

FURG
UNIVERSIDADE FEDERAL DO RIO GRANDE

PPGMC
PROGRAMA DE PÓS-GRADUAÇÃO EM
MODELAGEM COMPUTACIONAL



Defesa 18/2019 - Mestrado – Lucas Tubino Bonifácio Costa



**UNIVERSIDADE FEDERAL
DO RIO GRANDE**

Lucas Tubino Bonifácio Costa

**Análise de modelos explicáveis de
sistemas de classificação para
registros hidroacústicos em
ambientes marítimos**

Orientadora: Dr^a. Graçaliz Pereira Dimuro
Coorientador: Dr. Stefan Cruz Weigert

Rio Grande
2019

PPGMC

**Programa de Pós-Graduação em
Modelagem Computacional**

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO RIO GRANDE
PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM COMPUTACIONAL

**ANÁLISE DE MODELOS EXPLICÁVEIS DE SISTEMAS DE
CLASSIFICAÇÃO PARA REGISTROS HIDROACÚSTICOS EM
AMBIENTES MARÍTIMOS**

por

Lucas Tubino Bonifácio Costa
Orientadora: Graçaliz Pereira Dimuro
Coorientador: Stefan Cruz Weigert

Dissertação para obtenção do Título de
Mestre em Modelagem Computacional

Rio Grande, Dezembro, 2019

Ficha catalográfica

C837a Costa, Lucas Tubino Bonifácio.
Análise de modelos explicáveis de sistemas de classificação para registros hidroacústicos em ambientes marítimos / Lucas Tubino Bonifácio Costa. – 2019.
59 f.

Dissertação (mestrado) – Universidade Federal do Rio Grande – FURG, Programa de Pós-Graduação em Modelagem Computacional, Rio Grande/RS, 2019.
Orientadora: Dra. Graçaliz Pereira Dimuro.
Coorientador: Dr. Stefan Cruz Weigert.

1. Hidroacústica 2. Classificação de Peixes Marinhos
3. Explicabilidade 4. Sistemas de Classificação Baseados em Regras Fuzzy 5. Árvores de Decisão I. Dimuro, Graçaliz Pereira II. Weigert, Stefan Cruz III. Título.

CDU 551.46

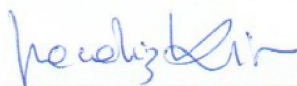
Lucas Tubino Bonifácio Costa

“Análise de modelos explicáveis de sistemas de classificação para registros hidroacústicos em ambientes marítimos”

Dissertação apresentada ao Programa de Pós Graduação em Modelagem Computacional da Universidade Federal do Rio Grande - FURG, como requisito parcial para obtenção do Grau de Mestre. Área concentração: Modelagem Computacional.

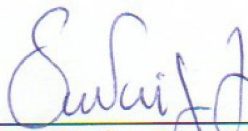
Aprovado em

BANCA EXAMINADORA



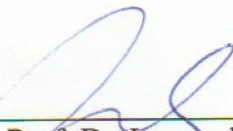
Prof.^a. Dr.^a. Graçaliz Pereira Dimuro

Orientadora – FURG



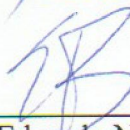
Prof. Dr. Stefan Cruz Weigert

Coorientador – FURG



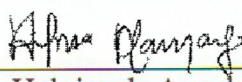
Prof. Dr. Leonardo Emmendorfer

FURG



Prof. Dr. Eduardo Nunes Borges

FURG



Prof.^a. Dr.^a. Heloisa de Arruda Camargo

UFSCar

Rio Grande - RS
2019

AGRADECIMENTOS

Agradeço a CAPES pelo financiamento da bolsa de mestrado recebida durante meu primeiro ano de Mestrado. Agradecimentos também aos membros do Laboratório de Tecnologia Pesqueira e Hidroacústica do Instituto de Oceanografia (Universidade Federal do Rio Grande - FURG) que forneceram os dados utilizados nesta dissertação, além de fornecer a ajuda necessária para a verificação dos resultados, e agradeço também aos membros do Laboratório de Gerenciamento de Informação e Computação Flexível do Centro de Ciencias Computacionais (C3-FURG) por auxiliar no aprendizado das ferramentas e teorias utilizadas neste trabalho.

RESUMO

Este trabalho visa desenvolver um modelo explicável de sistema de classificação de organismos marinhos a partir de dados obtidos por meio de ecossondas. As ecossondas funcionam normalmente acopladas na parte de baixo de uma embarcação, enviando sinais acústicos para detectar peixes, moluscos, zooplâncton ou outros objetos na coluna de água, gerando registros acústicos que aparecem nos ecogramas. Atualmente, o método utilizado para classificar os registros acústicos é a caracterização de ecotipos, onde certos padrões nos registros acústicos são identificados visualmente por um especialista que, tendo por base informações chamadas descritores, determina uma classe ao registro. Para fazer a classificação utilizando meios computacionais, descritores foram extraídos de ecogramas obtidos de pesquisa pesqueira, que então foram tratados para filtrar as características que descreviam melhor os registros contidos nos ecogramas. Estes dados foram utilizados para o gerar os modelos a partir de três algoritmos de classificação de fácil explicabilidade, um que utiliza árvores de decisão e dois baseados em regras *fuzzy*. Os modelos obtidos foram avaliados por medidas de acurácia baseadas nos desempenhos das classificações no treinamento. Os modelos foram também avaliados por um especialista da área, para análise de sua utilidade para descrever as espécies trabalhadas e o que poderia ser observado nestes resultados. Os modelos gerados pelos algoritmos tiveram um bom desempenho para os dados disponíveis, aonde os dados foram principalmente classificados por descritores geográficos e que, de acordo com um especialista da área, representam uma distribuição próxima da que realmente acontece para as espécies estudadas nessa região e época do ano trabalhadas.

Palavras-chaves: Hidroacústica, Classificação de Peixes Marinhos, Explicabilidade, Sistemas de Classificação Baseados em Regras Fuzzy, Árvores de Decisão.

ABSTRACT

This work aims to develop an explainable model of classification system of marine organisms from data obtained through ecosystems. Echo sounders typically work attached to the underside of a vessel, sending acoustic signals to detect fish, mollusks, zooplankton, or other objects in the water column, generating acoustic records that appear on echograms. Currently, the method used to classify acoustic records is the characterization of ecotypes, where certain patterns in acoustic records are visually identified by a specialist who, based on information called descriptors, determines a class for the record. To make the classification using computational means, descriptors were extracted from ecograms obtained from fishing research, which were then treated to filter the characteristics that best described the records contained in the ecograms. These data were used to generate the models from three easy-to-explain classification algorithms, one using decision trees and two fuzzy rules. The obtained models were evaluated by accuracy measures based on the performance of the training classifications. The models were also evaluated by an expert in the field, to analyze their usefulness to describe the worked species and what could be observed in these results. The models generated by the algorithms performed well for the available data, where the data were mainly classified by geographic descriptors and which, according to an expert in the field, represent a close distribution of what actually happens for the species studied in this region and time of the year of the data worked.

Keywords: Hydroacoustics, Marine Fish Classification, Explanability, Fuzzy Rule Based Classification Systems, Decision Trees.

ÍNDICE

1	Introdução	12
1.1	Objetivo	13
1.1.1	Objetivo geral	13
1.1.2	Objetivos específicos	13
1.2	Organização do Texto	14
2	Fundamentação Teórica	15
2.1	Registros acústicos de organismos marinhos e classificação de eco-registros . .	15
2.2	Classificação	17
2.3	Classificação Baseada em Árvores de Decisão	18
2.3.1	Algoritmo C4.5	18
2.4	Lógica <i>Fuzzy</i>	22
2.4.1	Regra <i>Fuzzy</i>	24
2.4.2	Sistemas de Classificação Baseados em Regras <i>Fuzzy</i>	25
2.4.3	FARC-HD	25
2.4.4	<i>Fuzzy Unordered Rule Induction Algorithm</i> - FURIA	29
3	Metodologia	35
3.1	Coleta de Dados	35
3.2	Pré-processamento	35
3.3	Estratégia de classificação	42
4	Resultados	45
4.1	Modelos Aprendidos	45
4.2	Avaliação dos Modelos	49
4.3	Interpretação dos Resultados	52
4.3.1	Classificação de ecotipos de cardumes pesqueiros baseada em regras fuzzy	52
4.3.2	Importância do número de instâncias e dos diferentes atributos na clas- sificação dos ecotipos de cardumes pesqueiros	52
5	Conclusão	54
5.1	Trabalhos Futuros	54
6	REFERÊNCIAS	56

LISTA DE FIGURAS

Figura 2.1: Exemplo de ecograma com ecoregistros. (Fonte: Ecograma do cruzeiro ECOSAR VI, realizados com a ecossonda SIMRAD EK500, em 2009) . . .	16
Figura 2.2: Método de Aprendizagem de um modelo supervisionado. (Adaptado de Lucca (2018))	18
Figura 2.3: Gráfico de uma função característica clássica. (Adaptado de Lucca (2018))	23
Figura 2.4: Gráfico de uma função característica <i>fuzzy</i> para números próximos de 2. (Adaptado de Lucca (2018))	23
Figura 2.5: Estrutura de um Sistema de Classificação Baseado em Regras <i>Fuzzy</i> . (Adaptado de Lucca (2018))	25
Figura 2.6: Um intervalo <i>fuzzy</i> I^F . (Adaptado de Hühn and Hüllermeier (2009)) . . .	31
Figura 3.1: Mapa mostrando a distribuição dos lances (Fonte: Laboratório de Tecnologia Pesqueira e Hidroacústica - IO - FURG)	36
Figura 3.2: Exemplo de ecograma obtido pela ecossonda SIMRAD EK500 (Fonte: Laboratório de Tecnologia Pesqueira e Hidroacústica - IO - FURG)	37
Figura 3.3: Ecograma obtido pela ecossonda SIMRAD EK500 com a área de pesca e os cardumes identificados (Fonte: Laboratório de Tecnologia Pesqueira e Hidroacústica - IO - FURG).	38
Figura 3.4: Exemplo de utilização da ferramenta <i>Identify</i> . (Fonte: Laboratório de Tecnologia Pesqueira e Hidroacústica - IO - FURG)	38
Figura 3.5: Exemplo de utilização da ferramenta <i>Zoom</i> . (Fonte: Laboratório de Tecnologia Pesqueira e Hidroacústica - IO - FURG)	39
Figura 3.6: Exemplo de ecograma com registro de cardume(s) disperso(s) obtido pela ecossonda SIMRAD EK500 (Fonte: Laboratório de Tecnologia Pesqueira e Hidroacústica - IO - FURG).	40
Figura 3.7: Imagem Representando o funcionamento da validação cruzada de k partições, com um exemplo de $k = 5$	43
Figura 4.1: Modelo gerado pelo Weka utilizando o algoritmo de árvores de decisão C4.5.	45

LISTA DE TABELAS

Tabela 1:	Medidas de dispersão antes da remoção dos cardumes anômalos.	40
Tabela 2:	Medidas de dispersão depois da remoção dos cardumes anômalos.	41
Tabela 3:	Resultados do teste de Kruskal-Wallis (K-W) são apresentados. Diferentes letras nas colunas indicam classes significativamente diferentes ($p < 0,05$), conforme o teste de comparação múltipla por postos de Siegel e Castellan.	42
Tabela 4:	Proposta de valores para as variáveis linguísticas referentes as funções de pertinência geradas pelo algoritmo FARC-HD.	47
Tabela 5:	Matriz de Confusão do C4.5.	50
Tabela 6:	Matriz de Confusão do FURIA.	50
Tabela 7:	Matriz de Confusão do FARC-HD.	50
Tabela 8:	Tabela com algumas medidas de acurácia por classe para cada um dos algoritmos. “R” se refere a Revocação, “P” para a Precisão e “F” para a Medida F.	51
Tabela 9:	Tabela com a contagem de regras em que um atributo está presente nos modelos gerados pelos algoritmos.	51

LISTA DE SÍMBOLOS

Classificação

e	Exemplo
E	Conjunto de Exemplos
$X(e)$	Aspecto/Variável/Característica do exemplo e
C_k	Classe k
C	Conjunto de Classes
M	Número de Classes do Problema
D	Modelo

Classificação Baseada em Árvores de Decisão

A	Atributo
d_h	Valor Atribuído a um Atributo
t	Limite Determinado no Nó

C4.5

TS	Treinamento
T	Conjunto de Casos Associados a um Nó

Lógica *Fuzzy*

U	Conjunto Universo
A	Subconjunto de U
X_A	Função com Domínio em U e a Imagem Contida em $\{0, 1\}$
F	Subconjunto <i>Fuzzy</i>
φ_F	Função de Pertinência de F
χ_F	Função Característica de F

Regra *Fuzzy*

R_j	Regra <i>Fuzzy</i>
A_{j1}	Conjunto <i>Fuzzy</i> Antecedente Representando um Termo Linguístico
C_j	Rótulo de Classe
RW_j	Peso da Regra
x_p	Exemplo a ser Classificado
L	Número de Regras no Banco de Regras
M	Número de Classes no Problema
$\mu_{A_j}(x_p)$	Grau de Correspondência
b_j^k	Grau de Associação
Y_k	Grau de Solidez
$F(Y_1, \dots, Y_M)$	Função de Decisão sobre o Grau de Solidez

FARC-HD

A e B	Conjuntos de Itens
$\emptyset$	Conjunto Vazio
T	Banco de Dados
X	Atributo
$ N $	Número de Transações em T
μ_A	Grau de Combinação

FURIA

I	Intervalo
A_i	Atributo
I_F	Intervalo <i>Fuzzy</i>
$\phi^{c,L}$	Limite Superior do Núcleo
$\phi^{c,U}$	Limite Inferior do Núcleo
$\phi^{s,L}$	Limite Superior do Suporte
$\phi^{s,U}$	Limite Inferior do Suporte
r^F	Regra <i>Fuzzy</i>
s	Suporte

LISTA DE ABREVIATURAS

XAI	Inteligência Artificial Explicável
FRBCS	Sistema de Classificação Baseado em Regras <i>Fuzzy</i>
MRF	Método de Raciocínio <i>Fuzzy</i>
FARC-HD	Método de classificação baseada em regras de associação <i>fuzzy</i>
FURIA	Algoritmo de Indução de Regra <i>Fuzzy</i> Não-Ordenada
IO	Instituto de Oceanografia
FURG	Universidade Federal do Rio Grande <i>Fuzzy</i>
K-W	Kruskal-Wallis
MDL	Comprimento Mínimo de Descrição
TP	Verdadeiro Positivo
FP	Falso Positivo
TN	Verdadeiro Negativo
FN	Falso Negativo

1 INTRODUÇÃO

Com o advento da computação, problemas de classificação de dados ou imagens que eram complicados ganharam um instrumento confiável para resolvê-los. Grande quantidade de dados ou características difíceis de serem descritas estão sendo resolvidos com o auxílio de sistemas inteligentes e técnicas de aprendizagem de máquina. Existem diversos métodos que podem ser aplicados nas mais diversas áreas de conhecimento, seja psicologia (Murphy and Stich, 2000), para xadrez (Quinlan, 1983), na medicina (Han et al., 2018) (Srividya and Arulmozhi, 2018), entre outras áreas (Aljawarneh et al., 2018) (Ashqar et al., 2019).

