

**UNIVERSIDADE FEDERAL DE JUIZ DE FORA**  
**INSTITUTO DE CIÊNCIAS EXATAS**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**

**Felipe Souza Amaral**

**Um classificador baseado na profundidade de Tukey para geração de uma  
aproximação do ponto de Bayes**

Juiz de Fora

2022

**Felipe Souza Amaral**

**Um classificador baseado na profundidade de Tukey para geração de uma aproximação do ponto de Bayes**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Raul Fonseca Neto

Coorientador: Prof. Dr. Saulo Moraes Villela

Juiz de Fora

2022

Ficha catalográfica elaborada através do Modelo Latex do CDC da UFJF  
com os dados fornecidos pelo(a) autor(a)

Amaral, Felipe Souza.

Um classificador baseado na profundidade de Tukey para geração de uma aproximação do ponto de Bayes / Felipe Souza Amaral. – 2022.

64 f. : il.

Orientador: Raul Fonseca Neto

Coorientador: Saulo Moraes Villela

Dissertação (Mestrado) – Universidade Federal de Juiz de Fora, Instituto de ciências exatas. Programa de Pós-Graduação em Ciência da Computação, 2022.

1. Comitês. 2. Perceptron. 3. Profundidade de Tukey. 4. Mediana de Tukey. I. Fonseca Neto, Raul, orient. II. Villela, Saulo Moraes, coorient. III. Título.

**Felipe Souza Amaral**

**Um classificador baseado na profundidade de Tukey para geração de uma aproximação do ponto de Bayes**

Dissertação  
apresentada ao  
Programa de Pós-  
graduação em Ciência  
da Computação  
da Universidade  
Federal de Juiz de Fora  
como requisito parcial  
à obtenção do título de  
Mestre em Ciência da  
Computação. Área de  
concentração: Ciência  
da Computação.

Aprovada em 11 de março de 2022.

BANCA EXAMINADORA

**Prof. Dr. Raul Fonseca Neto - Orientador**

Universidade Federal de Juiz de Fora

**Prof. Dr. Saulo Moraes Villela - Coorientador**

Universidade Federal de Juiz de Fora

**Prof. Dr. Carlos Cristiano Hasenclever Borges**

Universidade Federal de Juiz de Fora

**Prof. Dr. Marcelo Azevedo Costa**

Universidade Federal de Minas Gerais



Documento assinado eletronicamente por **Raul Fonseca Neto, Professor(a)**, em 14/06/2022, às 09:38, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).

---



Documento assinado eletronicamente por **Saulo Moraes Villela, Professor(a)**, em 17/06/2022, às 15:57, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).

---



Documento assinado eletronicamente por **Carlos Cristiano Hasenclever Borges, Professor(a)**, em 20/06/2022, às 15:19, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).

---



Documento assinado eletronicamente por **MARCELO AZEVEDO COSTA, Usuário Externo**, em 20/06/2022, às 21:26, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).

---



A autenticidade deste documento pode ser conferida no Portal do SEI-Ufjf ([www2.ufjf.br/SEI](http://www2.ufjf.br/SEI)) através do ícone Conferência de Documentos, informando o código verificador **0699968** e o código CRC **47CC7C0B**.

---

## AGRADECIMENTOS

Agradeço primeiramente a minha mãe Jacira Maria das Graças e Souza por todo amor, comprometimento e apoio incondicional para comigo. Agradeço a minha madrinha Joana Darque Ferreira Amaral pelo incentivo, perseverança e carinho. Agradeço minha irmã Talita Souza Amaral e aos meus amados sobrinhos, Yuri Fernando Amaral Sanchez e Davi Eduardo Amaral Sanchez pela paciência. E por fim, mas não menos importante, agradeço a todos os meus familiares.

Agradeço aos meus orientadores Raul Fonseca Neto e Saulo Moraes Villela pelo carinho, pela oportunidade de trabalhar em conjunto, pelos aprendizados não somente acadêmicos mas da vida, e principalmente pela enorme paciência. Agradeço a Universidade Federal de Juiz de Fora pelo fomento investido em mim.

Agradeço aos amigos adquiridos Alexandre e Marcos. E agradeço ao meu querido professor de graduação Fillipe Jabour pelo apoio, atenção e presteza quando requisitado.

“Eu não quero acreditar, eu quero saber.”

Carl Sagan

## RESUMO

Comitês (*ensembles*) para a criação de classificadores, no campo de aprendizado de máquina, têm sido largamente usados por serem uma via mais eficaz e mais acurada, ao se comparar a um único classificador. Esse aumento na acurácia dos comitês está diretamente relacionado à capacidade de combinar os conhecimentos aprendidos de componentes do classificador para gerar uma solução que supera as soluções de cada indivíduo. Um comitê pode ser interpretado como um conjunto de membros, por essa ótica, pode-se relacionar com o campo da estatística e portanto aplicar conceitos de profundidade de dados nesses membros. Uma concepção bastante conhecida é a profundidade de Tukey, também conhecida como profundidade de meio espaço, que é definida como a menor área de probabilidade de um meio espaço fechado que contenha o ponto analisado, resultando assim as profundidades dos dados em um conjunto multivariado. O ponto mais profundo em relação à profundidade de Tukey é comumente visto como a mediana de Tukey. Este trabalho tem como objetivo desenvolver ferramentas e análises para aplicar em um comitê de classificadores, sendo o espaço de versões criado através de Perceptrons balanceados, buscando encontrar o representante que mais ganha nas votações ocorridas, ou o membro considerado mais profundo, cunhando assim o Perceptron de Tukey, além de apresentar algoritmos, baseados nos conceitos de buscas em profundidade e largura, para identificação do membro mais ganhador das votações realizadas. Um estudo experimental foi conduzido para avaliação dos métodos propostos e os resultados mostram que os métodos apresentados são capazes de superar, na maior parte dos casos, outros algoritmos, como a Máquina de Vetores Suporte, Máquina de Ponto de Bayes, Máquina de Redução do Espaço de Versões e Classificador Evolucionário de Centro Analítico.

Palavras-chave: Comitês. Perceptron. Profundidade de Tukey. Mediana de Tukey.

## ABSTRACT

Ensembles for creating classifiers, in the area of machine learning, have been widely used since they are a more efficient and more accurate way when compared to a single classifier. This increase in the accuracy of ensembles is directly related to the capacity to combine the acquired knowledge from the classifier components to generate a solution that surpasses the solutions of each individual. An ensemble can be understood as a group of members and, from that point of view, it can be related to the field of statistics and, therefore, apply concepts of depth data to these members. A well-known concept is Tukey's depth, also known as the half-space depth, which is defined as the smallest area of the probability of a closed half-space that contains an analyzed point, thus resulting in the depths of data in a multivariate group. The deepest point regarding Tukey's depth is commonly seen as Tukey's median. This dissertation has as its main objective to develop tools and analyses to be applied to an ensemble of classifiers, being the version space created from balanced Perceptrons, aiming to find the representative member that wins the most in the voting or the deepest member, then coining Tukey's Perceptron, besides presenting the algorithms based on the concepts of the depth-first search and breadth-first search for identifying the winner with most votes. An experimental study has been performed to evaluate the proposed methods, and the results show that the introduced methods are able to overcome, in most cases, other algorithms such as Support Vector Machine, Bayes Point Machine, Version Space Reduction Machine, and Evolutionary Analytic Center Classifier.

Key-words: Ensembles. Perceptron. Tukey's Depth. Tukey's Median.

## LISTA DE FIGURAS

Figura 1 – <i>Convex Hull Peeling</i> . . . . .	17
Figura 2 – <i>Simplicial</i> . . . . .	18
Figura 3 – <i>Profundidade de Oja</i> . . . . .	19
Figura 4 – Topologia do modelo Perceptron. . . . .	24
Figura 5 – Três razões fundamentais para a construção de um <i>ensemble</i> . . . . .	28
Figura 6 – Espaço de versões . . . . .	32
Figura 7 – Ilustração do ponto de Bayes estimado no espaço tridimensional . . . . .	34
Figura 8 – Ilustração da estratégia empregada pelo VRSM . . . . .	35
Figura 9 – (a) Coordenadas esféricas de um sistema espacial em 3-dimensões. (b) Representação de um vetor unitário tridimensional em uma esfera unitária	36
Figura 10 – Hiperplano separador . . . . .	38
Figura 11 – Divisão do espaço de versões. . . . .	39
Figura 12 – Perceptron de Tukey por Busca em Profundidade. . . . .	40
Figura 13 – Perceptron de Tukey por Busca em Profundidade com Subcomitê por Vizi- nhança. . . . .	42
Figura 14 – Perceptron de Tukey por Busca em Largura . . . . .	44
Figura 15 – Perceptron de Tukey por Busca em Largura com Subcomitê por Votação	46
Figura 16 – Perceptron de Tukey por Busca em Largura com Subcomitê por Vizi- nhança . . . . .	47

## LISTA DE TABELAS

Tabela 1 – Valores da profundidade de Tukey . . . . .	20
Tabela 2 – Informações sobre as bases de dados consideradas. . . . .	48
Tabela 3 – Valores de dissimilaridade aplicados nas bases para o EBPd. . . . .	50
Tabela 4 – Algoritmo de Busca em Profundidade . . . . .	51
Tabela 5 – Algoritmo de Busca em Profundidade com Subcomitê por Vizinhança .	52
Tabela 6 – Algoritmo de Busca em Largura . . . . .	53
Tabela 7 – Algoritmo de Busca em Largura com Subcomitê por Votação . . . . .	54
Tabela 8 – Algoritmo de Busca em Largura com Subcomitê por Vizinhança . . . .	55
Tabela 9 – Comparação dos erros médios entre os classificadores. . . . .	57

## LISTA DE ABREVIATURAS E SIGLAS

AdaBoost	<i>Adaptive Boosting</i>
BFS	<i>Breadth-First Search</i>
BPM	<i>Bayes Point Machine</i>
DFS	<i>Depth-First Search</i>
EACC	<i>Evolutionary Analytic Center Classifier</i>
EBP	<i>Ensemble of Balanced Perceptrons</i>
IA	Inteligência Artificial
PB	Perceptron Balanceado
PK	Perceptron Kernel
RNA	Redes Neurais Artificiais
SVM	<i>Support Vector Machine</i>
TP	<i>Tukey's Perceptron</i>
TP-BFS	<i>Tukey's Perceptron by Breadth-First Search</i>
TP-BFS-NB	<i>Tukey's Perceptron by BFS with Subensemble by Neighborhood</i>
TP-BFS-SV	<i>Tukey's Perceptron by BFS with Subensemble by Voting</i>
TP-DFS	<i>Tukey's Perceptron by Depth-First Search</i>
TP-DFS-NB	<i>Tukey's Perceptron by DFS with Subensemble by Neighborhood</i>
VSRM	<i>Version Space Reduction Machine</i>

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO . . . . .</b>	<b>13</b>
1.1	MOTIVAÇÃO E ARGUMENTAÇÃO . . . . .	14
1.2	OBJETIVOS . . . . .	15
1.3	CONTRIBUIÇÕES . . . . .	15
1.4	ORGANIZAÇÃO . . . . .	16
<b>2</b>	<b>PROFUNDIDADE DE DADOS . . . . .</b>	<b>17</b>
2.1	PROFUNDIDADE <i>CONVEX HULL PEELING</i> . . . . .	17
2.2	PROFUNDIDADE SIMPLICIAL . . . . .	18
2.3	PROFUNDIDADE DE OJA . . . . .	18
2.4	PROFUNDIDADE DE MAHALANOBIS . . . . .	18
2.5	PROFUNDIDADE E MEDIANA DE TUKEY . . . . .	19
<b>3</b>	<b>APRENDIZADO DE MÁQUINA . . . . .</b>	<b>23</b>
3.1	CLASSIFICAÇÃO BINÁRIA . . . . .	23
3.2	PERCEPTRON . . . . .	24
<b>3.2.1</b>	<b>Formulação primal . . . . .</b>	<b>25</b>
3.3	MÉTODOS <i>ENSEMBLES</i> . . . . .	25
<b>3.3.1</b>	<b><i>Ensemble</i> de Perceptrons Balanceados . . . . .</b>	<b>29</b>
3.4	ESPAÇO DE VERSÕES . . . . .	32
<b>4</b>	<b>CLASSIFICADORES <i>BASELINES</i> . . . . .</b>	<b>33</b>
4.1	MÁQUINA DE VETORES SUPORTE . . . . .	33
4.2	MÁQUINA DO PONTO DE BAYES . . . . .	33
4.3	MÁQUINA DE REDUÇÃO DO ESPAÇO DE VERSÕES . . . . .	34
4.4	CLASSIFICADOR EVOLUCIONÁRIO DE CENTRO ANALÍTICO . . . . .	36
<b>5</b>	<b>PERCEPTRON DE TUKEY . . . . .</b>	<b>37</b>
5.1	BUSCA EM PROFUNDIDADE . . . . .	39
<b>5.1.1</b>	<b>Subcomitê por vizinhança . . . . .</b>	<b>41</b>
5.2	BUSCA EM LARGURA . . . . .	43
<b>5.2.1</b>	<b>Subcomitê por votação . . . . .</b>	<b>44</b>
<b>5.2.2</b>	<b>Subcomitê por vizinhança . . . . .</b>	<b>45</b>
<b>6</b>	<b>ANÁLISE EXPERIMENTAL E RESULTADOS . . . . .</b>	<b>48</b>
6.1	CONJUNTOS DE DADOS . . . . .	48
6.2	DEFINIÇÃO DE PARÂMETROS . . . . .	48
<b>6.2.1</b>	<b><i>K-fold cross-validation</i> . . . . .</b>	<b>48</b>
<b>6.2.2</b>	<b>Tamanho do <i>ensemble</i> . . . . .</b>	<b>49</b>
<b>6.2.3</b>	<b>Medida de dissimilaridade . . . . .</b>	<b>49</b>
6.3	RESULTADOS NUMÉRICOS . . . . .	50
<b>6.3.1</b>	<b>Algoritmos baseados na busca em profundidade . . . . .</b>	<b>51</b>

6.3.2	Algoritmos baseados na busca em largura . . . . .	53
6.3.3	Comparação dos algoritmos de classificação . . . . .	56
7	CONCLUSÃO . . . . .	58
	REFERÊNCIAS . . . . .	60

## 1 INTRODUÇÃO

No campo da Inteligência Artificial (IA), mais especificamente em Aprendizado de Máquina, ao formalizar a ideia de um grupo, de hipóteses que sejam capazes de interagir com problemas e resultar em respostas acuradas e, em certo grau, distintas dos demais componentes do grupo, é adotado o nome de comitês (HANSEN; SALAMON, 1990; DIETTERICH, 2000; KUNCHEVA, 2004).

O conceito de comitês em classificadores binários é usado para referenciar a respectivas hipóteses que são induzidas separadamente e através de um consenso fazem a classificação de novos dados. O objetivo principal, desse agrupamento, é aumentar a capacidade de generalização do modelo individual.

Atualmente, técnicas baseadas em métodos de comitê têm sido amplamente difundidas, resultando em uma área de pesquisa importante e ativa, por tratar-se de uma alternativa eficaz para a criação de classificadores mais acurados. De fato, resultados teóricos e empíricos demonstram que métodos comitê apresentam acurácia superior em relação aos classificadores individuais que os compõem (TUMER; GHOSH, 1996; DIETTERICH, 2000; KUNCHEVA, 2004). O possível aumento de acurácia está diretamente relacionado à capacidade que os métodos comitê tem de agregar o conhecimento obtido por cada um de seus componentes, determinando assim, uma solução global potencialmente superior à melhor das soluções individuais. Para a construção de métodos *ensembles* eficazes, duas premissas básicas devem ser obedecidas: a acurácia individual do classificador de base e a diversidade dos componentes do comitê (DIETTERICH, 2000).

Pode-se interpretar um comitê como um conjunto com uma amostra de vários indivíduos distribuídos em uma distribuição aleatória com uma probabilidade  $P$ , portanto, sob o campo da estatística, pode-se aplicar conceitos de profundidade de dados nesse conjunto. O conceito de profundidade de dados é antigo e amplamente usado, tornando muitos métodos estatísticos univariados para a configuração multivariada generalistas (CHEN; TYLER, 2002). Uma função de profundidade mede a centralidade de um ponto  $X$  em relação a um conjunto de dados ou a uma distribuição de probabilidade  $P$ , tornando possível definir uma ordem e uma versão das classificações para dados multivariados (MOSLER, 2013). Baseado na literatura é sabido que existe várias noções de profundidade de dados disponíveis, como por exemplo a Mahalanobis (MAHALANOBIS, 1936) e Oja (OJA, 1983). Contudo a profundidade do meio espaço de Tukey é uma das funções de profundidade mais populares, e usadas por vários pesquisadores (LIU, 1988; CHEN; TYLER, 2004; MOZHAROVSKYI, 2016; CHEN; TYLER, 2002). A construção de regiões centrais com base em cortes, estimativa robusta de localização multivariada de dados, testes de hipóteses estatísticas multivariadas e classificação supervisionada são alguns exemplos de sua ampla aplicação. Por isso a localização do ponto mais profundo através do

conceito de profundidade de Tukey pode ser uma forma de localizar o centro geométrico, também conhecido como ponto de Bayes, em um comitê.

## 1.1 MOTIVAÇÃO E ARGUMENTAÇÃO

O Classificador Bayesiano alcança a menor taxa de erro de generalização ou maior acurácia obtida através da votação majoritária de conjunto infinito de hipóteses que formam o espaço de versões. Pode-se associar este classificador ao ponto de Bayes deste mesmo espaço. Entretanto, a determinação deste ponto é, por vezes, proibitiva computacionalmente. Neste sentido, alguns trabalhos propõem uma aproximação do ponto de Bayes baseando-se no conceito de profundidade de dados mas, exatamente, da profundidade de Tukey, tendo a mediana de Tukey como um a hipótese a ser aproximada. Em Gilad-Bachrach e Burges (2013) é proposto um algoritmo polinomial que aproxima a mediana de Tukey. Em Gilad-Bachrach, Navot e Tishby (2004) é mostrado que, no espaço de classificadores baseados em hipóteses lineares, a mediana de Tukey é uma aproximação consistente do ponto de Bayes e, também, do centro geométrico do respectivo espaço de versões.

Dois dos principais trabalhos na área de aprendizado de *ensembles* (DIETTERICH, 2000; KUNCHEVA, 2004) são claros ao mencionar que não existe uma teoria unificada em relação aos métodos de construção, bem como não existe uma medida de diversidade mais adequada para cada classificador de base e cada problema avaliado. O consenso é de que não faz sentido a geração de componentes diversos se os classificadores de base que compõem o comitê não são acurados. Os métodos para a combinação das saídas dos componentes são inúmeros e, da mesma forma, não existem resultados sobre a otimalidade de algum.

Em particular, métodos comitê construídos a partir da combinação de RNAs, embora capazes de apresentar bons resultados, frequentemente tem um desempenho, em termos de capacidade de generalização, inferior a métodos como *Bagging* e *AdaBoost*. Uma justificativa para esse fato pode ser encontrada na acurácia das RNAs selecionadas ou ainda na diversidade dos componentes e forma de combinação das soluções. Trabalhos anteriores relacionados a comitê de RNAs incluem os métodos propostos por Hansen e Salamon (1990), Opitz, Shavlik et al. (1996), Zhou, Wu e Tang (2002).

As noções de profundidade para conjuntos de dados são importantes na análise estatística. Tukey e Cleveland (1988) popularizaram uma das principais definições para a profundidade de dados, ou também conhecida como profundidade de meio espaço, onde sugeriu o uso dos contornos de profundidade para expressar os limites de regiões de todos os pontos com profundidade igual. A mediana de Tukey pode servir como um estimador de ponto para o conjunto de dados (um “centro”) que é robusto contra *outliers*, não depende de distâncias e é invariante sob transformações afins (ROUSSEEUW; RUTS,

1998; FREUND; SCHAPIRE, 1996).

