente causa um desbalanceamento da representatividade das classes, problema esse que pode ser resolvido, se não amenizado, com a aplicação de técnicas de balanceamento como o *oversampling*. Todo esse contexto apresentado tem por finalidade treinar um classificador com método de hiperplano separador com o máximo de acurácia possível. A seleção de características é, conhecidamente, uma das formas de se aumentar a acurácia do classificador.

3 Abordagem proposta

É necessário uma abordagem que faça uso de algoritmos eficientes para que seja possível medir a acurácia do classificador ao final do processo e comparar com a acurácia obtida originalmente. Dessa forma, é possível verificar o quanto a seleção de característica impactou na classificação.

Primeiramente, tem-se o *dataset* em estado inicial com todas as suas características e amostras. O primeiro passo é, seguindo a estratégia do *one-against-all*, transformar o *dataset* em subproblemas de classificação binária.

Nesse momento é pertinente definir se alguma técnica de balanceamento será utilizada no conjunto de dados obtido após a separação do problema em subproblemas binários. Essa decisão é necessária pois, dentre as acurácias obtidas com os *datasets* originais, nas quais essa pesquisa utiliza como base de comparação, estão aquelas que foram obtidas sem nenhum tipo de balanceamento.

As seleções serão feitas utilizando um algoritmo classificador como um avaliador da performance da seleção e o classificador utilizado é do tipo que utiliza hiperplano separador de larga margem. Sendo assim, foi definido que os treinamentos durante a seleção de características devem ter sucesso com a mesma flexibilização da margem que resulta em sucesso para o treinamento do classificador com o *dataset* com todas as características e transformado em classificação binária. Seguindo essa premissa, é necessário encontrar a flexibilização mínima na qual o classificador consegue realizar o treinamento com sucesso. Nesse ponto do processo, executa-se o IMA_p para o conjunto de dados antes de aplicar qualquer tipo de seleção.

Para um dado subproblema de classificação binária, aplica-se um algoritmo de seleção que irá tentar reduzir a dimensionalidade do conjunto de dados. Nesse momento o importante é armazenar quais características foram mantidas após a seleção. Para tal, utiliza-se um vetor de dimensão m, onde m é o número de características do *dataset* original. Nesse vetor, para cada posição é colocado o valor 1 ou 0, onde 1 representa que aquela característica é importante e deve ser mantida para aquele hiperplano, e 0

representa que aquela característica foi considerada dispensável durante a seleção.

Para o processo de seleção de características, foram utilizados os seguintes métodos: Golub, RFE e AOS. Em todos os casos, a execução do algoritmo parou sempre que se encontrou a dimensão desejada ou então quando o treinamento do classificador falhou.

Esse procedimento é realizado para cada subproblema de classificação binária gerado com a estratégia *one-against-all*, e.g., num *dataset* com 3 classes, 3 processos de seleção são realizadas. Ao final de cada seleção, um vetor de tamanho m é gerado. Todos os vetores são unificados em uma matriz de dimensão $n \times m$, onde n é a quantidade de hiperplanos gerados, i.e., número de classes do *dataset*, e m é a dimensionalidade do *dataset*, i.e., o número de características. Os valores dessa matriz sempre são 1 ou 0, indicando que aquela característica daquele hiperplano foi mantida, ou não, respectivamente.

Como exemplo de uma seleção aplicada em um *dataset* com 6 classes ($n = 6$) e 9 características ($m = 9$), obtém-se a matriz:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Nesse ponto, é necessário definir quais características vão ser mantidas no *dataset*. Para tal, é necessário ter como saída um vetor de tamanho m, onde m é a dimensionalidade do *dataset* e os valores dos elementos do vetor, 1 ou 0, definem se aquela característica será mantida ou não. Para decidir se uma característica deve ser mantida ou não no vetor unificado, leva-se em consideração a representatividade daquela característica dentre os hiperplanos. Quanto mais representativa é a característica, maior a tendência dela ser importante para o *dataset*. Como estratégia de transformação da matriz de características dos hiperplanos em um vetor que será aplicado pra gerar um novo subconjunto de dados simplificado, levando em consideração a representatividade das características, foram usadas 4 abordagens:

- União: Se a característica aparece como necessária em pelo menos um hiperplano, ela é considerada importante para o *dataset*;
- Porcentagem 50: Se a característica aparece como necessária em pelo menos 50% dos hiperplanos, ela é considerada importante para o *dataset*;
- Porcentagem 80: Se a característica aparece como necessária em pelo menos 80% dos hiperplanos, ela é considerada importante para o *dataset*;
- Interseção: Para ser considerada necessária para o *dataset*, a característica deve se mostrar necessária para todos os hiperplanos.

A seguir, é possível ver o resultado das quatro abordagens aplicadas à matriz anteriormente exposta como exemplo:

União:

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

Porcentagem 50:

$$\begin{bmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$

Porcentagem 80:

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Interseção:

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Após essa unificação, um subconjunto é gerado, agora mantendo apenas as características consideradas importantes. Nesse ponto, tem-se um *dataset* simplificado, i.e., com as características consideradas irrelevantes retiradas. Com o *dataset* simplificado, é necessário treinar o classificador para obter a acurácia do mesmo com o conjunto de dados após a seleção de característica.

Como algoritmo de classificação, o IMA_p foi utilizado. A escolha deste algoritmo se deu por conta da pesquisa realizada por Silva e Villela (2021), a qual utilizou o IMA_p como classificador dos *datasets* originais, i.e., sem a remoção de nenhuma característica. Tal trabalho foi utilizado como comparação para as análises dos resultados da presente pesquisa.

O processo todo é descrito pelo fluxograma exibido na Figura 3.1.