O foco deste trabalho consiste em utilizar ferramentas e algoritmos de inteligência artificial para classificar registros hidroacústicos de peixes, no ambiente marítimo, baseando-se em suas características físicas, seu comportamento e distribuições espaço-temporais. Através do método hidroacústico, estruturas biológicas, que são capazes de refletir ondas sonoras em função da diferença de densidade entre sua composição e o meio, são detectadas ao longo do percurso de embarcações em toda a coluna de água através de instrumentos hidroacústicos como ecossondas, ecobatímetros e sonares (Calazans and Griep, 2015) (Soares et al., 2005). Os dados utilizados neste trabalho foram obtidos em cruzeiros de pesquisa pesqueira feitos pelo Laboratório de Tecnologia Pesqueira e Hidroacústica do Instituto Oceanográfico da FURG nos cruzeiros ECO-SAR VI e VII, entre 2009 e 2010, e foi fornecido em forma de dados brutos que são lidos e convertidos por programas para possam ser exibidos visualmente.

Os registros gerados por esses equipamentos são disponibilizados na forma de ecogramas, que podem ser classificados por um especialista operando o equipamento, com base em alguns descritores pré-definidos da imagem. Um dos métodos para interpretar esses registros é a classificação em ecotipos, na qual certos padrões são identificados visualmente por um especialista e, com auxílio de dados espaço-temporais, os registros são rotulados para diferentes classes (Soares et al., 2005) Weigert and Madureira (2011). Para criar sites rótulos são necessários lances de pesca, aonde se identifica quais peixes, entre outros seres marinhos, correspondem aos registros encontrados no ecograma.

Este tipo de situação em que a classificação é feita a partir de informações subjetivas, como a forma dos cardumes (se é redondo, alongados, denso ou mais disperso) ou comportamento (vários cardumes separados ou poucos juntos, cardumes próximos à superfície ou perto do fundo, etc.), torna possível a aplicação da lógica *fuzzy* (Zadeh, 1988), que é uma maneira de modelar essas interpretações imprecisas dos dados para possibilitar o uso de aplicações computacionais, que possam apoiar o especialista na sua tomada de decisão e reduzir a subjetividade da classificação. A lógica *fuzzy* possibilita resolver situações em que não se pode usar somente a lógica clássica de verdadeiro ou falso. Por exemplo, um navio pesqueiro separa somente peixes que forem grandes e descarta no mar os menores. Digamos que para um peixe ser considerado

grande ele tenha 50 cm, então um peixe de 49 cm não seria considerado grande, o que em alguns casos poderia não fazer sentido. Então, para resolver estes casos, pode-se dizer que um peixe pertence a uma classe (grande, médio ou pequeno) com uma diferentes intensidades, podendo assim existir peixes que são considerados mais “grandes” que outros, logo um peixe pertence a um grupo com certa intensidade ou pertinência, o que é a ideia central da teoria dos conjuntos *fuzzy*.

A classificação de registros hidroacústico pode ser tratada pela XAI (*Explainable Artificial Intelligence* - Inteligência Artificial Explicável), que procura desenvolver sistemas de inteligência artificial que sejam mais compreensíveis para os usuários (Adadi and Berrada, 2018). Por este motivo, decidiu-se analisar alguns métodos de classificação baseados em regras *Fuzzy*, que ajudam a transformar dados que seriam mais crús em variáveis linguísticas qualificadas por termos linguísticos de melhor entendimento (Zadeh, 1965), o que facilita ao usuário dos modelos gerados pelos classificadores, assim como os resultados obtidos.

1.1 Objetivo

1.1.1 Objetivo geral

Desenvolver um modelo explicável de sistema de classificação de eco-registros de organismos marinhos, obtidos em cruzeiros de pesquisa pesqueira, com ecossondas e a captura destes através de amostragem com rede de pesca de arrasto de meia água.

1.1.2 Objetivos específicos

- Aprofundar o conhecimento sobre métodos de classificação baseados em regras *fuzzy*, analisando-os criticamente;
- Estudar outros métodos de classificação que atendam à característica de explicabilidade.
- Analisar os ecotipos, no que se refere a forma de caracterização, assim como critérios utilizados por especialistas para sua classificação;
- Organizar e tratar os dados hidroacústicos que serão utilizados pelo sistema de classificação;
- Determinar a acurácia dos resultados para descobrir o desempenho do classificador;
- Realizar a classificação com diferentes métodos, identificando os que demonstram melhores resultados.

1.2 Organização do Texto

No Capítulo 2 são apresentadas as fundamentações teóricas que foram necessárias para o desenvolvimento do trabalho, respondendo as seguintes questões:

- O que é classificação e o problema de classificação?
- O que são registros hidroacústicos, quais são suas características e com são usualmente classificados?
- O que são árvores de decisão, como podem ser utilizadas na classificação e o algoritmo C4.5 que as utiliza?
- O que é a logica *fuzzy*, o que são regras *fuzzy*, como funciona um sistema de classificação baseado em regras *fuzzy* e os algoritmos FURIA e FARC-HD que aplicam essas teorias?

No Capítulo 3 é descrita a metodologia utilizada na realização do trabalho, explicando como foram coletados os dados, o processo desenvolvido para o pré processamento dos dados antes das classificações e a estratégia tomada para se classificar os dados. Em seguida, no Capítulo 4, são feitas as análises dos resultados obtidos, apresentando-se os modelos gerados pelos algoritmos utilizados, avaliando o desempenho deles e interpretando esses resultados. Por último, no Capítulo 5 se conclui o trabalho, descrevendo as contribuições geradas e os possíveis trabalhos que podem vir a ser feitos com base no que foi realizado.

2 FUNDAMENTAÇÃO TEÓRICA

Esta seção tem como objetivos explicar conceitos teóricos necessário para o entendimento do resto dos assuntos abordados nesta dissertação. Os tópicos discutidos nesta seção são: O que são registros acústicos de organismos marinhos e como é feita a sua classificação; o que é uma classificação e um sistema baseado nela; o que é e como são feitas classificações utilizando uma árvore de decisões e como funciona o algoritmo que a utiliza, o C4.5; o que é um sistema de classificação baseado em regras *fuzzy*, explicando os números e a lógica *fuzzy*, e de que maneira são utilizados em um regra *fuzzy*, além de explicar o funcionamento dos algoritmos baseados nestes conceitos e utilizados neste trabalho, o FARC-HD e o FURIA.

2.1 Registros acústicos de organismos marinhos e classificação de eco-registros

Registros acústicos de organismos marinhos são obtidos com uso de ecossondas científicas, que são instrumentos que utilizam os princípios da acústica, como o comportamento das ondas sonoras na água, para detectar peixes ou outros objetos na coluna de água, no oceano ou em outras massas de água (Madureira et al., 2015) (Madureira, 2004). As ecossondas funcionam pela emissão de um sinal elétrico que é transformado por um transdutor num pulso acústico, que é dirigido para baixo da embarcação. Quando este pulso atinge algum objeto na coluna de água, parte da energia acústica é reflectida e recebida pelo transdutor sob a forma de um eco, reconvertido em energia elétrica (Calazans and Griep, 2015). O tempo que medeia entre a emissão do pulso e a recepção do eco, conhecida a velocidade do som na água, fornece a distância a que o objeto se encontra do transdutor. O sinal acústico é então apresentado ao operador pela ecossonda com um registro visual ou armazenado de forma digital. Este registro é chamado de ecograma, que corresponde à representação gráfica dos dados detectados pela sonda na coluna de água, incluindo ou não o fundo (Madureira, 2004) (Weigert and Madureira, 2011).

Qualquer estrutura graficamente registrada nos ecogramas é chamada de eco-registro, consequentemente podem existir vários eco-registros em cada ecograma. As estruturas identificadas e definidas a partir de um conjunto de eco-registros que apresentam características comuns, e que apresentam padrões morfolologicamente consistentes, com base nos descritores utilizados, é chamado de ecotipo (Madureira, 2004) (Soares et al., 2005) (Weigert and Madureira, 2011).

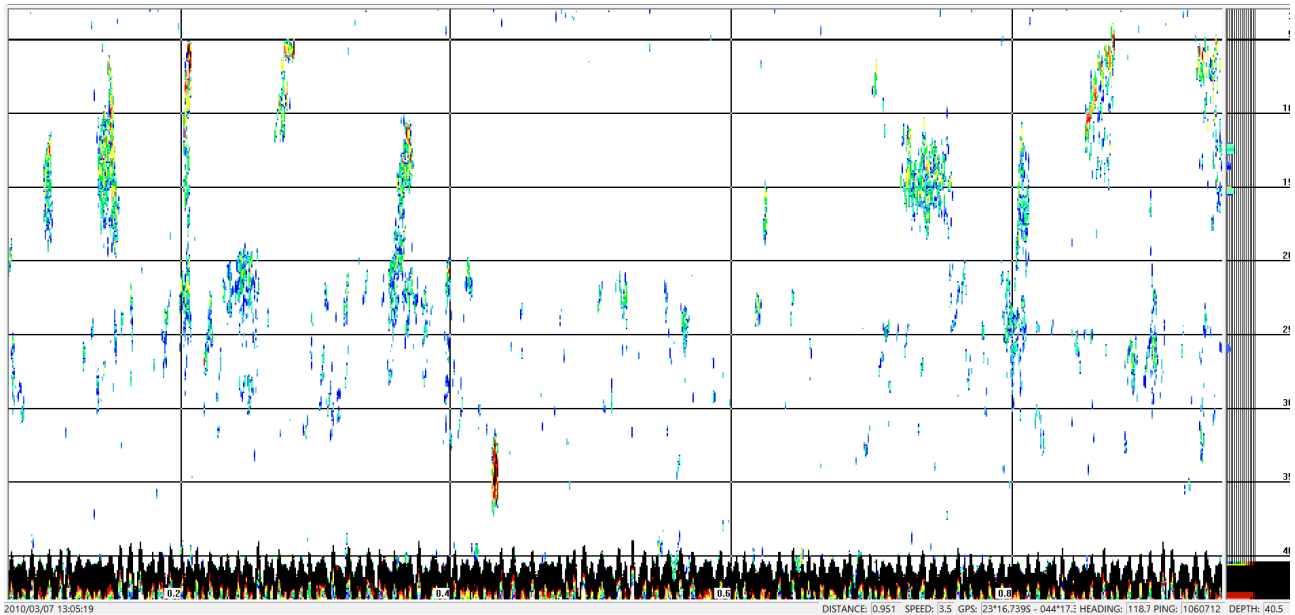


Figura 2.1: Exemplo de ecograma com ecoregistros. (Fonte: Ecograma do cruzeiro ECOSAR VI, realizados com a ecossonda SIMRAD EK500, em 2009)

Normalmente os eco-registros são caracterizados pelos seguintes descritores (Soares et al., 2005):

- Os descritores energéticos (densidade do eco-registro do padrão de cores das imagens, que representam graficamente o parâmetro Sv);
- Os descritores morfológicos (tipo de ecotraço, sendo a forma, extensão horizontal e vertical dele, e grau de agrupamento, como cardumes, camadas, dispersos e compactos);
- Os descritores espaciais (localização geográfica e de profundidade do eco-registro);
- Os descritores temporais (data e hora em que o eco-registro foi detectado);
- Os descritores biológicos (representatividade das capturas, ocorrência e percentual de ocorrência das espécies em peso, número nas capturas e sua frequência de ocorrência).

A definição de ecotipos é um problema de classificação de eco-registros, sendo que para validar os ecotipos identificados é necessária a obtenção da captura por pesca de amostras dos organismos, presentes no momento em que o ecograma está sendo obtido. Alguns descritores dos eco-registros podem ser inconsistentes ou apresentar um grande variação, como por exemplo os morfológicos e os biológicos, devido a modificações na propagação do som por alterações na temperatura da água e/ou migrações verticais dos peixes, suas presas e outros organismos presentes na coluna d'água (Weigert and Madureira, 2011) (Calazans and Griep, 2015). Por causa destas característica a utilização da lógica *fuzzy* foi vista como uma boa opção para poder ser feita a classificação de ecotipos diferente da lógica clássica que reconhece dois valores decisórios, verdadeiro ou falso, a lógica *fuzzy* (“difusa”) aceita múltiplos valores. Isto permite

a utilização em algoritmos de dados imprecisos ou incertos e de informação vaga e ambígua, favorecendo o processo de tomada de decisão.