Por causa das características que a mediana de Tukey possui a questão de projetar algoritmos eficientes para sua localização atraiu muita atenção de pesquisadores como Langerman e Steiger (2000a), Langerman e Steiger (2000b), Kreveld et al. (2008), Gil, Steiger e Wigderson (1992), Bern e Eppstein (2000), Aloupis et al. (2003), Aloupis et al. (2002).

Esse trabalho possui, como motivação principal, o desenvolvimento de classificadores baseados na formação de ensembles tendo como componentes hipóteses que possuam uma proximidade geométrica com a mediana de Tukey. Neste contexto comparou-se a acurácia dos classificadores propostos com outros classificadores baseados em pontos de centralidade do espaço de versões, como exemplo: centro da maior hiperesfera inscrita (SVM), centro de massa (VSRM) , centro médio (BPM) e centro analítico (EACC).

## 1.2 OBJETIVOS

O objetivo geral desse trabalho é desenvolver métodos capazes de identificar indivíduos em um comitê já definido que possuem uma grau de profundidade, e identificar aquele indivíduo que possui uma maior profundidade. O conceito de profundidade é baseado na profundidade de Tukey, que será explicada durante esse trabalho.

Como objetivo específicos pode-se pontuar:

1. Os métodos de buscas dos algoritmos utilizando o conceito de buscas por profundidade e por largura, buscando localizar a hipótese solução mais profunda.
2. Propor outros métodos de exploração do espaço de versões na região relacionada a profundidade de Tukey, no sentido de encontrar hipóteses de maior profundidade mais aprimoradas.
3. Desenvolver algoritmos capazes de solucionarem o problema proposto

## 1.3 CONTRIBUIÇÕES

As principais contribuições desse trabalho são apresentadas a seguir:

1. Apresentar o Perceptron de Tukey;
2. Introduzir os algoritmos baseados na Busca em Profundidade:
  - Perceptron de Tukey por Busca em Profundidade;
  - Perceptron de Tukey por Busca em Profundidade com Subcomitê por Vizinhaça.

3. Introduzir os algoritmos baseados na Busca em Largura:

- Perceptron de Tukey por Busca em Largura;
- Perceptron de Tukey por Busca em Largura com Subcomitê por Vizinhança;
- Perceptron de Tukey por Busca em Largura com Subcomitê por Votação.

## 1.4 ORGANIZAÇÃO

Essa dissertação está organizada em 7 capítulos. Após o capítulo de introdução, a fundamentação teórica e os conceitos principais sobre profundidade de dados são apresentados no Capítulo 2. O Capítulo 3 é destinado a abordar os principais conceitos de aprendizado de máquina, como o paradigma de aprendizado supervisionado e o problema de classificação binária. Além dos conceitos fundamentais referentes a teoria de aprendizado por comitê são apresentados. Isso constitui a base do trabalho proposto.

No Capítulo 4, é apresentado os principais classificadores que essa dissertação tem como objetivo relacionar com a Mediana de Tukey. Já no Capítulo 5 o objetivo é apresentar o Perceptron de Tukey para geração de uma hipótese mais profunda em um comitê de Perceptrons balanceados combinados pelo voto majoritário e os algoritmos desenvolvidos. O estudo experimental realizado, as bases de dados avaliadas, os resultados obtidos, bem como as discussões relevantes em relação a eles são apresentadas no Capítulo 6. Por fim, o Capítulo 7 apresenta as conclusões do trabalho em questão, algumas considerações finais e trabalhos futuros sugeridos.

## 2 PROFUNDIDADE DE DADOS

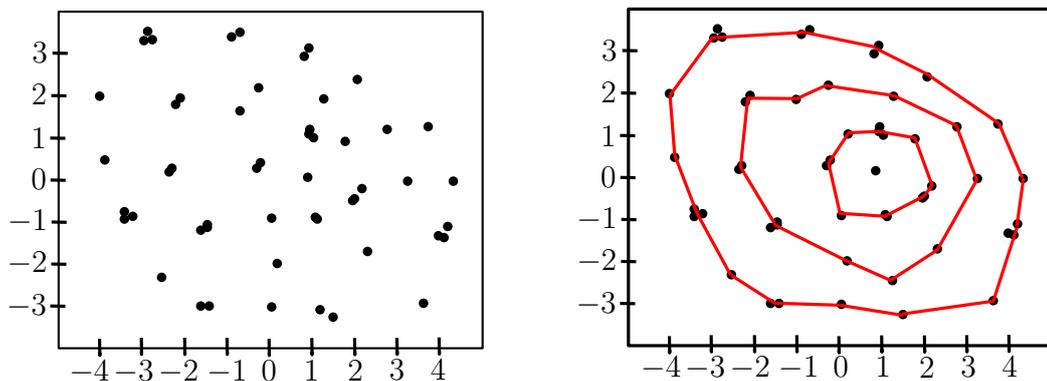
Segundo Liu (2003), a profundidade de dados é uma medida de profundidade ou centralidade de um determinado ponto em relação a uma distribuição multivariada, além de fornecer uma ordem natural do centro para fora dos pontos de dados em uma determinada amostra, é uma forma para definir uma ordem e uma versão de classificações para dados multivariados.

Várias noções de profundidade são encontradas na literatura, como a profundidade *Convex Hull Peeling* (PORZIO; RAGOZINI, 2007), a profundidade Simplicial (LIU, 1988), a profundidade de Oja (OJA, 1983) e a profundidade de Mahalanobis (MOSLER, 2013). Tais profundidades são abordadas adiante.

### 2.1 PROFUNDIDADE *CONVEX HULL PEELING*

O conceito de profundidade *Convex Hull Peeling* foi criado por Barnett (1976), segundo o autor a profundidade de um ponto  $x = (x_1, \dots, x_p) \in S_n \subset \mathbb{R}^d$ , relativamente de dados  $p$ -dimensionais  $S_n = \{x_i = (x_{i1}, \dots, x_{ip}); i = 1, \dots, n\}$ , corresponde ao nível da camada convexa onde este ponto se encontra, ou seja, se este ponto está situado na  $k$ -ésima camada então terá profundidade  $k$ , ou seja, ao analisar a Figura 1 é possível observar que existe 3 camadas de profundidade.

Figura 1 – *Convex Hull Peeling*.



Fonte: Elaborada pelo autor, 2022.

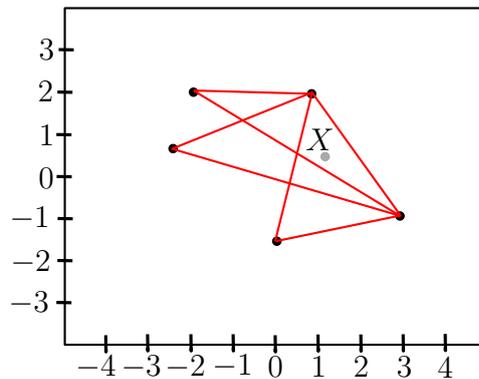
O objetivo do *convex hull* é a construção de camadas convexas sucessivas obrigando todos os pontos situar-se sobre essas várias camadas. O processo consiste em construir uma camada inicial com os pontos considerados mais externos, logo em seguida passa-se para a camada mais interna à anterior, e de maneira análoga, constrói-se essa camada até que todos os pontos estejam empregados. Por fim, tem-se uma estrutura com uma representação extremamente intuitiva pois um ponto pertencente à camada mais interna é mais profunda do que outro ponto pertencente à uma camada considerada mais externa.

## 2.2 PROFUNDIDADE SIMPLICIAL

O conceito de profundidade Simplicial foi introduzida por Liu et al. (1990). Esse tipo de profundidade baseia-se na ideia de *simplex*. O *simplex* é um conceito geométrico e pode ser definido como um involutório convexo de  $p + 1$  pontos em  $\mathbb{R}^d$  (JÚNIOR et al., 2003). De forma informal, pode-se explicar que o *simplex* é a generalização do conceito de triângulo a outras dimensões, ou seja, o *simplex* em  $\mathbb{R}$  é um segmento de reta, em  $\mathbb{R}^2$  é um triângulo e em  $\mathbb{R}^3$  é um tetraedro.

Portanto, a profundidade Simplicial de  $x = (x_1, \dots, x_p) \in S_n \subset \mathbb{R}^d$ , relativamente de dados  $p$ -dimensionais  $S_n = \{x_i = (x_{i1}, \dots, x_{ip}); i = 1, \dots, n\}$ , é determinada como o número de *simplex* fechados, com  $p + 1$  vértices pertencentes a  $S_n$ , que contêm  $X$  (LIU et al., 1990). Como é mostrado na Figura 2 onde a profundidade do ponto  $X$  é igual a 3.

Figura 2 – *Simplicial*.



Fonte: Adaptada de Wazneh (2015).

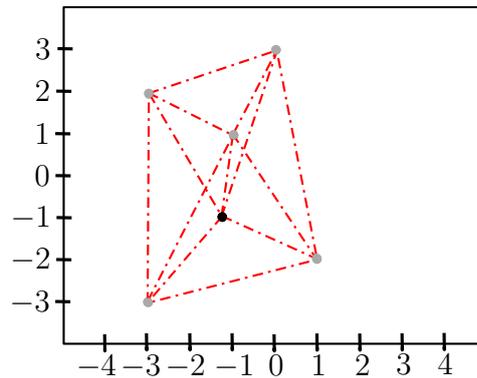
## 2.3 PROFUNDIDADE DE OJA

A profundidade de Oja, como o próprio nome indica, foi introduzida por Oja (1983). O conceito dessa profundidade baseia-se em um ponto  $x = (x_1, \dots, x_p) \in S_n \subset \mathbb{R}^d$ , relativamente de dados  $p$ -dimensionais  $S_n = \{x_i = (x_{i1}, \dots, x_{ip}); i = 1, \dots, n\}$ , onde a soma dos volumes de todos os *simplex* fechados que tem como vértices  $x$  e quaisquer outras  $p$  observações pertencentes ao conjunto de dados  $S_n$ . Para melhor compreensão, a profundidade de Oja de um ponto  $x$ , em um caso bivariado, relativamente a uma conjunto de dados  $S_n$  pode ser expressada pela soma das áreas de todos os triângulos cujos vértices sejam  $x, x_i, x_k$ , como  $x_i$  e  $x_k$  pertencentes a  $S_n$ .

## 2.4 PROFUNDIDADE DE MAHALANOBIS

A profundidade de Mahalanobis, como o próprio nome indica, é baseada na famosa distância de Mahalanobis (MOSLER, 2013) e, de acordo com Hamurkaroglu, Mert e

Figura 3 – Profundidade de Oja.



Fonte: Adaptada de Aloupis e McLeish (2005).

Saykan (2004), a profundidade de Mahalanobis ( $M_hD$  - *Mahalanobis Depth*) pode ser expressa a partir de um ponto  $x = (x_1, \dots, x_p) \in S_n$  para todo  $R^d$ , relativamente a um conjunto de dados  $p$ -dimensional  $S_n = \{x_i = (x_{i1}, \dots, x_{ip}); i = 1, \dots, n\}$ , é definida da seguinte forma:

$$M_hD(x; S_n) = [1 + (x - \bar{x})^T S^{-1} (x - \bar{x})]^{-1} \quad (2.1)$$

onde  $\bar{x}$  representa a média dos pontos  $x_1, \dots, x_n$  pertencentes a  $S_n$  e  $S$  é a matriz de covariância amostral de  $S_n$

## 2.5 PROFUNDIDADE E MEDIANA DE TUKEY

No ano de 1975, o estatístico estadunidense John Wilder Tukey introduziu o conceito sobre profundidade de dados. Esse conceito mede como os pontos profundos são incorporados a um conjunto de dados fornecendo uma ordem mais refinada de dados multivariados. Geralmente, a profundidade dos dados estatísticos é uma função que determina a localização central de um ponto em um aglomerado de dados.

A profundidade de Tukey, também conhecida como profundidade de meio espaço, é uma das mais importantes noções de profundidades e historicamente foi a primeira a ser mencionada.

Para seu entendimento, considere uma variável aleatória  $x$  com espaço amostral finito, com medida de probabilidade  $P$ , como  $\{x_1, x_2, \dots, x_n\}$  em  $R^d$ . A profundidade de Tukey em um ponto  $z \in R^d$ , é definido como a menor área de probabilidade de um meio espaço fechado contendo  $z$ , isto é:

$$D(\vec{x}; P) = \inf_H \{P(H) : H \text{ é um semi espaço fechado em } \mathbb{R}^d : \vec{x} \in H\}. \quad (2.2)$$

Com um visão mais lúdica, considere o lançamento de um dado não viciado de seis faces e que  $X$  representa o número de pontos da face de saída. Tem-se  $P(X = x_i =$

$i) = \frac{1}{6}, i = 1, \dots, 6$ . Pode-se afirmar que a profundidade de Tukey de  $x_2$  é calculada da seguinte forma:

$$\begin{aligned} D(\vec{x}_2; P) &= \inf_h \{P(H) : H \text{ é um semi espaço fechado em } \mathbb{R}^2 : \vec{x}_2 \in H\} \\ &= \inf_h \{P(x_2) = (x \leq 2); 1 - F(x_2) = P(x \geq 2)\} \\ &= \inf_h \left\{ \frac{2}{6}, 1 - \frac{1}{6} \right\} = \frac{2}{6} \end{aligned} \quad (2.3)$$

A Tabela 1 informa os valores calculados da profundidade para os outros valores. Percebe-se que os pontos  $x_3$  e  $x_4$  são os pontos mais profundos.

Tabela 1 – Valores da profundidade de Tukey

$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$
1/6	2/6	3/6	3/6	2/6	1/6

Fonte: Neto (2008).

Segundo Mozharovskyi (2016) a profundidade de Tukey possui algumas propriedades importantes como ser invariante afim, tender a zero no infinito, é monótono em relação ao ponto mais profundo, quase côncavo e semicontínuo superior.

Ou seja, segundo os autores Zuo e Serfling (2000), a propriedade de invariância afim é quando a profundidade de um ponto  $X \in \mathbb{R}^d$  não deve depender do sistema de coordenadas subjacente ou, em particular, das escalas das medições subjacentes. Já a propriedade de tender zero ao infinito ocorre quando a profundidade de um ponto  $X$  deve aproximar-se de zero como  $X$  se aproxima do infinito.

De acordo com Liu et al. (1990), um ponto é monótono em relação ao ponto mais profundo, quando um ponto  $X \in \mathbb{R}^d$  se afasta do ponto com maior profundidade ao longo de qualquer distância fixa no centro, a profundidade em  $X$  deve diminuir monotonicamente.

De acordo com o quarto postulado de Mosler (2013), onde afirma que a função de quase concavidade ocorre quando a distribuição de dados gera uma área convexa. E segundo Donoho, Gasko et al. (1992) a profundidade de Tukey é semicontínuo superior quando a probabilidade mínima de qualquer meio espaço contendo  $X$  é uma função semicontínua superior de  $X$ , se a probabilidade for absolutamente contínuo, então a probabilidade mínima é uma função contínua de  $X$ .

O ponto mais profundo ou mais interno em relação à profundidade de Tukey é comumente visto como uma mediana multivariada e, por isso, é referido a esse ponto mais profundo como mediana de Tukey. Pode-se definir a mediana de Tukey como o ponto com maior profundidade de Tukey; é uma generalização da mediana unidimensional para altas dimensões. O comportamento dessa mediana é bem compreendido sob o modelo de apresentado em Huber et al. (1973) no qual uma fração dos dados são *outliers* arbitrários. Na literatura existe numerosos artigos que estudam várias propriedades da profundidade e

mediana de Tukey. O mais notável é Donoho (1982), no qual o autor mostrou que o ponto de localização da mediana de Tukey pode ser tão alto quanto  $\frac{1}{3}$ , independentemente da dimensão  $d$  do dados.

Então, baseando-se no raciocínio da profundidade de Tukey, seja  $X^n = \{X_1, X_2, \dots, X_n\}$  representa um conjunto de dados em  $\mathbb{R}^d$  com o  $d \geq 2$  e seja  $S^{d-1} = \{u \in \mathbb{R}^d : u'u = 1\}$ , onde  $u$  é o vetor unitário do conjunto de todos os elementos em  $\mathbb{R}^d$ . Para um dado ponto  $X \in \mathbb{R}^d$ , considere a posição de sua projeção univariada  $u'x$  no conjunto de dados univariado  $u'X^{(n)} = \{u'X_1, \dots, u'X_n\}$  para alguns  $u \in S^{d-1}$ . Defina a profundidade de  $u'x$  dentro de  $u'X^n$  como o menor número de pontos em  $u'X^n$  que não são maiores que  $u'x$  e o número de pontos que não são menores que  $u'x$ . A profundidade do ponto multivariado  $X \in \mathbb{R}^d$  dentro de  $X^n$ , conforme definido em Mozharovskyi (2016) é então a menor profundidade possível. Mas formalmente, a profundidade de Tukey para um ponto  $x$  em relação a um conjunto de dados  $X^n$  é definido como:

$$\text{depth}(x; X^n) = \min_{u \in S^{d-1}} nF_{u,n}(u'x). \quad (2.4)$$

onde  $F_{u,n}$  é a função de distribuição cumulativa empírica de  $X^{(n)}$ . Para um dado  $u \in S^{d-1}$ ,  $nF_{u,n}(u'x)$  corresponde ao número de pontos em  $X^{(n)}$  que não são maiores que  $u'x$ , enquanto o número que não são menores que  $u'x$  é dado por  $nF_{u,n}(u'x)$ . Em uma dimensão, o ponto de profundidade máxima coincide com a mediana univariada de Tukey e, portanto, em dimensões superiores

$$\arg \max_{x \in \mathbb{R}^D} \text{depth}(x; X^{(n)}) \quad (2.5)$$

é geralmente referido como uma mediana multivariada ou, mais especificamente, como mediana de Tukey.

Para formalizar esse trabalho precisa-se de uma versão mais funcional da profundidade de Tukey. Para isso, seja então,  $X$  um vetor aleatório com medida de probabilidade  $P$  em  $\mathbb{R}^d$ , e  $F_u(y)$  representa a função de distribuição cumulativa de  $u'X$  para  $u \in S^{d-1}$ . Uma generalização da profundidade de Tukey para medidas de probabilidade, é dado por:

$$\pi(x; P) = \inf_{u \in S^{d-1}} F_u(u'x). \quad (2.6)$$

Se  $P_n$  representa uma probabilidade empírica derivada de  $X^n$ , então:

$$\text{depth}(x; X^n) = n\pi(x; P_n). \quad (2.7)$$

Pode-se referenciar a profundidade de Tukey, além da descrita na Equação (2.2), como

$$\pi(x; P) = \inf_{u \in S^{d-1}} P \{H(u, x)\}, \quad (2.8)$$

onde  $H(u, x) = \{y \in \mathbb{R}^d : u'y \leq u'x\}$  é o semi-espaço fechado em  $\mathbb{R}^d$  com limite ortogonal a  $y \in S^{d-1}$  e passa por  $x$ . Portanto, a mediana de Tukey pode ser expressa como

$$\arg \max_{x \in \mathbb{R}^d} \pi(x; P). \quad (2.9)$$

Formando assim uma versão funcional da mediana de Tukey que pode ser definida exclusivamente para uma medida de probabilidade  $P$  em  $\mathbb{R}^d$  como:

$$T(P) = \frac{\int_{\text{Med}(P)} y dy}{\int_{\text{Med}(P)} dy}, \quad (2.10)$$

onde a integral é em relação à medida de Lebesgue no plano gerado por  $\text{Med}(P)$ . A função  $T(P)$  é então equivariante afim. Ou seja, se  $P^*$  representa a medida de probabilidade de  $X^* = AX + b$  com  $A$  sendo não singular, então  $T(P^*) = AT(P) + b$ . Existem outras opções para definir uma média de preservação equivariante afim sobre  $\text{Med}(P)$ .