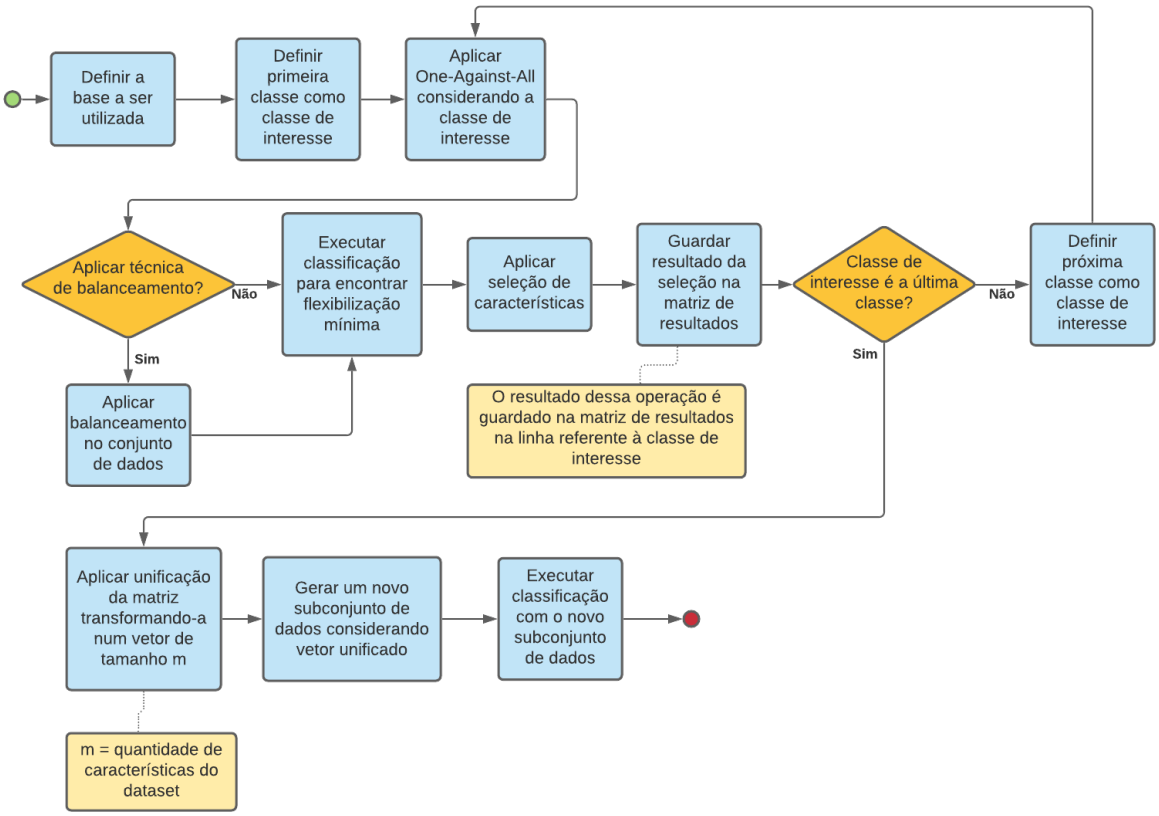


Figura 3.1: Fluxograma com a descrição da abordagem proposta.

4 Experimentos e resultados

4.1 Conjuntos de dados

Para os experimentos, foram utilizados 8 conjuntos de dados (*datasets*) já conhecidos na literatura. Esses conjuntos são os mesmos utilizados por Silva e Villela (2021). A Tabela 4.1 apresenta os *datasets*, com seus números de características, classes e amostras.

Tabela 4.1: *Datasets* e suas informações.

Dataset	Características	Classes	Amostras
iris	4	3	150
wine	13	3	178
vehicle	18	4	846
page-blocks	10	5	5473
glass	9	6	214
segment	18	7	2310
vowel	10	11	990
collins	19	30	100

A combinação dos *datasets* escolhidos, algoritmos de seleção, técnicas de balanceamento e abordagens de unificação de características resultou em 384 subconjuntos potencialmente distintos. Dependendo da abordagem de unificação de características, o subconjunto gerado pode possuir as mesmas características que o *dataset* original (normalmente acontece com união) ou não possuir característica nenhuma (normalmente acontece com interseção). No primeiro caso, não há necessidade de nova classificação, aproveitando a execução feita por Silva e Villela (2021). No segundo, a classificação não é executada pois o subconjunto está vazio.

A classificação teve como validação cruzada o método *k-fold* (KOHAVI, 1995) com $k = 10$, com 10 iterações. A acurácia média foi calculada a partir desse 10-10-*fold*.

4.2 Informações iniciais

Inicialmente, o critério de parada escolhido para os algoritmos de seleção foi a falha do treinamento do classificador com o valor de flexibilização fixo na maior flexibilização encontrada para o *dataset* após a transformação em problemas de classificação binária, aplicando a estratégia *one-against-all*. Esse treinamento para encontrar a flexibilização acontece antes da seleção, i.e., sempre antes de se executar a seleção em um *dataset* binário, encontra-se a menor flexibilização possível para que o IMA_p obtenha êxito no treinamento do classificador. Essa flexibilização é fixada para os treinamentos executados durante a seleção.

Para os algoritmos de seleção escolhidos, o IMA_p foi utilizado como classificador. O valor da norma q definido para a etapa de seleção foi $q = 1$, pois, segundo Villela et al. (2021), este valor gera melhores resultados para o processo de seleção de características. Além do classificador IMA_p, que foi utilizado nos três métodos de seleção, a parametrização de cada método foi definida segundo o trabalho desenvolvido por Villela et al. (2021). A parametrização foi a seguinte:

- Golub:

  Dimensão de parada (número de dimensões que se deseja alcançar): 2.

- RFE:

  Dimensão de parada: 2;

  Quantidade de características removidas por iteração: 1.

- AOS:

  Fator de ramificação: 5;

  Dimensão de parada: 2;

  Profundidade do corte: 5;

  Profundidade do *look ahead*: 2;

  Forma de ordenação: w;

  Forma de escolha: margem.