A classificação mais exata de ecotipos é importante pois possibilita a análise da vida marinha sem utilizar métodos invasivos (Weigert and Madureira, 2011), podendo assim estudar as populações sem causar danos á elas. Ela também pode ser utilizada na pesca de espécies específicas, aumentando as capturas por tempo de arrasto, bem como pode reduzir a pesca de outros organismos que não possuem valor comercial e normalmente seriam descartados, preservando assim diversas especies (Madureira et al., 2015).

Nas próximas seções explicamos o problema da classificação de objetos e sobre o sistema de classificação baseado em regras fuzzy.

2.2 Classificação

Segundo (Lucca, 2018), um problema de classificação é uma situação na qual é necessário prever um valor de uma variável categórica de um objeto baseando nas informações mensuráveis do mesmo. Para começar a trabalhar com um problema de classificação é necessário definir um critério de decisão, chamado de modelo ou classificador, e para isso são necessários exemplos corretamente classificados, conhecidos como conjunto de treinamento, no qual cada exemplo $e \in E$ é descrito pela valor de N aspectos (também chamados de variáveis, caraterísticas) $X(e) = (e_1, \dots, e_N)$. O processo de aprendizado indutivo extrai o modelo da informações contidas no conjunto de treinamento para ser capaz de classificar novos exemplos nas classes predefinidas conhecidas, $C_k \in C = (C_1, \dots, C_M)$, aonde M é o número de classes do problema.

O objetivo é construir um modelo, $D : X(e) \longrightarrow C$, o que seria o suficiente para prever a classe dos exemplos tendo uma taxa de erro o mais baixo possível. Para medir a qualidade do treino, é necessário utilizar dados classificados que não foram usados na etapa de treino, conhecido como dados de teste. Então a previsão dos dados de teste é feita pelo classificador aprendido, o que também pode ser usado nos dados de treinamento para checar se o modelo apresenta uma boa capacidade geral. Demonstramos um exemplo de um problema de classificação desde o ponto de vista supervisionado na Figura 2.2.

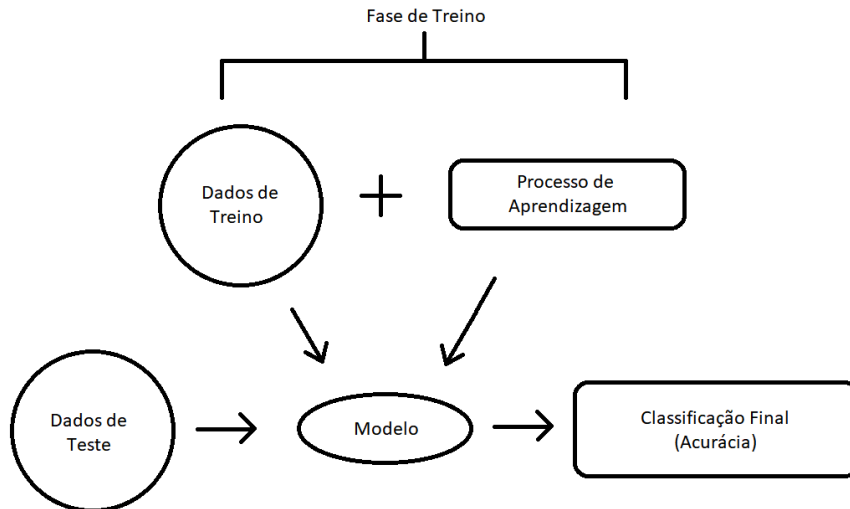


Figura 2.2: Método de Aprendizagem de um modelo supervisionado. (Adaptado de Lucca (2018))

2.3 Classificação Baseada em Árvores de Decisão

Uma técnica frequentemente utilizada para classificação de objetos é chamada árvore de decisão. Esta técnica é usada, por exemplo, para descobrir a que classe um caso (objeto) pode pertencer, levando em conta os valores de atributos associados a este caso. Segundo (Ruggieri, 2002), uma árvore de decisão é uma estrutura de dados árvore que consiste em nó de decisão e folhas: uma folha especifica o valor da classe; já um nó de decisão o teste sobre um dos atributos, chamado de atributo selecionado no nó. Para cada possível saída do teste existe um nó filho. Ou seja, para cada atributo A existem h saídas $A = d_1, \dots, A = d_h$, aonde $d_1, \dots, d_h$ são os valores conhecidos do atributo A . O teste em um atributo contínuo tem duas possíveis saídas, $A \leq t$ e $A > t$, onde t é o valor determinado no nó e é chamado de limite.

Uma árvore de decisão é usada para classificar um caso, por exemplo, para descobrir o valor de uma classe para um caso levando em conta o valor dos atributo daquele caso. O caminho desde a raiz até uma folha é a classe prevista pela árvore de decisão. Uma medida de performance sobre um conjunto de casos em uma árvore de decisão é chamado erro de classificação e é definido pela porcentagem de casos que foram classificados erroneamente, aonde foi predito uma classe que difere da real em um caso.

2.3.1 Algoritmo C4.5

O algoritmo C4.5 (Quinlan, 2014) é empregado para construir uma árvore de decisão utilizando a estratégia divisão e conquista, aonde cada nó da árvore é associado com um conjunto

de caso, então pesos são atribuídos a estes para levar em conta valores desconhecidos de algum atributo. No início só se tem a raiz que está associada com todo o conjunto de treinamento TS e cada peso de um caso é igual a 1.0. À cada nó, o algoritmo abaixo (Ruggieri, 2002) de divisão e conquista é executado para se encontrar a melhor escolha local, sem levar em conta resultados anteriores.

Programa 1: Pseudo-Código do algoritmo de construção de árvore C4.5

```

FormTree( $T$ )

1. ComputeClassFrequency;

2. if OneClass or FewCase
    return a leaf;
    create a decision node  $N$ ;

3. ForEach Attribute  $A$ 
    ComputeGain( $A$ );

4.  $N.test = \text{AttributeWithBestGain}$ ;

5. if  $N.test$  is continuous
    find Treshold

6. ForEach  $T'$  in the splitting of  $T$ 

7. if  $T'$  is Empty
    Child of  $N$  is a leaf
else

8.    Child of  $N = \text{FormTree}(T')$ ;

9. ComputeErrors of  $N$ ;

    return  $N$ 

```

Sendo T o conjunto de casos associados a um nó, a frequência ponderada $freq(C_i, T)$ é computada (Etapa 1) para casos em T onde a classe é C_i para $i \in [1, NClass]$. Se todos os casos (Etapa 2) em T pertencem a mesma classe C_j (ou o número de casos em T é menor que um certo valor), então o nó é a folha, com a classe associada C_j (respectivamente, a classe com maior frequência). O erro de classificação da folha é a soma ponderada dos casos em T que não são da classe C_j .

Se T contém casos que pertencem a duas ou mais classes (Etapa 3), então o ganho de informação de cada atributo é calculado. Para atributos descritivos, o ganho de informação é relativo aos casos de divisão em T em conjuntos com valores de atributos distintos. Para atributos contínuos, o ganho de informação é relacionado à divisão de T em dois subconjuntos, por exemplo, casos com um valor de atributo “menor do que” e casos com um “maior do que” um certo limite local, que é calculado durante o ganho de informação.

O atributo com o maior ganho de informação (Etapa 4) é selecionado para o teste no nó. Também, no caso de um atributo contínuo ser selecionado, o limite é computado (Etapa 5) de acordo com o maior valor do conjunto de todo o treinamento que esteja abaixo do limite local. Um nó de decisão tem s descendentes se $T_1, \dots, T_s$ são os conjuntos da divisão produzida pelo teste nos atributos selecionados (Etapa 6), aonde $s = 2$ quando os atributos selecionados são contínuos e $s = h$ para atributos discretos com um valor h conhecido. Para $i = [1, s]$, se T_i é vazio, (Etapa 7) o nó descendente é diretamente feito uma folha, sendo sua classe associada a classe mais frequente no nó pai e erro de classificação igual a 0.

Se $T - i$ não está vazio, a abordagem de divisão e conquista consiste em aplicar as mesmas operações recursivamente (Etapa 8) no conjunto que consiste em T_1 mais os casos em T com um valor desconhecido do atributo selecionado. Nota-se que neste ultimo caso se replica eles em cada um dos descendentes com seus pesos proporcionais à proporção de casos em T_i sobre os casos em T com um valor conhecido do atributo selecionado.

Por último, o erro de classificação (Etapa 9) do nó é calculado como a soma dos erros nos nós descendentes. Se o resultado é maior que o erro ao classificar todos os casos em T como pertencentes a classe mais frequente em T , então o nó é feito uma folha e todas suas sub-árvores são removidas.

2.3.1.1. Ganho de Informação

O ganho de informação que um atributo a tem para um conjunto de casos T é calculado pelo seguinte: Se a é discreto, e $T_1, \dots, T_s$ são os subconjuntos de T formado por casos com valores conhecidos distintos para o atributo a , então:

$$gain = info(T) - \sum_{i=1}^s \frac{|T_i|}{|T|} \times info(T_i), \quad (2.1)$$

aonde

$$info(T) = - \sum_{j=1}^{NClass} \frac{freq(C_j, T)}{|T|} \times \log_2 \left(\frac{freq(C_j, T)}{|T|} \right) \quad (2.2)$$

é a função da entropia. Mesmo tendo uma opção do ganho de informação, por padrão, o C4.5 considera o uma taxa do ganho de informação da divisão $T_1, \dots, T_s$, que é a taxa do ganho de informação para a sua divisão de informação:

$$Split(T) = - \sum_{i=1}^s \frac{|T_i|}{|T|} \times \log_2 \left(\frac{|T_i|}{|T|} \right). \quad (2.3)$$

Pode-se observar que se um atributo discreto foi selecionado em um nó anterior, o seu ganho e sua taxa de ganho são zero. Então, o C4.5 nem computa esse ganho de informação daqueles atributos. Se a é um atributo contínuo, casos em T com um valor de atributo conhecido são primeiramente ordenados utilizando o algoritmo de ordenação *Quicksort*. Assume-se que os valores ordenados são $v_1, \dots, v_m$. Considere para $i \in [1, m-1]$ o valor $v = (v_i + v_{i+1})$ e a divisão:

$$T_1^v = \{v_j | v_j \leq v\}; T_2^v = \{v_j | v_j > v\} \quad (2.4)$$

Para cada valor v , o ganho de informação $gain_v$ é computado levando em conta a divisão acima. O valores v' aonde $gain_{v'}$ é máximo é o limite máximo local e o ganho de informação para o atributo a é definido como $gain_{v'}$. Por padrão, de novo, o C4.5 considera a taxa de ganho de informação da divisão $T_1^{v'}, T_2^{v'}$. Finalmente, nota-se que, no caso de que o atributo seja selecionado no nó, o limite é calculado (Etapa 5) através de uma busca linear em todo o conjunto de treinamento TS do valor do atributo que melhor se aproxima do limite local v' por baixo (ou seja, não é maior que v'). Esse valor é feito o limite do nó.

Já que a árvore de decisão construída pode ser grande e sofrer problemas de *fitting*, o sistema C4.5 oferece uma árvore simplificada obtida cortando caminhos de acordo com um nível de confiança. Ambas as árvores de decisão e sua versão simplificada são avaliadas pela porcentagem de casos classificados erroneamente por elas. Também, essa avaliação pode ser feita sobre um conjunto de teste, que não foi utilizado durante a construção da árvore.

2.3.1.2. Número mínimo de instâncias por folha

Este parâmetro têm grande influencia no resultado. Como é dito em (Bonini, 2016), valores altos desse parâmetro resultam em árvores mais genéricas, menores e com menos atributos sendo analisados para a classificação das amostras. Com valores próximos de um, as árvores geradas são mais precisas e mais atributos são analisados para classificar a amostra. Neste trabalho foi escolhido o valor 10 pois valores muito pequenos criavam árvores muito extensas que reutilizavam o mesmo atributo várias vezes para a decisão

2.4 Lógica *Fuzzy*

A teoria dos conjuntos *fuzzy* foi introduzida em 1965 pelo matemático Lotfi Asker Zadeh (Zadeh, 1965) com o objetivo de dar um tratamento matemático a certos termos linguísticos subjetivos (“aproximadamente”, “em torno de”, etc.) para então poder armazenar e programar conceitos vagos em computadores e assim calcular utilizando com informações imprecisas como, por exemplo, a satisfação de uma pessoa ao realizar alguma tarefa.

Para a formalização matemática de um conjunto *fuzzy*, baseia-se no fato que um conjunto clássico pode ser caracterizado pela sua função característica (Zadeh, 1965), que pode ser definido como:

Definição 1.1. Seja U um conjunto e A um subconjunto de U . A *função característica* de A é dada por:

$$X_A(x) = \begin{cases} 1 & \text{se } x \in A \\ 0 & \text{se } x \notin A \end{cases} \quad (2.5)$$

Desta maneira X_A é uma função com domínio U e a imagem esta contida no conjunto $\{0, 1\}$, sendo que $X_A(x) = 1$ indica que x esta em A e $X_A(x) = 0$ indica que não esta em A . Assim, a função característica indica quais elementos do universo U pertencem no subconjunto A . No entanto, existem casos em que a pertinência entre elementos e conjuntos não é precisa, isto e, não sabemos dizer se um elemento pertence efetivamente a um conjunto ou não. É mais plausível dizer qual elemento do conjunto universo se enquadra “melhor” ao termo que caracteriza o subconjunto. Por exemplo, consideremos o subconjunto dos números reais “próximos de 2”:

$$A = \{x \in \mathbb{R} : x \text{ é próximo de } 2\} \quad (2.6)$$

se compararmos 7 e 2001 em relação a distancia a 2, é possível afirmar que 7 esta mais próximo que 2001 mas não se pode afirmar com certeza se ambos estão contido no subconjunto A dos números próximos de 2. Pra poder se definir isso é necessário a formalização dos subconjuntos *fuzzy*.