### 3 APRENDIZADO DE MÁQUINA

O campo de pesquisa de Inteligência Artificial (IA) propõe sistemas que simulem uma inteligência similar à humana, seus estudos começaram no final da década de 1940 como os pesquisadores Russell e Norvig (2003) mostram em seu trabalho. Nos dias atuais existem várias subáreas de pesquisas provenientes do campo de IA, dentre elas, a principal área de estudo desse trabalho, chamada de Aprendizado de Máquina. Essa subárea busca o estudo de reconhecimento de padrões e da teoria do aprendizado computacional. Uma boa definição sobre Aprendizado de Máquina foi feita por Weiss (1992) no ano de 1959 onde o autor afirma que essa subárea é responsável em proporcionar a habilidade de aprendizado aos computadores sem serem explicitamente programados.

Na subárea de Aprendizado de Máquina é sabido da existência de três paradigmas, que seriam: o aprendizado supervisionado, não supervisionado e o aprendizado por reforço. De acordo com Faceli et al. (2011) o aprendizado supervisionado baseia-se na ideia da presença de um “supervisor externo”, que conhece a saída dos rótulos para cada exemplo apresentado ao sistema de classificação. Com isso, esse “supervisor externo” pode avaliar a capacidade de predição de uma saída apresentando exemplos inéditos. O trabalho aqui apresentado está embasado no paradigma do aprendizado supervisionado e, em particular, para problemas de classificação binária.

Já o aprendizado não-supervisionado não possui os rótulos para os seus dados (FACELI et al., 2011). Entretanto, pode-se encontrar padrões existentes, gerando uma organização de dados em agrupamentos consistentes, a partir de similaridades e diferenças encontradas em seus padrões.

O aprendizado por reforço possui como objetivo reforçar ou recompensar uma ação considerada positiva para o sistema, e em contrapartida, deve-se punir uma ação considerada negativa. Um exemplo clássico é proposto por Shi et al. (2018) onde o autor ensina um robô encontrar a melhor trajetória entre dois pontos. Os algoritmos de aprendizado utilizados nessa tarefa, em geral, punem a travessia do robô por caminhos considerados ruins e recompensa a passagem por caminhos considerados bons.

#### 3.1 CLASSIFICAÇÃO BINÁRIA

Considere o paradigma de um aprendizado que é sabido, a priori, os rótulos dos exemplos apresentados para classificação. Uma classificação binária segue o seguinte raciocínio: seja o conjunto  $Z$  dos dados de entrada de cardinalidade  $m$ , constituído de um conjunto de vetores de entrada  $x_i$  e de um conjunto de escalares  $y_i$ . Defina  $Z = \{(x_i, y_i) : i \in \{1, 2, \dots, m\}\}$ , como o conjunto de treinamento de  $m$  amostras de um função desconhecida  $f$ . Cada elemento dos vetores de entrada é um valor pertencente ao conjunto  $\mathbb{R}$  e cada vetor de entrada está, portanto, inserido em um espaço de dimensão  $d$ , i.e.,  $x_i \in \mathbb{R}^d$ .

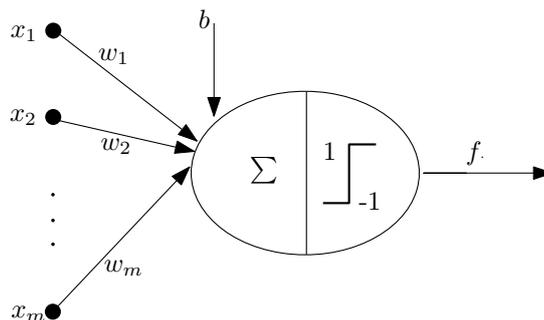
Esses elementos são denominados como atributos ou características, sendo representados como valores reais ou discretos. Cada vetor apresentado à entrada é rotulado por um escalar  $y_i$ . No caso do problema de classificação binária, os valores de  $y_i$  são estruturados em um conjunto discreto de classes, i.e.,  $y_i \in \{-1, +1\}$ , com o elemento  $i \in \{1, 2, \dots, m\}$ . Apresentado ao classificador um conjunto de treinamento, um algoritmo de aprendizado será capaz de construir um classificador, dado por uma hipótese (representada por um hiperplano) em relação à função  $f : \mathbb{R}^d \rightarrow \{-1, +1\}$ .

No momento da classificação, o classificador gerado, será responsável em reconhecer padrões de uma classe, predizendo os valores correspondentes de  $y$  à medida que amostras adicionais  $x$  são apresentadas ao classificador. Essas amostras adicionais formam o conjunto teste.

### 3.2 PERCEPTRON

No ano de 1958 foi implementada a primeira Rede Neural Artificial (RNA) criada por Rosenblatt (1958). Essa rede baseava-se no modelo matemático do neurônio desenvolvido por McCulloch e Pitts (1943) com a adição de um processo de treinamento para atualização dos pesos das RNAs. Ainda que essa rede fosse simplória, com apenas uma camada de neurônios, ela foi capaz de apresentar uma boa acurácia de predição quando apresentado problemas de classificação. A Figura 4 representa a estrutura simples de uma rede perceptron. Conforme é possível notar, os exemplos são apresentados à uma entrada de dados, que são pré-processados e então encaminhados à rede, que possui apenas um neurônio.

Figura 4 – Topologia do modelo Perceptron.



Fonte: Enes, Villela e Neto (2016).

Estruturalmente, o Perceptron realiza um mapeamento de um espaço de entrada de dimensão  $d$  para um espaço de saída de dimensão reduzida  $n$ , associando cada vetor de entrada,  $x_i$ , a um componente de um vetor de pesos  $w_i$  de dimensão  $d$ , com  $i \in \{1, 2, \dots, m\}$  e uma camada de saída  $y$  formada por  $n$  unidades. Quando empregado em problemas de classificação binária, é suficiente a existência de somente um elemento processado na camada de saída.

De acordo com Faceli et al. (2011) para a correção de possíveis erros, a rede perceptron é treinada de acordo com um algoritmo supervisionado e usa como forma de ativação do neurônio a função do tipo limiar. Na etapa de treinamento, os pesos  $w_i$  são ajustados da seguinte forma:

$$w_j^{t+1} = w_j^t + \eta x_{ij}(y_i - \hat{f}(x_i)), \quad (3.1)$$

em que  $w_j^t$  é o peso da  $j$ -ésima conexão de entrada no instante de tempo  $t$ ,  $\eta$  é chamado de taxa de aprendizado, essa taxa determina a magnitude do ajuste feito no valor de cada peso, já  $x_{ij}$  é o valor do  $j$ -ésimo atributo do vetor  $x_i$  apresentado na entrada da rede,  $\hat{f}(x_i)$  é a saída produzida pela rede no instante de tempo  $t$  e por último  $y_i$  é a saída desejada para a rede.

Alguns anos após a criação do Perceptron, Novikoff (1963) provou o teorema de convergência que em problemas linearmente separáveis o algoritmo converge com um número finito de correções tendo como limite superior a razão inversa do valor ao quadrado da margem de parada.

### 3.2.1 Formulação primal

Em termos matemáticos, a rede Perceptron é capaz de encontrar um hiperplano separador, com variáveis primais, através da solução de um sistema de inequações lineares, dado por:

$$f(x_i) = \begin{cases} +1, & \langle w, x_i \rangle + b \geq 0 \\ -1, & \langle w, x_i \rangle + b < 0, \end{cases} \quad (3.2)$$

no qual  $w$  é vetor dos pesos,  $b$  valor do *bias* e  $\langle w, x_i \rangle$  é o produto interno de  $w$  por  $x_i$ . Uma amostra de treinamento  $(x_i, y_i)$  é classificada erroneamente se  $\langle w, x_i \rangle + b < 0$ , quando isso ocorre é preciso aplicar a regra de correção da Equação (3.1) até que não haja mais erros.

A resposta final do algoritmo de aprendizado é obtida quando o hiperplano solução é capaz de classificar todas as amostras de treinamento corretamente. O pseudocódigo do Algoritmo 1 descreve a formulação primal relativa ao treinamento do Perceptron.

## 3.3 MÉTODOS *ENSEMBLES*

Hansen e Salamon (1990), foram os primeiros a proporem, um estudo combinando várias RNAs diferentes para solucionar problemas de classificação binária. A principal ideia era que, em geral, problemas de classificação binária possuem mais de um mínimo local. Por essa razão, ao se utilizar apenas uma única rede, poderia esta, estar restrita a exclusivos pontos de mínimo local. Dessa forma, a solução poderia ser potencializada caso outras redes fossem avaliadas em conjunto.

E de acordo com os autores, redes de comitê que possuem taxas de erro individuais distintas e menores que 50%, corroboram na melhoria da acurácia. Isso ocorre, já que

---

**Algoritmo 1:** Perceptron Primal
 

---

**Entrada:** conjunto de treinamento  $Z = \{(x_i, y_i)\}$  de cardinalidade  $m$ ;  
 taxa de aprendizado  $\eta$ ;  
 limite superior no número de iterações  $max$ ;  
**Saída:** vetor de pesos  $w$  e bias  $b$ ;

```

1 início
2   inicializar  $(w^0, b^0)$ ;
3    $j \leftarrow 0$ ;
4    $t \leftarrow 0$ ;
5    $stop \leftarrow \underline{falso}$ ;
6   enquanto  $j \leq max$  e  $\neg stop$  faça
7      $erro \leftarrow \underline{falso}$ ;
8     para  $i$  de 1 até  $m$  faça
9       se  $y_i (\langle w^t, x_i \rangle + b^t) < 0$  então
10         $w^{t+1} \leftarrow w^t + \eta x_i y_i$ ;
11         $b^{t+1} \leftarrow b^t + \eta y_i$ ;
12         $t \leftarrow t + 1$ ;
13         $erro \leftarrow \underline{verdadeiro}$ ;
14      fim se
15    fim para
16    se  $\neg erro$  então
17       $stop \leftarrow \underline{verdadeiro}$ ;
18    fim se
19     $j \leftarrow j + 1$ ;
20  fim enqto
21 fim
  
```

---

a probabilidade da saída do modelo em conjunto estar classificada erroneamente seria inferior a menor das taxas de erro dos seus componentes isoladamente. Ao realizar testes empíricos é possível validar o trabalho proposto dos autores, ao mostrar ganhos em termos da capacidade de generalização do modelo criado em relação a uma RNA individual. Além disso, no mesmo trabalho de Hansen e Salamon (1990), o termo comitê já é cunhado para designar a combinação dos classificadores.

Por definição, comitê é um paradigma de aprendizado em que um conjunto de possíveis propostas para a solução de um problema, denominadas componentes, são empregadas em conjunto na proposição de uma solução única. Nesse caso, cada componente representa, isoladamente, uma potencial solução para um problema Sollich e Krogh (1996). Essa área de pesquisa é recente, por isso não existe uma teoria unificada de aprendizado por *ensembles*. Contudo, existem muitas razões teóricas para combinar classificadores e ainda mais razões empíricas da eficácia dessa abordagem.

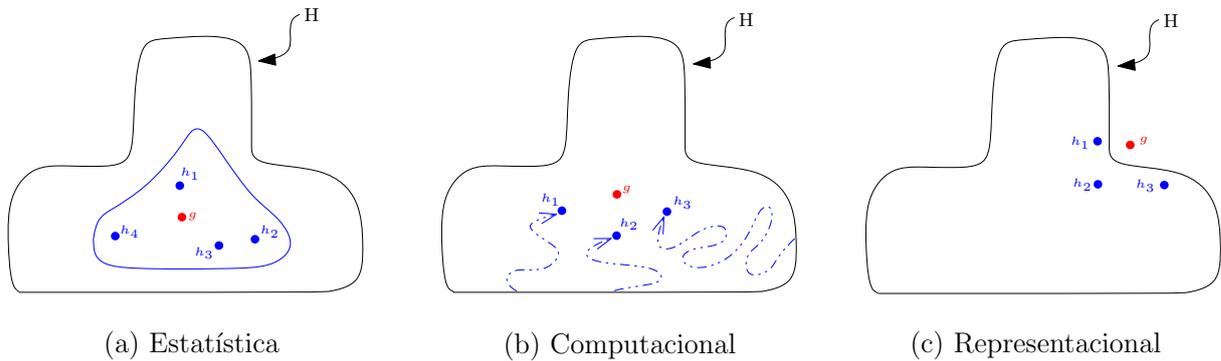
Para uma melhor explicação tome  $H$  como o espaço onde possui todas as possíveis soluções para um dado problema qualquer,  $h_i$  é uma possível solução e  $g$  é a solução hipotética ótima para o problema apresentado. De acordo com Dietterich (2000) é possível

construir *ensembles* baseado em três justificativas descritas a seguir:

- Estatística: suponha que os dados de treinamento referente ao problema seja limitado, então o espaço de soluções viáveis fica restrito às possíveis soluções geradas pelo conjunto de treinamento limitado. Observando a Figura 5-(a) pode-se perceber que o modelo é treinado e está restrito ao subconjunto formado apenas pelas observações restritas à marcação linear interna mostrada na imagem. O processo de geração de soluções pode determinar várias hipóteses com desempenhos semelhantes. Supondo que a seleção das hipóteses para a construção de um comitê é efetiva, ou seja, bons modelos são selecionados, a solução combinada tende a prover uma boa aproximação em relação à solução ótima. Como é mostrado na Figura 5, as quatro hipóteses sugeridas pelo mecanismo de seleção estão distribuídas em torno de  $g$  e o comitê gerado pela combinação das quatro hipóteses tende a se aproximar mais de  $g$  do que qualquer uma das hipóteses avaliadas isoladamente.
- Computacional: ao empregar algoritmos de aprendizados tenta-se realizar um refinamento do processo de busca em um ambiente local e limitado, se uma solução é encontrada então é proposto uma modificação para validar a melhoria do desempenho. Dessa forma, suas soluções geradas podem estar restritas à convergência para um ponto de ótimo local. Entretanto, problemas de aprendizado podem apresentar vários pontos de ótimo local e, mesmo que o conjunto de dados não seja restrito, não há garantias de que uma solução gerada seja a de ótimo global. E em relação a termo computacionais, encontrar um ótimo global pode ser muito difícil. Supondo, por exemplo, que o ponto de partida de cada solução apresentada seja diferente entre si, o comitê formado por diferentes processos de busca local podem proporcionar uma melhor aproximação para  $g$  do que um único componente gerado. Esse fato é observado na Figura 5-(b), na qual as linhas tracejadas representam as buscas locais.
- Representacional: ocorre, em grande parte das aplicações, que uma solução ótima para o problema em questão não seja representada por nenhuma das hipóteses pertencentes ao espaço de solução viável do modelo. Sendo assim, ao conciliar vários modelos factíveis em  $H$ , o espaço de hipóteses é estendido de modo a se aproximar de  $g$ . De acordo com a teoria de comitê, a combinação de classificadores pretende evitar que a capacidade de representação fique restrita ao conjunto finito de hipóteses, conforme é mostrado na Figura 5-(c).

Baseados nesses três fatores que justificam a utilização de comitês e ainda ilustram, pode-se concluir que o uso de comitês possuem a capacidade de generalização superior se comparado ao uso de apenas um componente individual. Contudo, a existência de vários resultados empíricos comprovem a eficácia, em termo de capacidade de generalização da construção de um comitê, não há um garantia de que a precisão do comitê será sempre

Figura 5 – Três razões fundamentais para a construção de um *ensemble*.



Fonte: Enes, Villela e Neto (2016)

superior ao melhor dos componentes. Sendo assim, a justificativa dessa abordagem é motivado principalmente pela possibilidade de aumento da capacidade de generalização do modelo quando somado em um comitê. Para que essa possibilidade ocorra é preciso respeitar duas premissas básicas:

- Acurácia individual: para se compor o comitê, cada classificador deve ter um bom desempenho. Quanto melhor o desempenho do classificador individual melhor tende a ser o resultado da composição dos classificadores.
- Diversidade dos componentes: de maneira clara, se todos os classificadores forem iguais não haverá ganho na capacidade de generalização do comitê. Dessa forma, outra premissa fundamental é a diversidade dos componentes para a construção dos comitês.

Sendo estipulados o modelo de classificador individual e as estratégias para geração de diversidade no modelo, para construção de um modelo comitê é preciso respeitar três etapas Kuncheva (2004):

- Geração de um conjunto de candidatos a componentes do comitê: a geração de componentes viáveis deve-se atentar para a acurácia de cada componente, sendo essa acurácia superior a 50% em relação a problemas de classificação binária.
- Seleção dos candidatos viáveis: em seguida a geração do conjunto candidatos, o processo de seleção funciona separando aqueles componentes que contribuem mais fortemente para a criação do modelo. A contribuição é feita em diversas formas, como por exemplo, pela introdução de medidas de dissimilaridade de componentes. O processo de seleção é ainda responsável por determinar a quantidade de componentes que formarão o comitê, com base no critério de seleção adotado.

- Combinação das saídas geradas: essa etapa é responsável avaliação de como os componentes serão analisados. Em geral, essa avaliação é feita baseada na média dos componentes e esquemas de votação, podendo ser ponderadas ou não.

A diferenciação de dois modelos comitê ocorre quando leva-se em conta a escolha do classificador de base na presença ou não de tratamento nos dados de entrada dos modelos e ainda o método de combinação adotado, como referenciado por Roli, Giacinto e Vernazza (2001), e não o processo de construção do mesmo. Isso ocorre porque as diferenças relativas à geração de um conjunto de componentes e seleção dos candidatos viáveis não são suficientes para garantir que os modelos gerados são distintos. Processos diferentes podem levar a construção de um mesmo modelo comitê.

A construção de bons classificadores que possuem boa capacidade de generalização e serem funcionais é uma tarefa difícil, por isso existem vários métodos para desenvolver *ensembles* que sigam as duas premissas básicas de combinação de classificadores e que sejam capazes de prover uma capacidade de generalização ótima. Um desses métodos que tem se destacado na literatura é chamado de *boosting* (FREUND; SCHAPIRE, 1997) e que possui como o principal representante o algoritmo chamado AdaBoost, introduzido por Freund e Schapire (1996).

O *AdaBoost*, que é um modelo baseado em *boosting*, é referenciado na literatura como o estado da arte em métodos comitê. Esse método combina vários classificadores até obter um classificador considerado robusto para solução do problema apresentado à ele. Outro tipo de método de classificador que usa comitês é o modelo do tipo *bagging*.

O trabalho desenvolvido usufrui do algoritmo do Comitê de Perceptrons Balanceados para a construção do espaço de versões. Esse algoritmo baseia-se em uma combinação de classificadores do tipo de Perceptron Balanceado (PB), selecionados conforme as medidas de diversidades.

### 3.3.1 *Ensemble de Perceptrons Balanceados*

O modelo de um perceptron padrão Rosenblatt (1958), devido à sua formulação, tende a manter o hiperplano de separação próximo às últimas amostras corrigidas, podendo gerar, em alguns casos, um hiperplano de separação desequilibrado. Por isso ao aumentar o valor da margem significativamente, gera um reposicionamento do hiperplano, mantendo uma mesma direção, resultando o hiperplano em uma posição equidistante das duas classes. Essa nova posição é a maior solução de margem para essa hipótese, o que provavelmente implica um melhor desempenho de generalização para esse modelo.

De acordo com Enes, Villela e Neto (2015), dado que o espaço de versões possui um lado com  $Z^+ = \{(x_i, y_i) \in Z : y_i = +1\}$  e outro lado com  $Z^- = \{(x_i, y_i) \in Z : y_i = -1\}$  e

considerando a distância entre o hiperplano de separação e as duas possíveis classes, as semi-margens são definidas da seguinte forma

$$\begin{aligned}\gamma^+ &= \min \{y_i(\langle w, x_i \rangle + b), \forall x_i \in Z^+\} \\ \gamma^- &= \min \{y_i(\langle w, x_i \rangle + b), \forall x_i \in Z^-\}.\end{aligned}\tag{3.3}$$

O deslocamento é determinado com base na média dos valores das margens. Assim, define-se o coeficiente de deslocamento como  $\Gamma = (\gamma^+ + \gamma^-)/2$ . Dado o valor de  $\Gamma$ , a atualização do valor do *bias* é dado por

$$b \leftarrow b - \gamma^- + \Gamma.\tag{3.4}$$

Se o vetor de pesos não é alterado, a solução final do PB é igual a solução do Perceptron clássico. Entretanto, ao introduzir o deslocamento do *bias* no modelo, um novo hiperplano é obtido mantendo a margem mais larga possível, considerando a mesma solução.

