



UNIVERSIDADE  
FEDERAL  
DE PERNAMBUCO

Universidade Federal de Pernambuco  
Centro de Informática

Graduação em Engenharia da Computação

AVALIAÇÃO DE UM CLASSIFICADOR MULTICLASSE COM  
ABORDAGEM ONE-VS-ONE USANDO DIFERENTES  
CLASSIFICADORES BINÁRIOS

Victor Carriço Santos  
TRABALHO DE GRADUAÇÃO

Recife

2017



Universidade Federal de Pernambuco  
Centro de Informática

Victor Carriço Santos

Avaliação de um Classificador Multiclasse com Abordagem One-vs-One  
Usando Diferentes Classificadores Binários

Trabalho apresentado ao Programa de Graduação em Engenharia da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Bacharel em Engenharia da Computação.

Orientador: Prof. Cleber Zanchettin

Recife

2017

## AGRADECIMENTOS

Primeiramente, agradeço aos meus pais, Andréa Azzi e Paulo Fernando, por sempre apoiarem as minhas escolhas e por sempre terem me dado todo o suporte necessário, me dando toda confiança que preciso para seguir em frente. Aos meus irmãos, Paulo André e Luise Carriço, por todo apoio e compreensão.

A todos os amigos que fiz na faculdade, em especial Danilo Dias, Diogo Medeiros, Diógenes, Egle, Hélio Meira Lins, Ricardo Mariz, Victor Casé pela ótima convivência e apoio mútuo que sempre existiu desde os primeiros períodos juntos.

A todos os meus amigos de fora da faculdade, que compreenderam os dias e noites que deixei de sair com eles para estudar e mesmo assim não me abandonaram. A minha namorada, Bruna Guedes, por ter me incentivado tanto e ter me aguentado nos momentos de tensão para conclusão deste trabalho.

Agradeço ao meu orientador Cleber Zanchettin, por ter confiado em mim para fazer o presente trabalho e por ter me guiado de maneira atenciosa. E a todos os professores do Centro de Informática que eu passei durante esses anos, por me inspirarem e pelo esforço feito por eles para tornarem o centro o que ele é hoje.

## RESUMO

Classificação é um tipo de problema recorrente na área de aprendizagem de máquina que consiste em inferir a qual classe um determinado objeto pertence dentro de um número finito de possibilidades. Em geral, problemas de classificação multiclasse são mais complexos que problemas com classes binárias. Para contornar esse problema, a abordagem *One-Against-One* pode ser aplicada e o problema de classificação com multiclasse é dividido em vários problemas de classificação binária. No entanto, essas soluções normalmente utilizam apenas um único tipo de classificador. O presente trabalho faz uma análise comparativa do método *Diversified One-Against-One (DOAO)* [1], com o *One-Against-One*, *Vote-One-Against-One*, *One-Against-Rest* e *Random Forest*. O método avaliado consiste, basicamente, em utilizar a abordagem um-contra-um, porém, com uma etapa anterior para a escolha do método de classificação binário a ser utilizado com cada par de classes do problema. Na etapa de treinamento, todos os classificadores candidatos são treinados e apenas os que apresentarem melhores erros de validação, para cada par, são escolhidos. Desta forma a uma grande chance de ser utilizado o melhor classificador para cada par de classes.

## ABSTRACT

Classification is a frequent problem in machine learning field and it consists of induce which class a determined object belongs given a finite number of classes. Generally, multiclass classification problems are more complexes than binary classification problems. The One-against-One approach is a common work-around to solve this problem, where the problem is divided in smaller binaries problems, however this approach uses only one type of classifier. The given study does an comparative analysis of the method Diversified One-Against-One (DOAD) [4], and it's basically a One-against-One approach but with a previous step to choose the binary classification method for every pair of the problem classes. The best candidates to solve each binary classification problem are chosen, ensuring that the best method for each pair will be used, among the given ones. They are then used in cooperation for the solution of the same multiclass problem.

## Lista de Abreviaturas e Siglas

|      |                                     |
|------|-------------------------------------|
| OAO  | <i>One-Against-One</i>              |
| OAR  | <i>One-Against-Rest</i>             |
| DOAO | <i>Diversified One-Against-One</i>  |
| ANN  | <i>Artificial Neural Network</i>    |
| DT   | <i>Decision Tree</i>                |
| RF   | <i>Random Forest</i>                |
| KNN  | <i>k-Nearest Neighbors</i>          |
| LDA  | <i>Linear Discriminant Analysis</i> |
| LR   | <i>Logistic Regression</i>          |
| SVM  | <i>Support Vector Machines</i>      |

## Sumário

|       |  |    |
|-------|--|----|
| 1     | Introdução . . . . .   | 8  |
| 1.1   | Objetivos . . . . .  | 9  |
| 2     | Aprendizado Supervisionado. . . . .                                | 11 |
| 2.1   | Classificação Multiclasse . . . . .                                | 11 |
| 2.1.2 | Árvores de Decisão . . . . .                                       | 12 |
| 2.1.3 | k-NN . . . . .   | 13 |
| 2.1.4 | Análise Discriminante Linear . . . . .                             | 14 |
| 2.1.5 | Regressão Logística . . . . .                                      | 14 |
| 2.1.6 | SVM . . . . .  | 15 |
| 2.1.7 | Random Forest . . . . .  | 15 |
| 2.2   | Decomposição de problemas multiclasse . . . . .                    | 16 |
| 2.2.1 | One-Against-One . . . . .  | 17 |
| 2.2.2 | One-Against-Rest . . . . .   | 18 |
| 2.3   | Combinação de Classificadores . . . . .                            | 19 |
| 2.3.1 | <i>VOTE One-Against-One</i> . . . . .                              | 20 |
| 3     | Um-Contra-Um Diversificado (Diversified One-Against-One) . . . . . | 22 |
| 3.1   | Algoritmo DOAO . . . . .   | 23 |
| 4     | Experimentos . . . . .   | 27 |
| 4.1   | Metodologia . . . . .  | 27 |
| 4.2   | Implementação . . . . .  | 28 |
| 5     | Resultados . . . . .   | 29 |
| 5.1   | DOAO vs OAOs . . . . .   | 30 |
| 5.2   | DOAO vs OARs . . . . .   | 31 |
| 5.3   | DOAO vs VOTE-OAO . . . . .   | 33 |
| 5.4   | DOAO vs Random Forest . . . . .                                    | 34 |
| 5.6   | Análise do Tempo de Execução . . . . .                             | 35 |
| 5.6   | Comparação das Implementações . . . . .                            | 38 |
| 6     | Conclusão . . . . .  | 41 |

|                               |    |
|-------------------------------|----|
| 6.1 Conclusão Final . . . . . | 41 |
| 7. Referências . . . . .      | 43 |

## 1. Introdução

Problemas de classificação abrangem grande parte dos problemas em aprendizagem de máquina. Suas aplicações são diversas e de grande importância quando se deseja que máquinas desempenhem funções que habitualmente são feitas por humanos, como reconhecimento de voz, reconhecimento facial, detecção de SPAM na caixa de e-mail [1] e categorização de células cancerígenas [6].

Nesses problemas, quando o número de classes é maior que dois, ele é chamado de problema de classificação multiclases. Podemos dividir os métodos para a resolução desses problemas em duas categorias, alguns deles são métodos inicialmente elaborados para problemas de classificação binária mas que podem ser naturalmente estendidos para o caso de multiclasse e outros em que o problema precisa de um tratamento especial para se adequar ao caso de várias classes.

A primeira categoria inclui as árvores de decisão, C4.5 [15] e CART [3], redes neurais [8], k-NN [9], Naive Bayes [10] e SVM [11] que podem ser diretamente estendidos para o caso multiclasse. A segunda abordagem parte do princípio que problemas multiclases são mais difíceis do que problemas de duas classes, pois as regiões de fronteiras de decisão tendem a ser mais complexas [4]. Tipicamente, nos casos dessa segunda categoria, as técnicas consistem em decompor o problema original de multiclases em um conjunto de problemas de classificação binária que podem ser resolvidos separadamente. Duas das abordagens mais utilizadas para esse tipo de decomposição são os métodos “Um-contra-um”, *One-Against-One* (OAO) e o método “Um-contra-todos”, *One-Against-Rest*.

Na abordagem OAO, a solução consiste basicamente em utilizar um algoritmo de classificação binário para treinar cada par de classes pertencentes ao problema e na predição utilizar algum método para agregar os resultados preditos por cada um deles e então inferir a classe do objeto. Uma estratégia de agregação simples é a *Voting strategy*, onde cada classificador binário dá seu voto para a classe predita, os votos são somados e a classe com mais votos ganha. Na decomposição *One-Against-Rest* (OAR), um problema de K classes é dividido em K problemas binários onde cada problema é resolvido por um classificador que é

responsável por distinguir uma classe de todas as outras. Um exemplo interessante é o modelo proposto em [16], onde cada neurônio de saída da rede neural é um discriminante entre a classe que ele representa e todas as outras classes. Assim como na OAO, o OAR também precisa de uma estratégia de agregação dos resultados dos classificadores na fase da teste, uma abordagem bastante comum é utilizar a confiança dos classificadores na predição da classe, onde a classe vencedora é a que obtém mais votos.

Além dos métodos citados, uma outra abordagem, bastante utilizada recentemente nos problemas de classificação, binários e de multiclasse, é a combinação de classificadores. Nesse campo de estudo, sistemas com múltiplos classificadores, se tornou bastante popular após a divulgação de estudos, na década de 1990, que mostraram que a combinação de classificadores podem trazer bons resultados [18]. Ela se baseia na premissa de que não existe um único classificador ótimo para todos os problemas. Logo, a abordagem em questão tenta unir uma variedade de classificadores e explorar o melhor de cada um para resolver diferentes problemas.

Recentemente, o método chamado *diversified one-against-one (DOAO)* [12] foi proposto, ele consiste numa variação do *one-against-one* utilizando o conceito de combinação de classificadores. Neste método, vários classificadores são usados para o treinamento de cada par de classes, apenas o melhor para cada par será escolhido e na validação será feita uma agregação dos resultados fornecidos pelos classificadores para enfim gerar o resultado final.

## 1.1 Objetivos

Este trabalho tem como objetivo geral a realização de uma análise experimental e comparativa do método de classificação DOAO com outros métodos de combinação de classificadores, métodos de redução de problemas de classificação multiclasse para classificação binária e métodos que segundo estudos recentes têm mostrado uma boa performance.

Dentre os métodos que serão analisados com o DOAO estarão o *One-Against-One* (OAO) e *One-Against-Rest* (OAR) utilizando os seguintes classificadores binários: Rede Neural Artificial (ANN), Árvore de Decisão (DT), k-Nearest-Neighbor (kNN), Análise discriminante linear (LDA), Regressão logística (LR) e Support Vector Machines (SVM). Estes algoritmos foram escolhidos pois os mesmos também foram utilizados pelos autores do método avaliado.

Juntamente com esses métodos, os mesmos experimentos também serão feitos com o *VOTE One-Against-One*, por ter uma abordagem semelhante ao DOAO, utilizando classificadores diferentes. O método *Random Forest* [20] também será avaliado e comparado ao DOAO, que pela sua versatilidade e simplicidade vem sendo utilizado na literatura. Além de ter se mostrado efetivo em diversas aplicações, como monitoramento de terras [21], prevenção de câncer [22], e de ser uma forma de comparar o algoritmo avaliado com a estratégia de combinação de classificadores.

## 2. APRENDIZADO SUPERVISIONADO

Os métodos de aprendizagem de máquina podem ser divididos em duas fases, o treinamento e a predição. Na fase de treinamento, é passado para o algoritmo um número de instâncias correspondente a um conjunto de dados, onde a partir desse treinamento ele poderá inferir qual classe cada instância pertence. Esse conjunto de dados é formado por diversas instâncias com um número de características (as quais representam esses dados no espaço de características), que podem ser nominais, contínuas e binárias. Cada uma dessas instâncias, naturalmente, pertence a uma classe, e quando esta informação é passada para o treinamento do algoritmo com a classe a qual ela corresponde, esse tipo de aprendizagem é chamado de aprendizagem supervisionado. Ou seja, no treinamento o método sabe a qual classe cada instância pertence. Ao contrário do aprendizado não-supervisionado, onde na etapa do treinamento o algoritmo desconhece a classe correta das instâncias e o método irá tentar dividir o conjunto de dados em grupos de classes.