Para a verificação da acurácia do classificador após a seleção, o algoritmo IMA_p também foi utilizado, porém dessa vez, definindo o valor da norma $q = 2$. Essa escolha foi para manter os parâmetros iguais à classificação executada com os *datasets* originais, por Silva e Villela (2021), os quais foram utilizados como base para as análises.

4.3 Resultados

4.3.1 Dimensionalidade

Um dado importante para se avaliar o desempenho de uma seleção de características é o quanto o algoritmo conseguiu reduzir de dimensionalidade de um *dataset*. Na Tabela 4.2 estão representadas as dimensões obtidas para cada base combinando técnicas de *oversampling*, com os algoritmos de seleção e as abordagens de unificação das seleções binárias (União, Porcentagem 50, Porcentagem 80 e Interseção).

4.3.2 Acurácia dos subconjuntos

É necessário avaliar o quanto de acurácia é possível obter ao executar o IMA_p em um subconjunto, além de quantas características o algoritmo de seleção conseguiu retirar.

Na Tabela 4.3 estão representados os resultados obtidos ao executar a classificação nos subconjuntos gerados pela combinação das técnicas de *oversampling* com as técnicas de unificação e o algoritmo de seleção Golub.

Na Tabela 4.4 estão representados os resultados obtidos ao executar a classificação nos subconjuntos gerados pela combinação das técnicas de *oversampling* com as técnicas de unificação e o algoritmo de seleção RFE.

Na Tabela 4.5 estão representados os resultados obtidos ao executar a classificação nos subconjuntos gerados pela combinação das técnicas de *oversampling* com as técnicas de unificação e o algoritmo de seleção AOS.

Tabela 4.2: Dimensionalidade de cada subconjunto gerado pela combinação dos algoritmos de seleção com as estratégias de *oversampling*.

Base	Balanc.	Dim	Golub				RFE				AOS			
			∪	50	80	∩	∪	50	80	∩	∪	50	80	∩
iris	Puro	4	4	4	2	1	3	3	3	2	4	4	3	2
	SMOTE		4	4	2	1	4	4	4	2	4	4	3	1
	Borderline 1		3	3	2	1	4	4	3	2	4	4	2	2
	Borderline 2		3	3	2	1	4	4	3	1	3	3	3	2
wine	Puro	13	8	8	5	0	8	8	5	1	9	9	4	0
	SMOTE		10	10	7	0	8	8	5	1	9	9	4	0
	Borderline 1		10	10	5	0	7	7	6	0	9	9	4	0
	Borderline 2		10	10	7	0	8	8	5	1	8	8	5	2
vehicle	Puro	18	18	7	1	1	18	18	18	17	18	18	13	5
	SMOTE		16	5	1	0	18	18	18	16	18	18	17	10
	Borderline 1		15	4	2	0	18	18	18	17	18	18	14	8
	Borderline 2		17	5	1	0	18	18	18	15	18	18	13	6
page-blocks	Puro	10	6	2	1	0	10	10	8	0	10	9	6	0
	SMOTE		10	10	2	1	10	10	10	9	10	10	7	5
	Borderline 1		9	6	2	1	10	10	9	9	10	10	3	0
	Borderline 2		9	7	2	2	10	10	9	9	10	10	6	2
glass	Puro	9	8	2	2	0	9	8	8	3	9	8	8	1
	SMOTE		9	5	4	0	9	9	9	4	9	9	9	4
	Borderline 1		9	2	2	0	9	9	9	4	9	9	9	4
	Borderline 2		7	2	2	0	9	9	9	7	9	9	9	1
segment	Puro	18	8	5	1	0	18	18	17	0	18	15	5	0
	SMOTE		14	8	3	0	18	18	17	0	18	18	12	0
	Borderline 1		12	8	3	0	18	18	17	1	18	18	11	0
	Borderline 2		10	6	1	0	18	18	18	3	18	18	8	0
vowel	Puro	10	7	2	1	0	10	10	9	1	10	9	3	0
	SMOTE		7	2	0	0	10	10	9	5	10	10	10	2
	Borderline 1		6	2	1	0	10	10	9	6	10	10	9	0
	Borderline 2		9	2	1	0	10	10	10	5	10	10	9	1
collins	Puro	19	17	0	0	0	19	19	18	0	19	0	0	0
	SMOTE		19	0	0	0	19	19	19	4	19	11	0	0
	Borderline 1		19	1	0	0	19	19	19	1	19	3	0	0
	Borderline 2		17	0	0	0	19	19	19	0	19	3	0	0

4.4 Análises

Entende-se como abordagem de unificação mais restritiva aquela que tende a retirar mais características durante o processo de unificação das seleções em problemas binários e, consequentemente, menos restritiva aquela que tende a retirar menos. No caso, tem-se a União, Porcentagem 50, Porcentagem 80 e Interseção ordenadas da menos restritiva para

Tabela 4.3: Acurácia média do classificador IMA_p nos subconjuntos gerados pela combinação do algoritmo de seleção Golub com estratégias de *oversampling*.