Para a geração de *ensembles* é preciso respeitar uma das premissas básicas, que seria a diversidade dos componentes do modelo. A medida de dissimilaridade é a mais adequada para a construção de *ensembles*. Cada componente do comitê é gerado a partir de uma nova permutação aleatória dos dados de entrada, e o vetor de pesos iniciais é preenchido com valores aleatórios. Essas estratégias são comuns na geração de *ensembles* de Perceptrons como é mostrado no trabalho do autor Kuncheva (2004).

Considerando algoritmos estáveis, como o Perceptron, simplesmente aleatorizar a ordem das amostras de entrada e inicializar o vetor de pesos com valores aleatório não é suficiente para produzir uma diversidade nos componentes de forma adequada, permitindo a geração de componentes similares. Nesse caso, ainda assim, o comitê gerado não é muito diverso.

Com o intuito de gerar um conjunto de hipóteses o mais diverso possível, uma outra estratégia é adotada. O algoritmo de *Ensemble* de Perceptrons Balanceados (*Ensemble of Balanced Perceptrons* – EBP) inicia-se definindo um valor,  $\varepsilon > 0$ , que representa a distância mínima que todos os componentes pertencentes ao comitê devem manter entre si. Em outras palavras, dado o conjunto de hipóteses  $\{h_1, h_2, \dots, h_j\}$  já aceitas no comitê, uma nova hipótese  $h_k$  é tomada como um componente válido se, e somente se

$$\min_{i \in \{1, \dots, j\}} \{\|h_i, h_k\|_2\} > \varepsilon.\tag{3.5}$$

Essa estratégia é chamada de *medida de dissimilaridade*.

No EBP, essa distância é calculada com base na distância Euclidiana entre as hipóteses aceitas no comitê. Vale ressaltar ainda que, comparar distâncias só faz sentido

caso o vetor de pesos estendido, i.e., o vetor de pesos  $w$  acrescido do valor do *bias*, esteja normalizado. Caso contrário, seria impossível definir valores de  $\varepsilon$  adequados, considerando, por exemplo, problemas de alta dimensão e o intervalo de valores possíveis para o preenchimento do vetor de pesos. O cálculo da norma, adotada aqui, é definido pela Eq. (3.6) e a normalização do vetor de pesos estendido é dada pela Eq. (3.7)

$$\|w\|_2 = \left( \sum_{i=1}^d w_i^2 \right)^{1/2}, \quad (3.6)$$

$$(w, b) = (w, b) / \|w\|_2. \quad (3.7)$$

O pseudocódigo do Algoritmo 2 descreve o treinamento do EBP.

---

**Algoritmo 2:** *Ensemble* de Perceptrons Balanceados

---

**Entrada:** conjunto de treinamento  $Z = \{(x_i, y_i)\}$  de cardinalidade  $m$ ;  
taxa de aprendizado  $\eta$ ;  
tamanho do comitê  $tam$ ;  
valor de dissimilaridade  $\varepsilon$ ;  
limite superior no número de iterações  $max$ ;  
**Saída:** componentes do comitê;

```

1 início
2   para  $k$  de 1 até  $tam$  faça
3     permutar  $Z$ ;
4      $(w_k, b_k) \leftarrow \text{PerceptronPrimal}(Z, \eta, max)$ ;
5      $\gamma^+ = \min \{y_i(\langle w_k, x_i \rangle + b_k), \forall x_i \in Z^+\}$ ;
6      $\gamma^- = \min \{y_i(\langle w_k, x_i \rangle + b_k), \forall x_i \in Z^-\}$ ;
7      $\Gamma = (\gamma^+ + \gamma^-)/2$ ;
8      $b_k = b_k - \gamma^- + \Gamma$ ;
9     normalizar  $(w_k, b_k)$ ;
10     $aceite \leftarrow \underline{\text{verdadeiro}}$ ;
11    para  $j$  de 1 até  $k - 1$  faça
12      se  $\|(w_k, b_k), (w_j, b_j)\|_2 < \varepsilon$  então
13         $k \leftarrow k - 1$ ;
14         $aceite \leftarrow \underline{\text{falso}}$ ;
15      fim se
16    fim para
17    se  $aceite$  então
18       $comite_k \leftarrow (w_k, b_k)$ ;
19    fim se
20  fim para
21 fim

```

---

### 3.4 ESPAÇO DE VERSÕES

De acordo com Herbrich (2001), ao apresentar um conjunto de amostras de treinamento  $Z$  e  $H$  sendo o conjunto de hipóteses soluções de um problema qualquer, pode-se definir  $V(Z)$  em função de  $H$  tal que

$$V_H(Z) \stackrel{def}{=} \{h \in H : i \in \{1, 2, \dots, m\} : h(x_i) = y_i\} \subseteq H, \quad (3.8)$$

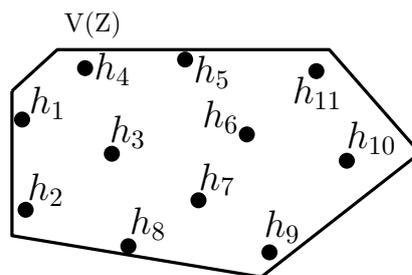
como o *espaço de versões*, i.e, o agrupamento de todos os classificadores aplicados com o conjunto de treinamento. Para classificadores lineares, de maneira particular, como o Perceptron, o espaço de versões também pode ser definido como o conjunto de vetores de pesos consistentes, dado por

$$V_W(Z) \stackrel{def}{=} \{(w, b) \in W : i \in \{1, 2, \dots, m\} : y_i(\langle w, x_i \rangle + b) > 0\} \subseteq W, \quad (3.9)$$

na qual  $W = \{(w, b) : \|(w, b)\|_2 = 1\}$  é a esfera unitária isomorfa ao espaço de hipóteses do problema em questão. Para facilitar o emprego da notação, o vetor de pesos  $(w, b)$ , equivalente a um ponto no espaço de versões, será representado simplesmente por  $w$ , partir de agora.

A Figura 6 ilustra o espaço de versões  $V(Z)$  produzido pelo comitê de Perceptrons Balanceados.

Figura 6 – Espaço de versões



Fonte: Adaptado de Enes, Villela e Neto (2016).

De acordo com Herbrich (2001), um ensemble bem distribuído é aquele que apresenta os seus componentes mais diversificado possível, facilitando assim a divisão do espaço de versões.

## 4 CLASSIFICADORES *BASELINES*

Esse capítulo tem como objetivo apresentar, de maneira sucinta, os principais algoritmos que foram confrontados pelos métodos desenvolvidos neste trabalho, que são Máquina de Vetores Suporte, Máquina do Ponto de Bayes, Máquina de Redução do Espaço de Versões e Evolutionary Analytic Center Classifier.

### 4.1 MÁQUINA DE VETORES SUPORTE

As Máquinas de Vetores Suporte (*Support Vector Machines* – SVMs) são classificadores de margem máxima que foram introduzidos por Boser, Guyon e Vapnik (1992), Vapnik (1995). Essa técnica visa separar o conjunto de treinamento por um hiperplano que maximiza a distância de integrantes de classes opostas. Para obter o hiperplano de margem máxima que classifica corretamente todos os padrões do conjunto de treinamento, é necessário resolver um problema de otimização.

Considerando a abordagem do espaço de versão, os SVMs fornecem como solução o centro da maior hipersfera inscrita em  $V(Z)$ . Normalmente, essa solução é razoável e, na maioria dos casos, mantém o erro de generalização menor. Porém, se o espaço de versão é alongado ou assimétrico, a solução fornecida por SVMs tende a apresentar uma diminuição notável do erro de generalização Tong e Koller (2001). Além disso, Herbrich, Graepel e Campbell (2001) mostram que os SVMs também podem ser vistos como uma aproximação do centro de massa do espaço da versão no cenário sem ruído.

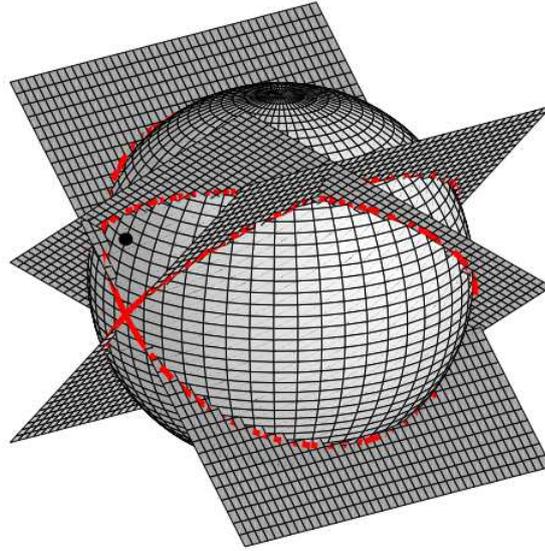
### 4.2 MÁQUINA DO PONTO DE BAYES

A Máquina de Ponto de Bayes (*Bayes Point Machine* – BPM) é um classificador que visa retornar uma aproximação para o centro de massa, seguindo a estratégia de classificação Bayesiana. Tal classificador foi originalmente apresentado em duas versões por Herbrich, Graepel e Campbell (2001), sendo o primeiro um algoritmo de amostragem estocástico e um método aproximativo baseado na média de um conjunto de Perceptrons.

A técnica proporcionada pelas BPMs funciona por meio da geração de um número fixo de Perceptrons, considerando para cada um  $\|w\|_2 = 1$ , treinados com diferentes permutações do conjunto de treinamento, e a média deles em um único classificador. A hipótese induzida por tal conjunto de classificadores se aproxima muito da estratégia de classificação Bayesiana (HERBRICH; GRAEPEL; CAMPBELL, 2001). As BPMs são considerados um ótimo método para classificadores ótimos de Bayes, uma vez que fornecem uma melhor aproximação do centro de massa, superando outras estratégias de aproximação como o SVM.

A Figura 7 representa um exemplo do ponto de Bayes em um espaço tridimensional.

Figura 7 – Ilustração do ponto de Bayes estimado no espaço tridimensional



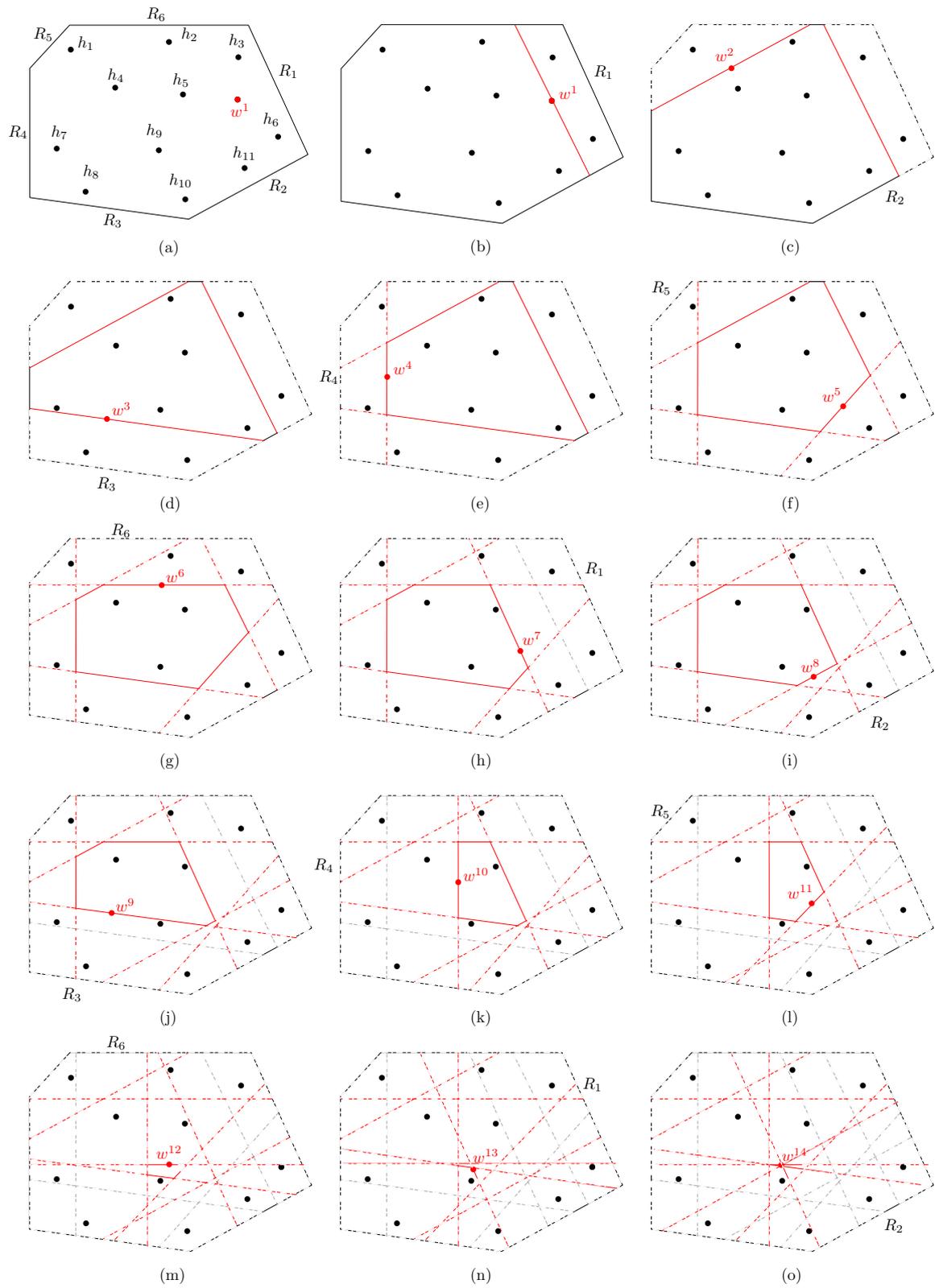
Fonte: Herbrich, Graepel e Campbell (2001)

#### 4.3 MÁQUINA DE REDUÇÃO DO ESPAÇO DE VERSÕES

No trabalho dos autores Enes, Villela e Neto (2016), foi proposto o método chamado Máquina de Redução do Espaço de Versões (*Version Space Reduction Machine – VSRM*), a fim de fornecer uma aproximação efetiva do centro de massa do espaço de versão, ou seja, o trabalho busca desenvolver um classificador ótimo de Bayes. O processo de redução aplicado no VSRM é baseado na introdução de planos de corte sucessivos, esse planos, quando adicionados, fazem o espaço de versão ser dividido em dois meios-espacos. Para uma correta redução desse espaço recém gerado, é necessário descartar a metade do espaço com o menor volume, após um número finito de iterações e a constante redução do espaço de versões, esse método converge para uma hipótese aproximada como solução, ou o centro de Bayes.

A Figura 8 mostra a convergência do VSRM, caracterizando o passo a passo do método em um problema hipotético com 6 hipóteses soluções em um espaço de versões. Por exemplo, considere o esquema de votação apresentado na Figura 8-(b). Obtemos, pelo processo de votação, 9 votos para a margem superior contra 2 para a margem inferior. Nesse caso, pode-se descartar o meio-espaço com menos membros. Por outro lado, se for considerado Figura 8-(c), sera descartado o meio-espaço com 10 membros da margem superior contra apenas 1 da margem inferior.

Figura 8 – Ilustração da estratégia empregada pelo VRSM



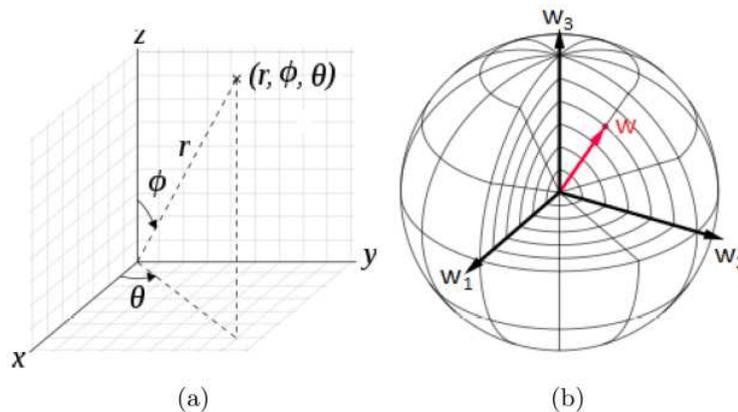
Fonte: Enes, Villela e Neto (2016)

#### 4.4 CLASSIFICADOR EVOLUCIONÁRIO DE CENTRO ANALÍTICO

Um algoritmo evolutivo para resolver o problema do centro analítico aplicado a tarefas de classificação binária é apresentado pelos autores Goulart et al. (2020) chamado Classificador Evolucionário de Centro Analítico (*Evolutionary Analytic Center Classifier – EACC*). O objetivo desse trabalho é criar um algoritmo para o desenvolvimento de uma população viável de hiperplanos para aproximar o centro analítico de um espaço de versões criado por perceptrons. Sendo assim o operador de recombinação tem como propósito mover a população em direção à solução ótima, ao mesmo tempo que preserva a viabilidade de cada indivíduo. O operador de mutação pode ser interpretado como um movimento de rotação de um hiperplano.

No trabalho foram usadas coordenadas hiperesféricas e as mesmas provaram serem essenciais para que ambos os operadores funcionassem corretamente. Os experimentos realizados mostraram que o EACC obteve bons resultados competindo em precisão e na maioria das vezes superando os classificadores como SVM e VSRM em conjuntos de dados linearmente separáveis, os mesmos usados nessa dissertação. Nesse sentido, o método proposto apresenta possibilidades reais de aplicação na solução de problemas práticos.

Figura 9 – (a) Coordenadas esféricas de um sistema espacial em 3-dimensões. (b) Representação de um vetor unitário tridimensional em uma esfera unitária



Fonte: Goulart et al. (2020)

## 5 PERCEPTRON DE TUKEY

A localização exata da mediana de Tukey é uma tarefa complexa sendo categorizada como um problema NP-difícil (JOHNSON; PREPARATA, 1978) e também com uma difícil aproximação (AMALDI; KANN, 1995). Para a dimensão  $d = 1$ , o problema da mediana de Tukey obviamente corresponde à mediana padrão e pode, portanto, ser resolvido em tempo  $O(n)$  (CORMEN et al., 2001); nesse caso a profundidade máxima aqui é exatamente  $\left\lceil \frac{n}{2} \right\rceil$ , já para dimensões maiores que  $d$ , a máxima profundidade de Tukey está entre  $\left\lceil \frac{n}{d+1} \right\rceil$  e  $\left\lceil \frac{n}{2} \right\rceil$  (CHAN, 2004).

Rousseeuw e Ruts (1998) propuseram dois algoritmos importantes para calcular a mediana de Tukey. O primeiro é relativamente simples, mas bastante lento e com complexidade igual a  $O(n^5 \log n)$ , enquanto o segundo (chamado HALFMED) é muito mais rápido, cuja complexidade é  $O(n^2 \log^2 n)$ , onde  $n$  é a quantidade de pontos analisados. Um pequeno estudo de simulação é realizado no trabalho e alguns exemplos, em um espaço bi-dimensional dos dados, evidencia que a medida que novas amostras são analisadas, a identificação da localização da mediana de Tukey torna-se possível. A evolução desse trabalho veio logo em seguida com o desenvolvimento do algoritmo chamado DEEPLOC (STRUYF; ROUSSEEUW, 2000), sendo esse último mais veloz que o seu antecessor e com complexidade  $O(n \log^2 n)$ .