A definição de subconjunto *fuzzy* F é obtida simplesmente ampliando-se no contra-domínio da função característica, que é o conjunto $0, 1$, para o intervalo $[0, 1]$. Pode se dizer que um conjunto clássico é um caso particular de um conjunto *fuzzy*, cuja função de pertinência φ_F é sua função característica χ_F . Um subconjunto clássico costuma ser denominado por subconjunto *crisp*. Um subconjunto *fuzzy* F é composto de elementos x de um conjunto clássico $U \rightarrow [0, 1]$, providos de um valor de pertinência a F , dado por $\varphi_F(x)$. Podemos dizer que um subconjunto *fuzzy* F de U é dado por um conjunto clássico de pares ordenados:

$$F = \{(x, \varphi_F(x)), \text{ com } x \in U\} \quad (2.7)$$

O subconjunto clássico de U definido por

$$\text{supp } F = \{(x \in U : \varphi_F(x) > 0)\} \quad (2.8)$$

é denominado suporte de F e tem papel fundamental na inter-relação entre teorias de conjuntos clássica e *fuzzy*.

Abaixo nós temos a representação gráfica de uma função característica clássica (Figura 2.3) e *fuzzy* (Figura 2.4).

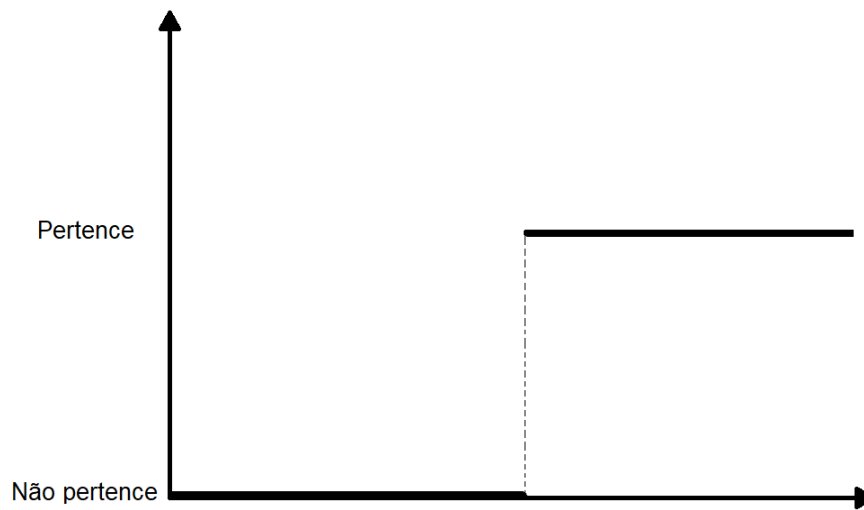


Figura 2.3: Gráfico de uma função característica clássica. (Adaptado de Lucca (2018))

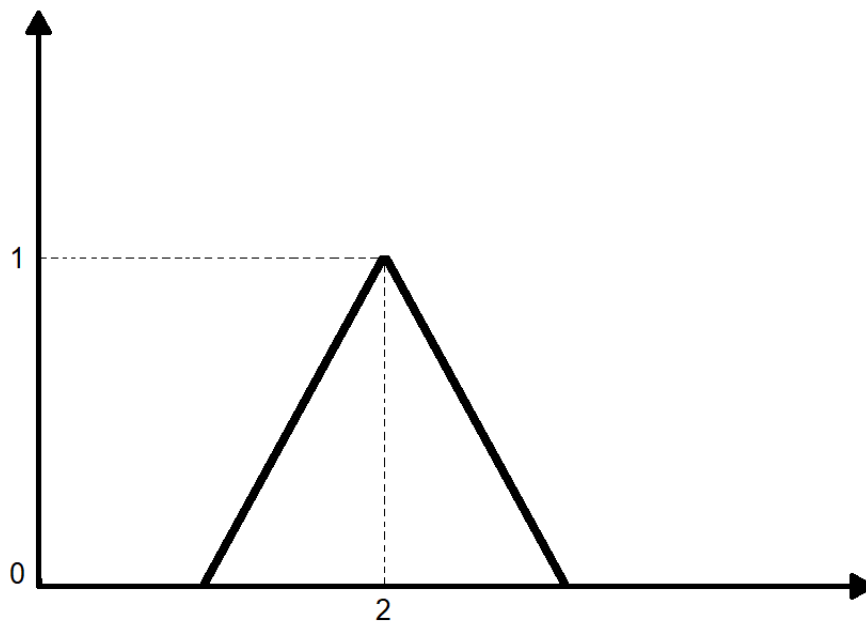


Figura 2.4: Gráfico de uma função característica *fuzzy* para números próximos de 2. (Adaptado de Lucca (2018))

2.4.1 Regra *Fuzzy*

A regra *fuzzy* é utilizada da seguinte maneira:

Regra R_j : Se x_1 é A_{j_1} e ... e x_n é A_{j_n} então $Classe = C_j$ com RW_j

aonde R_j é o rótulo da j -ésima regra, $x = (x_1, \dots, x_n)$ é um vetor de exemplos de dimensão n , A_{j_i} é um conjunto *fuzzy* antecedente representando um termo linguístico, que é uma ou mais palavras que descrevem o exemplo (i.e. pequeno, médio, grande), C_j é um rótulo de classe, e RW_j é o peso da regra, o que pode ser interpretado como a força que cada regra possui (Ishibuchi and Nakashima, 2001).

Já o Método de Raciocínio *Fuzzy* (MRF) (Cordón et al., 1999) é feito da seguinte maneira: Sendo $x_p = (x_{p1}, \dots, x_{pn})$ um novo exemplo a ser classificado, L o número de regras no Banco de Regras e M o número de classes do problema, as etapas MRF são as seguintes:

1. O *grau de correspondência*, que é a intensidade de ativação da parte “Se” de todas as regras no banco de regras com o exemplo X_p . É utilizada uma t-norma para ser computado.

$$\mu_{A_j}(x_p) = T(\mu_{A_{j1}}(x_{p1}), \dots, \mu_{A_{jn}}(x_{pn})), \quad j = 1, \dots, L \quad (2.9)$$

2. O *grau de associação*, que é o grau em que o exemplo x_p com classe de cada regra no banco de regras.

$$b_j^k = \mu_{A_j}(x_p) \times RW_j^k \quad k = Classe(R_j), \quad j = 1, \dots, L \quad (2.10)$$

3. O *grau de solidez da classificação do exemplo para todas as classes*, onde é usada uma função de agregação, que é uma função que combina as suas entradas que tipicamente são interpretadas como graus de pertinência em conjuntos *fuzzy* (Beliakov et al., 2007), que combina os graus de associação positivos da etapa anterior.

$$Y_k = M(b_j^k | j = 1, \dots, L \text{ e } b_j^k > 0), \quad k = 1, \dots, M \quad (2.11)$$

4. E por último a *classificação*, aonde se aplica função de decisão F sobre o grau de solidez da classificação do exemplo para todas as classes. Essa função determina a classe correspondente ao valor máximo.

$$F(Y_1, \dots, Y_M) = \arg \max_{k=1, \dots, M} (Y_k) \quad (2.12)$$

2.4.2 Sistemas de Classificação Baseados em Regras *Fuzzy*

Segundo Ishibuchi (2009), métodos de classificação baseado em Regras *Fuzzy* são bastante utilizados na mineração de dados, pois permitem a utilizar toda informação disponível no modelo do sistema, ou seja, conhecimento profissional, medidas empíricas ou modelos matemáticos. Algumas de suas vantagens é gerar um modelo interpretável e assim permitir a representação do conhecimento e ser compreensível para o usuário do sistema. Existem 2 principais componentes de um Sistema de Classificação Baseado em Regras *Fuzzy*: a Base de Conhecimento, que é composta da Base de Regras e da Base de Dados, nas quais as regras e as funções de pertinência são guardadas, respectivamente; e o Método de Raciocínio *Fuzzy*, que é o mecanismo utilizado para classificar os exemplos utilizando as informações na Base de conhecimento. Mostramos na Figura 2.5, o esquema de um FRBCS (Sistemas de Classificação Baseados em Regras *Fuzzy*).

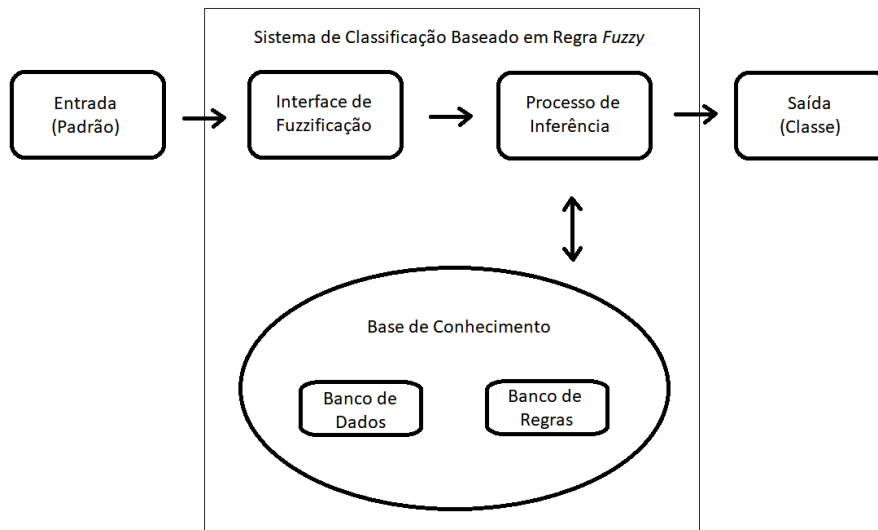


Figura 2.5: Estrutura de um Sistema de Classificação Baseado em Regras *Fuzzy*. (Adaptado de Lucca (2018))

2.4.3 FARC-HD

O FARC-HD (*Fuzzy Association Rule-based Classification method for High-Dimensional problems* - Método de classificação baseada em regras de associação *fuzzy* para problemas de grandes dimensões) é apresentado pelos autores (Alcala-Fdez et al., 2011) com o objetivo de se obter um classificador baseado em regra *fuzzy* compacto, preciso e com baixo custo computacional.

Primeiramente, é necessário explicar o que são regras de associação *fuzzy*. Elas são usadas para a representação e identificação de dependências entre itens em um banco de dados (Zhang and Zhang, 2002). Elas são expressões do tipo $A \rightarrow B$ aonde A e B são conjuntos de itens e $A \cap B = \emptyset$. Isso significa que, se todos os itens em A existem em uma transação então todos os itens de B têm uma grande probabilidade de também estarem presentes na mesma, e A e B não podem possuir itens em comum. O uso de conjuntos *fuzzy* para descrever associações

entre dados estende os tipos de relacionamentos que podem ser representados, facilitando a interpretação de regras em termos linguísticos, e evita limites não-naturais no particionamento do domínios de atributos.

Considerando um simples banco de dados T com dois atributos (X_1 e X_2) e três termos linguísticos (Pequeno, Médio e Alto) com suas funções de pertinências associadas. Baseado nesta definição, um simples exemplo de regra de associação *fuzzy* é $X_1 \text{ é Médio} \rightarrow X_2 \text{ é Alto}$.

Suporte e confiança são as medidas mais comuns de interesse de uma regra de associação. Essas medidas podem ser definidas para as regras de associação *fuzzy* da seguinte maneira:

$$Support(A \rightarrow B) = \frac{\sum_{x_p \in T} \mu_{AB}(x_p)}{|N|} \quad (2.13)$$

$$Confidence(A \rightarrow B) = \frac{\sum_{x_p \in T} \mu_{AB}(x_p)}{\sum_{x_p \in T} \mu_A(x_p)} \quad (2.14)$$

onde $|N|$ é o número de transações em T , $\mu_A(x_p)$ é o grau de combinação da transação x_p com a parte do antecedente da regra e $\mu_{AB}(x_p)$ é o grau de combinação da transação x_p com o antecedente e consequência da regra.

Uma regra de associação *fuzzy* pode ser considerada uma regra de classificação se o antecedente contém conjuntos de itens *fuzzy* e a parte consequente contém somente um rótulo de classe ($C = C_1, \dots, C_j, \dots, C_S$). Uma regra *fuzzy* de classificação associativa $A \rightarrow C_j$ pode ser medida diretamente em termos de suporte e confiança:

$$Support(A \rightarrow C_j) = \frac{\sum_{x_p \in Class C_j} \mu_A(x_p)}{|N|} \quad (2.15)$$

$$Confidence(A \rightarrow C_j) = \frac{\sum_{x_p \in Class C_j} \mu_A(x_p)}{\sum_{x_p \in T} \mu_A(x_p)} \quad (2.16)$$

O autor divide o FARC-HD em três etapas:

1. Extração da regra de associação *fuzzy* para a classificação: Uma árvore de busca é usada para listar todos os conjuntos de itens *fuzzy* possíveis frequentes e para gerar regras de associação *fuzzy* para a classificação, limitando a profundidade de cada ramo com o objetivo de achar um número pequeno, e por consequência mais simples, de regras *fuzzy*.
2. Prévia das regras candidatas: Mesmo sabendo que a ordem das associações é limitada na extração das regras de associação, o número de regras geradas pode ser muito grande. Para se diminuir o custo computacional da etapa genética de pós-processamento foi considerado o uso da descoberta de subgrupos baseado em uma medida de Acurácia Média Ponderada

melhorada (wWRAcc') para pré-selecionar as regras interessam mais por meio de um esquema de ponderamento de padrões (Kavšek and Lavrač, 2006).