Base	Balanc.	Original		Golub							
				∪		50		80		∩	
		Dim	Acurácia	Dim	Acurácia	Dim	Acurácia	Dim	Acurácia	Dim	Acurácia
iris	Puro	4	96.33 ± 0.65	4	96.33 ± 0.65	4	96.33 ± 0.65	2	95.27 ± 0.81	1	87.13 ± 4.28
	SMOTE		96.35 ± 0.90	4	96.35 ± 0.90	4	96.35 ± 0.90	2	95.80 ± 0.60	1	84.87 ± 6.36
	Borderline 1		96.29 ± 0.78	3	96.13 ± 0.72	3	96.60 ± 0.63	2	95.60 ± 0.61	1	80.20 ± 7.36
	Borderline 2		96.37 ± 0.94	3	93.20 ± 0.98	3	93.27 ± 1.65	2	95.20 ± 0.78	1	83.53 ± 6.25
wine	Puro	13	97.46 ± 0.72	8	96.71 ± 0.49	8	96.71 ± 0.49	5	95.72 ± 0.82	0	-
	SMOTE		97.53 ± 0.55	10	96.71 ± 0.68	10	96.43 ± 0.55	7	96.13 ± 0.86	0	-
	Borderline 1		97.44 ± 0.55	10	98.29 ± 0.62	10	98.60 ± 0.44	5	95.78 ± 0.77	0	-
	Borderline 2		97.36 ± 0.80	10	98.30 ± 0.66	10	97.27 ± 0.66	7	95.97 ± 0.61	0	-
vehicle	Puro	18	77.85 ± 0.56	18	77.85 ± 0.56	7	60.68 ± 0.94	1	25.84 ± 1.24	1	24.51 ± 0.48
	SMOTE		78.92 ± 0.67	16	76.70 ± 0.35	5	61.79 ± 1.00	1	26.25 ± 1.46	0	-
	Borderline 1		78.92 ± 0.67	15	72.28 ± 0.63	4	62.12 ± 1.41	2	40.74 ± 1.01	0	-
	Borderline 2		78.68 ± 0.75	17	77.85 ± 0.64	5	63.24 ± 1.31	1	28.05 ± 1.66	0	-
page-blocks	Puro	10	95.28 ± 0.09	6	94.45 ± 0.17	2	91.91 ± 0.18	1	89.76 ± 0.02	0	-
	SMOTE		88.18 ± 0.32	10	88.18 ± 0.32	10	88.18 ± 0.32	2	84.12 ± 1.10	1	74.51 ± 3.86
	Borderline 1		91.17 ± 0.29	9	92.36 ± 0.09	6	92.05 ± 0.25	2	87.32 ± 1.05	1	83.72 ± 1.62
	Borderline 2		88.98 ± 1.61	9	90.64 ± 0.27	7	90.08 ± 0.21	2	87.58 ± 0.69	2	87.40 ± 0.87
glass	Puro	9	48.25 ± 1.96	8	49.15 ± 2.81	2	21.08 ± 3.54	2	20.23 ± 2.17	0	-
	SMOTE		55.19 ± 2.19	9	55.19 ± 2.19	5	50.27 ± 3.23	4	47.74 ± 2.02	0	-
	Borderline 1		55.83 ± 2.04	9	55.83 ± 2.04	2	46.31 ± 2.02	2	47.36 ± 1.10	0	-
	Borderline 2		55.03 ± 2.25	7	53.71 ± 1.66	2	46.68 ± 1.65	2	46.04 ± 2.42	0	-
segment	Puro	18	84.02 ± 0.35	8	75.03 ± 0.85	5	74.92 ± 1.01	1	27.95 ± 1.03	0	-
	SMOTE		84.33 ± 0.60	14	83.36 ± 0.32	8	80.29 ± 0.62	3	48.04 ± 0.40	0	-
	Borderline 1		84.33 ± 0.60	12	83.50 ± 0.39	8	79.77 ± 1.07	3	64.79 ± 0.52	0	-
	Borderline 2		83.91 ± 0.92	10	80.94 ± 0.84	6	79.14 ± 0.95	1	25.88 ± 0.89	0	-
vowel	Puro	10	30.85 ± 1.45	7	26.39 ± 1.62	2	16.99 ± 0.62	1	10.68 ± 0.22	0	-
	SMOTE		35.38 ± 0.97	7	33.64 ± 1.27	2	19.88 ± 1.37	0	-	0	-
	Borderline 1		37.24 ± 1.07	6	33.93 ± 0.78	2	23.53 ± 1.23	1	10.81 ± 0.85	0	-
	Borderline 2		36.90 ± 0.84	9	36.82 ± 1.21	2	24.13 ± 0.80	1	12.31 ± 1.81	0	-
collins	Puro	19	20.05 ± 1.45	17	20.00 ± 0.83	0	-	0	-	0	-
	SMOTE		22.06 ± 0.96	19	22.06 ± 0.96	0	-	0	-	0	-
	Borderline 1		22.95 ± 0.83	19	22.95 ± 0.83	1	9.47 ± 0.71	0	-	0	-
	Borderline 2		22.39 ± 0.97	17	21.97 ± 1.06	0	-	0	-	0	-

a mais restritiva.

4.4.1 Golub

Analisando as acurácias obtidas com o algoritmo Golub (Tabela 4.3) percebe-se que, na maioria da bases, nenhuma abordagem de unificação obteve uma acurácia maior do que a obtida com o *dataset* original. Nos casos que houve aumento na acurácia, o mesmo não foi tão significativo. Outro ponto que pode-se destacar é que as abordagens de unificação menos restritivas tendem a gerar acurácias bem próximas da original e, em alguns casos, com uma redução moderada da dimensionalidade, chegando a retirar entre 4 e 6 características da base *segment*, por exemplo, e manter uma acurácia bem próxima da original. Quanto mais restritiva é a estratégia de unificação, maior a queda na acurácia

Tabela 4.4: Acurácia média do classificador IMA_p nos subconjuntos gerados pela combinação do algoritmo de seleção RFE com estratégias de *oversampling*.