Seguindo na linha de desenvolver algoritmos que possam encontrar uma solução heurística, esta dissertação apresenta o conceito de Perceptron de Tukey (*Tukey's Perceptron* – TP), principal contribuição desse trabalho, e a apresentação e descrição dos algoritmos desenvolvidos para fazer a localização aproximada da mediana de Tukey, para determinação da solução exata é uma tarefa com alta complexidade, por isso é um trabalho considerado difícil.

Em um espaço de versões já definido é possível encontrar o representante que possui uma maior profundidade, aproximando-se assim do centro de massa (GILAD-BACHRACH; NAVOT; TISHBY, 2004). A identificação desse melhor candidato é através do paradigma da profundidade de Tukey, onde é encontrado a hipótese solução mais profunda no espaço de versões. Com base nesse paradigma esse trabalho apresenta sua primeira contribuição, cunha-se então o Perceptron de Tukey.

Para obter o Perceptron de Tukey é preciso, primeiramente, encontrar os conjuntos de hipóteses soluções considerados majoritários do espaço. A Figura 10 representa como classificar as hipóteses soluções  $h_i$  e  $h_j$  como pertencentes ao grupo classificados como ganhadores ou perdedores. Note que nessa mesma figura é mostrado o ponto  $h_0$ , esse ponto é o cálculo do ponto médio das hipóteses soluções pertencentes a figura, e é obtido através do cálculo representado pela Equação (5.1). A reta tracejada é o hiperplano separador

que é traçado acima do ponto médio das hipóteses soluções.

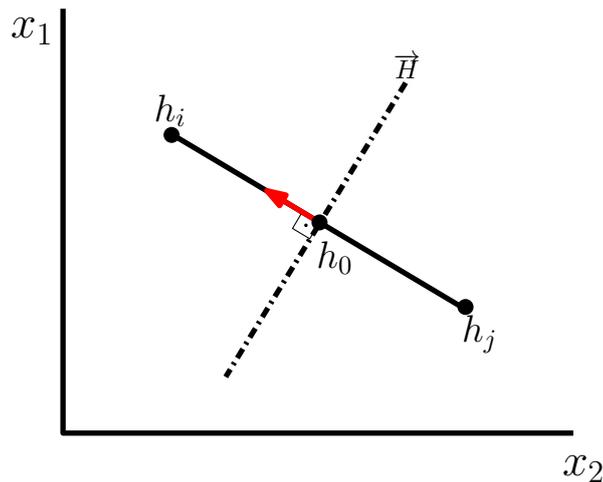
$$h_0 = \frac{h_i + h_j}{2}. \quad (5.1)$$

Logo após a identificação dos ponto médio e traçado o hiperplano separador, é possível inferir a Equação (5.2) que representa a equação geral do hiperplano separador.

$$\vec{H} = (h_0 - \frac{h_i + h_j}{2} \times (h_i - h_j)). \quad (5.2)$$

Ao aplicar a Equação (5.2) na hipótese solução  $h_i$ , é obtido que a mesma está no sentido positivo do vetor, como mostrado é mostrado na Figura 10 através da seta da cor vermelha. De maneira análoga ocorre com a hipótese solução  $h_j$ , mas a mesma é encontra-se no sentido negativo do vetor.

Figura 10 – Hiperplano separador



Fonte: Elaborada pelo autor (2022).

Conclui então que a hipótese  $h_i$  encontra-se no lado que é considerado positivo do hiperplano, e a hipótese  $h_j$  no lado negativo.

O Algoritmo 3 descreve o pseudocódigo responsável por fazer a divisão dos lados majoritário e minoritário do espaço de versões através de duas hipóteses selecionadas no comitê apresentado.

A Figura 11 ilustra como são os primeiros passos para localizar o Perceptron de Tukey. A Fig. 11-(a) a representa o espaço de versões com as hipóteses soluções de  $h_1$  até  $h_{11}$ , já na Fig. 11-(b) as hipóteses soluções  $h_1$  e  $h_2$  são as primeiras selecionadas para realizar a divisão do hiperplano separador em *ensembles* majoritários e minoritários com a reta vermelha representando o hiperplano que delimita o limite dessa divisão.

---

**Algoritmo 3:** Computa Sinal
 

---

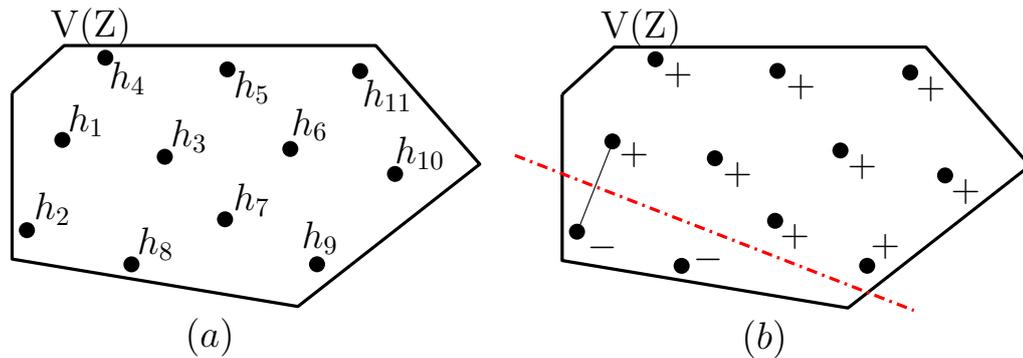
**Entrada:** hipóteses do espaço de versões  $h_1, h_2$  e  $h_3$ ;  
**Saída:** subespaço em que se encontra a hipótese  $h_3$ ;

```

1 início
2   resultado  $\leftarrow$  0;
3   para  $j$  de 1 até  $d$  faça
4     | resultado  $\leftarrow$  resultado +  $(h_{1j} - h_{2j}) * [h_{3j} - (h_{1j} + h_{2j})/2]$ ;
5   fim para
6   se resultado  $\geq$  0 então
7     | retorna +1;
8   senão
9     | retorna -1;
10  fim se
11 fim
  
```

---

Figura 11 – Divisão do espaço de versões.



Fonte: Elaborada pelo autor (2022).

## 5.1 BUSCA EM PROFUNDIDADE

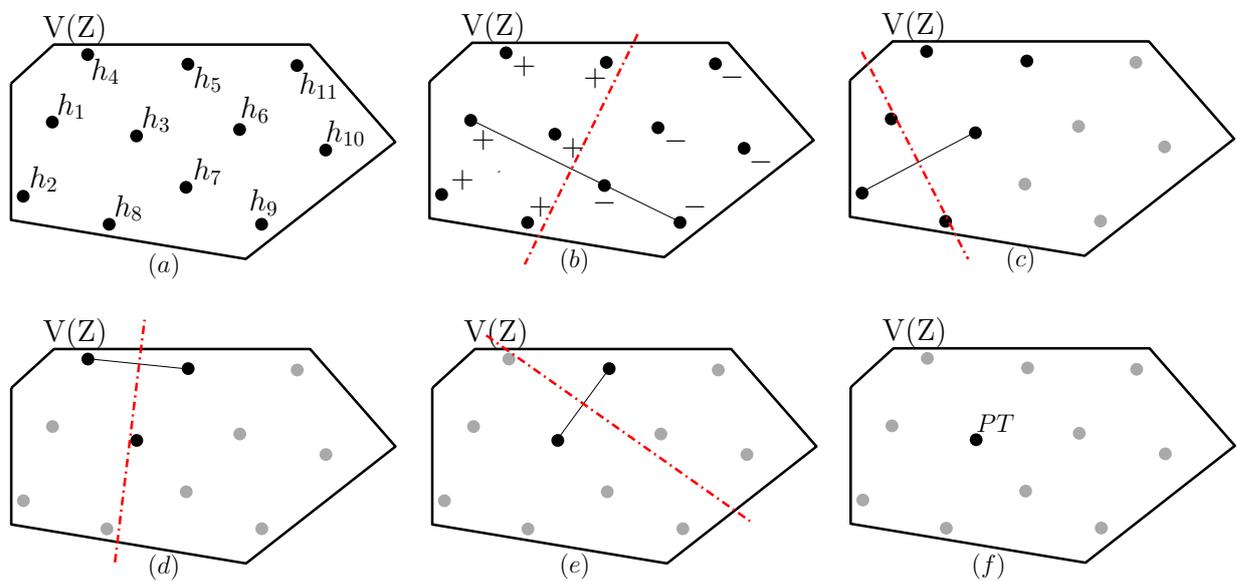
Os algoritmos de busca possuem uma variedade de estratégias que são executadas em ações com o objetivo de buscar solucionar um problema proposto, como por exemplo o percurso ou a busca por um item específico. As sequências dessas ações possíveis tem início em um nó chamado raiz, e a partir deste estado inicial é formado uma árvore de busca. Os algoritmos mais conhecidos são os de busca em largura e busca em profundidade.

O algoritmo de busca em profundidade (*Depth-First Search* – DFS), a partir do nó raiz da árvore, é capaz de explorar o nó recém descoberto que ainda possui nós inexplorados, uma vez que todos os caminhos foram analisados a busca retorna para o nó anterior afim de buscar outros caminhos que saem do respectivo nó. Se ainda houver um nó não explorado, a busca em profundidade o seleciona e repete todo o processo até descobrir todos os nós (BHARGAVA, 2016).

Um das propostas para a localizar o Perceptron de Tukey é o uso do algoritmo de busca em profundidade, com base na estratégia que esse algoritmo usa para percorrer

os seus nós foi desenvolvido o algoritmo Perceptron de Tukey por Busca em Profundidade (*Tukey's Perceptron by Depth-First Search* – TP-DFS). Essa operação inicia-se baseado nas hipóteses soluções encontradas no espaço de versões com suas respectivas profundidades iguais e zeradas, como é demonstrado na Figura 12-(a). Logo em seguida, é realizado a escolha de duas hipóteses soluções iniciais, de maneira aleatória, para fazerem o cálculo da divisão do hiperplano que resulta a segmentação entre conjuntos majoritários e minoritários. Os membros do comitê do conjunto classificado como ganhador, ou seja, os membros do conjunto majoritário são incrementado em uma unidade na sua profundidade, como representando na Figura 12-(b).

Figura 12 – Perceptron de Tukey por Busca em Profundidade.



Fonte: Elaborada pelo autor (2022).

Para a próxima iteração do algoritmo Perceptron de Tukey por busca em profundidade as hipóteses definidas como conjunto de membros classificados anteriormente como majoritário, irão fazer a nova segmentação dos membros do comitê, para isso é escolhido novamente de maneira aleatória outras duas hipóteses soluções realizando a nova divisão do hiperplano separador, surgindo outro conjunto de hipóteses soluções ganhadoras e um novo incremento de profundidade no grupo considerado ganhador é realizado. Essas etapas estão representadas na Figura 12-(c), (d) e (e). Por fim, esse procedimento é repetido até o Perceptron de Tukey ser encontrado, como mostra Figura 12-(f) e todo esse processo é descrito no pseudocódigo do Algoritmo 4.

Essa proposta, de um algoritmo de busca baseado em profundidade, é mais viável em custo computacional, pois menos etapas são executadas para determinar o Perceptron de Tukey. Uma desvantagem desse algoritmo é que a hipótese encontrada como Perceptron de Tukey não necessariamente resulta o melhor membro das hipóteses soluções, por

---

**Algoritmo 4:** Perceptron de Tukey por Busca em Profundidade
 

---

**Entrada:** comitê de hipóteses;  
 tamanho do comitê  $tam$ ;  
**Saída:** hipótese com a profundidade máxima;

```

1 início
2   para  $k$  de 1 até  $tam$  faça
3     |  $profundidade_k \leftarrow 0$ ;
4   fim para
5    $n \leftarrow tam$ ;
6   enquanto  $n > 1$  faça
7     |  $i \leftarrow$  sorteio de uma das hipóteses com a atual profundidade máxima;
8     |  $j \leftarrow$  sorteio de uma das hipóteses com a atual profundidade máxima;
9     |  $qtdePos \leftarrow 0$ ;
10    |  $qtdeNeg \leftarrow 0$ ;
11    | para  $k$  de 1 até  $tam$  faça
12      |  $controle_k \leftarrow$  ComputaSinal( $h_i, h_j, h_k$ );
13      | se  $controle_k = 1$  então
14        |  $qtdePos++$ ;
15      | senão
16        |  $qtdeNeg++$ ;
17      | fim se
18    | fim para
19    | para  $k$  de 1 até  $tam$  faça
20      | se  $qtdePos > qtdeNeg$  e  $controle_k = 1$  então
21        |  $profundidade_k++$ ;
22      | senão se  $qtdeNeg > qtdePos$  e  $controle_k = -1$  então
23        |  $profundidade_k++$ ;
24      | fim se
25    | fim para
26    |  $n \leftarrow$  número de hipóteses com a atual profundidade máxima;
27  fim enqto
28 fim
  
```

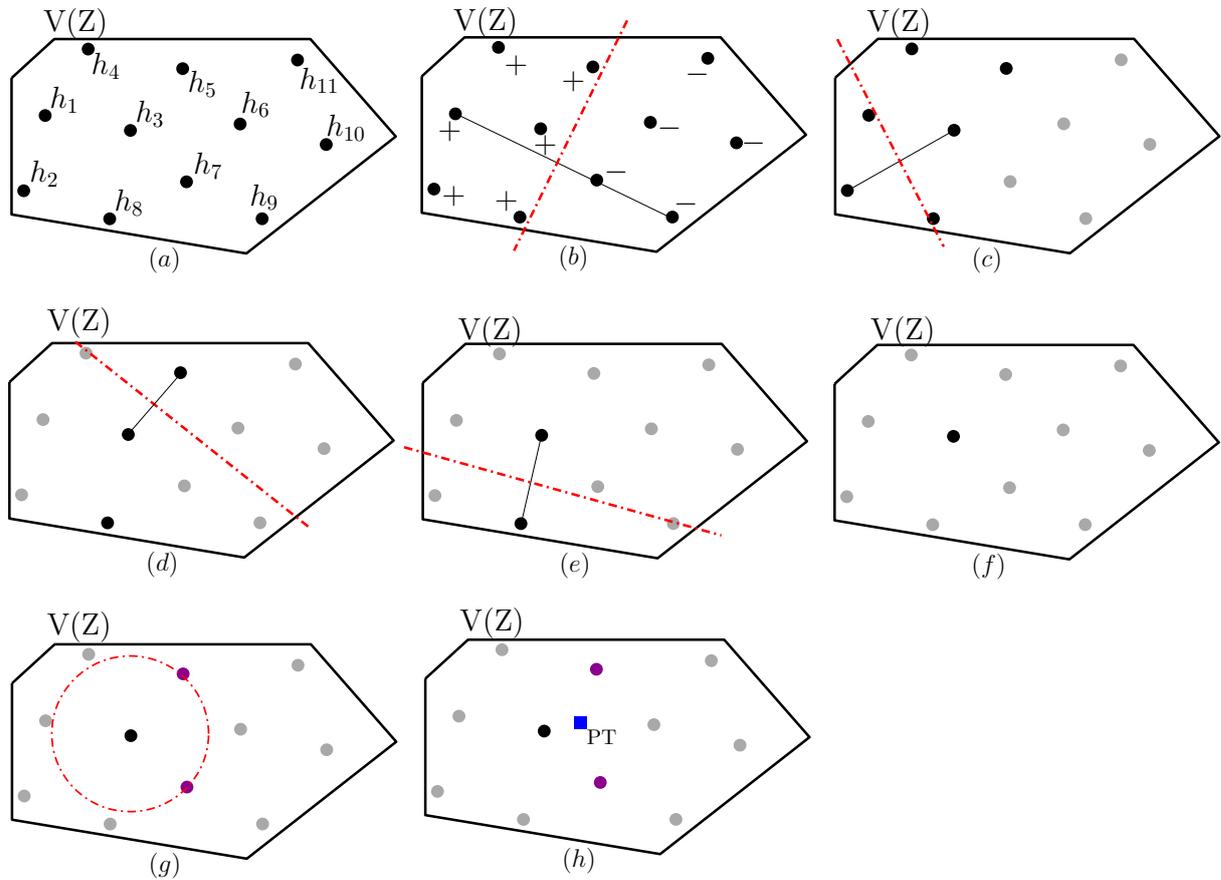
---

exemplo, se escolher aleatoriamente duas hipóteses para fazer a segmentação do plano separador, pode resultar uma ocorrer uma má seleção.

### 5.1.1 Subcomitê por vizinhança

Outra proposta apresentada neste trabalho é o algoritmo de Perceptron de Tukey por Busca em Profundidade com Subcomitê por Vizinhança (*Tukey's Perceptron by Depth-First Search with Subensemble by Neighborhood – TP-DFS-NB*), uma proposta de melhoria do algoritmo de Perceptron de Tukey por Busca em Profundidade. Sua inspiração tem como início o TP-DFS, ou seja, é escolhido inicialmente duas hipóteses soluções aleatórias para o cálculo do hiperplano separador que resultará nos conjuntos majoritários e minoritários, como é representado na Figura 13-(a), (b), (c), (d) e (e).

Figura 13 – Perceptron de Tukey por Busca em Profundidade com Subcomitê por Vizinhança.



Fonte: Elaborada pelo autor (2022).

Após identificar a hipótese solução com maior profundidade, como é mostrado na Figura 13-(f), procura-se uma vizinhança das hipóteses soluções mais próximas da hipótese solução mais profunda, ou seja, é analisado  $n$  hipóteses soluções próxima da mais profunda, como é mostrado na Figura 13-(g). Essa imagem trás uma representação didática de um círculo com a hipótese solução centrada, e então ocorre uma expansão do raio ao encontro das hipóteses soluções que possuem grau de similaridade com o Perceptron de Tukey. Ao localizar as hipótese soluções mais próximas da hipótese solução mais profunda é calculado a média desses pontos, gerando assim uma hipótese solução em potencial, não apresentada antes por causa da quantidade de hipóteses criadas, mas tornando o novo Perceptron de Tukey.

O algoritmo TP-DFS-NB possui como melhoria sobre o TP-DFS a proposta de expor um novo ponto que pode ser membro do comitê ou não, portanto, depois de identificado as melhores hipóteses soluções é calculado o ponto médio entre os mesmos. Esse novo ponto é chamado de Ponto Médio -  $PM$ , como é representado na Figura 13-(h). O  $PM$  representa uma melhoria na identificação da melhor hipótese solução, pois é um

ponto mais próximo ao centro de massa da região.

Apesar de possuir um passo a mais que o TP-DFS essa proposta busca um representante mais fidedigno para melhor hipótese solução.