Algoritmos supervisionados podem ser divididos em duas categorias: Regressão e Classificação. Ambas as categorias são de grande importância e amplamente utilizados na área de aprendizagem de máquina. Os métodos de regressão nos permitem fazer previsões de alvos contínuos, como qual vai ser a temperatura de determinada cidade em um dado dia, ou quanto vai estar custando uma ação da bolsa de valores. Enquanto que na classificação as respostas valores categóricos, onde os dados podem ser separados em classes específicas, como classificar um tumor como cancerígeno ou não, ou classificar uma imagem como imprópria ou não para postagem numa rede social.

### 2.1 Classificação Multiclasse

Classificação multiclasse é o nome dado a categoria dos problemas de classificação onde existem mais de duas classes a ser inferida. Ao contrário dos

problemas binários onde existem apenas duas classes a serem inferidas, e.g. prever se um e-mail é um *spam* ou não, na classificação multiclasse o método irá inferir, por exemplo, qual o gênero do filme que está sendo analisado.

Existem vários algoritmos que se propõem em reduzir esse problemas multiclasse em problemas de classificação binária. Alguns desses algoritmos de classificação são naturalmente binários mas podem ser utilizados para resolver problemas multiclasse como por exemplo o Naive Bayes, Regressão linear e k-NN, outros precisam de uma formulação diferente para que possam resolver problemas multiclasse. Nessa última categoria usualmente os problemas são reduzidos em problemas binários para então serem aplicados os algoritmos de classificação binária, e.g. máquinas de vetor de suporte. Nas próximas seções serão apresentados alguns classificadores para múltiplas classes que serão investigados neste trabalho.

### 2.1.1 Redes Neurais

Ao contrário do *perceptron*, que não pode ser treinado para inferir uma classe dentre várias outras classes. Uma rede neural do tipo *multilayer perceptron*, pode ser estendida para a resolução de problemas multiclases, ao invés de utilizar apenas um neurônio na camada de saída da rede, utilizamos N neurônios binários[14].

A primeira abordagem pode ser utilizada cada neurônio de saída correspondendo a uma determinada classe, então a saída no neurônio identificador da classe A seria 1 e nas outras seria 0. Então para um problema de 4 classes os vetores da saída seriam os apresentados na tabela 1.

Tabela 1 - Saída neurônio abordagem 1

|          |      |
|----------|------|
| Classe A | 1000 |
| Classe B | 0100 |
| Classe C | 0010 |
| Classe D | 0001 |

Uma segunda abordagem seria referenciar um vetor binário de 0 a  $2^N - 1$ , onde N é o número de neurônios de saída. Então a rede daria como resposta para uma determinada instância um vetor que mais se aproximasse ao vetor da sua classe, esse cálculo de quão próximo a resposta está, pode ser feito calculando a distância, e.g distância de Hamming. Por exemplo, um problema de 4 classes utilizando N=5 poderia ter uma saída como demonstrado na tabela 2. Se para alguma instância de teste a rede dá como resposta o vetor 11101, a distância de Hamming é calculada e é inferido que ela corresponde a classe C.

Tabela 2 - Saída rede neural abordagem 2

|          |       |
|----------|-------|
| Classe A | 00000 |
| Classe B | 00111 |
| Classe C | 11001 |
| Classe D | 11110 |

### 2.1.2 Árvores de Decisão

Árvore de decisão é um método de fácil entendimento e mas bastante poderoso, que pode ser naturalmente estendido para classificação multiclasse. A construção da árvore consiste basicamente em dividir o conjunto de treinamento que mais separa as classes. Ou seja, cada divisão é feita de modo que haja maior ganho de informação.

Os algoritmos de árvore de decisão mais utilizados são o ID3/C4.5[2] e CART[3]. A diferença entre esses algoritmos é a forma como é calculada a função de custo, ou seja, como é escolhida as características para a divisão dos dados. No ID3 e C4.5 é utilizada a entropia de Shannon e as características com maior ganho de informação são escolhidas como nós. No caso do algoritmo CART é utilizado o cálculo de impureza *Gini*. Uma medida de homogeneidade ou pureza dos nós, onde um nó onde só existem instâncias de uma mesma classe é considerado puro.

Na implementação, foi escolhido o algoritmo CART que também foi utilizado por [4] para fazer o estudo comparativo do DOAO.

### 2.1.3 k-NN

O algoritmo k-NN é bastante simples e de fácil implementação, porém muito eficiente e efetivo na área de reconhecimento de padrões, categorização de textos e reconhecimento de objetos [6].

Essa é uma técnica que não precisa do treinamento do conjunto de dados e a inferência das instâncias de teste é dada pela proximidade das 'k' instâncias do conjunto de teste mais próximas da instância de teste.

### 2.1.4 Análise Discriminante Linear

O algoritmo LDA é um método estatístico que basicamente busca achar uma combinação linear das variáveis de do conjunto de treinamento a fim de maximizar a distância entre as classes diferentes e reduzindo a distância entre classes iguais. Ver figura 1.

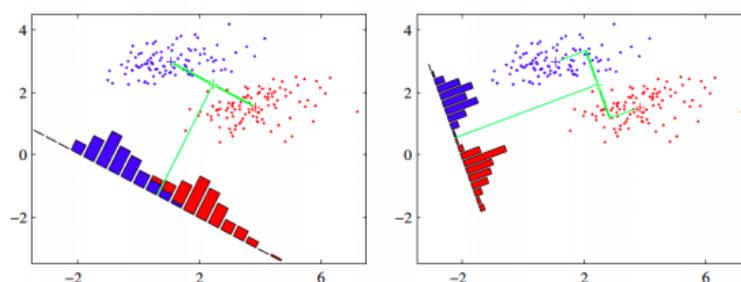


Figura 1 - Representação do algoritmo LDA - adaptado de [7]

Como pode-se observar na figura 1, o gráfico nos mostra que o conjunto de dados original apresenta uma grande sobreposição entre classes tanto no eixo X como no Y. Porém na transformação da esquerda essa sobreposição diminui, e na projeção da direita, correspondente ao LDA, essa sobreposição não existe mais.

### **2.1.5 Regressão Logística**

A regressão logística [27] é um modelo estatístico utilizado em diversas áreas como medicina, instituições financeiras e econometria. A técnica constrói uma função linear a partir da combinação linear das variáveis de entrada, onde os parâmetros podem ser encontrados utilizando a estimativa de máxima verossimilhança.

### **2.1.6 SVM**

Máquinas de vetor de suporte é um dos métodos de classificação mais robusto existentes [8]. Inicialmente, ele é designado apenas para problemas de classificação binária, porém existem diversas propostas para utilizá-lo em problemas multiclases.

A idéia principal do algoritmo consiste encontrar a borda máxima do hiperplano que separa as duas classes. Este mesmo algoritmo pode ser utilizado para problemas de classificação não linear utilizando funções kernel que mapeiem o espaço de entrada em um espaço de dimensão maior.

### **2.1.7 Random Forests**

O *Random Forest* [20] é um método de classificação supervisionado relativamente novo, proposto em 2001, que ficou muito popular por suas vantagens. Alguma de suas maiores vantagens são sua simplicidade de implementação e versatilidade, por funcionar bem com problemas de classificação e de regressão [26].

Este método é um tipo de combinação de classificadores que usa como algoritmo base o método *Decision Tree*. Como o nome do algoritmo deixa claro, no final da implementação tem-se um grupo de Árvores de decisão, formando uma floresta. Para formar essa floresta, inicialmente, o algoritmo divide o conjunto de dados inicial em diversos subconjuntos aleatórios. A partir de cada um desses

subconjuntos é construído uma árvore de decisão, o algoritmo abaixo demonstra como é feita essa construção.

**Algoritmo 1.** Construção de uma árvore do *Random Forest*

**Entrada:** Subconjunto de treinamento  $D = \{f_{i1}, f_{i2}, f_{i3}, \dots, f_{iM}, y_i\}_{i=1}^M$ , sendo 'f' as variáveis e 'y' a classe correspondente.

1: **procedimento** RandomForestTREE:

2:     **para** <cada nó> **faça**:

3:         Escolha aleatoriamente m variáveis, sendo  $m \ll M$

4:         Dentre as m variáveis ache a melhor bifurcação.

5:     **fim\_para**

6: **fim\_procedimento**

Após executar o algoritmo 1 para cada um dos N subconjuntos, a floresta está pronta. Na fase da predição, a nova entrada é executada por todas as árvores da floresta e então cada floresta dará seu voto para a classe predita, a resposta final da *Random Forest* será a classe que obtiver o voto majoritário.

## 2.2 Decomposição de Problemas Multiclasse

Problemas de multiclases são normalmente mais complexos que problemas de classificação binária. Isto se deve às suas fronteiras de decisão que tendem a ser mais complexas [13]. Diversos métodos já foram desenvolvidos na área de binarização de problemas multiclases. Uma das técnicas mais conhecidas consiste em dividir o conjunto de dados original em subconjuntos de duas classes onde para cada subconjunto é usado um método binário e a saída desses classificadores binários são combinadas para a decisão final.

Neste trabalho, serão utilizadas para comparação com o DOAO as duas das formas mais utilizadas para a decomposição em problemas binários, o *One-Against-One* e *One-Against-Rest*.

### 2.2.1 One-Against-One

Neste tipo de decomposição, as classes de um problema de  $K$  classes são combinadas dois a dois de forma que ele seja dividido em  $\frac{K(K-1)}{2}$  problemas binários. Ou seja, ao invés de resolver um problema, provavelmente mais complexo, de  $K$  classes, ele irá resolver  $\frac{K(K-1)}{2}$  problemas de duas classes. A figura 2 ilustra como funciona a divisão, mostrando um problema de classificação de três classes sendo dividido em 3 problemas binários.

Na fase do treinamento desse algoritmo, são treinados  $\frac{K(K-1)}{2}$  classificadores binários e são apresentados a eles apenas um subconjunto do conjunto de treinamento contendo apenas instâncias que correspondem a uma das duas classes que o classificador está sendo treinado. Sendo assim, cada um dos classificadores são treinados para distinguir instâncias dentre duas classes.

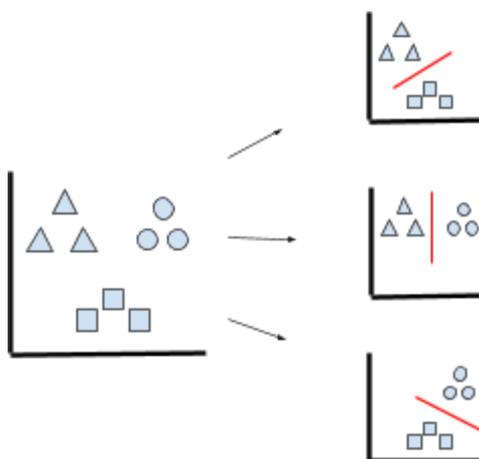


Figura 2 - Ilustração do funcionamento do OAO

Na fase da validação, uma instância é passada para cada um desses classificadores binários e então os resultados são agregados de alguma forma antes de decidir o resultado final. Existem diversas formas de agregar essas respostas dadas pelos classificadores. Uma forma bastante popular é basear-se na confiança da resposta dos classificadores, então a resposta final do método será a mesma do classificador que tiver maior confiança em sua resposta.

Outros métodos para agregação dos resultados são a regra *Max-Wins* [22] (também conhecida como *Voting*) e o *Weighted voting* [22]. Na abordagem

*Max-Wins* [22], cada classificador binário dará um voto para a classe predita, os votos são contados e a classe com mais votos ganha. O *Weighted voting* funciona de maneira semelhante ao *Max-Wins*, porém cada voto tem um peso, onde esse peso é dado pela confiabilidade da predição de cada classificador.