Base	Balanc.	Original		RFE							
				∪		50		80		∩	
		Dim	Acurácia	Dim	Acurácia	Dim	Acurácia	Dim	Acurácia	Dim	Acurácia
iris	Puro	4	96.33 ± 0.65	3	96.67 ± 0.60	3	96.27 ± 0.53	3	96.60 ± 0.87	2	95.33 ± 0.79
	SMOTE		96.35 ± 0.90	4	96.35 ± 0.90	4	96.35 ± 0.90	4	96.35 ± 0.90	2	95.53 ± 0.67
	Borderline 1		96.29 ± 0.78	4	96.29 ± 0.78	4	96.29 ± 0.78	3	96.00 ± 0.67	2	95.60 ± 0.61
	Borderline 2		96.37 ± 0.94	4	96.37 ± 0.94	4	96.37 ± 0.94	3	93.80 ± 0.85	1	83.53 ± 8.45
wine	Puro	13	97.46 ± 0.72	8	98.81 ± 0.16	8	98.74 ± 0.22	5	95.37 ± 0.88	1	66.83 ± 1.60
	SMOTE		97.53 ± 0.55	8	98.53 ± 0.52	8	98.69 ± 0.27	5	95.99 ± 0.91	1	54.18 ± 6.29
	Borderline 1		97.44 ± 0.55	7	98.36 ± 0.29	7	98.09 ± 0.43	6	97.57 ± 0.70	0	-
	Borderline 2		97.36 ± 0.80	8	98.18 ± 0.40	8	98.24 ± 0.60	5	95.81 ± 0.66	1	43.40 ± 2.17
vehicle	Puro	18	77.85 ± 0.56	18	77.85 ± 0.56	18	77.85 ± 0.56	18	77.85 ± 0.56	17	78.35 ± 0.42
	SMOTE		78.92 ± 0.67	18	78.92 ± 0.67	18	78.92 ± 0.67	18	78.92 ± 0.67	16	78.73 ± 0.72
	Borderline 1		78.92 ± 0.67	18	78.92 ± 0.67	18	78.92 ± 0.67	18	78.92 ± 0.67	17	77.73 ± 0.75
	Borderline 2		78.68 ± 0.75	18	78.68 ± 0.75	18	78.68 ± 0.75	18	78.68 ± 0.75	15	76.80 ± 0.98
page-blocks	Puro	10	95.28 ± 0.09	10	95.28 ± 0.09	10	95.28 ± 0.09	8	94.75 ± 0.17	0	-
	SMOTE		88.18 ± 0.32	10	88.18 ± 0.32	10	88.18 ± 0.32	10	88.18 ± 0.32	9	89.11 ± 0.29
	Borderline 1		91.17 ± 0.29	10	91.17 ± 0.29	10	91.17 ± 0.29	9	92.27 ± 0.25	9	92.34 ± 0.13
	Borderline 2		88.98 ± 1.61	10	88.98 ± 1.61	10	88.98 ± 1.61	9	90.88 ± 0.29	9	90.95 ± 0.38
glass	Puro	9	48.25 ± 1.96	9	48.25 ± 1.96	8	47.39 ± 2.49	8	47.54 ± 2.14	3	32.81 ± 5.75
	SMOTE		55.19 ± 2.19	9	55.19 ± 2.19	9	55.19 ± 2.19	9	55.19 ± 2.19	4	47.85 ± 2.97
	Borderline 1		55.83 ± 2.04	9	55.83 ± 2.04	9	55.83 ± 2.04	9	55.83 ± 2.04	4	52.98 ± 1.67
	Borderline 2		55.03 ± 2.25	9	55.03 ± 2.25	9	55.03 ± 2.25	9	55.03 ± 2.25	7	52.92 ± 2.78
segment	Puro	18	84.02 ± 0.35	18	84.02 ± 0.35	18	84.02 ± 0.35	17	84.26 ± 0.56	0	-
	SMOTE		84.33 ± 0.60	18	84.33 ± 0.60	18	84.33 ± 0.60	17	83.67 ± 0.58	0	-
	Borderline 1		84.33 ± 0.60	18	84.33 ± 0.60	18	84.33 ± 0.60	17	83.95 ± 0.40	1	31.16 ± 5.06
	Borderline 2		83.91 ± 0.92	18	83.91 ± 0.92	18	83.91 ± 0.92	18	83.91 ± 0.92	3	62.87 ± 1.56
vowel	Puro	10	30.85 ± 1.45	10	30.85 ± 1.45	10	30.85 ± 1.45	9	29.96 ± 1.33	1	9.30 ± 0.25
	SMOTE		35.38 ± 0.97	10	35.38 ± 0.97	10	35.38 ± 0.97	9	36.47 ± 0.88	5	28.35 ± 1.32
	Borderline 1		37.24 ± 1.07	10	37.24 ± 1.07	10	37.24 ± 1.07	9	37.40 ± 1.20	6	33.49 ± 0.93
	Borderline 2		36.90 ± 0.84	10	36.90 ± 0.84	10	36.90 ± 0.84	10	36.90 ± 0.84	5	32.02 ± 0.83
collins	Puro	19	20.05 ± 1.45	19	20.05 ± 1.45	19	20.05 ± 1.45	18	19.41 ± 0.66	0	-
	SMOTE		22.06 ± 0.96	19	22.06 ± 0.96	19	22.06 ± 0.96	19	22.06 ± 0.96	4	10.73 ± 0.45
	Borderline 1		22.95 ± 0.83	19	22.95 ± 0.83	19	22.95 ± 0.83	19	22.95 ± 0.83	1	3.09 ± 0.36
	Borderline 2		22.39 ± 0.97	19	22.39 ± 0.97	19	22.39 ± 0.97	19	22.39 ± 0.97	0	-

em relação à acurácia original. Isso acontece para todos os *datasets* analisados, e aumenta independente da técnica de balanceamento. Em bases mais complexas, muitos casos de subconjuntos vazios acontecem, i.e., a abordagem de unificação gerou um conjunto vazio, principalmente em abordagens mais restritivas.

4.4.2 RFE

Diferente do Golub, o RFE gerou menos casos de subconjuntos vazios - apenas 6 em 128, contra 32 do Golub. A queda na acurácia é menos significativa conforme a complexidade do *dataset* e a restrição da abordagem de unificação aumentam. Percebe-se, porém, que em bases mais complexas, não houve uma remoção tão significativa das características, mesmo em unificações mais restritivas. Entende-se com isso que, o RFE, conseguiu retirar,

Tabela 4.5: Acurácia média do classificador IMA_p nos subconjuntos gerados pela combinação do algoritmo de seleção AOS com estratégias de *oversampling*.