## 5.2 BUSCA EM LARGURA

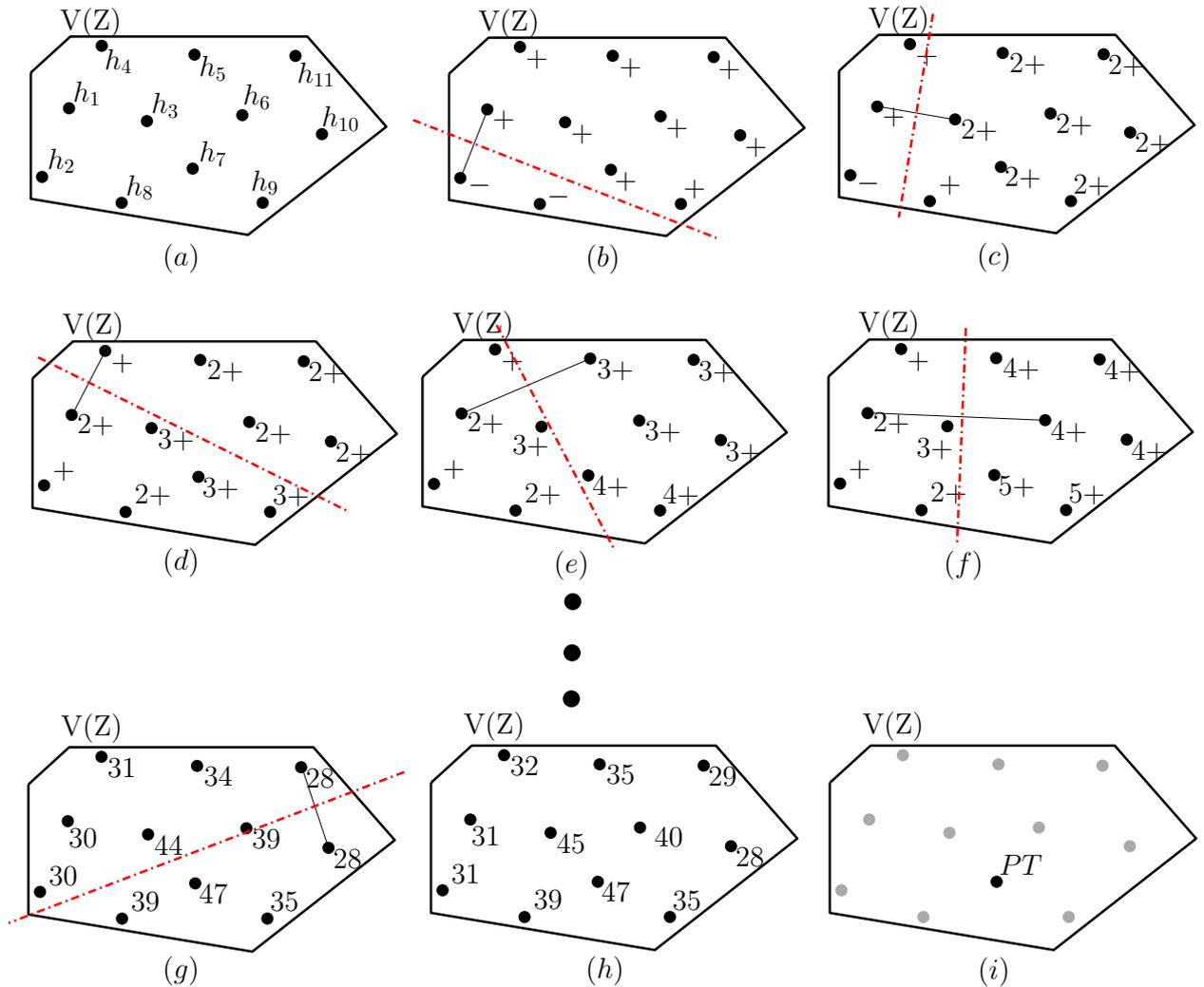
A busca sem informação, também conhecida como busca cega, como exemplo a busca em largura (*Breadth-First Search* – BFS), não utiliza qualquer informação adicional acerca dos estados dos nós, além daquelas a priori. Esse tipo de algoritmo é uma estratégia simples em que o nó da raiz da árvore é expandido primeiro e logo em seguida todos os sucessores da raiz também são expandidos, em seguida os sucessores desses primeiros nós e assim por diante. Ou seja, todos os nós de cada profundidade da árvore de busca são expandidos, para então explorar todos os nós no nível posterior (RUSSELL; NORVIG, 2003).

O algoritmo Perceptron de Tukey por Busca em Largura (*Tukey's Perceptron by Breadth-First Search* – TP-BFS) tem como propósito percorrer todos os possíveis pares de hipóteses soluções até identificar o Perceptron de Tukey. A Figura 14 demonstra os primeiros passos desse algoritmo, não foram representados todos os passos na Figura 14, pois a mesma ficaria muito grande tornando inviável a sua representação nesse trabalho. No início todas as hipóteses soluções apresentadas no espaço de versões possuem profundidades iguais e zeradas, como pode-se ver na Figura 14-(a). Logo após o algoritmo seleciona a primeira hipótese solução e a hipótese subsequente, como é mostrado na letra (b) da Figura 14 é selecionado a hipótese  $h_1$  e  $h_2$ , para então realizar o cálculo da segmentação dos hiperplanos, representado pela reta vermelha, gerando assim conjuntos majoritários e minoritários. O conjunto de hipóteses soluções majoritários é acrescido em uma unidade sua profundidade, já o conjunto minoritário é mantido a profundidade atual. Esse percurso de seleção de hipóteses soluções é feita de maneira ordenada, até todas as hipóteses sejam percorridas, como pode-se perceber no decorrer da Figura 14-(b), (c), (d), (e) e (f).

Após a escolha de todas as hipóteses soluções, é possível identificar o Perceptron de Tukey, cuja sua profundidade é a maior de todas, tornando-se a melhor hipótese solução para esse espaço de versões apresentado. Todo esse processo é descrito no pseudocódigo do Algoritmo 5.

Por ser uma abordagem que faz todas as combinações possíveis para o cálculo da profundidade, é considerada muito custosa computacionalmente, tornando-se inviável em grandes comitês, como é mostrado nas respectivas Tabelas 6 e 7 que usam esse algoritmo para encontrar o Perceptron de Tukey, mas tal abordagem pode trazer uma hipótese solução mais fidedigna para o espaço de versões.

Figura 14 – Perceptron de Tukey por Busca em Largura



Fonte: Elaborada pelo autor (2022).

### 5.2.1 Subcomitê por votação

A partir do algoritmo Perceptron de Tukey em Busca por Largura é proposto uma segunda versão com o objetivo de melhorar a identificação do Perceptron de Tukey chamado algoritmo Perceptron de Tukey por Busca em Largura com Subcomitê por Votação (*Tukey's Perceptron by Breadth-First Search with Subensemble by Voting* – TP-BFS-SV). O mesmos passos do TP-BFS são executados nessa nova proposta, como é representado na Figura 15-(a-f).

Após observar a hipótese solução mais profunda no espaço de versões apresentado, é possível identificar também  $n$  subvotações. As subvotações são as hipóteses soluções que obtiveram profundidades menores que a hipótese solução mais profunda, mas as mais próximas da maior profundidade encontrada. A Figura 15-(h) representa a hipótese solução mais profunda e 2 subvotações no espaço de versões proposto. Portanto, após localizar a

---

**Algoritmo 5:** Perceptron de Tukey por Busca em Largura
 

---

**Entrada:** comitê de hipóteses;  
 tamanho do comitê  $tam$ ;  
**Saída:** hipótese com a profundidade máxima;

```

1 início
2   para  $k$  de 1 até  $tam$  faça
3     |  $profundidade_k \leftarrow 0$ ;
4   fim para
5   para  $i$  de 1 até  $tam$  faça
6     | para  $j$  de 1 até  $tam$  faça
7       | |  $qtdePos \leftarrow 0$ ;
8       | |  $qtdeNeg \leftarrow 0$ ;
9       | | para  $k$  de 1 até  $tam$  faça
10      | | |  $controle_k \leftarrow \text{ComputaSinal}(h_i, h_j, h_k)$ ;
11      | | | se  $controle_k = 1$  então
12      | | | |  $qtdePos++$ ;
13      | | | senão
14      | | | |  $qtdeNeg++$ ;
15      | | | fim se
16      | | fim para
17      | | para  $k$  de 1 até  $tam$  faça
18      | | | se  $qtdePos > qtdeNeg$  e  $controle_k = 1$  então
19      | | | |  $profundidade_k++$ ;
20      | | | senão se  $qtdeNeg > qtdePos$  e  $controle_k = -1$  então
21      | | | |  $profundidade_k++$ ;
22      | | | fim se
23      | | fim para
24      | fim para
25   fim para
26 fim
  
```

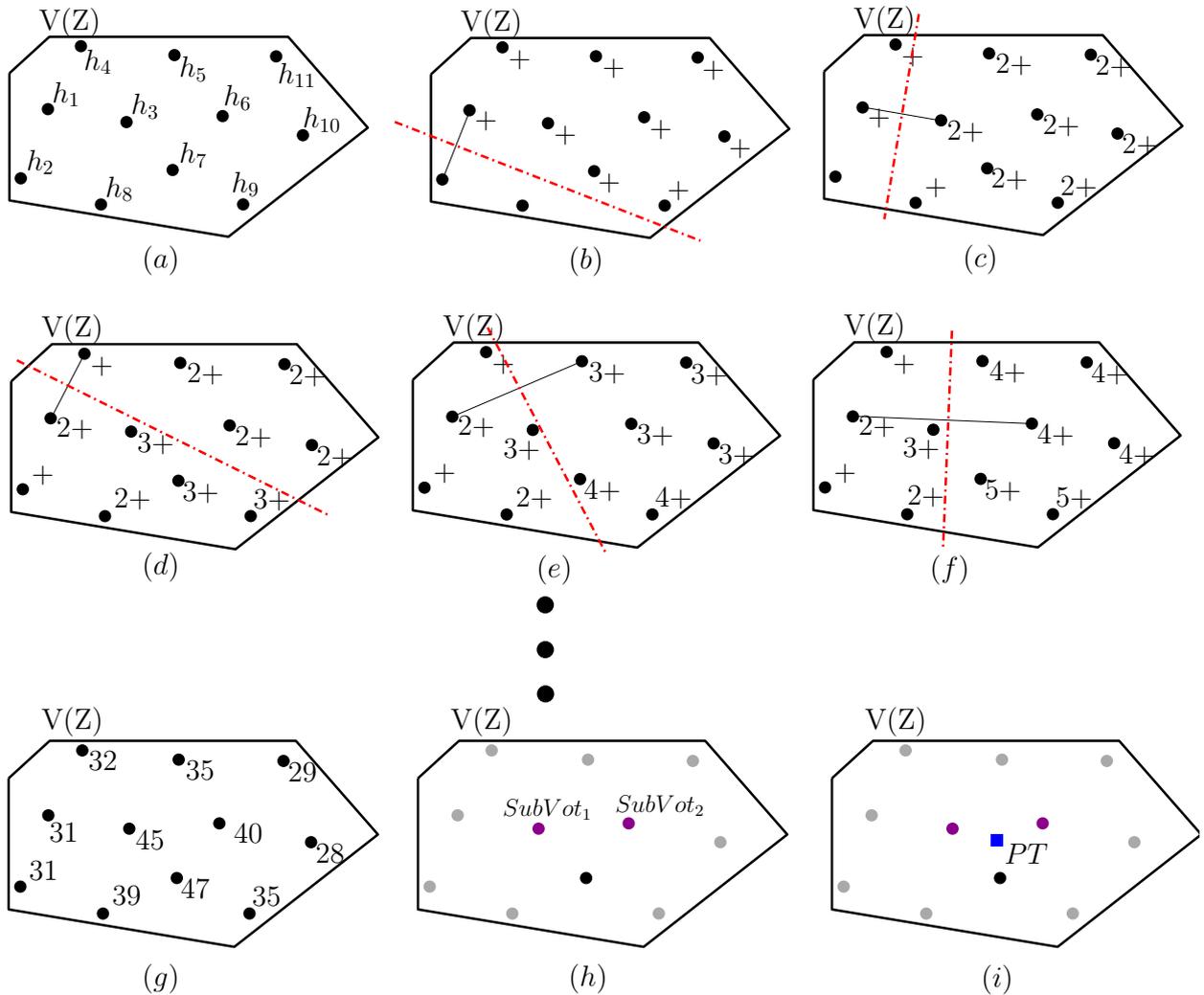
---

hipótese solução mais profunda e as respectivas  $n$  subvotações, é calculado o ponto médio entre essas hipóteses, localizando então o Perceptron de Tukey. Como mencionado no algoritmo TP-BFS, essa hipótese mais profunda é um ponto em potencial, não gerado anteriormente por causa da quantidade de pontos que foram representados na criação do espaço de versões.

### 5.2.2 Subcomitê por vizinhança

De forma similar ao algoritmo Perceptron de Tukey por Busca em Largura (TP-BFS) uma nova abordagem é proposta, o algoritmo em questão é chamado Perceptron de Tukey por Busca em Largura com Subcomitê por Vizinhança (*Tukey's Perceptron by Breadth-First Search with Subensemble by Neighborhood* – TP-BFS-NB). Essa conduta executa os mesmos passos do TP-BFS, como é representado na Figura 16-(a-h).

Figura 15 – Perceptron de Tukey por Busca em Largura com Subcomitê por Votação

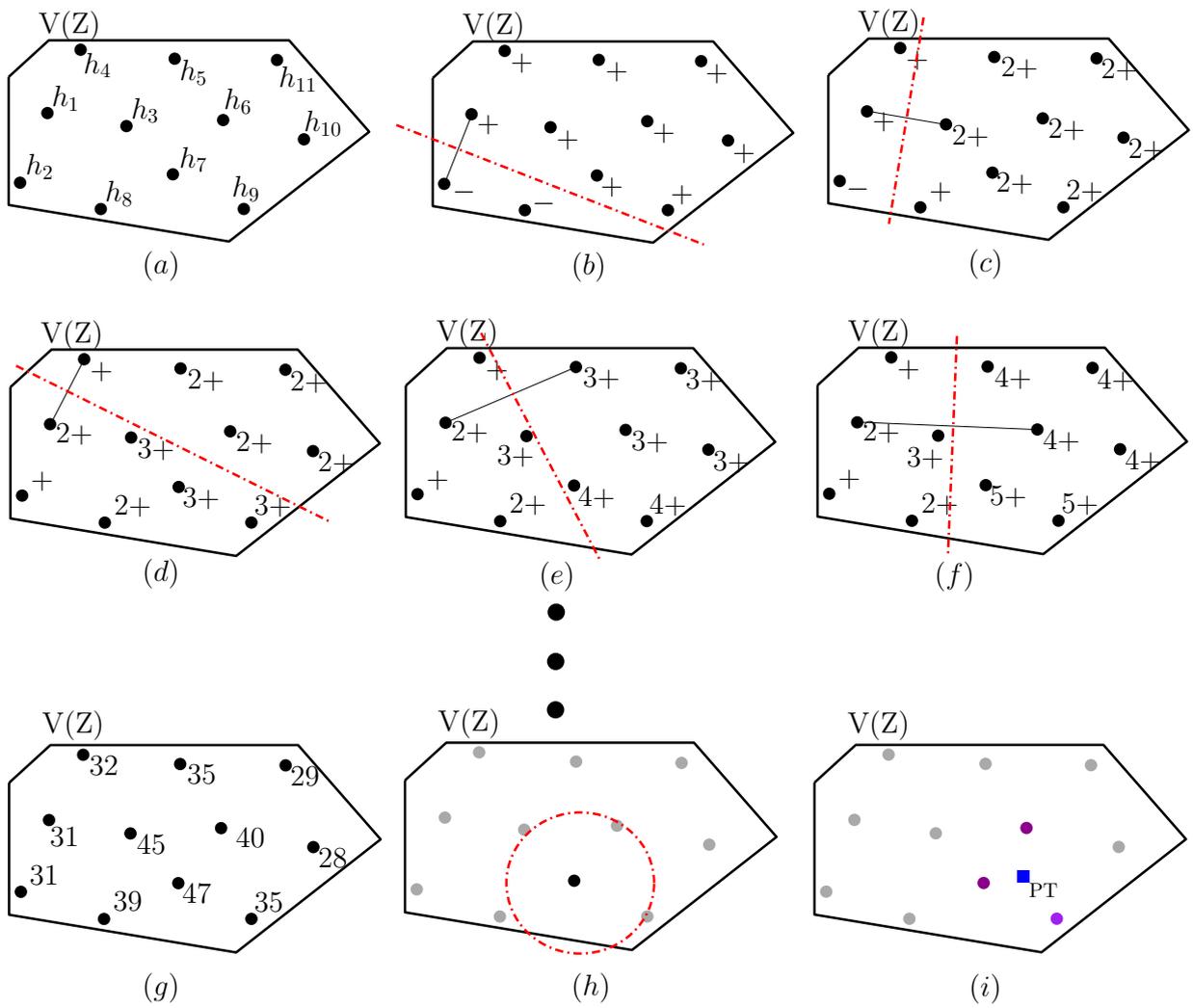


Fonte: Elaborada pelo autor (2022).

Após localizar a hipótese mais profunda traça-se um raio, centrada na mesma, buscando encontrar os vizinhos mais próximas, como é mostrado na Figura 16 letra (i), para então calcular o ponto médio entre essas hipóteses localizando então o Perceptron de Tukey. Assim como os outros algoritmos, TP-DFS-NB e TP-BFS-SV, que baseiam-se em hipóteses soluções com profundidade menores do que a hipótese solução mais profunda, o algoritmo de TP-BFS-NB gera um hipótese em potencial, não gerada anteriormente por causa da quantidade de pontos que foram representados na criação do espaço de versões.

As abordagens dos algoritmos baseados em busca por largura possuem a vantagem de gerar soluções baseadas em comitês, tornando esse métodos mais reais em relação a determinar uma hipótese solução mais profunda.

Figura 16 – Perceptron de Tukey por Busca em Largura com Subcomitê por Vizinhança



Fonte: Elaborada pelo autor (2022).

## 6 ANÁLISE EXPERIMENTAL E RESULTADOS

### 6.1 CONJUNTOS DE DADOS

Ao realizar a análise dos resultados, o trabalho desenvolvido utilizou 6 bases de dados, todas linearmente separáveis e consideradas *microarrays*, as bases foram utilizadas para a avaliação dos algoritmos desenvolvidos nesse trabalho e apresentados no capítulo anterior. Os repositórios podem ser encontrados nos trabalhos Glaab et al. (2012), Zhu, Ong e Dash (2007) e Golub et al. (1999). A Tabela 2 apresenta de maneira sumarizadas as principais informações das bases.

Tabela 2 – Informações sobre as bases de dados consideradas.

Base	Atributos	Amostras		
		+1	-1	Total
Breast	12625	10	14	24
Colon	2000	22	40	62
DLBCL	5468	58	19	77
Leukemia	7129	47	25	72
Prostate	12600	50	52	102
CNS	7129	21	39	60

Fonte: Elaborada pelo autor (2022).

Todas as bases não sofreram qualquer tipo de pré-processamento, ou foi realizado algum tipo de redução ou normalização nos dados e não possuem dados faltantes, portanto estão como obtidas a partir das fontes anteriormente citadas. As bases selecionadas também são compostas por valores reais e discretos.

### 6.2 DEFINIÇÃO DE PARÂMETROS

#### 6.2.1 *K-fold cross-validation*

Ao realizar algum tipo de predição em um problema inteligência computacional, pode-se usar a técnica de validação cruzada *cross-validation*, que tem como objetivo avaliar a capacidade de generalização de um modelo dado um conjunto de dados a ser testado (KOHAVI et al., 1995). O *cross-validation* foi proposto por Geisser (1975), Wahba e Wold (1975) e revisada por Tukey e Cleveland (1988).

O método funciona a partir da geração de uma permutação aleatória do conjunto de treinamento  $Z$ , de cardinalidade  $m$ , e posterior subdivisão em  $k$  subconjuntos, portanto *k-fold cross-validation*. Assim, para todo  $i \in \{1, 2, \dots, k\}$ ,  $Z_i$  tem tamanho  $m/k$ . Para a geração dos  $k$  modelos segue a seguinte forma: para cada  $i$ , o  $i$ -ésimo modelo é gerado tomando  $Z - Z_i$ , como conjunto de treinamento. O modelo obtido é então testado no conjunto  $Z_i$ .

O *cross-validation* é então estabelecido. Para o último passo, calcula-se a média dos erros obtidos para cada um dos conjuntos de teste  $Z_i$ ,  $i \in \{1, 2, \dots, k\}$ . Portanto, não é necessário dividir previamente o conjunto de dados de entrada em amostras de treino e teste e todos os dados disponíveis são usados nas suas respectivas fases.

Um desdobramento da técnica original é o método de *k-fold* estratificado. Esse método emprega uma estratificação do conjunto de dados de entrada visando obter modelos mais justos e com melhores resultados (KOHAVI et al., 1995). Estratificar os dados implica em dividi-los de forma que em cada conjunto seja mantida a proporcionalidade entre as classes do problema. Essa divisão é feita verificando a porcentagem de dados disponíveis para cada classe no conjunto de amostras. O processo de treinamento e teste dos dados permanece o mesmo, o que muda é a forma de se dividir o conjunto de dados em  $k$  partes de tamanho  $m/k$  Kohavi et al. (1995), Efron e Tibshirani (1997).