Para este trabalho, foi utilizada a regra *Max-Wins*, pois a mesma também foi utilizada pelo autor do DOAO [4] quando comparou o mesmo com o OAO. Outra vantagem da regra *Max-Wins* é ter uma implementação mais simples diante dos outros métodos e ser possível utilizá-lo no DOAO [4]. Esta mesma regra também foi utilizada para todos os métodos de decomposição, garantindo que a escolha do método de agregação não influencie para um melhor desempenho, ou não, do algoritmo avaliado.

### **2.2.2 One-Against-Rest**

O método de decomposição *One-Against-Rest* (OAR) divide o problema de K classes em K problemas binários. Onde K classificadores são treinados para distinguir uma classe de todas as outras. Ou seja, cada classificador irá inferir se determinada instância pertence ou não pertence a determinada classe. A figura 3 ilustra como funciona a decomposição do OAR.

Ao contrário do *One-Against-One*, que na fase de treinamento só recebem um subconjunto do conjunto de dados, o *One-Against-Rest* recebe todo o conjunto de dados, ou um subconjunto do conjunto de dados inicial, porém contendo instâncias pertencentes a todas as classes do problema.

Na fase de validação, uma instância é dada para cada um dos classificadores binários e o classificador com a resposta positiva será a resposta final do método. Assim como na decomposição OAO, também existem diversas formas de agregar os resultados dos classificadores binários. Porém para o presente trabalho preferiu-se utilizar o mesmo que foi utilizado em [4], por ser de mais fácil implementação e se assemelha ao *Max-Wins* utilizado no OAO e no DOAO [4].

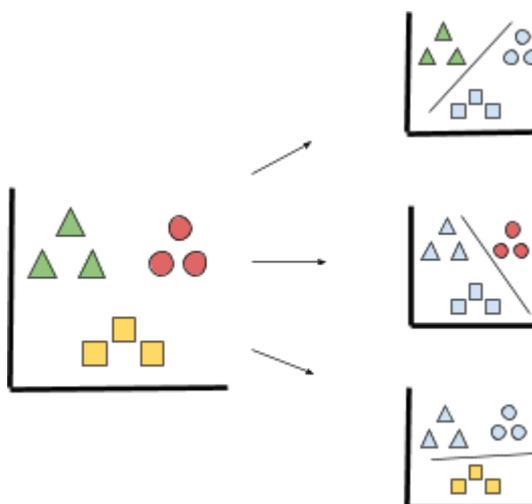


Figura 3 - Ilustração da decomposição *One-Against-Rest*

### 2.3 Combinação de Classificadores

A idéia básica por trás da combinação de classificadores pode ser resumida como um método que cria um conjunto de diferentes classificadores e a partir das decisões de cada um desses definir uma resposta final. Esse tipo de abordagem é bem intuitivo já que imita nosso comportamento natural de levar em consideração várias opiniões antes de tomar uma decisão crucial [23].

De acordo com [24] existem três principais motivações para combinar classificadores. A motivação do “pior caso”, onde é possível evitar um classificador muito ruim fazendo uma média com vários classificadores. A motivação do “melhor caso”, onde em alguns casos particulares, a combinação de classificadores diferentes podem melhorar o desempenho do melhor classificador individual. E a motivação computacional, onde alguns algoritmos de otimização, como o *Backpropagation*, são inicializados com valores aleatórios e podem cair em mínimos locais sendo necessário iniciá-los diversas vezes. Frequentemente, nestes algoritmos, são utilizadas várias inicializações para determinar um classificador ótimo, o que pode ser custoso. Enquanto que a combinação desses tipos de classificadores mostraram uma ganho da estabilidade e uma melhora na performance.

Dois dos métodos de combinação mais conhecidos são o *Bagging* [28] e o *Boosting* [30], onde o Boosting em especial possui uma grande quantidade de variações, como o *AdaBoost* [29], *GradientBoost* [5], entre outros. Porém os mesmos tem uma abordagem bastante diferente do DOAO [4]. Pois o *Bagging* e o *Boosting* ao contrário do DOAO, buscam construir um conjunto de modelos de classificação usando o mesmo algoritmo. Para o presente trabalho, foi escolhido o *Random Forest* [20] (um algoritmo do tipo *Bagging*) para ser experimentado e comparado ao método DOAO [4].

No entanto, no artigo em que o DOAO [4] foi proposto, também foram feitos os mesmos experimentos com um outro método de combinação, servindo como base de comparação. O método em questão é o VOTE One-Against-One, o mesmo também foi avaliado pelo presente estudo e será melhor explicado na seção 2.3.1.

### **2.3.1 VOTE One-Against-One**

O método VOTE-OAO foi experimentado pelo autor do DOAO [4] para servir de comparação com o mesmo. Pois ambos tem uma abordagem semelhante, combinando vários classificadores em conjunto para a solução de um problema multiclasse. Porém ele é um método mais “ingênuo” que o DOAO e espera-se que ele se mostre menos efetivo.

O VOTE-OAO, assim como no OAO, divide o problema de  $K$  classes em  $K(K-1)/2$  problemas binários. Onde, na fase de treinamento, para cada problema binário serão treinados todos os classificadores candidatos. Sendo assim, se tivermos  $C$  classificadores candidatos, teremos um total de  $\frac{C \times K(K-1)}{2}$  classificadores treinados. Na fase da validação, a instância a ser predita será apresentada para cada um desses classificadores, que votarão na classe e então a classe com mais votos será a resposta final do classificador.

Como pode-se perceber, não existe nenhuma escolha dos melhores classificadores para cada par de classe e todos os classificadores treinados são usados na fase da validação. Veremos na seção 3, que essa é a principal diferença

do DOAO, pois ele irá utilizar o erro de validação para selecionar os melhores classificadores para cada par de classe.

### 3. DIVERSIFIED ONE-AGAINST-ONE

O método avaliado consiste na construção de um classificador para problemas multi-classes baseado na abordagem *one-against-one*, chamado DOAO. Diferente de classificadores OAO que usam apenas um classificador como base, treinado par-a-par, o DOAO pode ser formado com diferentes tipos de classificadores, sendo estes os melhores classificadores para cada subconjunto de dados “um-contra-um”. Um par de classe OAO é entendido como um subconjunto dos dados  $D$ , na qual apenas estão presentes as instâncias nas quais a classe objetivo é uma classe do par  $D_{ij}$ .

Supondo que um conjunto de dados com  $N$  instâncias é dado,  $D = \{x_i, y_j\}_{i=1}^N$ , onde  $x_i$  é o vetor de atributos de entrada e  $y_j$  o vetor de classes com  $y_j \in \{1, \dots, c\}$ . Então para cada par de classes  $D_{ij}$  são construídos classificadores candidatos a partir de um conjunto previamente definido  $\{A_1, \dots, A_M\}$ . Logo, a partir dos classificadores candidatos, erros de validação são computados e o classificador com o menor erro é escolhido. Assim, como no OAO, são escolhidos  $c(c-1)/2$  classificadores, sendo cada um o mais competente para distinguir o par de classes correspondente.

#### Algoritmo 2. DOAO

**Entrada:** conjunto de treinamento  $D = \{x_i, y_j\}_{i=1}^N$ ,  $y_j \in \{1, \dots, c\}$ , algoritmos de classificação candidatos  $\{A_1, \dots, A_M\}$ .

**Saída:** Conjunto de classificadores  $\mathbb{C}$

1: **procedimento** DOAO:

2:  $\mathbb{C} \leftarrow \phi$

3: **para** par de classes  $(i,j)$  **faça**

4:  $D_{i,j} \leftarrow$  Conjunto de dados onde os elementos são das classes  $i$  ou  $j$

5: treinar cada classificador candidato

6: obtenha o erro de validação pra cada candidato

7: achar o classificador  $C_{A\_best, D_{ij}}$  que corresponde ao erro de validação mínimo

8:  $\mathbb{C} \leftarrow \mathbb{C} \cup C_{A\_best, D_{ij}}$

9: **fim\_para**

10: **fim\_procedimento**

A figura 4 apresenta um exemplo ilustrativo do método proposto em [4] para um problema de três classes. Na figura, círculos, triângulos e retângulos representam as classes 1, 2 e 3, respectivamente. As figuras 1(a), 1(b) e 1(c) apresentam os limites de decisão obtidos a partir de classificadores OAO baseados em um único algoritmo de aprendizagem. A partir de uma análise visual da figura, se pode perceber que os classificadores  $A_1$ ,  $A_2$  e  $A_3$  são os melhores candidatos para distinguir os pares de classes (1,2), (2,3), (1,3), respectivamente. Logo, o método proposto seleciona o algoritmo mais competente para cada fronteira de decisão, como mostrado na figura 1(d).

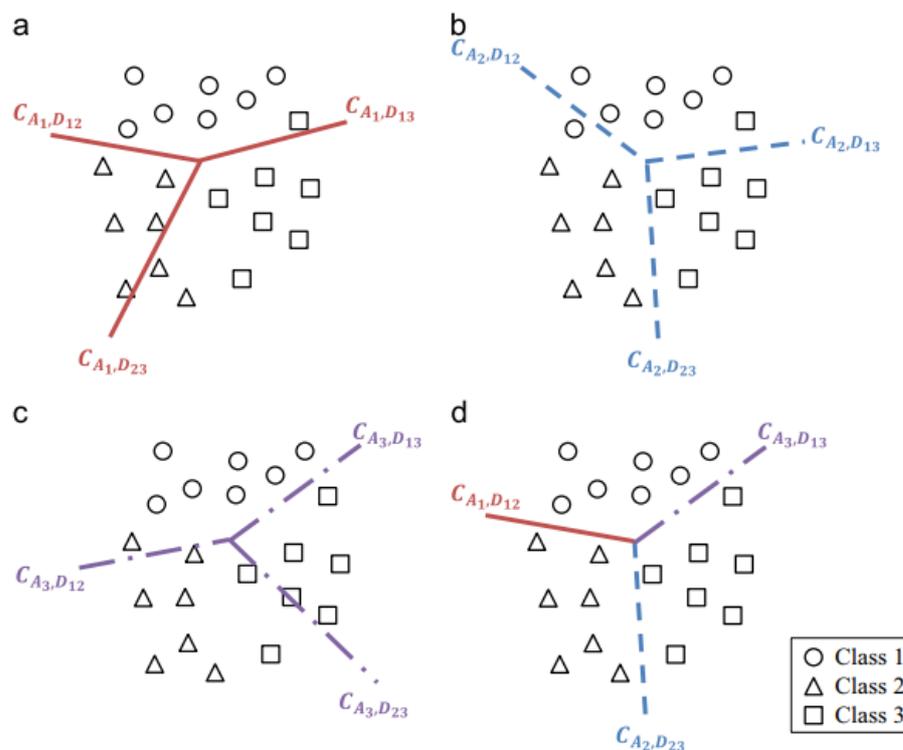


Figura 4 - Exemplo ilustrativo da construção do classificador DOAO - adaptado de [4]

## 4 EXPERIMENTOS

### 4.1 Metodologia

O DOAO foi comparado com seis classificadores OAO baseados em um único algoritmo de classificação. São eles o ANN-OAO, DT-OAO, kNN-OAO, LDA-OAO, LR-OAO e SVM-OAO, em que cada um desses métodos usa a estratégia de decomposição *One-Against-One* usando como classificador binário o respectivo algoritmo. Por exemplo, o método DT-OAO decompõe o problema com o *One-Against-One* e utiliza Árvore de Decisão para classificar os problemas binários. De modo similar, o DOAO também foi comparado com seis classificadores *One-Against-Rest* (ANN-OAR, DT-OAR, kNN-OAR, LDA-OAR, LR-OAR e SVM-OAR) e também, com o método *Random Forest*. A configuração dos parâmetros para cada um desses algoritmos foi a mesma escolhida em [4] e estão na tabela 3.