Base	Balanc.	Original		AOS							
				∪		50		80		∩	
		Dim	Acurácia	Dim	Acurácia	Dim	Acurácia	Dim	Acurácia	Dim	Acurácia
iris	Puro	4	96.33 ± 0.65	4	96.33 ± 0.65	4	96.33 ± 0.65	3	96.47 ± 0.52	2	95.47 ± 1.02
	SMOTE		96.35 ± 0.90	4	96.35 ± 0.90	4	96.35 ± 0.90	3	96.00 ± 0.84	1	83.20 ± 7.90
	Borderline 1		96.29 ± 0.78	4	96.29 ± 0.78	4	96.29 ± 0.78	2	95.60 ± 0.61	2	95.87 ± 0.27
	Borderline 2		96.37 ± 0.94	3	92.87 ± 0.73	3	92.80 ± 1.19	3	94.13 ± 0.98	2	91.80 ± 1.40
wine	Puro	13	97.46 ± 0.72	9	98.92 ± 0.39	9	98.92 ± 0.39	4	92.47 ± 0.59	0	-
	SMOTE		97.53 ± 0.55	9	98.87 ± 0.66	9	98.86 ± 0.77	4	92.02 ± 0.95	0	-
	Borderline 1		97.44 ± 0.55	9	98.69 ± 0.45	9	98.86 ± 0.35	4	92.25 ± 0.99	0	-
	Borderline 2		97.36 ± 0.80	8	98.12 ± 0.63	8	98.47 ± 0.50	5	96.37 ± 0.95	2	69.88 ± 1.43
vehicle	Puro	18	77.85 ± 0.56	18	77.85 ± 0.56	18	77.85 ± 0.56	13	69.48 ± 0.49	5	59.01 ± 0.58
	SMOTE		78.92 ± 0.67	18	78.92 ± 0.67	18	78.92 ± 0.67	17	79.16 ± 0.70	10	73.12 ± 0.66
	Borderline 1		78.92 ± 0.67	18	78.92 ± 0.67	18	78.92 ± 0.67	14	77.54 ± 0.66	8	71.50 ± 0.48
	Borderline 2		78.68 ± 0.75	18	78.68 ± 0.75	18	78.68 ± 0.75	13	72.85 ± 0.53	6	67.15 ± 1.03
page-blocks	Puro	10	95.28 ± 0.09	10	95.28 ± 0.09	9	95.23 ± 0.12	6	91.54 ± 0.05	0	-
	SMOTE		88.18 ± 0.32	10	88.18 ± 0.32	10	88.18 ± 0.32	7	87.45 ± 1.19	5	78.99 ± 1.05
	Borderline 1		91.17 ± 0.29	10	91.17 ± 0.29	10	91.17 ± 0.29	3	83.62 ± 0.93	0	-
	Borderline 2		88.98 ± 1.61	10	88.98 ± 1.61	10	88.98 ± 1.61	6	91.72 ± 0.55	2	24.53 ± 2.86
glass	Puro	9	48.25 ± 1.96	9	48.25 ± 1.96	8	46.19 ± 2.35	8	44.31 ± 2.58	1	43.48 ± 1.48
	SMOTE		55.19 ± 2.19	9	55.19 ± 2.19	9	55.19 ± 2.19	9	55.19 ± 2.19	4	47.62 ± 3.06
	Borderline 1		55.83 ± 2.04	9	55.83 ± 2.04	9	55.83 ± 2.04	9	55.83 ± 2.04	4	45.00 ± 1.65
	Borderline 2		55.03 ± 2.25	9	55.03 ± 2.25	9	55.03 ± 2.25	9	55.03 ± 2.25	1	18.93 ± 3.77
segment	Puro	18	84.02 ± 0.35	18	84.02 ± 0.35	15	81.06 ± 0.79	5	32.63 ± 2.16	0	-
	SMOTE		84.33 ± 0.60	18	84.33 ± 0.60	18	84.33 ± 0.60	12	80.41 ± 0.48	0	-
	Borderline 1		84.33 ± 0.60	18	84.33 ± 0.60	18	84.33 ± 0.60	11	82.48 ± 0.55	0	-
	Borderline 2		83.91 ± 0.92	18	83.91 ± 0.92	18	83.91 ± 0.92	8	66.50 ± 1.07	0	-
vowel	Puro	10	30.85 ± 1.45	10	30.85 ± 1.45	9	28.47 ± 1.12	3	17.31 ± 0.99	0	-
	SMOTE		35.38 ± 0.97	10	35.38 ± 0.97	10	35.38 ± 0.97	10	35.38 ± 0.97	2	19.97 ± 0.86
	Borderline 1		37.24 ± 1.07	10	37.24 ± 1.07	10	37.24 ± 1.07	9	37.18 ± 1.38	0	-
	Borderline 2		36.90 ± 0.84	10	36.90 ± 0.84	10	36.90 ± 0.84	9	37.07 ± 1.32	1	10.54 ± 0.70
collins	Puro	19	20.05 ± 1.45	19	20.05 ± 1.45	0	-	0	-	0	-
	SMOTE		22.06 ± 0.96	19	22.06 ± 0.96	11	17.47 ± 0.70	0	-	0	-
	Borderline 1		22.95 ± 0.83	19	22.95 ± 0.83	3	6.93 ± 0.48	0	-	0	-
	Borderline 2		22.39 ± 0.97	19	22.41 ± 0.85	3	8.38 ± 0.89	0	-	0	-

em geral, praticamente as mesmas características em todas as seleções binárias realizadas. Algumas exceções estão na interseção que chegou a retirar mais de 10 características em alguns *datasets* com dimensionalidade maior, e.g., *collins*, *segment*, *wine*. Nesses casos, a queda na acurácia seguiu a queda da dimensionalidade.