3. Seleção de regras genéticas e afinamento lateral: Por último, são utilizados algoritmos genéticos para selecionar e afinar um conjunto compacto de regras de associação *fuzzy* com alta acurácia de classificação com o objetivo de considerar a sinergia positiva conhecida que ambas técnicas apresentam (seleção e afinamento).

Na primeira etapa, para que o banco de regras seja gerado, é utilizada uma árvore de busca para listar todos os conjuntos de itens *fuzzy* de uma classe. A raiz da árvore é um conjunto vazio, se assume que todos os atributos têm uma ordem, e os conjuntos de um item que correspondem aos atributos são listados no primeiro nível da árvore de acordo com sua ordem. Se um atributo tem j possíveis resultados (q_j termos linguísticos para cada atributo quantitativo), ele vai ter j conjuntos de um item listados no seu primeiro nível. Os descendentes de um nó com um item para um atributo A são os conjuntos de dois itens que contêm o conjunto de um item do atributo A para outro atributo antes do A na ordem, e assim em diante. Se um atributo tem $j > 2$ possíveis resultados, ele pode ser substituído por j variáveis binárias para se ter certeza que não mais de um desses atributos binários j aparecerão no mesmo nó de um árvore de busca.

Um conjunto de itens com um suporte maior do que o suporte mínimo é um conjunto de itens frequente. Se o suporte de um conjunto com n itens em um nó J é menor que o suporte mínimo, ele não precisa ser estendido mais pois o suporte de qualquer conjunto de itens em um nó na sub-árvore proveniente do nó J também será menor que o suporte mínimo. Da mesma maneira, se um conjunto de itens candidato gerar uma regra de classificação com uma confiança maior do que a confiança máxima, essa regra alcançou o nível de qualidade demandado pelo usuário e novamente é desnecessário estendê-la ainda mais. Levar em conta essas propriedades reduz bastante o número de nós necessários para a pesquisa.

O suporte *fuzzy* de um conjunto de itens pode ser calculado como:

$$Support(A) = \frac{\sum_{x_p \in T} \mu_A(x_p)}{|N|} \quad (2.17)$$

onde $\mu_A(x_p)$ é o grau de combinação de um padrão x_p com o conjunto de itens. Este grau de combinação para diferentes regiões *fuzzy* é computado utilizando um operador de conjunção, que neste caso é uma T-norma.

No momento em que todos os conjuntos de itens *fuzzy* frequentes são obtidos, as regras de associação *fuzzy* candidatas para a classificação podem ser geradas, colocando os conjuntos de itens *fuzzy* no antecedente das regras e as correspondentes classes após. Este processo é repetido

para cada classe. O número de conjuntos de itens *fuzzy* extraídos depende diretamente do suporte mínimo. Ele normalmente é calculado considerando o número total de padrões que cada classe em um conjunto de dados, mas o número de padrões para cada classe pode ser diferente. Por causa disso, este algoritmo determina o suporte mínimo para cada classe pela distribuição delas no conjunto de dados. Por isso, o suporte mínimo para a classe C_j é definido por:

$$MinimumSupport_{C_j} = minSup * f_{C_j} \quad (2.18)$$

onde $minSup$ é o suporte mínimo determinado pelo especialista e f_{C_j} é a taxa de padrões da classe C_j .

Nesta etapa foi gerada uma grande quantidade de regras de associação *fuzzy* candidatas para a classificação. Isto é, no entanto, muito difícil para usuários humanos manejarem este grande volume regras *fuzzy* geradas e intuitivamente entender longas regras com diversas condições antecedentes. Então a profundidade limite das árvores são limitadas por um valor fixo ($Depth_{max}$), determinado pelo especialista.

Para diminuir o custo computacional da etapa 3, na segunda etapa (a prévia das regras candidatas) é utilizada a descoberta de subgrupos para pré-selecionar as regras mais interessantes do banco de regras obtido na etapa anterior por meio do esquema de ponderamento de padrões Kavšek and Lavrač (2006). Este esquema trata os padrões de maneira que os positivos cobertos não são deletados quando a melhor regra atual é selecionada. Ao invés disso, cada vez que uma regra é selecionada, o algoritmo guarda um contador i para cada padrão de quantas vezes, e com quantas das regras selecionadas, o padrão foi coberto.

Os pesos dos padrões positivos cobertos pela regra seleciona diminuem de acordo com a formula

$$w(e_j, i) = \frac{1}{i + 1}. \quad (2.19)$$

Na primeira iteração, todos os padrões da classe alvo são atribuídos com um mesmo peso $w(e_j, 0) = 1$, por enquanto que as iterações seguintes a contribuição dos padrões são inversamente proporcionais à sua cobertura pelas regras previamente selecionadas. Desta maneira os padrões que já foram cobertos por uma ou mais regras selecionados diminuem seus pesos por enquanto que os padrões da classe alvo que não foram encontrados os quais os pesos não foram diminuídos terão uma maior chance de ser cobertos nas iterações seguintes. Padrões cobertos são completamente eliminados quando eles foram cobertos mais de k_t vezes.

Então a cada iteração do processo as regras são ordenadas de acordo com o critério de avaliação de melhor a pior. A melhor regra é selecionada, padrões cobertos têm seus pesos recalculados, e o procedimento repete essas etapas até um dos critérios de parada sejam satisfeitos: ou todos os padrões foram cobertos mais de k_t vezes ou não há mais regras no banco de

regras. Este processo é repetido para cada classe.

$wWRAcc'$ foi utilizado para avaliar a qualidade das regras intervalar em APRIORI-SD (Kavšek and Lavrač, 2006). Essa medida é definida pelo seguinte:

$$wWRAcc'(A \rightarrow C_j) = \frac{n'(A)}{N'} \cdot \left(\frac{n'(A \cdot C_j)}{n'(A)} - \frac{n(C_j)}{N} \right) \quad (2.20)$$

onde N' é a soma dos pesos de todos os padrões, $n'(A)$ é a soma dos pesos de todos os padrões cobertos, $n'(A \cdot C_j)$ é a soma dos pesos de todos os padrões corretamente cobertos, $n(C_j)$ é o número de padrões da classe C_j , e N o número de padrões no total. Esta medida foi modificada pelo autor do FURIA para levar em conta as regras *fuzzy*. A nova medida ficou definida como:

$$wWRAcc''(A \rightarrow C_j) = \frac{n''(A \cdot C_j)}{n'(C_j)} \cdot \left(\frac{n''(A \cdot C_j)}{n''(A)} - \frac{n(C_j)}{N} \right) \quad (2.21)$$

onde $n''(A)$ é a soma do produto de todos os pesos para todos os padrões cobertos pelos graus de combinação (com a parte antecedente da regra), $n''(A \cdot C_j)$ é a soma do produto de todos os pesos dos padrões corretamente cobertos por seus graus de combinação com a parte antecedente das regras, e $n'(C_j)$ é a soma dos pesos dos padrões de C_j . Além disso, o primeiro termo na definição do $wWRAcc'$ foi substituído por $\frac{n''(A \cdot C_j)}{n'(C_j)}$ para recompensar regras que cobrem padrões que não foram eliminados da classe C_j .

Na terceira etapa, foi considerada a utilização de algoritmos genéticos para selecionar e afinar um conjunto compacto de regras de associação *fuzzy* com alta acurácia de classificação obtidas na etapa anterior. Foi considerada a abordagem proposta em Alcalá et al. (2007) aonde as regras são baseadas na representação em 2-tupla linguística (Herrera and Martínez, 2000). Essa representação permite o deslocamento lateral dos rótulos considerando somente um parâmetro, o de tradução simbólica, o qual envolve a simplificação do espaço de afinamento de pesquisa que deixa mais fácil a derivação de modelos otimizados, em particular quando combinado com a seleção de regras, no mesmo processo fazendo possível ela tirar vantagem da sinergia positiva de ambas técnicas presentes.

2.4.4 *Fuzzy Unordered Rule Induction Algorithm* - FURIA

Este algoritmo é uma modificação e extensão do que faz aprendizagem de regras RIPPER (Cohen, 1995), onde o FURIA (Hühn and Hüllermeier, 2009) aprende as regras *fuzzy* e conjuntos de regras não ordenadas ao invés das regras convencionais e das listas de regras. Além disso, para tratar exemplos que não foram incluídos, ele faz o uso de um método que amplia as regras.

A primeira modificação no RIPPER foi o tipo de modelo que é aprendido e o uso das regras padrão. aprender um lista de decisão e usar somente uma classe como predição padrão possui

algumas desvantagens, pois cria-se uma tendencia à classe padrão. Para contornar este problema, Boström (2004) propôs uma versão não ordenada do predecessor IREP (Fürnkranz and Widmer, 1994). Da mesma maneira, o FURIA propõe aprender um conjunto de regras para cada uma das classes, usando uma decomposição “um contra o resto”. Consequentemente, o programa aprende a separar cada uma das classes de todas as outras, o que significa que nenhuma classe padrão é utilizada e a ordem das classes é irrelevante.

Dois problemas podem acontecer na conexão entre a classificação e a nova questão da instância quando se utiliza um conjunto de regras não ordenado sem uma regra padrão: primeiramente, um conflito pode acontecer já que a instância é igualmente bem coberta por regras de diferentes classes, mas esse problema raramente acontece e pode ser facilmente resolvido. O segundo problema é que questão pode ser coberta por nenhuma das regras, que para ser resolvido foi proposto um método que amplia as regras.

O algoritmo RIPPER pode ser dividido nas fases de construção e de otimização. A construção das regras é feita utilizando o algoritmo IREP. No FURIA esta etapa foi omitida pois as estratégias de poda do IREP tinham uma influencia negativa na performance. Então, ao invés disso, é aprendido o conjunto de regras inicial de todos os dados de treinamento.

Na fase de otimização, a poda foi mantida, pois retirar-la não seria benéfico. O FURIA ainda aplica a poda quando são criadas as regras de reposição e de revisão. Nesta parte a estratégia original de poda ainda é aplicada, exceto no caso em que a poda tenta remover todos os antecedentes para uma regra, assim gerando um regra padrão. Neste caso, a poda será abortada e as regras não podadas são utilizadas na comparação do MDL (*Minimum Description Length* - Comprimento Mínimo de Descrição) (Quinlan, 1995) na fase de otimização. Estas estratégias são o suficiente para prevenir o *overfitting*, ou seja, ele não se torna ineficaz para a classificação de dados que não estejam no conjunto de treinamento, então a remoção da fase de poda que existia na parte do IREP nem tem impacto negativo na acurácia da classificação.

Um seletor relacionado a um atributo A_i (com domínio $\mathbb{D}_i = \mathbb{R}$) em uma regra RIPPER pode ser expressa na forma $(A_i \in I)$, aonde $I \subseteq \mathbb{R}$ é um intervalo: $I = (-\infty, v]$ se a regra contém um seletor $(A_i \leq v)$, $I = [u, +\infty)$ se contém um seletor $(A_i \geq u)$, e $I = [u, v]$ se contém ambos (neste ultimo caso os dois seletores são combinados).

Um regra *fuzzy* é obtido substituindo estes intervalos por intervalos *fuzzy*, no caso conjuntos *fuzzy* com função de pertinência trapezoidal.

Um intervalo *fuzzy* como este é especificado por quatro parâmetros e é escrito como $I^F = (\phi^{s,L}, \phi^{c,L}, \phi^{c,U}, \phi^{s,U})$:

$$I^F(v) \stackrel{df}{=} \begin{cases} 1 & \phi^{c,L} \leq v \leq \phi^{c,U} \\ \frac{v - \phi^{s,L}}{\phi^{c,L} - \phi^{s,L}} & \phi^{s,L} < v < \phi^{c,L} \\ \frac{\phi^{s,U} - v}{\phi^{s,U} - \phi^{c,U}} & \phi^{c,U} < v < \phi^{s,U} \\ 0 & \text{else} \end{cases} \quad (2.22)$$

$\phi^{c,L}$ e $\phi^{c,U}$ são, respectivamente, os limites superior e inferior do núcleo (elementos com pertinência 1) do conjunto *fuzzy*; da mesma maneira, $\phi^{s,L}$ e $\phi^{s,U}$ são, respectivamente, os limites superior e inferior do suporte (elementos com pertinência > 0) como é visto na Figura 2.6.

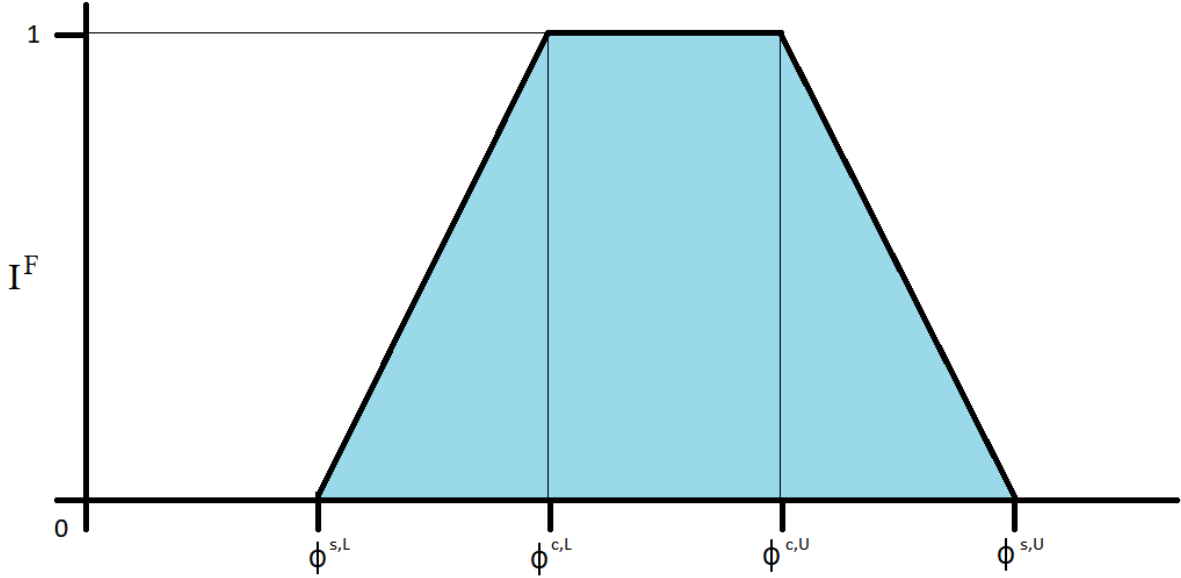


Figura 2.6: Um intervalo *fuzzy* I^F . (Adaptado de Hühn and Hüllermeier (2009))

Note que, como em um caso não-*fuzzy*, um intervalo *fuzzy* pode ser aberto para um lado ($\phi^{s,L} = \phi^{c,L} = -\infty$ ou $\phi^{c,U} = \phi^{s,U} = +\infty$). De fato, os antecedentes *fuzzy* que são sucessivamente aprendidos pelo FURIA são meio-intervalos *fuzzy* têm este comportamento.