O método *k-fold* é uma técnica amplamente difundida no meio acadêmico e de pesquisa, mas não possui uma definição sobre a quantidade de  $k$ . É um consenso entre os trabalhos utilizar a utilização de  $k = 10$  (GEISSER, 1993; KOHAVI et al., 1995). Considerando a natureza aleatória dos modelos abordados, em todos os experimentos foi empregada uma estratégia de *cross-validation* em 10x10-10-fold, i.e, 10 execuções independentes de 10 execuções de um 10-fold. Em relação à estratégia estratificada de *cross-validation*, em cada conjunto sempre foi mantida a porcentagem de dados referentes a cada classe (KOHAVI et al., 1995). Visando comparações mais precisas, para cada base de dados, sempre foram selecionados os mesmos 10 subconjuntos do *cross-validation*, preservando a geração da semente associada ao processo aleatório.

### 6.2.2 Tamanho do *ensemble*

Com o objetivo de analisar a variação do tamanho do comitê na capacidade de generalização do modelo obtido, foram realizados testes nas bases de dados já apresentadas anteriormente. Para isso usou-se a estratégia de *cross-validation* com a configuração 10x10-10-fold considerando as duas formas de combinação de saída empregadas nesse estudo e aplicada ao método dos algoritmos propostos.

Para uma visualização mais fácil optou-se por condensar os resultados apresentando apenas três valores principais: 11, 101 e 1001 do tamanho dos comitê quando possível, ou apenas dois valores principais: 11 e 101. Esses números foram propostos como ímpares, pois é possível obter-se lados majoritários e minoritários nos comitê criados.

### 6.2.3 Medida de dissimilaridade

O parâmetro de dissimilaridade não foi fixado no modelo criado nesse trabalho, ou seja, para cada tipo de problema usa-se uma medida de dissimilaridade diferente, como é mostrado na Tabela 3. Esse parâmetro é responsável por imprimir diversidade na

construção do comitê. A escolha de um valor adequado depende tanto da complexidade do problema quanto do tamanho pretendido de comitê. Para a construção de comitês com muitos componentes, valores pequenos devem ser estabelecidos para garantir admissão dos componentes requisitados para construção do comitê, independentemente da medida de distância empregada por cada modelo.

No decorrer do estudo experimental, a medida de dissimilaridade foi definida com base em testes empíricos para cada algoritmo analisado. O parâmetro foi estabelecido conforme o maior valor (aproximado) possível capaz de construir o comitê com a quantidade de componentes desejado.

A Tabela 3 apresenta os valores do parâmetro da medida de dissimilaridade empregados para cada base de dados, para todas as bases testadas.

Tabela 3 – Valores de dissimilaridade aplicados nas bases para o EBPd.

Base	Tamanho do comitê		
	11	101	1001
Prostate	10,0	1,00	0,55
Breast	70,0	8,00	1,30
Colon	11,0	1,50	0,70
Leukemia	97,0	17,0	2,50
DLBCL	150,0	15,0	3,00
CNS	125,0	10,0	0,70

Fonte: Elaborada pelo autor (2022).

A partir da Tabela 3 é possível observar que, de fato, à medida que o número de componentes no comitê aumenta, o valor da medida de dissimilaridade diminui. Esse fato é esperado, uma vez que, a diminuição do valor do parâmetro de dissimilaridade reflete na relaxação da restrição, possibilitando a aceitação de mais componentes.

### 6.3 RESULTADOS NUMÉRICOS

Para facilitar a compreensão, os resultados numéricos serão discutidos separadamente em relação aos métodos propostos. Inicialmente, discute-se os resultados obtidos para os algoritmos baseados em busca em profundidade para as 6 bases de dados linearmente separáveis analisadas, optou-se por discutir os resultados obtidos para as melhores configurações de cada um dos métodos. Logo em seguida é discutido os resultados para os para os algoritmos baseados em busca por largura para as mesmas bases do algoritmo por busca em profundidade

### 6.3.1 Algoritmos baseados na busca em profundidade

Em uma primeira análise, a Tabela 4, referencia-se ao algoritmo por busca em profundidade, que apresenta os resultados obtidos em relação ao percentual de amostras classificadas incorretamente dado pelo erro médio e desvio padrão do método adotado para cada base de dados avaliada durante a fase de teste. Pode-se perceber que foi possível executar para todos os *ensembles* propostos: 11, 101 e 1001, notando-se uma redução do erro calculado pelo *k-fold* a medida que a quantidade de *ensembles* aumenta.

Tabela 4 – Algoritmo de Busca em Profundidade

Base	Tamanho	10- <i>fold</i>	Tempo (s)
Breast	11	21.71±5.38	533.86
	101	21.54±4.89	2215.82
	1001	21.48±4.72	19463.12
CNS	11	35.72±3.85	8647.47
	101	35.18±2.03	72836.07
	1001	35.01±1.94	125784.69
Colon	11	17.38±2.95	279.90
	101	17.26±2.13	2230.04
	1001	17.02±2.58	33422.32
DLBCL	11	5.07±1.68	575.23
	101	5.02±1.45	3429.32
	1001	5.05±1.40	45574.61
Leukemia	11	5.48±2.03	256.57
	101	5.45±1.76	4236.52
	1001	5.60±1.73	27277.71
Protaste	11	11.75±1.72	4486.78
	101	11.80±1.76	22969.54
	1001	11.81±1.49	45714.66

Fonte: Elaborada pelo autor (2022).

A Tabela 5, assim como a tabela anterior, apresenta os resultados obtidos em relação ao percentual de amostras classificadas incorretamente dado pelo erro médio e desvio padrão do método adotado para cada base de dados avaliada durante a fase de teste, mas agora usando o método Algoritmo por Busca em Profundidade com Vizinhança.

Para realização dos testes com o algoritmo TP-DFS-NB foi considerado os vizinhos de 10% até 35%, com a variação de 5% em 5%. É possível notar também que de acordo com a variação da porcentagem da vizinhança do algoritmo, ocorre uma redução no erro médio e desvio padrão até um limite de 35%, quando ocorre justamente um aumento do erro médio e desvio padrão, tornando assim o limiar para se considerar o quantia de vizinhos.

Tabela 5 – Algoritmo de Busca em Profundidade com Subcomitê por Vizinhaça

Base	Tam.	Busca em Profundidade		Vizinhaça 10%		Vizinhaça 15%		Vizinhaça 20%		Vizinhaça 25%		Vizinhaça 30%	
		10-fold	Tempo	10-fold	Tempo	10-fold	Tempo	10-fold	Tempo	10-fold	Tempo	10-fold	Tempo
Breast	11	21.71±5.38	533.86	20.67±3.60	373.00	20.80±3.98	473.18	19.12±4.00	498.40	19.11±3.98	490.03	19.11±3.80	489.33
	101	21.54±4.89	2215.82	19.58±2.73	1608.08	19.43±3.08	1826.61	19.15±2.74	1673.50	19.14±2.72	1776.40	19.04±2.12	1828.40
	1001	21.48±4.72	19463.12	19.82±1.95	20379.13	19.82±1.88	20172.87	19.78±1.99	20159.69	19.89±1.90	20263.29	19.90±1.32	19963.09
CNS	11	35.72±3.85	8647.47	35.54±3.16	8274.40	34.14±3.25	8375.88	33.83±3.87	7589.17	33.74±3.17	7881.11	33.78±3.25	7901.17
	101	35.18±2.03	72836.07	34.98±2.96	69743.55	33.78±2.96	69801.55	33.67±2.34	67726.17	33.55±2.04	65726.17	33.05±1.98	65429.81
	1001	35.01±1.94	125784.69	34.05±1.04	122873.90	33.95±1.04	123003.98	33.11±1.75	123784.70	33.17±1.59	124581.70	33.28±1.01	124712.12
Colon	11	17.38±2.95	279.90	16.84±2.05	282.00	15.98±2.26	268.70	15.59±2.64	249.52	15.87±2.01	278.22	15.74±2.38	282.87
	101	17.26±2.13	2230.04	15.08±2.28	2287.11	14.82±2.74	2201.92	14.68±2.28	2218.41	14.36±2.14	2214.08	14.45±2.51	2198.82
	1001	17.02±2.58	33422.32	14.55±2.47	32873.87	14.48±2.39	32873.87	14.40±2.30	33004.19	14.36±2.22	32984.98	14.39±2.27	32746.66
DLBCL	11	5.07±1.68	575.23	4.29±1.27	659.76	4.13±1.34	467.53	4.14±0.81	547.63	4.34±0.92	568.33	3.75±0.92	535.66
	101	5.02±1.45	3429.32	3.74±0.75	3346.13	3.76±0.65	3435.99	3.81±0.66	3262.26	3.66±0.66	3850.58	3.77±0.70	3297.58
	1001	5.05±1.40	45574.61	3.86±0.60	47138.25	3.84±0.57	47567.11	3.84±0.54	47424.49	3.83±0.60	46725.61	3.74±0.63	52424.94
Leukemia	11	5.48±2.03	256.57	5.42±2.11	266.05	5.05±2.27	275.54	4.76±1.50	318.26	4.26±1.14	307.16	4.16±1.28	277.02
	101	5.45±1.76	4236.52	5.32±1.60	4158.52	4.57±1.27	4203.52	4.16±1.00	4177.02	4.05±1.01	4701.03	4.15±1.02	4304.02
	1001	5.60±1.73	27277.71	5.55±1.25	27589.70	4.15±1.78	28307.78	3.87±0.90	27257.97	3.55±0.95	27807.90	3.07±0.99	27792.00
Protaste	11	11.75±1.72	4486.78	11.55±1.27	4203.33	11.05±1.12	4237.17	9.64±1.37	4148.45	9.63±1.22	4119.44	9.62±1.25	4239.64
	101	11.80±1.76	22969.54	11.72±1.60	22170.50	11.02±1.22	23042.67	9.46±1.25	23118.55	9.16±1.27	23898.71	9.27±1.33	23487.70
	1001	11.81±1.49	45714.66	11.11±1.30	45782.27	11.02±1.02	44821.97	9.64±0.99	43519.87	9.42±1.02	44726.07	9.21±0.98	43972.64

Fonte: Elaborada pelo autor (2022).

### 6.3.2 Algoritmos baseados na busca em largura

Em uma primeira análise, a Tabela 6, pode-se perceber que o uso do Algoritmo por Busca em Largura tornou-se inviável para um comitê com 1001 hipóteses soluções, por isso não possui o valor referente ao *k-fold* para todos os bancos de dados analisados. Comprovando assim uma desvantagem por essa abordagem proposta.

Sendo assim a respectiva tabela organiza os resultados obtidos em relação ao percentual de amostras classificadas incorretamente dado pelo erro médio e desvio padrão do método adotado para cada base de dados avaliada durante a fase de teste. Assim como as outras tabelas, esse método mostra uma redução no erro médio calculado a medida que o comitê aumenta.

Tabela 6 – Algoritmo de Busca em Largura

Base	Tamanho	10-fold	Tempo
Breast	11	21.11±4.93	495.86
	101	20.88±4.99	12978.76
CNS	11	34.63±3.55	11315.47
	101	33.18±2.03	115781.07
Colon	11	16.05±2.98	298.07
	101	15.46±2.55	3644.76
DLBCL	11	4.22±1.83	604.96
	101	4.03±0.71	10018.73
Leukemia	11	4.97±1.86	284.09
	101	3.89±1.05	14889.62
Protaste	11	9.82±1.70	6309.81
	101	9.70±1.05	50054.01

Fonte: Elaborada pelo autor (2022).

Ao analisar o Algoritmo por Busca em Largura com Subcomitês por Votação em comparação Algoritmo por Busca em Largura é possível notar uma melhora nos *k-Folds* dos *ensembles* a medida que a porcentagem representativa dos comitês aumenta, ocorrendo uma piora nos erros e desvio padrão com a representatividade de 50% do subcomitê como é mostrado na Tabela 7. Na execução do algoritmo TP-BFS-SV foram realizados testes considerando os subcomitês dos *ensembles* de 90% até 50%, variando de 10% em 10%.

O último algoritmo aplicado em testes foi o Algoritmo por Busca em Largura com SubComitês por Vizinhança, cuja a tabela é apresentada na Tabela 8, nessa abordagem uma melhora nos *k-Folds* dos *ensembles* a medida que a porcentagem representativa dos comitês aumenta, ocorrendo uma piora nos erros e desvio padrão com a representatividade de 30% do subcomitê por vizinhos. Na execução do algoritmo TP-BFS-NB foram realizados testes considerando os subcomitês dos *ensembles* de 10% até 30%, variando de 5% em 5%.

Tabela 7 – Algoritmo de Busca em Largura com Subcomitê por Votação

Base	Tam	Busca em Largura		Votação 10%		Votação 20%		Votação 30%		Votação 40%		Votação 50%	
		10-fold	Tempo	10-fold	Tempo	10-fold	Tempo	10-fold	Tempo	10-fold	Tempo	10-fold	Tempo
Breast	11	21.11±4.93	495.86	19.35±4.12	3614.18	19.53±4.01	5716.23	19.28±3.27	3033.60	19.32±3.00	3327.72	19.73±2.87	3005.18
	101	20.88±4.99	12978.76	19.55±2.38	40124.33	19.55±2.46	40422.87	19.57±2.26	27061.28	19.62±2.46	90953.13	19.65±2.16	91237.80
CNS	11	34.63±3.55	11315.47	35.08±3.63	18714.25	33.83±2.73	16823.64	33.37±2.74	18685.55	33.28±2.42	18532.55	33.08±2.30	18824.81
	101	33.18±2.03	115781.07	35.01±2.78	201689.58	33.62±2.33	200899.56	33.57±2.62	200489.58	33.60±2.44	201180.08	33.58±2.29	201719.00
Colon	11	16.05±2.98	298.07	15.52±2.68	431.84	15.38±2.64	464.69	14.65±2.57	394.76	14.91±2.36	428.12	14.99±2.49	462.45
	101	15.46±2.55	3644.76	14.67±2.28	8901.45	14.50±2.40	8997.04	14.74±2.36	8717.61	14.57±2.38	9112.49	14.57±2.39	9148.11
DLBCL	11	4.22±1.83	604.96	4.07±1.20	960.12	3.72±0.82	1187.01	3.83±0.73	1364.66	3.78±0.67	1101.92	3.75±0.63	982.64
	101	4.03±0.71	10108.73	3.73±0.75	21276.32	3.67±0.63	21722.80	3.68±0.66	20973.66	3.93±0.46	21783.84	3.96±0.39	20753.91
Leukemia	11	4.97±1.86	284.09	5.22±1.72	555.35	4.46±1.38	649.72	4.90±1.32	598.12	4.28±1.18	549.06	4.00±1.09	539.23
	101	3.89±1.05	14889.62	4.20±0.98	31881.04	3.78±0.99	29101.13	3.93±0.90	29874.69	3.96±0.85	29887.84	3.80±0.94	30549.33
Prostate	11	9.82±1.70	6309.81	11.38±1.61	11173.32	9.72±1.41	11119.77	9.79±1.41	11586.91	9.62±1.29	11459.35	9.50±1.39	11784.71
	101	9.70±1.05	50054.01	9.67±1.25	72968.11	9.44±1.21	73223.68	9.58±1.12	722289.80	9.48±1.11	72843.36	9.55±1.14	71998.64

Fonte: Elaborada pelo autor (2022).

Tabela 8 – Algoritmo de Busca em Largura com Subcomitê por Vizinhaça

Base	Tamanho	Busca em Largura		Vizinhaça 10%		Vizinhaça 15%		Vizinhaça 20%		Vizinhaça 25%		Vizinhaça 30%	
		10- <i>fold</i>	Tempo	10- <i>fold</i>	Tempo	10- <i>fold</i>	Tempo	10- <i>fold</i>	Tempo	10- <i>fold</i>	Tempo	10- <i>fold</i>	Tempo
Breast	11	21.11±4.93	495.86	20.22±3.86	738.76	20.13±4.15	830.81	19.58±3.97	704.66	19.55±3.45	847.64	20.40±4.01	852.73
	101	20.88±4.99	12978.76	19.28±2.52	19639.29	19.23±3.13	20911.01	19.28±3.97	19982.61	19.40±3.82	20145.31	19.58±3.63	19872.37
CNS	11	34.63±3.55	11315.47	32.98±2.04	16315.47	32.63±2.93	16248.55	32.73±2.03	15302.72	32.03±3.29	16005.37	34.63±3.32	16581.67
	101	33.18±2.03	115781.07	32.28±2.43	180854.27	32.53±2.03	181423.19	32.13±2.79	182372.55	32.08±2.48	180658.21	33.17±2.29	191012.51
Colon	11	16.05±2.98	298.07	15.85±2.31	361.08	15.74±2.73	408.29	15.62±2.65	405.98	15.02±2.35	398.07	15.95±2.64	485.66
	101	15.46±2.55	3644.76	14.78±2.60	6070.18	14.71±2.45	6159.63	14.70±2.61	6588.02	14.64±2.61	7012.33	14.73±2.31	6970.12
DLBCL	11	4.22±1.83	604.96	3.98±0.93	817.43	3.99±1.13	1165.26	3.90±1.05	1157.53	3.88±1.25	965.87	4.01±1.25	1111.27
	101	4.03±0.71	10018.73	3.63±0.67	13078.19	3.61±0.68	13418.07	3.56±0.66	13598.02	3.44±0.61	13512.67	3.62±0.69	13452.38
Leukemia	11	4.97±1.86	284.09	4.69±1.55	364.53	4.60±1.49	404.85	4.47±1.52	412.11	4.17±1.62	488.31	5.12±1.53	409.54
	101	3.89±1.05	14889.62	3.93±0.95	16206.60	4.15±1.02	15747.98	4.11±1.07	19676.48	4.06±1.06	19182.83	4.26±1.34	19174.55
Protaste	11	9.82±1.70	6309.81	9.67±1.57	7489.96	9.67±1.88	7617.76	9.65±1.51	7804.01	9.52±1.41	9745.64	9.85±1.61	7821.73
	101	9.70±1.05	50054.01	9.64±1.36	63704.98	9.54±1.33	55817.44	9.44±1.23	55115.66	9.20±1.40	52781.96	9.58±1.26	55611.01

Fonte: Elaborada pelo autor (2022).

### 6.3.3 Comparação dos algoritmos de classificação

A Tabela 9 compara os resultados dos algoritmos do SVM, BPM, VRSM, EACC, TP-DFS, TP-DFS-NB, TP-BFS, TP-BFS-NB e TP-BFS-SV quando empregado um 10x10-10-*fold cross-validation*. Essas abordagens foram adotadas a fim de reduzir o enviesamento dos métodos analisados. Foi utilizada uma estratégia de validação cruzada estratificada, onde cada *fold* mantém a porcentagem de pontos de dados para cada classe. Para comparações mais precisas foram sempre selecionados, para cada conjunto de dados, os mesmos conjuntos de treinamento e teste e também os mesmos 10 subconjuntos para as validações cruzadas, preservando assim um processo aleatório. Os resultados para as abordagens BPM e VRSM foram retirados do trabalho Enes, Villela e Neto (2016) e os resultados do EACC foram do trabalho Goulart et al. (2020). Os melhores resultados para cada conjunto de dados foram marcados em negrito.

Os algoritmos baseados nas estratégias do Perceptron de Tukey conseguiram serem melhores em cinco das seis bases analisadas, sendo duas para o algoritmo TP-DFS-NB e três para o algoritmo TP-BFS-NB. O SVM conseguiu uma performance melhor apenas na base Leukemia.

Tabela 9 – Comparação dos erros médios entre os classificadores.