Tabela 3 - Configuração dos parâmetros dos métodos

| Algoritmos | Configuração dos Parâmetros  |
|------------|--|
| ANN        | No. <i>Hidden Nodes</i> =3,4,5,...,20<br>Max. Iterações = 300                            |
| DT         | Min. atributos em uma nó folha=1,2,3,5<br>Min. atributos em um nó pai=5,10<br>Prune=True |
| k-NN       |  |
| LDA        | Sem parâmetros   |
| LR         | Sem parâmetros   |
| SVM        | $C = 2^{-3}, \dots, 2^{10}$<br>Tipo do Kernel=RBF<br>$\alpha = 2^{-5}, \dots, 2^5$       |
| RF         | Número de atributos=2,3<br>Profundidade Máxima = Ilimitada                               |

Também foi comparado com o DOAO o método baseado em votação VOTE-OAO, explicado na seção 2.3.1, onde não existe uma escolha do melhor algoritmo de classificação (como no DOAO) para cada par de classe. O VOTE-OAO

faz uma votação entre os classificadores para uma determinada instância e infere a classe pelo voto majoritário. Ou seja, enquanto o DOAO escolhe um conjunto de classificadores (os classificadores especialistas para cada par), o VOTE-OAO não escolhe um conjunto de classificadores, e sim, ele coleta os votos (resultados das classificações individuais de cada classificador candidato) e avalia o resultado majoritário.

Para comparar os classificadores o conjunto de dados foi dividido pelo método *k-Fold cross validation* com  $k=10$ , onde o *k-Fold* divide os conjuntos de dados em  $k$  conjuntos disjuntos (com um décimo do tamanho do conjunto de dados) e de mesmo tamanho. Neste processo 9 folds (partição do conjunto de dados) são utilizados para treinamento e 1 fold para teste. Este processo é realizado de modo que cada fold seja usado uma vez como conjunto de teste, assim, se repete 10 vezes e todo conjunto de dados é avaliado.

Para cada execução de um *fold* é computado um erro, sendo este a soma de predições falhas do classificador sobre uma instância do conjunto de teste, o somatório do erro de cada *fold* é então dividido pelo tamanho do conjunto de dados (total de instâncias) e multiplicado por 100 para representar o valor em porcentagem. Assim, este se torna o critério de performance para avaliar os classificadores, onde a Taxa de Erro (em %) fica definida como na equação abaixo.

$$\text{Taxa de Erro} = \frac{1}{N} \sum_{i=1}^N 1_{(se\ y_i \neq y'_i)} \times 100$$

No qual  $N$  é o número de instâncias no conjunto de dados,  $y$  é a classe real e  $y'$  é a classe predita pelo classificador.

Ou seja, no DOAO, na fase do treinamento de cada par de classes é calculado o erro de validação para cada classificador candidato. Vale observar que, como mostrado na tabela 3, para cada classificador existem diversas configurações dos parâmetros e o erro de validação foi calculado para cada uma dessas configurações, a fim de se determinar a melhor configuração para cada um deles. Isto é, para o kNN, por exemplo, foi calculado o erro de validação para  $k$  igual a

1,3,5,7,10,20 e 30 e apenas o kNN com a configuração com menor erro de validação foi comparada aos outros classificadores candidatos.

Esse erro de validação é calculado utilizando o *10-fold cross validation*, como explicado no parágrafo anterior, o classificador que tiver menor erro de validação é escolhido como o melhor para aquele determinado par de classes. Caso ocorra empate entre os classificadores, um deles será escolhido aleatoriamente. A taxa de erro percentual foi definida como a soma dos erros para cada fold de teste dividido pelo número de folds de teste utilizados, no nosso caso 10. Por fim, todos os atributos dos conjuntos de dados foram normalizados no intervalo de [-1, 1], seguindo o método *Feature Scaling*, apresentado na equação [1].

$$x_{ij}' = \frac{2 \times (x_{ij} - x_{iMin})}{(x_{iMax} - x_{iMin})} - 1, \text{ onde } x_{ij} = \text{atributo a ser normalizado.} \quad [1]$$

Também foi feito um teste de hipótese ao final dos experimentos a fim de verificar se os métodos DOAO, VOTE-OAO e *Random Forest* têm a mesma precisão ou tem precisão diferentes. O teste foi feito utilizando o *T-Pair Test* bilateral no *k-Fold Cross Validation*. Primeiramente supõe-se uma hipótese nula de que os dois métodos tem precisão igual e uma hipótese alternativa de que algum dos dois métodos tem precisão melhor que o outro. Representado abaixo:

$$H_0 : \mu_{DOAO} = \mu_{RF}$$

$$H_1 : \mu_{DOAO} \neq \mu_{RF}$$

Em seguida mede-se o erro de validação de cada método para cada *Fold* em um dos banco de dados. E por fim aplica-se a equação do *T-pair Student*.

$$t = \frac{\bar{d}}{S_{cd} \times \sqrt{k}} \quad S_{cd} = \sqrt{\frac{\sum_{j=1}^k (d_j - \bar{d})^2}{(k-1)}}$$

Sendo  $\bar{d}$  a média da diferença amostral e  $k$  o número de *Folds*. Por fim, após o cálculo de 't', a partir da tabela de distribuição t-Student, para  $k=10$  e uma significância de 5%, temos a curva da figura 4, onde a área pintada representa valores de 't' em que a hipótese nula é rejeitada. O resultado do teste de hipótese dos métodos DOAO e *Random Forest* será demonstrado na seção 5.

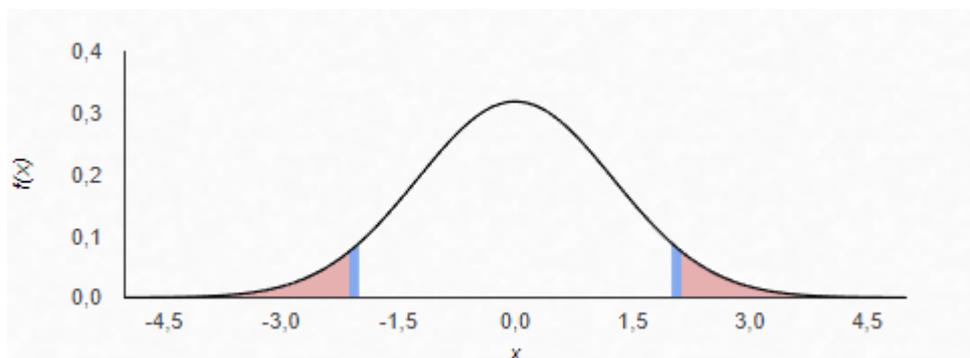


Figura 4 - Área onde a hipótese nula é rejeitada

## 4.2 Implementação

A implementação de todos os métodos citados na seção anterior foi feita em Python 3.6 com o auxílio das bibliotecas Scikit Learn [24] e NumPy [25]. Os equipamentos utilizados possuem uma configuração de máquina usual, processador: Intel® Core™ i5-4200U CPU @ 1.60GHz × 4, memória: 4 GB.

Os banco de dados utilizados nos experimentos foram todos obtidos no repositório de *datasets* do UCI, são eles: *Zoo*, *Iris*, *Wine*, *Seed*, *Glass*, *Ecoli*, *Movement*, *Balance*, *Landcover*, *Vehicle*, *Annealing*, *Vowel*, *Yeast*, *Car Evaluating*, *Segment*. O número de instâncias, atributos e classes de cada um desses conjunto de dados estão resumidos na tabela 4.

Tabela 4 - Bancos de dados utilizados no experimento

| <i>Dataset</i> | No. Instâncias | No. Atributos | No. Classes |
|----------------|----------------|---------------|-------------|
| Zoo            | 101            | 16            | 7           |
| Iris           | 150            | 4             | 3           |
| Wine           | 178            | 13            | 3           |
| Seed           | 210            | 7             | 3           |
| Glass          | 214            | 9             | 6           |
| Ecoli          | 336            | 7             | 8           |
| Movement       | 360            | 90            | 15          |
| Balance        | 625            | 4             | 3           |
| LandCover      | 675            | 147           | 9           |
| Vehicle        | 846            | 18            | 4           |
| Vowel          | 990            | 10            | 11          |
| Yeast          | 1484           | 8             | 10          |
| Segment        | 2310           | 19            | 7           |

## 5. RESULTADOS

Nesta seção serão analisados os desempenhos para cada um dos algoritmos implementados. Onde o DOAO será comparado com cada uma das categorias de algoritmos implementados.

A tabela 5 nos mostra uma visão geral dos erros percentuais obtidos pela realização dos experimentos de todos os algoritmos em todos os banco de dados. Os valores em negrito correspondem ao menor erro de validação calculado para determinado banco de dados.