Um seletor *fuzzy* ($A_i \in I_i^F$) cobre uma instancia $x = (x_1, \dots, x_n)$ ao grau $I_i^F(x_i)$. Um regra *fuzzy* r^F envolvendo k seletores ($A_i \in I_i^F$), $I = 1, \dots, K$, cobre x ao grau

$$\mu_{r^f}(x) = \prod_{i=1, \dots, k} I_i^F(x_i) \quad (2.23)$$

Para se obter as regras *fuzzy*, as regras finais geradas pelo algoritmo RIPPER modificado são “fuzzificadas”, ou seja, usando o conjunto de treinamento $D_T \subseteq \mathbb{D}$ para a avaliação dos candidatos, se procura a melhor extensão *fuzzy* para cada regra, aonde esta é entendida como uma regra de mesma estrutura mas com intervalos substituídos pelos *fuzzy*. Levando em conta os intervalos I_i das regras originais como os núcleos $[\phi^{c,L}, \phi^{c,U}]$ do intervalo *fuzzy* procurado I_i^F , o problema é encontrar os limites ideais para os respectivos suportes, ou seja, determinar $\phi^{s,L}$ e $\phi^{s,U}$.

Para a fuzzificação de um certo antecedente ($A_i \in I_i$) é importante considerar o único dado de treinamento relevante D_T^i , ou seja, ignorar instâncias que são excluídas por qualquer antecedente ($A_i \in I_j^F$), $j \neq i$:

$$D_T^i = \left\{ x = (x_1, \dots, x_k) \in D_T \mid I_j^F(x_j) > 0 \text{ para todo } j \neq i \right\} \subseteq D_T \quad (2.24)$$

Particiona-se D_T^i em um subconjunto de instâncias positivas, D_{T+}^i , e negativas, D_{T-}^i . Para medir a qualidade da fuzzificação, a pureza da regra é utilizada:

$$pur = \frac{p_i}{p_i + n_i}, \quad (2.25)$$

onde

$$p_i \stackrel{df}{=} \sum_{x \in D_{T+}^i} \mu_{A_i}(x) \quad (2.26)$$

$$n_i \stackrel{df}{=} \sum_{x \in D_{T-}^i} \mu_{A_i}(x) \quad (2.27)$$

Algorithm 1: Algoritmo de fuzzificação do antecedente para uma única regra r

```

Seja  $A$  o conjunto numérico de antecedentes de  $r$ ;
while  $A \neq \emptyset$  do
     $a_{max} \leftarrow null$   $a_{max}$  é o antecedente com maior pureza;
     $pur_{max} \leftarrow 0$   $pur_{max}$  é o maior valor de pureza até o momento;
    for  $i \leftarrow 1$  to  $size(A)$  do
        computar a melhor fuzzificação de  $A[i]$  em termos de pureza;
         $pur_{A[i]} \leftarrow$  a pureza da melhor fuzzificação;
        if  $pur_{A[i]} > pur_{max}$  then
             $pur_{max} \leftarrow pur_{A[i]}$ ;
             $a_{max} \leftarrow A[i]$ ;
        end
    end
     $A \leftarrow A \setminus a_{max}$ ;
    Atualiza  $r$  com  $a_{max}$ ;
end

```

As regras são fuzzificadas de maneira gulosa com mostra o Algoritmo 1 (Hühn and Hüllermeier, 2009). A cada iteração, uma fuzzificação é computada para cada antecedente, no caso a melhor fuzzificação. Isso é feito testando todos valores

$$\left\{ x_i | x = (x_1, \dots, x_k) \in D_T^i, x_i < \phi^{c,L} \right\} \quad (2.28)$$

como candidatos para $\phi^{s,L}$ e, da mesma maneira, todos valores

$$\left\{ x_i | x = (x_1, \dots, x_k) \in D_T^i, x_i > \phi^{c,u} \right\} \quad (2.29)$$

como candidatos para $\phi^{s,L}$. Conexões são quebradas em favor de conjuntos *fuzzy* maiores, ou seja, com maiores distâncias do núcleo.

A fuzzificação é então feita para o antecedente com maior pureza. Isso é repetido para todos os antecedentes que foram fuzzificados.

Suponha que as regras *fuzzy* $r_1^{(j)}, \dots, r_k^{(j)}$ foi aprendida para a classe λ_j . Para uma nova instância de pesquisa x , o suporte dessa classe é definido por

$$s_j(x) \stackrel{df}{=} \sum_{i=1, \dots, k} \mu_{r_i^{(j)}}(x) \cdot CF(r_i^{(j)}), \quad (2.30)$$

aonde $CF(r_i^{(j)})$ é o fator de certeza da regra $r_i^{(j)}$. Ele é definido da seguinte maneira:

$$CF(r_i^{(j)}) = \frac{2 \frac{|D_T^{(j)}|}{|D_T|} + \sum_{x \in D_T^{(j)}} \mu_{r_i^{(j)}}(x)}{2 + \sum_{x \in D_T} \mu_{r_i^{(j)}}(x)} \quad (2.31)$$

aonde $D_T^{(j)}$ denota o subconjunto de instâncias de treinamento com rótulo λ_j .

A classe predita pelo FURIA é aquela que tem o suporte máximo. Em caso de empate, a decisão é feita a partir da maior frequência entre as classes. No caso de x ser coberto por nenhuma regra, o que significa que $s_j(x) = 0$ para toda classe λ_j , a decisão de classificação é feita de outra maneira.

Para tratar este problema, Eineborg and Boström (2001) substitui todas as regras por suas “generalizações mínimas” para a instância. Essa generalização ou “extensão” de uma regra é obtida por deletar um ou mais de seus antecedentes, e a generalização é mínima se ela não deleta mais antecedentes do que o necessário para cobrir a instância da pesquisa. Então, a generalização mínima de uma regra é simplesmente obtida por deletar todos antecedentes que não são satisfeitos pela pesquisa. Para o FURIA, foi proposta uma aproximação alternativa que explora a ordem em que os antecedentes são aprendidos, tratando eles como uma lista $(\alpha_1, \alpha_2, \dots, \alpha_m)$ ao invés do conjunto $\alpha_1, \alpha_2, \dots, \alpha_m$. A ideia é que a ordem reflete a importância do antecedente, o que é claramente justificado à luz do algoritmo subjacente de aprendizagem de regras. Como generalizações, só é permitido listas na forma $(\alpha_1, \alpha_2, \dots, \alpha_k)$, onde $k \leq m$. Para a generalização mínima, k é simplesmente dado por $j - 1$, onde α_j é o primeiro antecedente que não é satisfeita pela instância da pesquisa. Para reavaliar as regras de generalização, se usa a medida

$$\frac{p + 1}{p + n + 2} \times \frac{k + 1}{m + 2}, \quad (2.32)$$

aonde p é o número de exemplos positivos e n o número de negativos cobertos pela regra. O segundo fator a se levar em conta para o grau de generalização: Regras altamente podadas não são levadas em conta, pois a poda faz com que seja muito provável a diminuição da relevância da regra na pesquisa. Ainda mais, por fazer uma correção com Laplace o número relativo de antecedentes sobrando, k/m , a preferência é dada para regras mais longas e, portanto, mais específicas.

Computacionalmente, essas extensão de regras é mais eficiente que a anterior. Além disso, já que a avaliação de todas as generalizações de uma regra podem ser calculadas e armazenadas diretamente ao longo do processo de aprendizado de regras, no qual os antecedentes são aprendidos de maneira sucessiva, não há necessidade de se armazenar os dados de treinamento.

3 METODOLOGIA

3.1 Coleta de Dados

Dados hidroacústicos e de capturas de pescado foram fornecidos pelo Laboratório de Tecnologia Pesqueira e Hidroacústica do Instituto de Oceanografia (Universidade Federal do Rio Grande - FURG - <http://www.io.furg.br/>). Os dados hidroacústicos e de lances de pesca foram obtidos durante os cruzeiros ECOSAR VI e VII, realizados entre 2009 e 2010, realizados a bordo do Navio de Pesquisa Atlântico Sul, na região da plataforma da costa sudeste e sul da costa brasileira.

Para a coleta de dados hidroacústicos foi utilizada uma ecossonda científica digital com ecoinTEGRADOR SIMRAD EK500, acoplada a um transdutor de casco tipo *split beam*, operando na frequência de 38 kHz. Os dados hidroacústicos foram digitalmente armazenados sob a forma de dados acústicos brutos, os quais podem ser vistos em forma de imagens de ecogramas através do software MOVIES + versão 3.4b (IFREMER). Lances de pesca com rede de arrasto de meia água foram realizados sempre que detectados registros com elevada densidade de organismos nectônicos (que se movem ativamente na coluna de água). Através da análise de descritores energéticos, morfológicos, espaciais, temporais, comportamentais e biológicos, os ecoregistros detectados foram caracterizados em ecotipos, de acordo com características comuns e padrões morfolologicamente consistentes, conforme trabalhos realizados por Soares et al. (2005), Madureira (2004), Rossi-Wongtschowski et al. (2014), Madureira et al. (2015).

Em função dos objetivos do projeto de pesquisa, os dados têm por característica uma restrição quanto à área geográfica, entre cabo de Santa Marta (SC; 28° 36 S) e o cabo de São Tomé (RJ; 22° 02' S)(Figura 3.1), quanto ao tipo de sonda utilizada, no caso a SIMRAD EK500, quanto a frequência de operação (38 kHz) e quanto ao horário do dia, pois a maioria dos lances de pesca foram realizados em períodos diurno.

3.2 Pré-processamento

O primeiro processo de filtragem feito sobre os dados foi separar somente os ecogramas relacionados aos lances de pesca feitos nos cruzeiros ECOSAR VI e VII (Figura 3.1). Somente foram usados os dados dos registros acústico identificados que foram pescados, pois esse é o melhor método disponível para confirmar as espécies relacionadas aos registros feitos nestes cruzeiros.

Para que pudesse ser utilizado no processo de treinamento do modelo, os dados necessitavam um rótulo que já fosse previamente conhecido, o que permite relacionar os dados à uma classe. Durante estas pescas, os tripulantes elaboraram planilhas de dados de quantidade (número e

biomassa) e porcentagem da biomassa total das espécies presentes em cada lance de pesca com registros de ecograma e, a partir destes dados, foram identificadas as espécies predominantes de peixes em cada lance, o que foi utilizado como a classe alvo posteriormente.

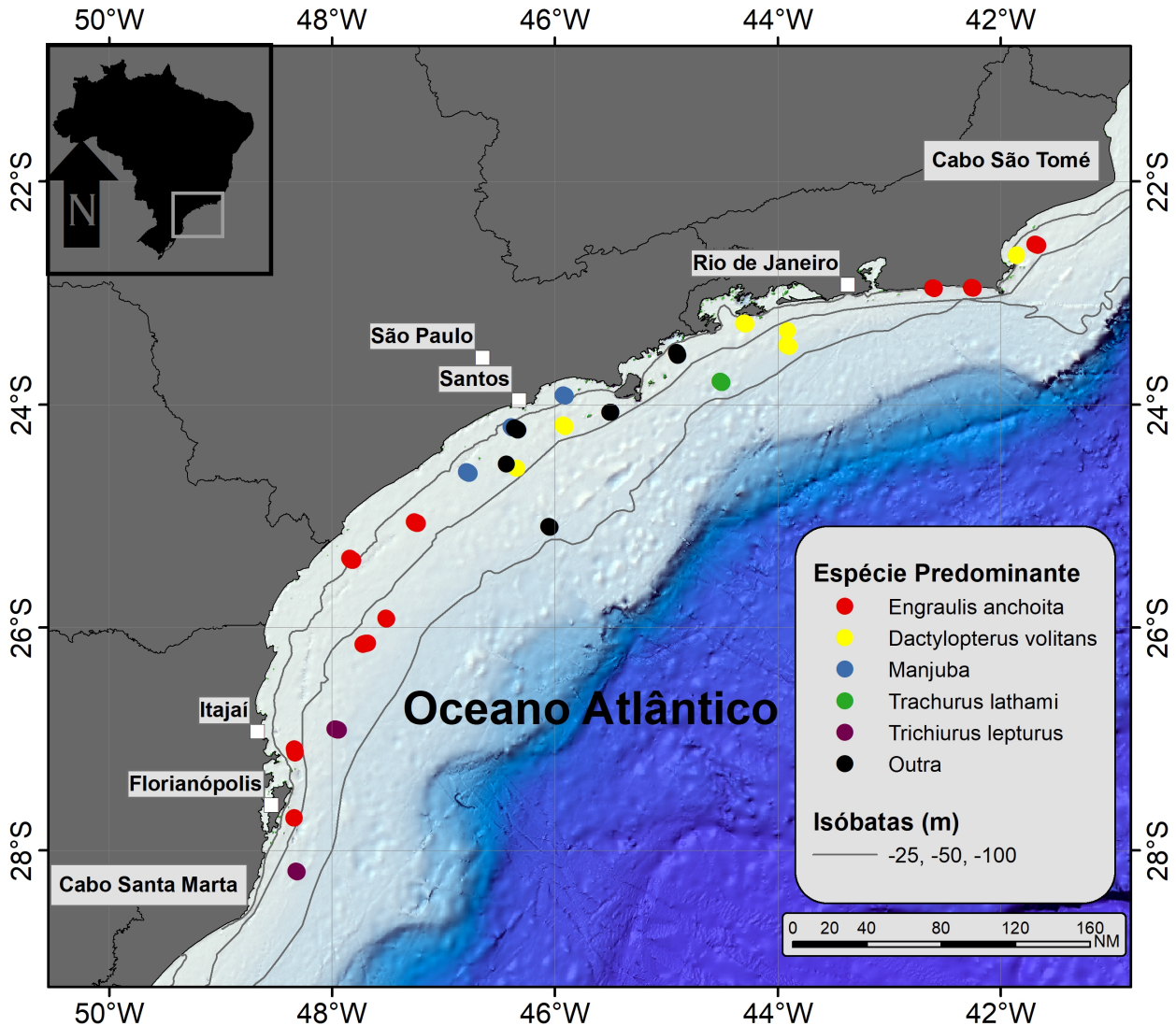


Figura 3.1: Mapa mostrando a distribuição dos lances (Fonte: Laboratório de Tecnologia Pesqueira e Hidroacústica - IO - FURG)