4.4.3 AOS

As acurácias obtidas nas classificações executadas em subconjuntos gerados pela seleção utilizando o AOS em geral foram próximas às acurácias originais, principalmente em abordagens de unificação menos restritivas, seguindo o padrão do Golub e RFE. Diferente do RFE, houve mais casos de subconjuntos vazios porém, menos do que com o Golub - 20 em 128. O AOS, em geral não retirou as mesmas características de maneira uniforme nas

seleções binárias. Até a abordagem de unificação Porcentagem 50 foram raros os casos de queda muito acentuada na dimensionalidade. Já para a Porcentagem 80, houve vários casos em que pelo menos metade das características foram removidas. Nesses casos de queda acentuada na dimensionalidade, percebe-se, como em outros casos, uma queda na acurácia também.

4.4.4 Balanceamento

Como já citado anteriormente, o desbalanceamento durante as classificações tende a ser danoso ao processo de classificação. Nos resultados obtidos, nas bases originais, a diferença da acurácia do classificador sem balanceamento para a acurácia de quando se aplica alguma técnica de *oversampling*, não costuma ser muito grande na maioria das bases testadas. Porém, em muitos casos após a seleção, essa diferença aumenta muito.

Percebe-se que, em geral, os algoritmos de seleção em bases desbalanceadas têm uma redução mais significativa na dimensionalidade do que quando se aplica algum técnica de *oversampling*. Isso pode explicar a queda mais significativa na acurácia do classificador também.

4.4.5 Conclusões gerais da análise

Em geral, a interseção não se mostra uma boa abordagem de unificação, tendo apenas dois *datasets*, *iris* e *page-blocks*, onde, independentemente do algoritmo de seleção, a acurácia se manteve ou apresentou uma queda pequena. Abordagens menos restritivas, como a união, podem ser consideradas mais seguras, se for analisado o aspecto da acurácia, pois retiram menos características e, por isso, tendem a manter a acurácia próxima da original. Porém, no aspecto de diminuir a dimensionalidade deixam a desejar. A porcentagem 80 se mostrou uma abordagem mais interessante no sentido de retirar uma quantidade considerável de características e apresentar uma acurácia razoavelmente próxima da original e, em alguns casos, até apresentar acurácia maior.

Em geral, os piores resultados são obtidos quando se executa a seleção e classificação sem nenhuma técnica de balanceamento, i.e., as técnicas de balanceamento se mostram uma agregação positiva ao processo de seleção.

O Golub não apresentou resultados satisfatórios. Gerou muitos subconjuntos vazios, quedas muito acentuadas na dimensionalidade e na acurácia para a maioria das bases. O RFE já apresentou resultados mais interessantes, pois, apresentou menos subconjuntos vazios e foi o que apresentou mais acurácias aproximadas da original na abordagem Interseção. O AOS também apresentou resultados mais satisfatórios que o Golub, retirando muitas características na abordagem Porcentagem 80 e, mesmo assim, apresentando acurácias bem aproximadas da original.

5 Considerações finais

Problemas de seleção de características são sempre muito desafiadores por se tratarem de algoritmos cujo objetivo principal é retirar características, logo, retirar dados das bases, sem que a informação contida em cada instância se perca. Quando se extrapola esse problema para classificações multiclasse, o desafio aumenta.

O objetivo ideal da seleção de características é sempre reduzir o número de características de um *dataset* mantendo a capacidade daquele subconjunto gerado em treinar um classificador eficiente. É necessário ter o cuidado para não comprometer a base de dados fazendo com que a mesma seja responsável por um treinamento ineficiente do classificador.

Nota-se nas tabelas apresentadas no Capítulo 4 a maldição de dimensionalidade atuando, uma vez que, sempre que a quantidade de características de um *dataset* diminuía demais em relação ao original, a acurácia também diminuía, por muitas vezes de forma significativa.

Ainda assim, nota-se que é possível diminuir a dimensionalidade de bases em problemas de classificação multiclasse e se obter resultados satisfatórios no treinamento do classificador.

Em geral, a abordagem Porcentagem 80 apresentou mais resultados positivos do que negativos, apesar de alguns casos de subconjuntos vazios. Em apenas um *dataset* (*collins*) a combinação com o AOS se mostrou insatisfatória, gerando subconjuntos vazios. Em todos os outros ao menos com uma técnica de balanceamento, a acurácia se manteve próxima da original, sempre diminuindo a dimensionalidade de forma significativa.

Nos *datasets* utilizados neste trabalho, com relativamente poucas características, a mudança sensível na porcentagem da abordagem de unificação se torna obsoleta pois gera o mesmo resultado, por isso a decisão de analisar as porcentagens 50 e 80. Porém, experimentos em *datasets* com mais características podem ser feito com valores de porcentagem mais refinados, com diferenças mais sensíveis.

Uma percepção possivelmente equivocada que se pode ter, é que o RFE é melhor

do que o AOS, pois em geral os resultados do RFE são mais satisfatórios do que o AOS. Porém, segundo Villela et al. (2021), apesar de alguns resultados satisfatórios, o algoritmo RFE continua sendo um método aquém em relação ao AOS. Uma possível explicação para o fato do RFE apresentar melhores resultados de acurácia do que o AOS, é o fato dele ter retirado menos características em todas as seleções binárias, gerando assim, uma seleção unificada mais modesta. Já o AOS foi mais agressivo ao retirar características nas seleções binárias e, assim, acabou gerando subconjuntos de seleção binária mais heterogêneos. O resultado disso foi uma retirada maior de características, prejudicando o desempenho do classificador.