Tabela 5 - Erro (%) de cada método analisado

|          | Zoo                   | Iris                  | Wine                  | Seed                  | Glass                   | Ecoli                  | Mov.                   | Bala.                 | L.Cov.                 | Veh.                   | Vowel                 | Yeast                  | Seg.                  |
|----------|-----------------------|-----------------------|-----------------------|-----------------------|-------------------------|------------------------|------------------------|-----------------------|------------------------|------------------------|-----------------------|------------------------|-----------------------|
| ANN-OAR  | 57,42<br>± 12,42      | 16,00<br>± 9,04       | 51,68<br>± 9,82       | 34,76<br>± 9,77       | 64,48<br>± 3,98         | 57,29<br>± 7,40        | 47,89<br>± 3,89        | 43,24<br>± 5,44       | 83,40<br>± 3,89        | 78,25<br>± 2,28        | 56,55<br>± 2,82       | 53,49<br>± 7,12        | 19,11<br>± 5,19       |
| DT-OAR   | 5,94<br>± 4,02        | 5,33<br>± 6,96        | 11,80<br>± 6,61       | 9,05<br>± 8,46        | 31,31<br>± 10,53        | 19,35<br>± 6,70        | 33,33<br>± 5,45        | 19,20<br>± 6,02       | 20,15<br>± 4,45        | 29,55<br>± 6,45        | 19,39<br>± 4,69       | 46,02<br>± 4,11        | 11,91<br>± 4,09       |
| KNN-OAR  | 3,96<br>± 3,20        | 4,67<br>± 4,42        | 5,06<br>± 3,51        | 5,71<br>± 7,43        | 29,91<br>± 4,98         | 15,18<br>± 9,49        | 15,28<br>± 7,99        | 12,00<br>± 3,47       | 22,67<br>± 4,52        | 28,84<br>± 3,48        | 1,01<br>± 1,52        | 42,12<br>± 4,65        | 12,86<br>± 4,00       |
| LDA-OAR  | 6,93<br>± 6,23        | <b>2,00</b><br>± 4,26 | 1,69<br>± 2,21        | <b>3,81</b><br>± 5,65 | 39,25<br>± 8,78         | 13,99<br>± 4,28        | 34,72<br>± 5,47        | 13,76<br>± 4,10       | 20,74<br>± 5,15        | 22,22<br>± 4,06        | 39,29<br>± 3,93       | 41,24<br>± 3,31        | 9,05<br>± 2,50        |
| LR-OAR   | 5,94<br>± 15,32       | 10,00<br>± 4,00       | 1,69<br>± 2,59        | 9,05<br>± 5,73        | 42,06<br>± 8,97         | 14,29<br>± 6,41        | 28,89<br>± 3,33        | 14,24<br>± 3,47       | 18,37<br>± 4,25        | 25,77<br>± 4,35        | 47,58<br>± 2,64       | 42,45<br>± 10,49       | 13,33<br>± 6,04       |
| SVM-OAR  | 28,71<br>± 12,04      | 8,00<br>± 4,00        | 4,49<br>± 2,76        | 6,19<br>± 5,30        | 30,37<br>± 9,41         | 35,12<br>± 8,48        | 16,94<br>± 6,47        | <b>3,84</b><br>± 1,57 | 16,15<br>± 4,00        | <b>16,08</b><br>± 2,80 | 2,15<br>± 0,92        | 40,50<br>± 9,12        | 12,86<br>± 5,17       |
| ANN-OAO  | 5,94<br>± 15,00       | 16,67<br>± 9,45       | 51,69<br>± 12,16      | 24,76<br>± 9,77       | 42,06<br>± 3,98         | 14,29<br>± 7,49        | 28,89<br>± 3,09        | 14,24<br>± 6,62       | 18,37<br>± 3,89        | 25,77<br>± 2,28        | 47,58<br>± 3,70       | 42,45<br>± 6,11        | 13,33<br>± 4,03       |
| DT-OAO   | 5,94<br>± 6,43        | 5,33<br>± 6,96        | 11,80<br>± 12,82      | 10,47<br>± 8,46       | 32,24<br>± 10,53        | 20,54<br>± 6,70        | 33,33<br>± 5,45        | 19,20<br>± 5,71       | 21,19<br>± 4,45        | 29,43<br>± 5,26        | 19,39<br>± 4,21       | 46,77<br>± 8,19        | 11,91<br>± 3,19       |
| KNN-OAO  | 3,96<br>± 6,63        | 4,67<br>± 4,42        | 5,06<br>± 3,51        | 6,19<br>± 7,43        | 29,91<br>± 4,98         | 19,64<br>± 9,49        | 15,28<br>± 7,99        | 12,00<br>± 5,48       | 22,67<br>± 4,52        | 28,84<br>± 3,84        | 1,90<br>± 1,52        | 46,77<br>± 10,01       | 12,86<br>± 5,94       |
| LDA-OAO  | 6,93<br>± 10,44       | <b>2,00</b><br>± 4,26 | 1,69<br>± 2,28        | <b>3,81</b><br>± 5,65 | 39,25<br>± 8,78         | 13,99<br>± 4,28        | 34,72<br>± 5,48        | 13,76<br>± 4,10       | 20,74<br>± 5,15        | 22,22<br>± 4,06        | 39,29<br>± 3,93       | 41,24<br>± 4,98        | 9,05<br>± 1,12        |
| LR-OAO   | 5,94<br>± 6,63        | 10,00<br>± 4,00       | 1,69<br>± 2,59        | 7,14<br>± 5,73        | 42,06<br>± 8,97         | 14,29<br>± 6,41        | 28,89<br>± 3,33        | 14,24<br>± 4,47       | 18,37<br>± 4,25        | 25,77<br>± 4,35        | 47,58<br>± 2,64       | 42,45<br>± 6,10        | 13,33<br>± 2,90       |
| SVM-OAO  | 28,71<br>± 15,01      | 8,00<br>± 4,00        | 14,05<br>± 6,12       | 9,04<br>± 8,09        | 30,84<br>± 9,41         | 35,42<br>± 7,99        | 27,22<br>± 9,78        | <b>3,84</b><br>± 0,73 | 15,85<br>± 4,02        | 18,68<br>± 2,81        | 16,06<br>± 3,49       | 40,97<br>± 5,76        | 22,86<br>± 4,70       |
| VOTE-OAO | 8,91<br>± 14,10       | 4,67<br>± 8,00        | 1,69<br>± 3,33        | 4,76<br>± 3,49        | 51,40<br>± 16,58        | 21,13<br>± 6,79        | 47,22<br>± 9,06        | 20,80<br>± 5,16       | 24,74<br>± 4,44        | 23,52<br>± 1,81        | 25,35<br>± 2,22       | 72,17<br>± 7,02        | 10,95<br>± 9,77       |
| RF       | <b>3,81</b><br>± 8,30 | 4,66<br>± 4,46        | 1,10<br>± 2,26        | 6,66<br>± 6,75        | <b>19,63</b><br>± 10,53 | <b>12,21</b><br>± 6,95 | <b>14,16</b><br>± 5,93 | 17,11<br>± 6,05       | <b>14,92</b><br>± 4,65 | 24,94<br>± 4,08        | 1,71<br>± 2,34        | <b>36,65</b><br>± 5,60 | <b>2,34</b><br>± 1,33 |
| DOAO     | 5,94<br>± 8,30        | 10,00<br>± 5,20       | <b>0,56</b><br>± 2,28 | 4,76<br>± 6,63        | 29,91<br>± 11,13        | 14,29<br>± 7,85        | 14,72<br>± 7,99        | <b>3,84</b><br>± 0,73 | 16,15<br>± 4,04        | 18,68<br>± 2,81        | <b>1,01</b><br>± 1,52 | 39,89<br>± 4,94        | 11,91<br>± 7,51       |

Pelos resultados apresentados pode ser observado que o DOAO produziu as menores taxas de erro, em comparação com os demais métodos analisados no artigo onde o DOAO foi proposto [4], para a 3 conjuntos de dados.

### 5.1 DOAO vs OAOs

Pela tabela 6 podemos observar que em cerca de 46% datasets o DOAO foi pior que os algoritmos OAOs. Isso indica que DOAO nem sempre seleciona o melhor classificador para cada par de classes, que é uma consequência por se basear na taxa de erro de validação que nem sempre leva a um baixo erro de teste.

Tabela 6 - Erros (%) do DOAO e dos métodos OAOs

| Dataset   | ANN-OAO          | DT-OAO           | KNN-OAO                | LDA-OAO                | LR-OAO          | SVM-OAO                | VOTE-OAO         | DOAO                    |
|-----------|------------------|------------------|------------------------|------------------------|-----------------|------------------------|------------------|-------------------------|
| Zoo       | 5,94<br>± 15,00  | 5,94<br>± 6,43   | <b>3,96</b><br>± 6,63  | 6,93<br>± 10,44        | 5,94<br>± 6,63  | 28,71<br>± 15,01       | 8,91<br>± 14,10  | 5,94<br>± 8,30          |
| Iris      | 16,67<br>± 9,45  | 5,33<br>± 6,96   | 4,67<br>± 4,42         | <b>2,00</b><br>± 4,26  | 10,00<br>± 4,00 | 8,00<br>± 4,00         | 4,67<br>± 8,00   | 10,00<br>± 5,20         |
| Wine      | 41,69<br>± 12,16 | 11,80<br>± 12,82 | 5,06<br>± 3,51         | 1,69<br>± 2,28         | 1,69<br>± 2,59  | 14,05<br>± 6,12        | 1,69<br>± 3,33   | <b>0,56</b><br>± 2,28   |
| Seed      | 34,05<br>± 9,77  | 9,05<br>± 8,46   | 6,19<br>± 7,43         | <b>3,81</b><br>± 5,65  | 9,05<br>± 5,73  | 5,24<br>± 8,09         | 4,76<br>± 3,49   | 5,24<br>± 6,63          |
| Glass     | 42,06<br>± 3,98  | 32,24<br>± 10,53 | <b>29,91</b><br>± 4,98 | 39,25<br>± 8,78        | 42,06<br>± 8,97 | 30,84<br>± 9,41        | 51,40<br>± 16,58 | <b>29,91</b><br>± 11,13 |
| Ecoli     | 14,29<br>± 7,49  | 20,54<br>± 6,70  | 19,64<br>± 9,49        | <b>13,99</b><br>± 9,49 | 14,29<br>± 6,41 | 35,42<br>± 7,99        | 21,13<br>± 6,79  | 14,29<br>± 7,85         |
| Movement  | 28,89<br>± 3,09  | 33,33<br>± 5,45  | 15,28<br>± 7,99        | 34,72<br>± 5,48        | 28,89<br>± 3,33 | 27,22<br>± 9,78        | 47,22<br>± 9,06  | <b>14,72</b><br>± 7,99  |
| Balance   | 14,24<br>± 6,62  | 19,20<br>± 5,48  | 12,00<br>± 5,48        | 13,76<br>± 4,10        | 14,24<br>± 4,47 | <b>3,84</b><br>± 0,73  | 20,80<br>± 5,16  | <b>3,84</b><br>± 0,73   |
| Landcover | 18,37<br>± 3,89  | 21,19<br>± 4,45  | 22,67<br>± 4,52        | 20,74<br>± 5,15        | 18,37<br>± 4,25 | <b>15,85</b><br>± 4,02 | 24,74<br>± 4,44  | 16,15<br>± 4,04         |
| Vehicle   | 55,77<br>± 2,28  | 29,43<br>± 5,26  | 28,84<br>± 3,84        | 22,22<br>± 4,06        | 25,77<br>± 4,35 | <b>18,68</b><br>± 2,81 | 23,52<br>± 1,81  | <b>18,68</b><br>± 2,81  |
| Vowel     | 47,58<br>± 3,70  | 19,39<br>± 4,21  | 1,01<br>± 1,52         | 39,29 ±<br>3,93        | 47,58<br>± 2,64 | 16,06<br>± 3,49        | 25,35<br>± 2,22  | <b>1,01</b><br>± 1,52   |
| Yeast     | 42,45<br>± 6,11  | 46,77<br>± 8,19  | 46,77<br>± 10,01       | 41,24<br>± 4,98        | 42,45<br>± 6,10 | 40,97 ±<br>5,76        | 72,17<br>± 7,02  | <b>39,89</b><br>± 4,94  |
| Segment   | 13,33<br>± 4,03  | 11,91<br>± 3,19  | 12,86<br>± 5,94        | <b>9,05</b><br>± 1,12  | 13,33<br>± 2,90 | 22,86<br>± 4,70        | 10,95<br>± 9,77  | 11,91<br>± 7,51         |

Os resultados também mostram que em 46,15% dos casos os OAOs foram melhores ou empataram com o DOAO, contra 53,8% de casos onde o DOAO foi melhor ou empatou com algum OAO. O gráfico representado na figura 5 nos mostra mais claramente a porcentagem de vitórias, empates e derrotas do DOAO em relação aos OAOs.

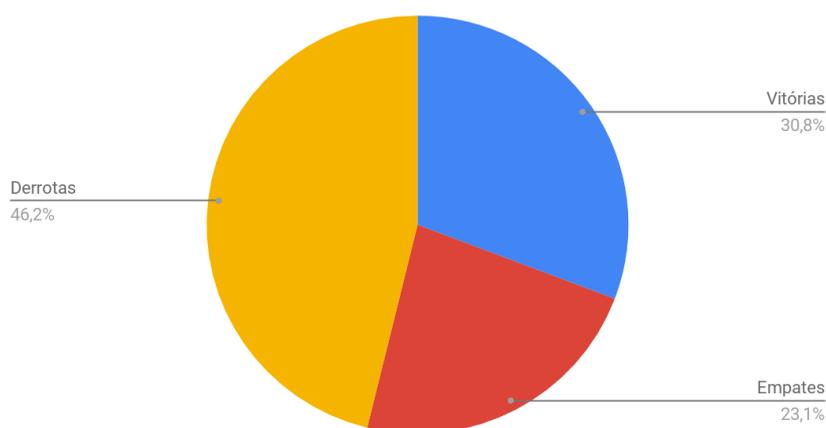


Figura 5 - Gráfico pizza DOAO vs OAOs

De modo geral, apesar do número de derrotas do DOAO em relação aos OAOs, verifica-se uma grande eficiência do mesmo. Pois se comparado dois a dois com cada um dos algoritmos OAO, percebe-se que o método proposto terá um erro percentual menor na maioria dos casos. Como por exemplo comparado ao SVM-OAO, que teve o melhor desempenho, o DOAO apresentou melhores resultados em 61% dos banco de dados e empatou em 23% dos casos.

## 5.2 DOAO vs OARs

Analisando a tabela 7, observa-se que tem-se um comparativo muito semelhante ao DOAO vs OAOs. Onde o DOAO apresentou melhores resultados ou resultados iguais em aproximadamente 53,84% dos casos e resultado pior em 46,15% dos casos analisados.