Na figura 3.1 podemos ver os locais onde foram feitos os lances de pesca durante os cruzeiros, mostrando as espécies predominante em cada um deles. Foram obtidos dados de principalmente de quatro espécies de peixes pelágicos: *Engraulis anchoita* (Anchoita), *Dactylopterus volitans* (Coio), *Trachurus lathami* (Xixarro) e *Trichiurus lepturus* (Peixe-espada). O item “Manjuba” mostrado no mapa se refere a peixes da família *Engraulidae*, outros que não a Anchoita, que não foram identificados no momento da pesca e por isso foram agregados sobre este nome. O item “Outra” se refere ao conjunto de cardumes que não possuíam uma espécie predominante, ou seja cardumes onde as espécies predominantes totalizavam cerca de 50% do total de indivíduos, ou espécies que tiveram muita pouca incidência e por isso foram agregadas em uma única

classe. As isóbatas que são mostradas no mapa são as linhas que definem a regiões em que a profundidade local é, respectivamente da costa em direção ao oceano, 25, 50 e 100 metros.

Durante os cruzeiros, todos os dados hidroacústicos foram armazenados em formato específico que podia ser lido pelo programa utilizados pelos profissionais que coletaram(o *software Movies+*). Este software possibilita o usuário visualizar o dados em forma de ecograma, disponibilizando diversas ferramentas para se extrair informações das imagens geradas. Na figura 3.2 se pode ver um exemplo de ecograma obtido neste processo.

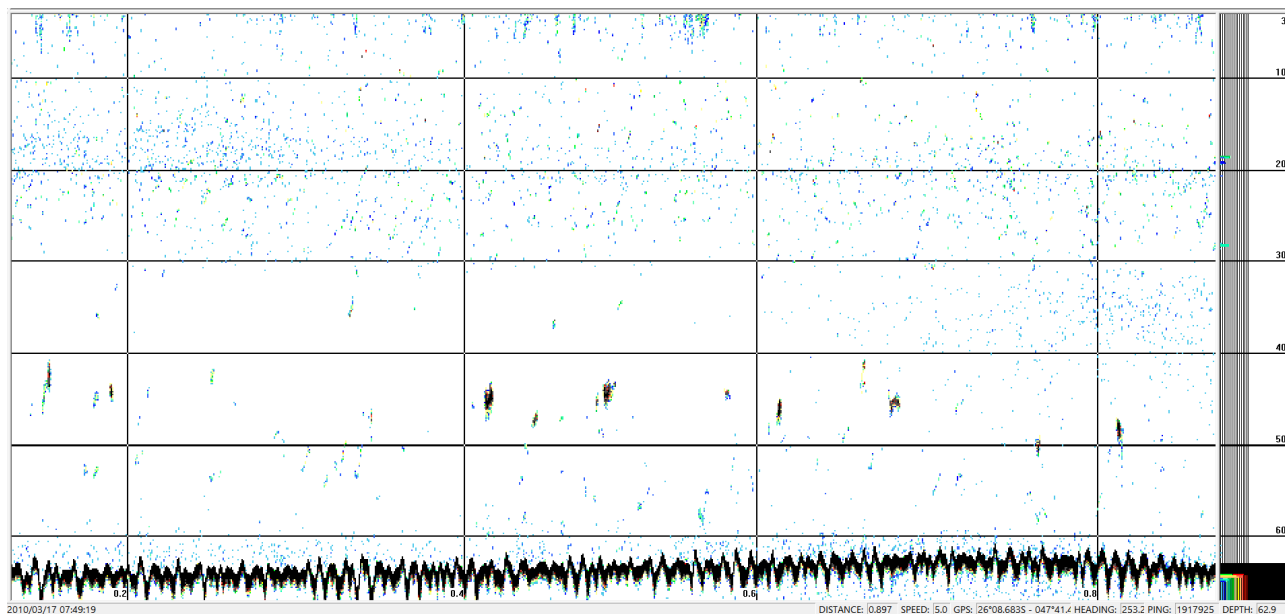


Figura 3.2: Exemplo de ecograma obtido pela ecossonda SIMRAD EK500 (Fonte: Laboratório de Tecnologia Pesqueira e Hidroacústica - IO - FURG)

Coletou-se dados complementares relacionados aos ecogramas dos lances de pesca, disponibilizados pelo Laboratório de Recursos Pesqueiros Pelágicos do Instituto de Oceanografia da FURG, que estavam presentes em diários de bordo e tabelas de informações. Nestes documentos foram obtidos dados de cada lance de pesca, como por exemplo quantidade de cada espécie pescada, profundidade média do local e de operação. Com o auxílio destes dados coletados, os ecogramas foram processados através da análise dos eco-registros de cardumes presentes dentro da área de operação e assim foram identificados quais eram relevantes. Ao final desse procedimento um total de 415 ecogramas foram selecionados para o estudo. Na figura 3.3 esta representado o processo feito para a identificação dos cardumes, aonde as linhas horizontais vermelhas representam os limites da área de operação e os registros circulados em vermelhos são os cardumes identificados.

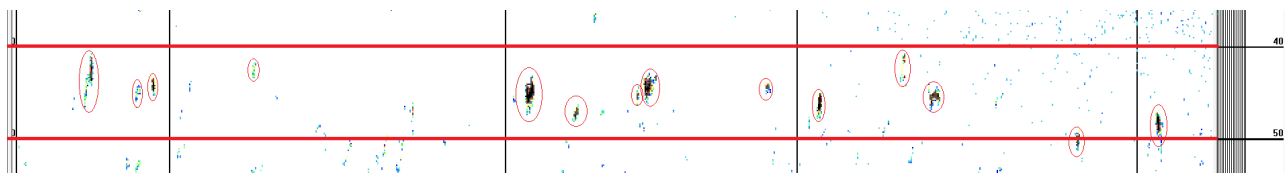


Figura 3.3: Ecograma obtido pela ecossonda SIMRAD EK500 com a área de pesca e os cardumes identificados (Fonte: Laboratório de Tecnologia Pesqueira e Hidroacústica - IO - FURG).

Após a identificação dos cardume, começou a coleta dos dados relacionados com o local do cardume (latitude, longitude, profundidade do local, profundidade em que o cardume se encontrava), os horários que foram encontrados e a morfologia (altura e comprimento) dos eco-registros. Estes dados foram obtidos através das ferramentas disponíveis no *software Movies*. A ferramenta *Zoom* foi utilizada para o enquadramento do eco-registro e maior precisão em suas medições morfológicas. A ferramenta *Identify* permitem identificar latitude, longitude, profundidade, hora e data do ponto em que está o cursor. Nas figuras 3.4 e 3.5 estão exemplificados o uso das ferramentas *Identify* e *Zoom*, respectivamente.

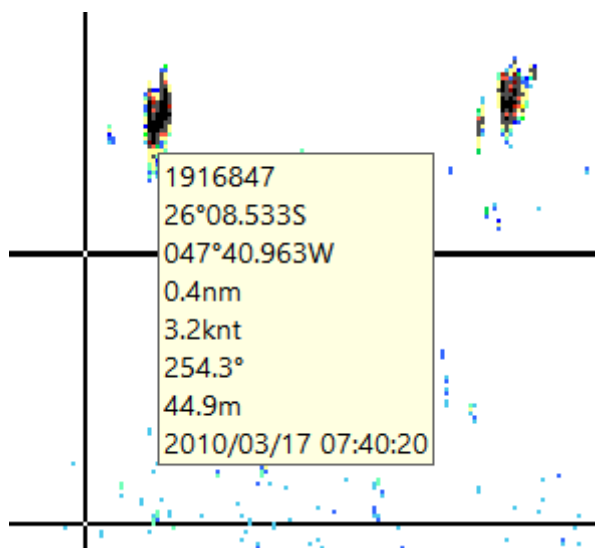


Figura 3.4: Exemplo de utilização da ferramenta *Identify*. (Fonte: Laboratório de Tecnologia Pesqueira e Hidroacústica - IO - FURG)

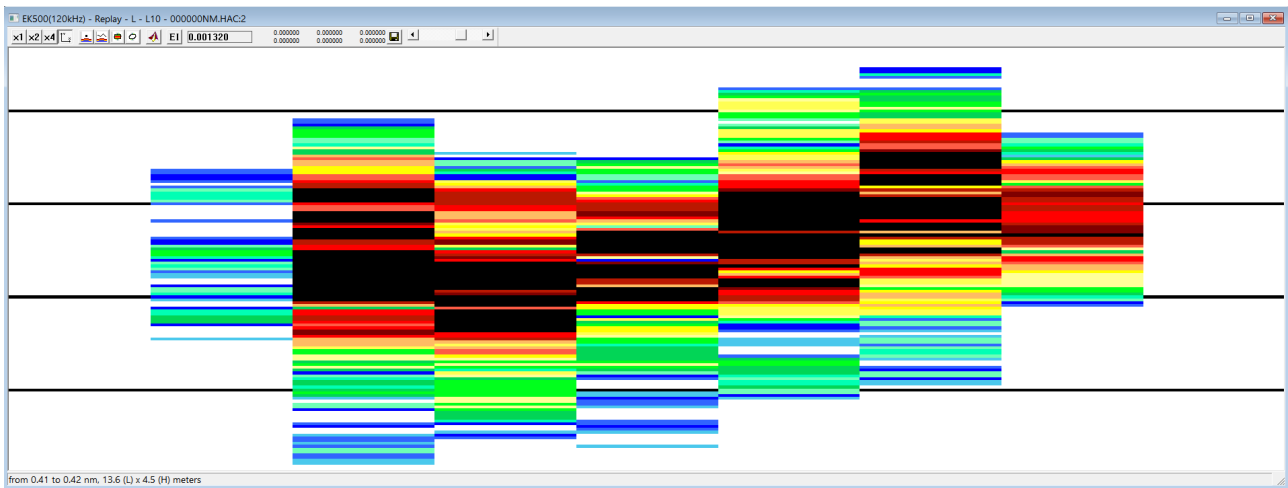


Figura 3.5: Exemplo de utilização da ferramenta *Zoom*. (Fonte: Laboratório de Tecnologia Pesqueira e Hidroacústica - IO - FURG)

O banco de dados foi testado e com auxílio de especialista experiente na aplicação de hidroacústica aplicada à pesca, que auxiliou na definição dos descritores que melhor representavam o conjunto de dados, conseguindo um total de 7 parâmetros descritores (Latitude, Longitude, Profundidade do Local, Profundidade do Cardume Superior e Inferior, Altura e Largura do Cardume). Alguns destes atributos que eram mais relacionados ao tipo de cruzeiro foram removidos: a profundidade de operação da rede de arraste, que foi utilizada na identificação dos cardume no ecograma. A latitude e longitude foram convertidas de graus e minutos (i.e. 37°12') para um atributo em graus com decimais (i.e 37,2). Também foram removidos os atributos relacionados ao horário das pescas, pois mesmo que o tempo possa influenciar o comportamento da espécie, ele estava mais relacionado as características do cruzeiro e não poderia se afirmar que a presença dos peixes naquele local estava relacionada com horário.

Durante os testes iniciais de aplicação dos algoritmos de classificação (veja seção 3.3), observou-se que algumas classes identificadas com poucas instâncias (registros) muitas vezes possuíam acerto nulo, isto é seus registros não eram associados corretamente a estas classes. Nestes casos a quantidade pequena de exemplos impedia que as classes fossem reconhecidas pelo modelo. Para solucionar este problema classes raras foram agregadas em uma única classe. Em regiões oceânicas, espécies formadoras de cardumes tendem a se dispersar ao longo da coluna d'água ao entardecer para se alimentar à noite, desagregando ou tornando muito dispersas as estruturas (eco-registros) formadas durante o dia (Soares et al., 2005). Na Figura 3.6 temos um exemplo de um dos cardumes dispersos que foram removidos do bancos de dados.