Base	SVM	BPM	VRSM	EACC	TP-DFS	TP-DFS-NB	TP-BFS	TP-BFS-NB	TP-BFS-SV
Breast	20.67±2.49	20.35±3.42	19.83±2.39	19.33±1.54	21.54±4.89	<b>19.04±2.12</b>	20.88±4.99	19.40±3.82	19.55±2.46
CNS	33.50±1.99	33.50±1.99	32.87±2.15	32.33±3.02	35.18±2.03	33.05±1.98	33.18±2.03	<b>32.08±2.48</b>	33.62±2.33
Colon	18.69±2.32	15.68±2.35	15.44±2.29	15.29±2.57	17.26±2.13	<b>14.45±2.51</b>	15.46±2.55	14.64±2.61	14.50±2.40
DLBCL	3.81±0.68	3.86±0.79	3.49±0.87	4.38±1.20	5.02±1.45	3.77±0.70	4.03±0.71	<b>3.44±0.61</b>	3.67±0.63
Leukemia	2.75±0.88	5.50±1.32	3.39±1.30	4.91±1.52	5.45±1.76	4.15±1.02	3.89±1.05	4.06±1.06	3.78±0.99
Protaste	9.58±1.35	10.46±1.57	9.46±1.77	8.75±1.05	10.80±1.76	9.27±1.33	9.70±1.05	<b>9.20±1.40</b>	9.44±1.21

Fonte: Elaborada pelo autor (2022).

## 7 CONCLUSÃO

Esse trabalho buscou apresentar abordagens para cálculo de um ponto próximo ao centro de massa do espaço de versões através de uma abordagem estatística baseada na profundidade de Tukey.

Em relação a primeira contribuição do trabalho, o Perceptron de Tukey, é possível notar que ao aplicar o paradigma do campo estatístico para profundidade de dados é possível identificar a hipótese solução mais profunda. Isso ocorre a medida que os membros considerados como ganhadores na divisão do hiperplano separador são incrementados em uma unidade na sua profundidade, o resultado é o encontro de uma hipótese solução mais profunda, cunhada nesse trabalho como Perceptron de Tukey, ou o perceptron mais próximo do centro de massa do espaço de versões. Portanto um estudo experimental foi realizado, apresentando assim os algoritmos de busca em profundidade e largura para encontrar o Perceptron de Tukey.

Para a estratégia dos algoritmo de busca baseados em profundidades duas versões foram apresentadas. A versão do Perceptron de Tukey por busca em profundidade mostrou-se ser uma estratégia simples porém interessante pois através desse método é possível encontrar a hipótese solução, ou o ponto mais profundo para grandes *ensembles*. Neste trabalho foram realizados testes para a geração de 1001 hipóteses soluções notando uma redução do erro calculado pelo *k-fold* à medida que a quantidade foi aumentada. Uma desvantagem que pode ser destacada desse método é a possível má escolha inicial do sorteio das hipóteses soluções podendo ser pontos próximos a fronteira do espaço de versões e sendo assim resultar em um Perceptron de Tukey ruim.

A segunda versão desenvolvida nesse trabalho baseado em profundidade de dados foi o Perceptron de Tukey por busca em profundidade com vizinhança, abordagem segue que os mesmos passos do TP-BFS, porém esse método, além de poder ser aplicado a grandes *ensembles*, gera uma nova solução baseada na hipótese mais profunda centrada em um circulo trazendo o ponto mais fidedigno ao centro de Bayes do espaço de versões. Ao analisar os resultados encontrados por essas abordagem percebe-se que o Perceptron de Tukey por busca em profundidade com vizinhança possui melhores resultados comparado com Perceptron de Tukey por busca em profundidade em todas as bases analisadas, sendo a vizinhança com 30% com os melhores resultados.

Outra estratégia realizada nesse trabalho para encontrar o ponto mais profundo foi baseada em algoritmos de busca por largura, por usar essa estratégia como busca é preciso comentar que o esforço computacional é maior, visto que nessa abordagem todos as hipóteses soluções são analisadas para encontro do Perceptron de Tukey, tornando assim uma desvantagem ao usar essa abordagem para localizar o Perceptron de Tukey. O primeiro algoritmo desenvolvido, Perceptron de Tukey por busca em Largura, percorre

todos os possíveis pares das hipóteses soluções até encontrar a hipótese mais profunda, como mostrado nesse trabalho identificando uma hipótese próxima ao ponto de Bayes do espaço de versões. Já a abordagem do algoritmo Perceptron de Tukey por busca em Largura com subcomitê por votação além de repetir os mesmos passos do algoritmo anterior, seleciona em teoria os melhores representantes para então propor um novo Perceptron de Tukey. De maneira análoga o algoritmo de Perceptron de Tukey por busca em Largura com subcomitê por vizinhança baseia-se no primeiro algoritmo de busca em largura, mas como resultado propõem um novo Perceptron de Tukey a partir das hipóteses soluções mais próxima do Perceptron de Tukey calculado.

Como algumas possibilidade de trabalhos futuros, quanto o desenvolvimento de algoritmos Duais e introdução a funções *kernel*, o que permite também a solução de problemas não-linearmente separáveis, sugere-se a implementação do método segundo essas abordagens. Quanto ao processo de escolha das hipóteses soluções para localizar o membro do comitê mais profundo, não existe garantia de que essa estratégia sempre é tomada de maneira correta. Por essa razão desenvolver métodos que trazem maior certeza ao percorrer todas as hipóteses soluções em busca do ponto mais profundo. Quanto às distribuições das hipóteses, propor novas distribuições com medidas de similaridades diferentes, garantindo assim uma maior eficiência em criar os espaços de versões. Por fim, em relação aos dados usados para comparação dos modelos, observou-se que os dados analisados não apresentavam grandes desbalanceamentos em relação as classes. Por essa razão, um estudo experimental para a avaliação do método comitê proposto em bases de dados desbalanceados pode ser realizado.

## REFERÊNCIAS

- ALOUPIIS, Greg; CORTÉS, Carmen; GÓMEZ, Francisco; SOSS, Michael; TOUSSAINT, Godfried. Lower bounds for computing statistical depth. **Computational Statistics & Data Analysis**, Elsevier, v. 40, n. 2, p. 223–229, 2002.
- ALOUPIIS, Greg; LANGERMAN, Stefan; SOSS, Michael; TOUSSAINT, Godfried. Algorithms for bivariate medians and a fermat–torricelli problem for lines. **Computational Geometry**, Elsevier, v. 26, n. 1, p. 69–79, 2003.
- ALOUPIIS, Greg; MCLEISH, Erin. A lower bound for computing oja depth. **Information processing letters**, Elsevier, v. 96, n. 4, p. 151–153, 2005.
- AMALDI, Edoardo; KANN, Viggo. The complexity and approximability of finding maximum feasible subsystems of linear relations. **Theoretical computer science**, Elsevier, v. 147, n. 1-2, p. 181–210, 1995.
- BARNETT, Vic. The ordering of multivariate data. **Journal of the Royal Statistical Society: Series A (General)**, Wiley Online Library, v. 139, n. 3, p. 318–344, 1976.
- BERN, Marshall; EPPSTEIN, David. Multivariate regression depth. In: **Proceedings of the sixteenth annual symposium on Computational geometry**. [S.l.: s.n.], 2000. p. 315–321.
- BHARGAVA, Aditya. **Grokking Algorithms: An illustrated guide for programmers and other curious people**. [S.l.]: Manning Publications Co., 2016.
- BOSER, Bernhard E; GUYON, Isabelle M; VAPNIK, Vladimir N. A training algorithm for optimal margin classifiers. In: **Proceedings of the fifth annual workshop on Computational learning theory**. [S.l.: s.n.], 1992. p. 144–152.
- CHAN, Timothy M. An optimal randomized algorithm for maximum tukey depth. In: **SODA**. [S.l.: s.n.], 2004. v. 4, p. 430–436.
- CHEN, Zhiqiang; TYLER, David E. The influence function and maximum bias of tukey’s median. **The Annals of Statistics**, Institute of Mathematical Statistics, v. 30, n. 6, p. 1737–1759, 2002.
- CHEN, Zhiqiang; TYLER, David E. On the behavior of tukey’s depth and median under symmetric stable distributions. **Journal of statistical planning and inference**, Elsevier, v. 122, n. 1-2, p. 111–124, 2004.
- CORMEN, Thomas H; LEISERSON, Charles E; RIVEST, Ronald L; STEIN, Clifford. Introduction to algorithms second edition. **The Knuth-Morris-Pratt Algorithm**, 2001.
- DIETTERICH, Thomas G. Ensemble methods in machine learning. In: SPRINGER. **International workshop on multiple classifier systems**. [S.l.], 2000. p. 1–15.
- DONOHU, David L. **Breakdown properties of multivariate location estimators**. [S.l.], 1982.

- DONOHO, David L; GASKO, Miriam et al. Breakdown properties of location estimates based on halfspace depth and projected outlyingness. **The Annals of Statistics**, Institute of Mathematical Statistics, v. 20, n. 4, p. 1803–1827, 1992.
- EFRON, Bradley; TIBSHIRANI, Robert. Improvements on cross-validation: the 632+ bootstrap method. **Journal of the American Statistical Association**, Taylor & Francis, v. 92, n. 438, p. 548–560, 1997.
- ENES, Karen Braga; VILLELA, Saulo Moraes; NETO, Raul Fonseca. A novel ensemble approach based on balanced perceptrons applied to microarray datasets. In: IEEE. **2015 Brazilian Conference on Intelligent Systems (BRACIS)**. [S.l.], 2015. p. 186–191.
- ENES, Karen Braga; VILLELA, Saulo Moraes; NETO, Raul Fonseca. Version space reduction based on ensembles of dissimilar balanced perceptrons. In: **IJCAI**. [S.l.: s.n.], 2016. p. 1448–1454.
- FACELI, Katti; LORENA, Ana Carolina; GAMA, João; CARVALHO, André Carlos Ponce de Leon et al. *Inteligência artificial: Uma abordagem de aprendizado de máquina*. 2011.
- FREUND, Yoav; SCHAPIRE, Robert E. Experiments with a New Boosting Algorithm. In: **International Conference on Machine Learning**. [S.l.: s.n.], 1996. p. 148–156.
- FREUND, Yoav; SCHAPIRE, Robert E. A decision-theoretic generalization of on-line learning and an application to boosting. **Journal of computer and system sciences**, Elsevier, v. 55, n. 1, p. 119–139, 1997.
- GEISSER, Seymour. The predictive sample reuse method with applications. **Journal of the American statistical Association**, Taylor & Francis Group, v. 70, n. 350, p. 320–328, 1975.
- GEISSER, S. **An Introduction to Predictive Inference**. [S.l.]: Chapman and Hall, New York, 1993.
- GIL, Joseph; STEIGER, William; WIGDERSON, Avi. Geometric medians. **Discrete Mathematics**, Elsevier, v. 108, n. 1-3, p. 37–51, 1992.
- GILAD-BACHRACH, Ran; BURGESS, Chris JC. The median hypothesis. In: **Empirical Inference**. [S.l.]: Springer, 2013. p. 161–175.
- GILAD-BACHRACH, Ran; NAVOT, Amir; TISHBY, Naftali. Bayes and tukey meet at the center point. In: SPRINGER. **International Conference on Computational Learning Theory**. [S.l.], 2004. p. 549–563.
- GLAAB, E; BACARDIT, J; GARIBALDI, JM; KRASNOGOR, N. Using rule-based machine learning for candidate disease gene prioritization and sample. 2012.
- GOLUB, Todd R; SLONIM, Donna K; TAMAYO, Pablo; HUARD, Christine; GAASENBEEK, Michelle; MESIROV, Jill P; COLLIER, Hilary; LOH, Mignon L; DOWNING, James R; CALIGIURI, Mark A et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. **science**, American Association for the Advancement of Science, v. 286, n. 5439, p. 531–537, 1999.

GOULART, Renan Motta; VILLELA, Saulo Moraes; BORGES, Carlos Cristiano Hasenclever; NETO, Raul Fonseca. An evolutionary analytic center classifier. In: SPRINGER. **Brazilian Conference on Intelligent Systems**. [S.l.], 2020. p. 34–48.

HAMURKAROGLU, Canan; MERT, Mehmet; SAYKAN, Yasemin. Nonparametric control charts based on mahalanobis depth. **Hacettepe Journal of Mathematics and Statistics**, v. 33, p. 57–67, 2004.

HANSEN, Lars Kai; SALAMON, Peter. Neural network ensembles. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 12, n. 10, p. 993–1001, 1990.

HERBRICH, Ralf. **Learning kernel classifiers: theory and algorithms**. [S.l.]: MIT press, 2001.

HERBRICH, Ralf; GRAEPEL, Thore; CAMPBELL, Colin. Bayes point machines. **Journal of Machine Learning Research**, v. 1, n. Aug, p. 245–279, 2001.

HUBER, Peter J et al. Robust regression: asymptotics, conjectures and monte carlo. **The annals of statistics**, Institute of Mathematical Statistics, v. 1, n. 5, p. 799–821, 1973.

JOHNSON, David S.; PREPARATA, Franco P. The densest hemisphere problem. **Theoretical Computer Science**, Elsevier, v. 6, n. 1, p. 93–107, 1978.

JÚNIOR, Osmar Abílio de Carvalho; CARVALHO, Ana Paula Ferreira de; GUIMARÃES, Renato Fontes; MENESES, Paulo Roberto; SHIMABUKURO, Yosio Edemir. Mistura espectral:(i) detecção dos membros finais utilizando a geometria do simplex. **Revista Espaço e Geografia**, v. 6, n. 1, 2003.

KOHAVI, Ron et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: MONTREAL, CANADA. **Ijcai**. [S.l.], 1995. v. 14, n. 2, p. 1137–1145.

KREVELD, Marc van; MITCHELL, Joseph SB; ROUSSEEUW, Peter; SHARIR, Micha; SNOEYINK, Jack; SPECKMANN, Bettina. Efficient algorithms for maximum regression depth. **Discrete & computational geometry**, Springer, v. 39, n. 4, p. 656–677, 2008.

KUNCHEVA, L. Combining pattern classifiers methods and algorithms. john wiley&sons. **Inc. Publication, Hoboken**, 2004.

LANGERMAN, Stefan; STEIGER, William. The complexity of hyperplane depth in the plane. In: CITESEER. **Japan Conference on Discrete and Computational Geometry**. [S.l.], 2000.

LANGERMAN, Stefan; STEIGER, William. Computing a maximal depth point in the plane. In: CITESEER. **Proceedings 4th Japan Conference on Discrete and Computational Geometry, Lecture Notes in Computer Science**, to appear. [S.l.], 2000.

LIU, Regina Y. On a notion of simplicial depth. **Proceedings of the National Academy of Sciences**, National Acad Sciences, v. 85, n. 6, p. 1732–1734, 1988.

- LIU, Regina Y. Data depth: center-outward ordering of multivariate data and nonparametric multivariate statistics. In: **Recent advances and trends in nonparametric statistics**. [S.l.]: Elsevier Inc., 2003. p. 155–167.
- LIU, Regina Y et al. On a notion of data depth based on random simplices. **The Annals of Statistics**, Institute of Mathematical Statistics, v. 18, n. 1, p. 405–414, 1990.
- MAHALANOBIS, Prasanta Chandra. On the generalized distance in statistics. In: NATIONAL INSTITUTE OF SCIENCE OF INDIA. [S.l.], 1936.
- MCCULLOCH, Warren S; PITTS, Walter. A logical calculus of the ideas immanent in nervous activity. **The bulletin of mathematical biophysics**, Springer, v. 5, n. 4, p. 115–133, 1943.
- MOSLER, Karl. Depth statistics. In: **Robustness and complex data structures**. [S.l.]: Springer, 2013. p. 17–34.
- MOZHAROVSKYI, Pavlo. Tukey depth: linear programming and applications. **arXiv preprint arXiv:1603.00069**, 2016.
- NETO, Maria Raquel. **O Conceito de Profundidade em Estatística**. Dissertação (Mestrado) — Universidade Técnica de Lisboa Instituto Superior Técnico Departamento de Matemática, 2008.
- NOVIKOFF, Albert B. **On convergence proofs for perceptrons**. [S.l.], 1963.
- OJA, Hannu. Descriptive statistics for multivariate distributions. **Statistics & Probability Letters**, Elsevier, v. 1, n. 6, p. 327–332, 1983.
- OPITZ, David W; SHAVLIK, Jude W et al. Generating accurate and diverse members of a neural-network ensemble. **Advances in neural information processing systems**, Citeseer, p. 535–541, 1996.
- PORZIO, Giovanni C; RAGOZINI, Giancarlo. Convex hull probability depth. In: **International Workshop on Robust and Nonparametric Statistical Inference. Hejnice, Czech Republic**. [S.l.: s.n.], 2007.
- ROLI, Fabio; GIACINTO, Giorgio; VERNAZZA, Gianni. Methods for designing multiple classifier systems. In: SPRINGER. **International Workshop on Multiple Classifier Systems**. [S.l.], 2001. p. 78–87.
- ROSENBLATT, Frank. The perceptron: a probabilistic model for information storage and organization in the brain. **Psychological review**, American Psychological Association, v. 65, n. 6, p. 386, 1958.
- ROUSSEEUW, Peter J; RUTS, Ida. Constructing the bivariate tukey median. **Statistica Sinica**, JSTOR, p. 827–839, 1998.
- RUSSELL, Stuart; NORVIG, Peter. **Inteligência artificial: uma abordagem moderna**. ed. Campus, 2ª Edição. São Paulo, Brazil, 2003.
- SHI, Haobin; LIN, Zhiqiang; HWANG, Kao-Shing; YANG, Shike; CHEN, Jialin. An adaptive strategy selection method with reinforcement learning for robotic soccer games. **IEEE Access**, IEEE, v. 6, p. 8376–8386, 2018.

- SOLLICH, Peter; KROGH, Anders. Learning with ensembles: How overfitting can be useful. In: **Advances in neural information processing systems**. [S.l.: s.n.], 1996. p. 190–196.
- STRUYF, Anja; ROUSSEEUW, Peter J. An algorithm for the multivariate tukey median. In: SPRINGER. **COMPSTAT**. [S.l.], 2000. p. 463–468.
- TONG, Simon; KOLLER, Daphne. Support vector machine active learning with applications to text classification. **Journal of machine learning research**, v. 2, n. Nov, p. 45–66, 2001.
- TUKEY, John Wilder; CLEVELAND, William S. **The collected works of John W. Tukey. 5. Graphics: 1965-1985**. [S.l.]: Wadsworth & Brooks/Cole, 1988.
- TUMER, Kagan; GHOSH, Joydeep. Analysis of decision boundaries in linearly combined neural classifiers. **Pattern recognition**, Elsevier, v. 29, n. 2, p. 341–348, 1996.
- VAPNIK, Vladimir N. The nature of statistical learning. **Theory**, springer, 1995.
- WAHBA, Grace; WOLD, Svante. A completely automatic french curve: Fitting spline functions by cross validation. **Communications in Statistics-Theory and Methods**, Taylor & Francis, v. 4, n. 1, p. 1–17, 1975.
- WAZNEH, Hussein. **Approches flexibles et optimales en analyse fréquentielle régionale des crues en se basant sur les fonctions de profondeur**. Tese (Doutorado) — Université du Québec, Institut national de la recherche scientifique, 2015.
- WEISS, Eric A. Biographies: Eloge: Arthur lee samuel (1901-90). **IEEE Annals of the History of Computing**, IEEE, v. 14, n. 3, p. 55–69, 1992.
- ZHOU, Zhi-Hua; WU, Jianxin; TANG, Wei. Ensembling neural networks: many could be better than all. **Artificial intelligence**, Elsevier, v. 137, n. 1-2, p. 239–263, 2002.
- ZHU, Zexuan; ONG, Yew-Soon; DASH, Manoranjan. Markov blanket-embedded genetic algorithm for gene selection. **Pattern Recognition**, Elsevier, v. 40, n. 11, p. 3236–3248, 2007.
- ZUO, Yijun; SERFLING, Robert. General notions of statistical depth function. **Annals of statistics**, JSTOR, p. 461–482, 2000.