Tabela 7 - Erros (%) do DOAO e dos métodos OARs

| Dataset   | ANN-OAR          | DT-OAR           | KNN-OAR               | LDA-OAR                | LR-OAR           | SVM-OAR                | DOAO                    |
|-----------|------------------|------------------|-----------------------|------------------------|------------------|------------------------|-------------------------|
| Zoo       | 57,42<br>± 12,42 | 5,94<br>± 4,02   | <b>3,96</b><br>± 3,20 | 6,93<br>± 6,23         | 5,94<br>± 15,32  | 28,71<br>± 12,04       | 5,94<br>± 8,30          |
| Iris      | 16,00<br>± 9,04  | 5,33<br>± 6,96   | 4,67<br>± 4,42        | <b>2,00</b><br>± 4,26  | 10,00<br>± 4,00  | 8,00<br>± 4,00         | 10,00<br>± 5,20         |
| Wine      | 51,68<br>± 9,82  | 11,80<br>± 6,61  | 5,06<br>± 3,51        | 1,69<br>± 2,21         | 1,69<br>± 2,59   | 4,49<br>± 2,76         | <b>0,56</b><br>± 2,28   |
| Seed      | 34,76<br>± 9,77  | 9,05<br>± 8,46   | 5,71<br>± 7,43        | <b>3,81</b><br>± 5,65  | 9,05<br>± 5,73   | 6,19<br>± 5,30         | 5,24<br>± 6,63          |
| Glass     | 64,48<br>± 3,98  | 31,31<br>± 10,53 | 29,91<br>± 4,98       | 39,25<br>± 8,78        | 42,06<br>± 8,97  | 30,37<br>± 9,41        | <b>29,91</b><br>± 11,13 |
| Ecoli     | 57,44<br>± 7,40  | 19,35<br>± 6,70  | 15,18<br>± 9,49       | <b>13,99</b><br>± 4,28 | 14,29<br>± 6,41  | 35,12<br>± 8,48        | 14,29<br>± 7,85         |
| Movement  | 47,89<br>± 3,89  | 33,33<br>± 5,45  | 15,28<br>± 7,99       | 34,72<br>± 5,47        | 28,89<br>± 3,33  | 16,94<br>± 6,47        | <b>14,72</b><br>± 7,99  |
| Balance   | 43,04<br>± 5,44  | 19,20<br>± 6,02  | 12,00<br>± 3,47       | 13,76<br>± 4,10        | 14,24<br>± 3,47  | <b>3,84</b><br>± 1,57  | <b>3,84</b><br>± 0,73   |
| Landcover | 83,40<br>± 3,89  | 20,15<br>± 4,45  | 22,67<br>± 4,52       | 20,74<br>± 5,15        | 18,37<br>± 4,25  | <b>16,15</b><br>± 4,00 | 16,15<br>± 4,04         |
| Vehicle   | 78,25<br>± 2,28  | 29,55<br>± 6,45  | 28,84<br>± 3,48       | 22,22<br>± 4,06        | 25,77<br>± 4,35  | <b>16,08</b><br>± 2,80 | <b>18,68</b><br>± 2,81  |
| Vowel     | 26,55<br>± 2,82  | 19,39<br>± 4,69  | <b>1,01</b><br>± 1,52 | 39,29<br>± 3,93        | 47,58<br>± 2,64  | <b>1,01</b><br>± 0,92  | <b>1,01</b><br>± 1,52   |
| Yeast     | 53,49<br>± 7,12  | 46,02 ±<br>4,11  | 42,12<br>± 4,65       | 41,24<br>± 3,31        | 42,45<br>± 10,49 | 40,50<br>± 9,12        | <b>39,89</b><br>± 4,94  |
| Segment   | 19,11<br>± 5,19  | 11,91<br>± 4,09  | 12,86<br>± 4,00       | <b>9,05</b><br>± 2,50  | 13,33<br>± 6,04  | 12,86<br>± 5,17        | 11,91<br>± 7,51         |

O gráfico pizza da figura 6 representa melhor a quantidade de vitórias, derrotas e empates do DOAO em relação aos OARs.

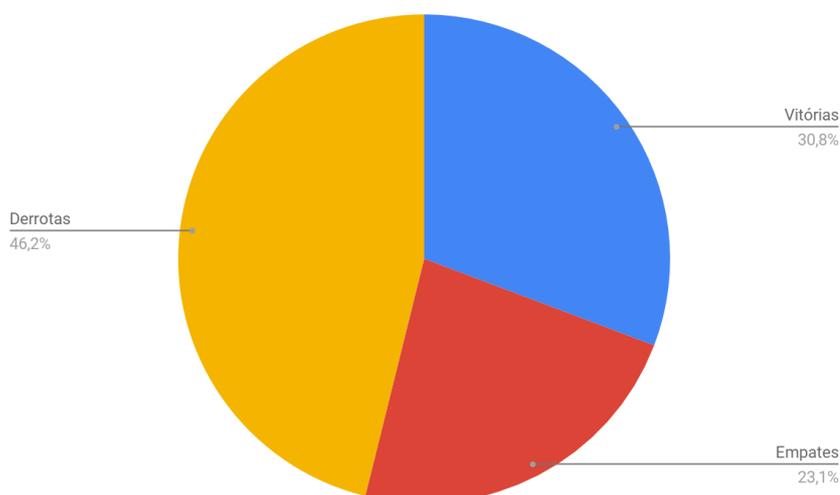


Figura 6 - Gráfico pizza DOAO vs OARs

Assim como comparado aos OAOs, apesar do maior número de derrotas do DOAO, percebe-se que o algoritmo proposto demonstra uma performance aceitável. Pois percebe-se que o DOAO apresenta melhor resultado ao ser comparado dois a dois com cada um dos métodos OARs.

### 5.3 DOAO vs VOTE-OAO

Analisando a tabela 8 constata-se que, apesar de ambos os algoritmos explorarem vários classificadores, verifica-se que o VOTE-OAO obteve piores taxas de erro em aproximadamente 77% dos banco de dados analisados.

Tabela 8 - VOTE OAO vs DOAO

| Dataset   | VOTE-OAO            | DOAO                 |
|-----------|---------------------|----------------------|
| Zoo       | 8,91 ± 14,10        | 5,94 ± 8,30          |
| Iris      | <b>4,67</b> ± 8,00  | 10,00 ± 5,20         |
| Wine      | 1,69 ± 3,33         | <b>0,56</b> ± 2,28   |
| Seed      | <b>4,76</b> ± 3,49  | 5,24 ± 6,63          |
| Glass     | 51,40 ± 16,58       | <b>29,91</b> ± 11,13 |
| Ecoli     | 21,13 ± 6,79        | 14,29 ± 7,85         |
| Movement  | 47,22 ± 9,06        | <b>14,72</b> ± 7,99  |
| Balance   | 20,80 ± 5,16        | <b>3,84</b> ± 0,73   |
| Landcover | 24,74 ± 4,44        | 16,15 ± 4,04         |
| Vehicle   | 23,52 ± 1,81        | <b>18,68</b> ± 2,81  |
| Vowel     | 25,35 ± 2,22        | <b>1,01</b> ± 1,52   |
| Yeast     | 72,17 ± 7,02        | <b>39,89</b> ± 4,94  |
| Segment   | <b>10,95</b> ± 9,77 | 11,91 ± 7,51         |

Logo, apesar do VOTE-OAO ter mostrado resultados aceitáveis para alguns banco de dados, no geral, quando comparado com o DOAO ele teve uma performance muito abaixo do que conseguiu-se com o algoritmo proposto. Esse era um comportamento esperado, visto que o VOTE-OAO não escolhe inicialmente os melhores métodos dentre os candidatos. Logo, na fase de predição o VOTE-OAO leva em consideração votos de classificadores não ideais para o problema, consequentemente piorando o desempenho geral do método.

Para demonstrar que o VOTE-OAO e o DOAO tem precisões diferentes, o teste de hipótese abaixo foi calculado e os resultados apresentados na tabela 9.

$$H_0 : \mu_{DOAO} = \mu_{VOTE-OAO}$$

$$H_1 : \mu_{DOAO} \neq \mu_{VOTE-OAO}$$

Tabela 9 - Testes de hipótese do método DOAO e VOTE-OAO

| Conjunto de Dados | H <sub>0</sub> | t     |
|-------------------|----------------|-------|
| Zoo               | Não Rejeitada  | -1,45 |
| Iris              | Não Rejeitada  | -0,66 |
| Wine              | Não Rejeitada  | -0,26 |
| Seed              | Não Rejeitada  | 0,89  |
| Glass             | Rejeitada      | -2,91 |
| Ecoli             | Rejeitada      | -2,91 |
| Movement          | Rejeitada      | -5,41 |
| Balance           | Rejeitada      | -5,97 |
| Landcover         | Rejeitada      | -3,12 |
| Vehicle           | Não Rejeitada  | -2,77 |
| Vowel             | Rejeitada      | -10,4 |
| Yeast             | Rejeitada      | -5,31 |
| Segment           | Não Rejeitada  | 1,01  |

#### 5.4 DOAO vs Random Forest

A tabela 10 mostra o comparativo dos resultados obtidos pelo DOAO e pelo Random Forest. Verificamos que o DOAO perde em 61% dos casos analisados e ganha em aproximadamente 39% dos bancos. Apesar do *Random Forest* apresentar tais resultados, foi feito um teste de hipótese para verificar se existe uma real diferença de precisão entre os dois métodos.

Tabela 10 - DOAO vs Random Forest

| Dataset   | DOAO         | Random Forest |
|-----------|--------------|---------------|
| Zoo       | 5,94         | <b>3,81</b>   |
| Iris      | 10,00        | <b>4,66</b>   |
| Wine      | <b>0,56</b>  | 1,10          |
| Seed      | <b>5,24</b>  | 6,66          |
| Glass     | 29,91        | <b>19,63</b>  |
| Ecoli     | 14,29        | <b>12,21</b>  |
| Movement  | 14,72        | <b>14,16</b>  |
| Balance   | <b>3,84</b>  | 17,11         |
| Landcover | 16,15        | <b>14,92</b>  |
| Vehicle   | <b>18,68</b> | 24,94         |
| Vowel     | <b>1,01</b>  | 1,71          |
| Yeast     | 39,89        | <b>36,65</b>  |
| Segment   | 11,91        | <b>2,34</b>   |

Foi feito o Teste T-Pareado bilateral em que, inicialmente estabeleceu-se que a hipótese nula é a que os dois métodos têm a mesma precisão e a hipótese alternativa é a de que um dos dois métodos tem melhor precisão em relação ao outro. Ou seja:

$$H_0 : \mu_{DOAO} = \mu_{RF}$$

$$H_1 : \mu_{DOAO} \neq \mu_{RF}$$

A Tabela 11 apresenta os resultados dos testes de hipótese para cada um dos banco de dados.

Tabela 11 - Testes de hipótese do método DOAO e *Random Forest*

| Conjunto de Dados | H <sub>0</sub> | t     |
|-------------------|----------------|-------|
| Zoo               | Não Rejeitada  | 0,80  |
| Iris              | Rejeitada      | 2,44  |
| Wine              | Não rejeitada  | -1,15 |
| Seed              | Não rejeitada  | 4,00  |
| Glass             | Rejeitada      | 3,69  |
| Ecoli             | Não rejeitada  | -1,04 |
| Movement          | Não rejeitada  | -0,73 |
| Balance           | Rejeitada      | -8,64 |
| Landcover         | Não rejeitada  | -2,12 |
| Vehicle           | Rejeitada      | -3,49 |
| Vowel             | Não rejeitada  | -2,04 |
| Yeast             | Não rejeitada  | -1,07 |
| Segment           | Rejeitada      | 2,85  |

Para o teste de hipótese realizado  $-2.228 < t < +2.228$  para que a hipótese nula seja aceita. Observando os dados da tabela 8 temos que para 5 conjunto de dados os dados analisados foram suficientes para rejeitar a hipótese de que os dois métodos têm igual precisão.

No caso dos banco de dados *Iris*, *Glass* e *Segment* o método *Random Forest* apresentou melhores resultados e nos bancos de dados *Vehicle* e *Balance* o método DOAO [4] apresentou melhores resultados. A partir desta observação podemos perceber que para alguns bancos de dados um dos métodos se mostra significativamente superior ao outro.

## 5.5 Análise de Tempo de Execução

A tabela 12.1 e 12.2 mostram os tempos de execução de cada algoritmo em que os melhores tempos estão em negrito. Estas medidas envolvem o tempo de execução para treinamento e validação, lembrando que para os algoritmos que foram testados com mais de uma configuração de parâmetros para o DOAO, também foram testadas e validadas para os algoritmos OAO e OAR. Ou seja, para o método KNN-OAR foi calculado o erro de validação de cada configuração, para então a melhor ser escolhida. Isso explica os altos tempos de execução do ANN-OAO e ANN-OAR, pois os mesmos foram executados com 3,4,5 até 20 *hidden nodes*. Desde modo podemos ter uma comparação mais justa, já que o DOAO sempre terá que calcular o erro de validação para todos os classificadores candidatos.