Alguns trabalhos futuros podem ser desenvolvidos a partir deste. Um deles tem relação com a disparidade do AOS e o RFE. Melhorar as dimensões de parada dos algoritmos, i.e., o número de características desejadas na seleção binária, principalmente do AOS, pode acabar gerando subconjuntos menos heterogêneos entre si, e assim, otimizar as estratégias de unificação mais restritivas como Porcentagem 80 e Interseção.

Outro trabalho que pode ser desenvolvido é implementar outras formas de unificação dos vetores de seleção binária, além das quatro já testadas. Um exemplo seria levar em consideração algum tipo de peso nas características ao unificar os vetores, principalmente relacionadas ao vetor w. Dessa forma, levando em consideração outros fatores para aplicar a unificação, pode-se evitar a retirada de características muito importantes.

Percebe-se que não foram obtidos resultados muito satisfatórios no quesito redução significativa da dimensionalidade com aumento da acurácia do classificador. Um dos possíveis motivos é que todas as bases utilizadas os testes parecem estar no lado esquerdo do pico retratado na Figura 1.1. Um trabalho futuro que pode ser desenvolvido é o teste com bases com dimensionalidade maior do que as já utilizadas. O objetivo é testar com bases que estejam bem à direita do pico retratado na Figura 1.1 para que seja possível observar o efeito da seleção de características na performance do classificador.

Bibliografia

BELLMAN, R. Dynamic programming. *Science*, American Association for the Advancement of Science, v. 153, n. 3731, p. 34–37, 1966. ISSN 0036-8075. Disponível em: ⟨https://science.sciencemag.org/content/153/3731/34⟩.

CASTRO, A. F. d. *Estudo da utilização de métodos de seleção de características aplicados ao problema de seleção de marcadores genômicos*. 2016.

CHAWLA, N. V. et al. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, v. 16, p. 321–357, 2002.

CLARK, P.; NIBLETT, T. The cn2 induction algorithm. *Mach. Learn.*, Kluwer Academic Publishers, USA, v. 3, n. 4, p. 261–283, mar. 1989. ISSN 0885-6125. Disponível em: ⟨https://doi.org/10.1023/A:1022641700528⟩.

FERREIRA, J. B. *Mineração de Dados na Retenção de Clientes em Telefonia Celular*. Dissertação (Mestrado) — Faculdade de Engenharia Elétrica, Pontíficia Universidade Católica do Rio de Janeiro, 2005.

GOLUB, T. R. et al. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, v. 286, n. 5439, p. 531–537, 1999. Disponível em: ⟨https://science.sciencemag.org/content/286/5439/531⟩.

GUYON, I. et al. Gene selection for cancer classification using support vector machines. Machine Learning, v. 46, n. 5439, p. 531–537, 2002. Disponível em: ⟨https://doi.org/10.1023/A:1012487302797⟩.

HAN, H.; WANG, W.-Y.; MAO, B.-H. Borderline-smote: A new over-sampling method in imbalanced data sets learning. In: *Proceedings of the 2005 International Conference on Advances in Intelligent Computing - Volume Part I*. [S.l.]: Springer-Verlag, 2005. p. 878–887.

Hughes, G. On the mean accuracy of statistical pattern recognizers. *IEEE Transactions on Information Theory*, v. 14, n. 1, p. 55–63, 1968.

JR., D. C. M. *Redução de Dimensionalidade Utilizando Entropia Condicional Média Aplicada a Problemas de Bioinformática e de Processamento de Imagens*. Dissertação (Mestrado) — Universidade de São Paulo, 2004.

KOHAVI, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1995. (IJCAI'95), p. 1137–1143.

KOHAVI, R.; JOHN, G. H. Wrappers for feature subset selection. *Artificial Intelligence*, v. 97, n. 1, p. 273–324, 1997. ISSN 0004-3702. Relevance. Disponível em: ⟨https://www.sciencedirect.com/science/article/pii/S000437029700043X⟩.

LEE, H. D. *Seleção de atributos importantes para a extração de conhecimento de bases de dados*. Tese (Doutorado) — Instituto de Ciências Matemáticas e de Computação, 2005.

LIU, H.; MOTODA, H. *Feature Selection for Knowledge Discovery and Data Mining*. 2. ed. MA, USA: Springer, 1998. (Kluwer International Series in Engineering and Computer Science). ISBN 978-1-4615-5689-3.

MILGRAM, J.; CHERIET, M.; SABOURIN, R. "one against one" or "one against all": Which one is better for handwriting recognition with svms? 2006.

PERVLOVSKY, L. I. *Conundrum of combinatorial complexity*. [S.l.: s.n.], 1998. v. 20. 666–670 p. (IEEE Trans. On Pattern Analysis and Machine Intelligence, v. 20).

QUINLAN, J. R. *Machine learning: An Artificial Intelligence Approach*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1983.

QUINLAN, J. R. *C4.5: programs for machine learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993. ISBN 1-55860-238-0. Disponível em: ⟨http://portal.acm.org/citation.cfm?id=152181⟩.

SILVA, W. A.; VILLELA, S. M. Improving the one-against-all binary approach for multi-class classification using balancing techniques. *Applied Intelligence*, v. 4, n. 51, p. 396–415, 8 2021.

VILLELA, S. M.; LEITE, S. C.; NETO, R. F. Feature selection from microarray data via an ordered search with projected margin. In: *Proceedings of the 24th International Joint Conference on Artificial Intelligence*. [S.l.]: AAAI Press, 2015. p. 3874–3881. ISBN 978-1-57735-738-4.

VILLELA, S. M.; LEITE, S. C.; NETO, R. F. Incremental p-margin algorithm for classification with arbitrary norm. *Pattern Recognition*, v. 55, p. 261–272, 2016.

VILLELA, S. M. et al. An ordered search with a large margin classifier for feature selection. *Applied Soft Computing*, v. 100, p. 106930, 2021. ISSN 1568-4946.