Podemos observar também, uma predominância nos melhores tempos pelos algoritmos que não possuem parâmetros a ser variados. Ou seja, o LDA-OAR, LR-OAR, LDA-OAO, LR-OAO. O fato de que eles não precisam ser testados em outras configurações faz com que surja essa grande discrepância entre o tempo deles e o tempo dos outros métodos de decomposição avaliados.

Tabela 12.1 - Tempo de execução em segundos

|          | Zoo          | Iris         | Wine         | Seed         | Glass        | Ecoli        | Tempo Médio |
|----------|--------------|--------------|--------------|--------------|--------------|--------------|-------------|
| ANN-OAR  | 38,824       | 21,739       | 25,923       | 28,237       | 62,371       | 122,958      | 50,01       |
| DT-OAR   | 0,406        | 0,243        | 0,361        | 0,369        | 0,81         | 1,331        | 0,59        |
| KNN-OAR  | 2,049        | 1,272        | 0,936        | 1,074        | 2,239        | 4,55         | 2,02        |
| LDA-OAR  | <b>0,220</b> | 0,081        | 0,098        | 0,096        | <b>0,211</b> | <b>0,344</b> | <b>0,18</b> |
| LR-OAR   | 0,227        | 0,072        | 0,095        | 0,098        | 0,253        | 0,523        | 0,21        |
| SVM-OAR  | 12,393       | 6,911        | 11,294       | 11,455       | 24,959       | 49,926       | 19,49       |
| ANN-OAO  | 43,397       | 14,488       | 18,154       | 19,907       | 59,678       | 136,529      | 48,69       |
| DT-OAO   | 0,753        | 0,192        | 0,258        | 0,273        | 1,01         | 1,987        | 0,75        |
| KNN-OAO  | 1,832        | 0,925        | 0,695        | 0,804        | 2,752        | 4,93         | 1,99        |
| LDA-OAO  | 0,509        | 0,079        | <b>0,092</b> | <b>0,090</b> | 0,43         | 0,977        | 0,36        |
| LR-OAO   | 0,586        | <b>0,066</b> | 0,092        | 0,089        | 0,529        | 1,523        | 0,48        |
| SVM-OAO  | 15,914       | 4,725        | 6,979        | 7,072        | 21,251       | 43,836       | 16,63       |
| VOTE-OAO | 64,338       | 19,997       | 25,679       | 27,545       | 78,902       | 165,795      | 63,71       |
| RF       | 1,399        | 1,309        | 1,361        | 1,495        | 1,552        | 1,842        | 1,49        |
| DOAO     | 58,406       | 19,426       | 24,594       | 26,626       | 76,296       | 161,459      | 61,13       |

O método *Random Forest* apresentou bons resultados, pois apesar de só ter o melhor tempo para o banco de dados *Movement*, apresentou uma grande estabilidade. Mantendo uma média de tempo baixa mesmo nos conjuntos de dados com muitos atributos e muitas instâncias, como o *LandCover*, *Movement* e *Segment*.

O método DOAO [4], junto com o VOTE-OAO, ANN-OAR e ANN-OAO, foi um dos que apresentou piores tempos de execução. Como foi mencionado no primeiro parágrafo desta seção, o fato do método fazer a validação de todos os classificadores candidatos com todas as configurações possíveis aumenta muito o seu custo computacional.

Tabela 12.2 - Tempo de execução em segundos

| Dataset  | Movement    | Balance     | L.Cover       | Vehicle       | Vowel  | Yeast       | Segment       | Tempo Médio |
|----------|-------------|-------------|---------------|---------------|--------|-------------|---------------|-------------|
| ANN-OAR  | 500,22      | 104,03      | 97,97         | 198,51        | 603,40 | 821,76      | 71,83         | 342,53      |
| DT-OAR   | 14,67       | 1,11        | 964,00        | 3,15          | 9,47   | 8,91        | 1,07          | 143,20      |
| KNN-OAR  | 12,54       | 5,05        | 3,83          | 7,20          | 22,07  | 30,92       | 2,51          | 12,02       |
| LDA-OAR  | 3,72        | 235,00      | 243,00        | <b>683,00</b> | 1,32   | <b>1,65</b> | <b>271,00</b> | 205,53      |
| LR-OAR   | 11,91       | 0,29        | 277,00        | 1,02          | 3,85   | 4,18        | 405,00        | 100,46      |
| SVM-OAR  | 369,58      | 89,46       | 89,77         | 268,24        | 588,10 | 1.089,47    | 32,40         | 361,00      |
| ANN-OAO  | 625,63      | 67,72       | 67,26         | 156,09        | 714,88 | 899,90      | 77,60         | 372,73      |
| DT-OAO   | 27,15       | 729,00      | 717,00        | 2,65          | 14,47  | 13,46       | 1,41          | 215,02      |
| KNN-OAO  | 22,47       | 3,13        | 2,82          | 6,35          | 37,42  | 50,10       | 3,40          | 17,95       |
| LDA-OAO  | 16,67       | <b>0,21</b> | <b>209,00</b> | 711,00        | 4,31   | 4,70        | 653,00        | 228,41      |
| LR-OAO   | 76,25       | 248,00      | 242,00        | 1,27          | 17,01  | 16,06       | 1,04          | 85,95       |
| SVM-OAO  | 185,50      | 45,12       | 45,15         | 124,33        | 308,60 | 661,72      | 25,94         | 199,48      |
| VOTE-OAO | 680,87      | 115,03      | 114,13        | 279,82        | 845,74 | 1,27        | 100,00        | 305,27      |
| RF       | <b>2,06</b> | 2,62        | 3,06          | 3,38          | 4,13   | 5,46        | 1,45          | <b>3,16</b> |
| DOAO     | 636,00      | 110,79      | 110,53        | 273,28        | 831,67 | 1,369,951   | 96,35         | 343,10      |

## 5.6 Comparação das Implementações

A partir da tabela 13 é possível observar a quantidade de vezes que cada algoritmo de classificação foi selecionado pela validação do DOAO como o mais eficiente, no artigo original do DOAO e na replicação. O número de seleção indica que DOAO selecionou o algoritmo para determinado número de pares de classes. Como se pode observar, apesar dos números entre o artigo original e a replicação serem diferentes para todos os bancos de dados, o SVM e o ANN foram os mais selecionados para quase todos os banco de dados. Sendo no artigo original em

100% dos banco de dados o algoritmo mais selecionado foi o SVM ou o ANN, contra aproximadamente 76% na replicação.

Tabela 13 - Vezes que cada algoritmo foi selecionado

| Dataset   | No. Classes | No. Pares de Classes | Experimento original |     |      |     |     |             | Replicação  |             |             |            |     |             |
|-----------|-------------|----------------------|----------------------|-----|------|-----|-----|-------------|-------------|-------------|-------------|------------|-----|-------------|
|           |             |                      | ANN                  | DT  | k-NN | LDA | LR  | SVM         | ANN         | DT          | k-NN        | LDA        | LR  | SVM         |
| Zoo       | 7           | 21                   | <b>7,9</b>           | 2,7 | 3,9  | 1,0 | 2,4 | 3,1         | <b>16,0</b> | 3,0         | 2,0         | 0,0        | 0,0 | 0,0         |
| Iris      | 3           | 3                    | <b>1,2</b>           | 0,4 | 0,2  | 0,2 | 0,4 | 0,6         | <b>2,0</b>  | 0,0         | 0,0         | 0,0        | 0,0 | 1,0         |
| Wine      | 3           | 3                    | <b>1,2</b>           | 0,0 | 0,2  | 0,4 | 0,3 | 0,9         | <b>1,0</b>  | 0,0         | 0,0         | <b>1,0</b> | 0,0 | <b>1,0</b>  |
| Seed      | 3           | 3                    | <b>1,1</b>           | 0,3 | 0,3  | 0,2 | 0,9 | 0,2         | 1,0         | 0,0         | 0,0         | 0,0        | 0,0 | <b>2,0</b>  |
| Glass     | 6           | 15                   | <b>9,0</b>           | 1,5 | 1,1  | 0,6 | 0,3 | 2,5         | 0,0         | 4,0         | 1,0         | 1,0        | 0,0 | <b>9,0</b>  |
| Ecoli     | 8           | 28                   | <b>18,2</b>          | 1,2 | 1,9  | 2,2 | 1,9 | 2,6         | <b>13,0</b> | 2,0         | 7,0         | 1,0        | 0,0 | 5,0         |
| Movement  | 15          | 105                  | <b>50,7</b>          | 5,0 | 17,3 | 2,9 | 0,7 | 28,4        | 23,0        | 7,0         | <b>38,0</b> | 1,0        | 5,0 | 31,0        |
| Balance   | 3           | 3                    | 0,0                  | 0,0 | 0,0  | 0,0 | 0,0 | <b>3,0</b>  | 0,0         | 0,0         | 0,0         | 0,0        | 0,0 | <b>3,0</b>  |
| Landcover | 9           | 36                   | <b>25,3</b>          | 2,6 | 2,5  | 0,7 | 0,0 | 4,9         | 7,0         | <b>11,0</b> | 8,0         | 1,0        | 0,0 | 9,0         |
| Vehicle   | 4           | 6                    | 1,8                  | 0,0 | 0,0  | 0,0 | 0,0 | <b>4,2</b>  | 0,0         | 0,0         | 0,0         | 0,0        | 0,0 | <b>6,0</b>  |
| Vowel     | 11          | 55                   | 13,7                 | 0,5 | 14,9 | 2,0 | 3,9 | <b>20,0</b> | 4,0         | 4,0         | <b>42,0</b> | 0,0        | 0,0 | 5,0         |
| Yeast     | 10          | 45                   | <b>32,9</b>          | 1,9 | 3,0  | 0,5 | 1,3 | 5,4         | 5,0         | 2,0         | 7,0         | 0,0        | 0,0 | <b>31,0</b> |
| Segment   | 7           | 21                   | <b>7,4</b>           | 2,7 | 2,8  | 1,2 | 1,6 | 5,3         | <b>10,0</b> | 3,0         | 0,0         | 3,0        | 1,0 | 4,0         |

A fim de demonstrar a fidelidade da implementação do presente trabalho com a implementação onde o artigo foi proposto [4]. Também foi avaliada a diferença absoluta entre os erros obtidos para cada método analisado (erro percentual), ou seja, a diferença entre os erros dos resultados do experimento original do artigo e a replicação do experimento. Essas diferenças podem ser observadas no gráfico da figura 7, para os classificadores (ANN-OAO, DT-OAO, kNN-OAO, LDA-OAO, LR-OAO, SVM-OAO, VOTE-OAO e DOAO).

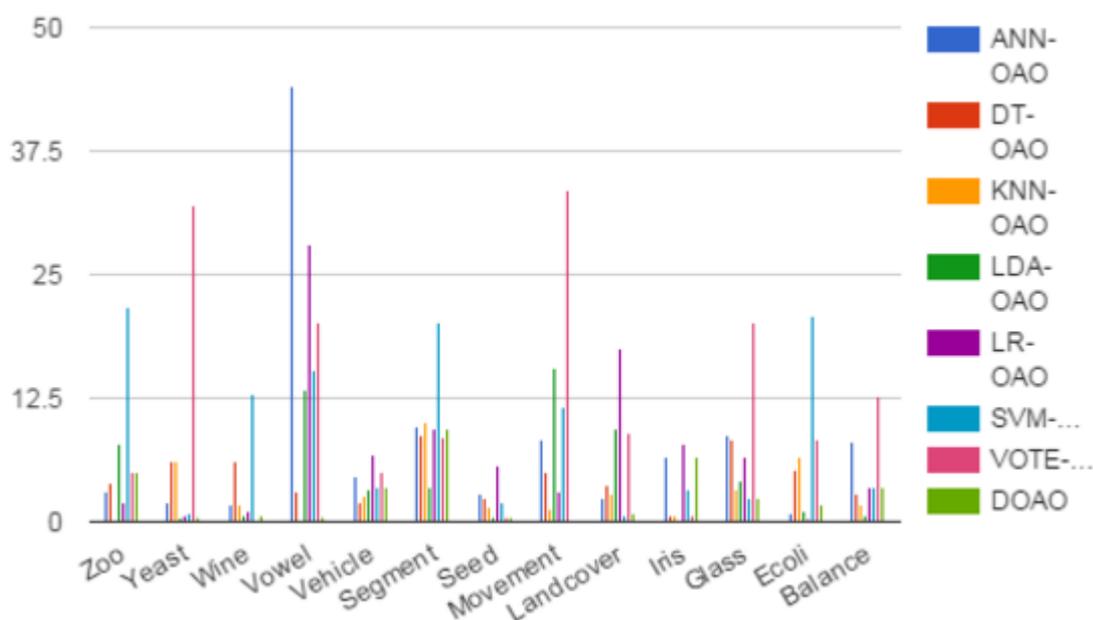


Figura 7 - Diferença absoluta do erro (pontos %)

Pode-se notar que nenhum erro da replicação teve uma diferença de mais de 50 pontos percentuais comparado com o experimento original. E para a grande maioria dos datasets, o erro absoluto não foi superior a 25 pontos percentuais. De modo similar, a diferença absoluta foi avaliada apenas para o classificador proposto DOAO, e na condução dos experimentos da replicação, representado na tabela 14.

Tabela 14 - Erros de validação da implementação no artigo original e na replicação

| Dataset   | DOAO (Artigo original) | DOAO (Replicação) |
|-----------|------------------------|-------------------|
| Zoo       | 0,99                   | 5,94              |
| Iris      | 3,33                   | 10,00             |
| Wine      | 1,12                   | 0,56              |
| Seed      | 4,76                   | 5,24              |
| Glass     | 32,24                  | 29,91             |
| Ecoli     | 12,50                  | 14,29             |
| Movement  | 14,72                  | 14,72             |
| Balance   | 0,32                   | 3,84              |
| Landcover | 15,25                  | 16,15             |
| Vehicle   | 15,24                  | 18,68             |
| Vowel     | 0,50                   | 1,01              |
| Yeast     | 40,36                  | 39,89             |
| Segment   | 2,38                   | 11,91             |

A diferença não foi maior do que 10 pontos percentuais. O maior erro foi para o dataset “segment”, com 9.5 pontos percentuais de diferença. Como pode ser visto na Figura 8.

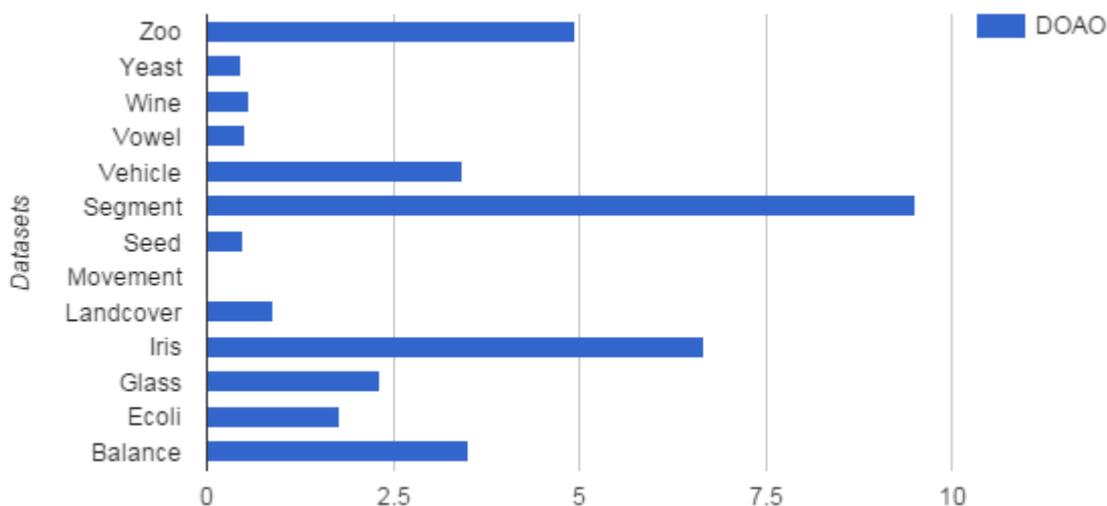


Figura 8 - Diferença absoluta do erro (pontos %)

Sendo assim, essas diferenças nos resultados das implementações pode-se dever a falta de mais informações a respeito dos parâmetros utilizados nos métodos candidatos, ou mesmo pelo fator aleatório na escolha dos *folds* para o cálculo do erro de validação para cada candidato. Porém apesar disso, foi possível replicar o algoritmo DOAO e obter taxas semelhantes as esperadas.

## 6. CONCLUSÃO

Neste trabalho foi replicado um método de experimento para um classificador originalmente proposto por [4], onde foi implementado um o método chamado *diversified one-against-one* (DOAO) para solucionar problemas de classificação multi-classes com a construção de um classificador multi-classes utilizando a abordagem *one-against-one* com diferentes algoritmos de classificação.

A proposta do método visa obter melhores resultados de classificação pela seleção de melhores algoritmos de classificação para cada par de classes. Os resultados experimentais mostraram que de modo geral obteve-se resultados equivalentes ao artigo original, como por exemplo menores taxas de erros para os datasets.

Algumas suposições tiveram que ser assumidas para conduzir os experimentos diante algumas dificuldades encontradas, pois algumas informações não estavam explícitas, como por exemplo alguns parâmetros dos classificadores (como taxa de aprendizagem), ou a forma de divisão de cada fold no 10-Fold validation, por exemplo.

Os experimentos foram coerentes com os resultados originais, apresentado erros em pontos percentuais para o método proposto (DOAO) inferiores a 10%. Também foi interessante notar a diferença do erro (DOAO), mesmo para os classificadores sendo implementados de formas não exatamente iguais (limitação da replicação), pois há uso de tecnologias diferentes (Python x Matlab) e o código fonte original não foi disponibilizado.

A respeito do método implementado e analisado, pode-se concluir que o mesmo apresentou resultados condizentes com outros métodos já conhecidos. Mesmo quando comparado com o *Random Forest*, que tem apresentado bons resultados em diversos estudos nos últimos anos, o algoritmo proposto apresentou resultados muito próximos dos obtidos com o Random Forest, e até mesmo melhor em alguns *datasets*, apesar de não ter se sobreposto na maioria dos *datasets* analisados.

Porém, foi observada uma desvantagem significativa no método proposto. Por ser necessário o treinamento do mesmo conjunto de dados diversas vezes variando os pares de classes, métodos de classificação binária e parâmetros de cada método, o mesmo mostrou necessitar de um custo computacional muito maior que os outros métodos analisados. Podendo ficar para um estudo futuro a análise comparativa mais profunda com foco na performance do método DOAO em relação aos outros métodos.

## 6.1 Trabalhos Futuros

Os seguintes trabalhos futuros podem ser sugestões de como continuar o presente trabalho e melhorá-lo:

- Uma análise mais profunda do DOAO com métodos de *Bagging*, mais e maiores banco de dados
- Aplicação do método avaliado em algum problema real a fim de comprovar a viabilidade de utilizá-lo e identificar possíveis novos problemas
- Otimização da implementação do DOAO a fim de diminuir o custo computacional do mesmo

## 7. REFERÊNCIAS

- [1] Awad, W.A.; Elseuofi, S.M.; Machine learning methods for spam e-mail classification, *International Journal of Computer Science & Information Technology*, 2011, Vol 3, No 1, P. 173-183.
- [2] Quinlan, J. R.; *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.
- [3] Breiman, L.; Friedman, J.; Stone, C.; R.A Olshen. *Classification and Regression Trees*. Taylor & Francis, EUA, 1984. 364 p.
- [4] Kang, S.; Cho, S.; Kang, P. Constructing a multi-class classifier one-against-one approach with different binary classifiers, *Neurocomputing*, Volume 149, Parte B, 3 de Fevereiro de 2015, Pág. 677-682
- [5] Breiman, L.; *Arcing The Edge*, *Technical Report 486*, *University of California*, Agosto de 1997.
- [6] Bhatia, N.; Vandana. Survey of Nearest Neighbor Techniques, *International Journal of Computer Science and Information Security*, Vol 8, No. 2, 2010, P.302-305.
- [7] Li, C.; Wang, B. Fisher Linear Discriminant Analysis, 31 de Agosto, 2014.
- [8] Bishop C. M.; *Neural Networks for Pattern Recognition*. *Oxford University Press*, 1995.
- [9] Bay, S.D.; *Combining nearest neighbor classifiers through multiple feature subsets*. In *Proceedings of the 17th International Conference on Machine Learning*, P. 37–45, Madison, WI, 1998.
- [10] Rish, I.; An empirical study of the naive bayes classifier. In *IJCAI Workshop on Empirical Methods in Artificial Intelligence*, 2001.
- [11] Cortes, C.; Vapnik, V.; Support-vector networks. *Machine Learning*, P. 273–297, 1995.

- [12] Belekar, V., Karthik, L., Prabha, G.; Classification of breast cancer resistant protein and non-Inhibitors using Machine learning approaches. *Combinatorial Chemistry & High Throughput screening*, 2015, Vol. 18, P. 476-485.
- [13] Galar M.; Fernández A.; Barrenechea E.; Bustince H.; Herrera F.; An overview of ensemble methods for binary classifiers in multi-class problems: experimental study on one-vs-one and one-vs-all schemes, *Pattern Recognit*, 2011, P. 1761–1776.
- [14] Aly, M.; *Survey on Multiclass Classification Methods*, The University of Utah, School of Computing, 2005.
- [15] Quinlan, R.; *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, 1993, San Mateo, California.
- [16] Rumelhart, D. E., Hinton G. E., Williams R.J.; *Nature*, Vol 323, 1986, P. 533-536.
- [17] Hastie, T., Tibshirani R.; Classification by Pairwise coupling, *The Annals of Statistics*, 1998, Vol. 26, Número 2, P. 451-471.
- [18] L. Kuncheva; *Combining pattern classifiers: methods and algorithms*. Wiley-Interscience, 2004.
- [19] M, P. Ponti; *Combining Classifiers: from the creation of ensembles to the decision fusion*, *Graphics, Patterns and Images Tutorials*, 2011.
- [20] Breiman, L.; *Random Forests*, *Journal Machine Learning*, Vol. 45, 2001, P. 5-32.
- [21] Rodriguez-Galliano, V.F.; Ghimire, B.; Rogan, J.; Chica-Olmo, M.; Rigol-Sanchez, J.P.; *An assessment of the effectiveness of a random forest classifier for land-cover classification*, *Journal of Photogrammetry and Remote Sensing*, Volume 67, 2012, P. 93-104.
- [22] J.H. Friedman; Another approach to polychotomous classification. Technical report, Department of Statistics, Stanford University, 1996.
- [23] L. Rokach; *Pattern Classification using Ensemble Methods*, World Scientific, Vol 75, 2010.
- [24] <http://scikit-learn.org/>
- [25] <http://www.numpy.org/>

- [26] De'ath, G., and K. E. Fabricius.; Classification and regression trees: a powerful yet simple technique for ecological data analysis, Ecological Society of America, 2000, Vol. 81, P. 3178-3192.
- [27] Cabrera, A. F.; Logistic Regression Analysis in Higher Education: An Applied Perspective, Agathon Press, 1994, Vol 10, New York, P. 225-256.
- [28] Breiman, L.; Bagging predictors. Journal Machine Learning, 1996, Vol. 24(2), P. 123-140.
- [29] Freund, Y.; Schapire, R. E.; *A Decision-Theoretic Generalization of on-Line Learning and an Application to Boosting*, Journal of Computer and System Sciences, 1997, Vol. 55, P. 119-139.
- [30] Shapire, Robert E.; *The Strength of Weak Learnability*, Machine Learning, Vol. 5, 1990, P. 197-227.