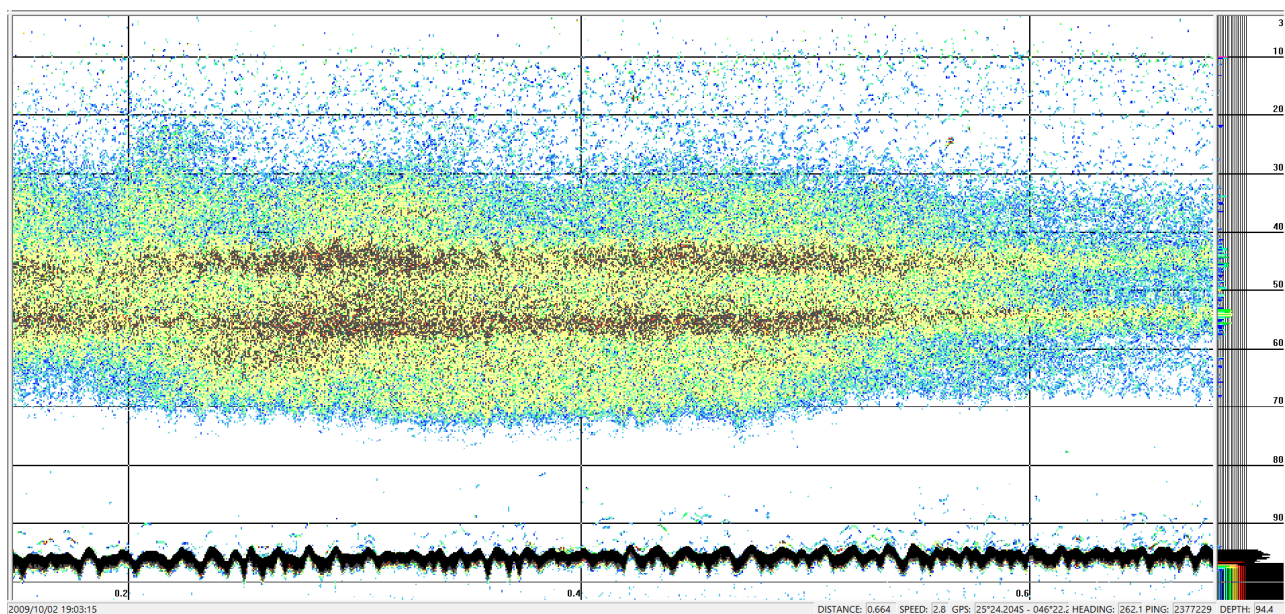


Figura 3.6: Exemplo de ecograma com registro de cardume(s) disperso(s) obtido pela ecossonda SIMRAD EK500 (Fonte: Laboratório de Tecnologia Pesqueira e Hidroacústica - IO - FURG).

Nas tabelas 1 e 2 estão representadas algumas medidas de dispersão, no caso a variância e o desvio padrão, e medidas de centralidade, com a média aritmética, a moda e a mediana, mostrando antes, considerando todos os registros, e depois da remoção destes cardumes dispersos, considerando somente os que sobraram. No total de 415 eco-registros, foram removida 7, sobrando 408.

	Médias	Moda	Mediana	Desvio Padrão	Variância	Máximo	Mínimo
LatitudeS	24.8942	25.067	24.6126	1.3308	1.7711	22.5632	28.6044
LongitudeO	46.3574	48.3265	46.7775	1.7079	2.9169	41.6664	48.6500
ProfLocal (m)	43.1364	32.5000	34.4000	18.1295	328.6771	97.0000	19.4000
ProfInicial(m)	24.8354	24.6000	21.7000	13.5662	184.0428	58.0000	3.8000
ProfFinal(m)	28.3535	26.8000	24.4000	13.6055	185.1098	80.1000	6.3000
Altura(m)	3.5246	1.6000	2.3000	4.4143	19.4857	61.5000	0.5000
Largura(m)	45.5629	7.4000	7.7000	285.1938	81335.5271	3184.4000	0.900

Tabela 1: Medidas de dispersão antes da remoção dos cardumes anômalos.

	Médias	Moda	Mediana	Desvio Padrão	Variância	Máximo	Mínimo
LatitudeS	24.8708	25.067	24.6123	1.2979	1.6845	22.5632	28.1939
LongitudeO	46.3506	48.3265	46.7773	1.6991	2.8869	41.6664	48.3383
ProfLocal (m)	42.8569	32.5000	34.0000	17.6523	311.6050	86.2000	19.4000
ProfInicial(m)	25.0027	24.6000	21.8000	13.5780	184.3629	58.000	3.8000
ProfFinal(m)	28.1625	26.8000	24.3500	13.2570	175.7476	62.4000	6.3000
Altura(m)	3.1713	1.6000	2.3000	2.5047	6.2736	15.9000	0.5000
Largura(m)	10.9206	7.4000	7.6000	11.9705	143.2938	129.2000	0.9000

Tabela 2: Medidas de dispersão depois da remoção dos cardumes anômalos.

Pode-se ver nas tabelas acima que, antes de se remover os eco-registros previamente citados, o desvio padrão e variância da largura do cardume eram extremamente altos em comparação com as informações obtidas após a remoção, enquanto que medidas como a moda e mediana não sofreram quase nenhuma mudança.

Também vale notar que por estes cardumes ocorrerem a noite ou ao entardecer, normalmente eram cardumes de espécies bastante misturadas, com prevalência de algumas espécies que possuíam comportamento em cardume bastante diferente ao dia. Então, para que se pudesse remover este efeito causado pelo horário do dia, estes cardumes foram deixados de lado para que o modelo pudesse ser gerado com maior confiabilidade.

Após a eliminação dos cardumes dispersos dos ecogramas selecionados para o estudo, os valores médios de cada um dos atributos foram comparados entre as classes de ecotipos através do teste de Kruskal-Wallis (Figura 3). O teste de Kruskal-Wallis é um teste não paramétrico, que permite comparar três ou mais classes/populações a partir de suas posições médias em uma ordenação de todos os valores do atributo considerado (Siegel and Castellan Jr, 1975). A determinação de quais os pares de classes eram diferentes foi realizada pelo teste de comparação múltipla por postos proposto por Siegel e Castellan.

Atributo Classe	LatitudeS	LongitudeO	ProfLocal	ProfInicial	ProfFinal	Altura	Largura
Dactylopterus volitans	23.5288d	44.4574d	47.0939b	10.4303c	17.3394c	6.9091a	29.0212a
Engraulis anchoita	25.2666b	46.7094b	39.1709c	24.5873b	27.5321b	2.9642bc	9.0327b
Trachurus lathami	23.7997d	44.5086d	77.8692a	14.7692c	17.8192c	3.0500abc	4.3538c
Manjuba	24.3312c	46.4647c	28.3265d	20.7980b	23.1510b	2.3569c	8.6039b
Outra	24.0698cd	45.5620d	44.1179bc	27.7641b	30.3154b	2.5513c	8.9256b
Trichiurus lepturus	27.0358a	47.9854a	65.9023a	51.4372a	55.0767a	3.6651ab	15.5488a
K-W	211,1000	224,2000	241,6000	187,5000	165,8000	55,6000	85,9000
p	< 0,001	< 0,001	< 0,001	< 0,001	< 0,001	< 0,001	< 0,001

Tabela 3: Resultados do teste de Kruskal-Wallis (K-W) são apresentados. Diferentes letras nas colunas indicam classes significativamente diferentes ($p < 0,05$), conforme o teste de comparação múltipla por postos de Siegel e Castellan.

Todos os atributos mostraram diferenças significativas ($p < 0,001$) entre as classes, sendo várias classes diferentes entre si. A classe “*Trichiurus lepturus*” distribuiu-se em maiores profundidades, latitudes e longitudes (afastados da costa), e junto com a classe “*Dactylopterus volitans*”, mostrou maiores altura e largura de seu eco-registro. As diferenças detectadas apontaram para um bom potencial de discriminação entre as classes na análise seguinte com os algoritmos de classificação.

3.3 Estratégia de classificação

Os programas utilizados para gerar os modelos de classificação foram o Weka (Hall et al., 2009), que é uma coleção de algoritmos de aprendizado de máquina feitos para tarefas de mineração de dados, e o KEEL (*Knowledge Extraction based on Evolutionary Learning* - Extração de Conhecimento baseado em Aprendizado Evolucionário) (Alcalá-Fdez et al., 2011), que é um software de ferramenta Java de código aberto que pode ser usada para um grande número de diferentes tarefas de descoberta de dados de conhecimento.

Foram escolhidos três algoritmos: o FARC-HD, o FURIA e o C4.5. O método utilizado para se fazer a classificação foi a validação cruzada de k partições (*k-folds cross-validation*) (Stone, 1974), com $k = 5$. Este método divide os dados em k subconjuntos mutuamente exclusivos de mesmo tamanho e então se utiliza um destes subconjuntos para teste e os outros $k - 1$ para o treino e se repete este processo k -vezes de forma circular. Após todas iterações, a acurácia é

calculada a partir dos erros calculados em cada um dos k modelos gerados, obtendo-se assim uma medida mais confiável sobre a capacidade do modelo de classificar corretamente os dados. Este processo está representado na Figura 3.7.

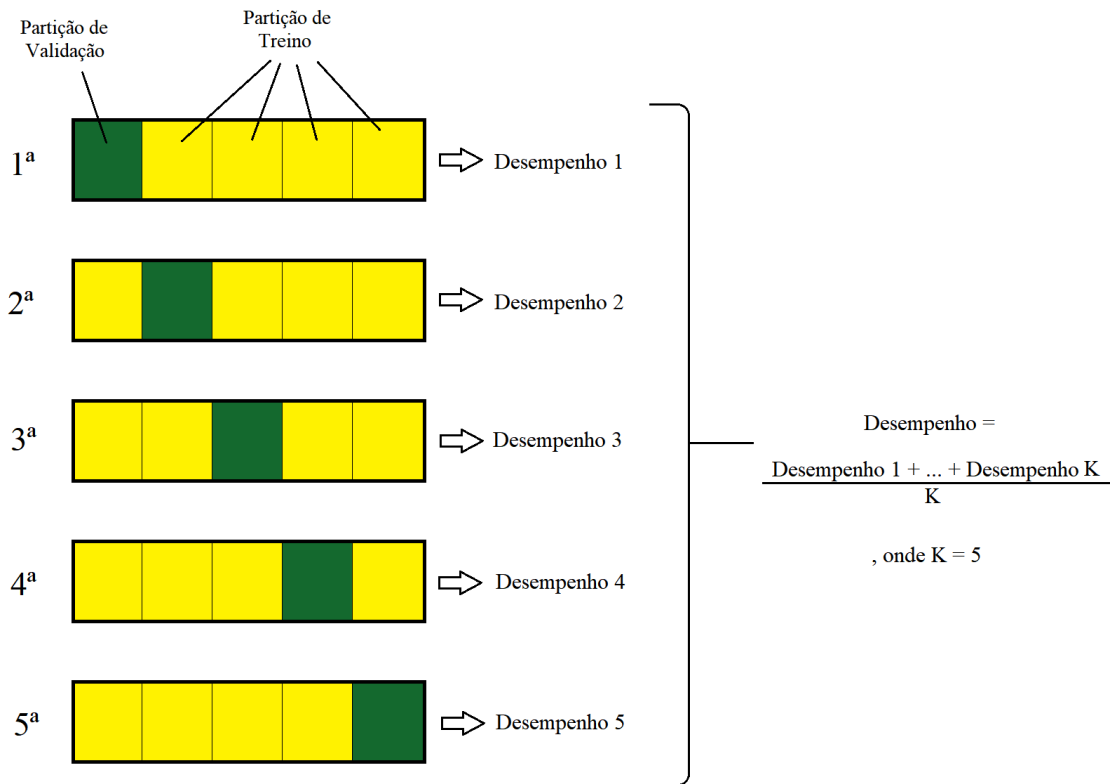


Figura 3.7: Imagem Representando o funcionamento da validação cruzada de k partições, com um exemplo de $k = 5$.

Diversos testes foram feitos para estes algoritmos, com o objetivo de descobrir os melhores parâmetros iniciais dos algoritmos, o que permitiu melhorar o desempenho. A partir destes testes foram feitas as seguintes mudanças: Para o algoritmo C4.5 foi modificado o parâmetro do número mínimo de instancias por folha (*minNumObj*), reduzindo seu valor de 10 para 5, a fim de se obter árvores menos genéricas; No caso do FARC-HD, a variável relativa ao número de variáveis linguísticas, que esta relacionada diretamente a quantos triângulos aparecem nas funções de pertinência das classes, foi modificada de 5 para 6; Por fim, foram modificadas no FURIA as variáveis a quantidade de otimizações, 2 para 1, e a quantidade de *folds*, de 3 para 4.

Na avaliação dos modelos, foram utilizadas as matrizes de confusão deles (Powers, 2011), que mostram quantos acertos de classificação aconteceram e se houve erros ela mostra qual foi a classe em que essas instâncias foram classificadas erroneamente. Nela é possível extrair valores como verdadeiros (TP) e falsos (FP) positivos, que são, respectivamente, quando uma classe classifica corretamente um exemplo ou quando ela está classificando um exemplo que

não é da classe, e os verdadeiros (TN) e falsos (FN) negativos, que são, respectivamente, os exemplos que realmente não são da classe específica e os exemplos que são de uma classe só que estão classificados como de outra classe. A partir destes valores é possível calcular medidas de acurácia por classe, que levam em conta estes acertos e erros feitos na classificação. As medidas escolhidas para serem usadas neste trabalho foram precisão (*Precision*), a revocação (*Recall*) e a medida F (*F-measure*) (Powers, 2011), que utilizam essas medidas são calculadas da seguinte maneira:

$$Precision = \frac{TP}{TP + FP}, \quad (3.1)$$

$$Recall = \frac{TP}{TP + FN}, \quad (3.2)$$

$$FMeasure = \frac{2 * Recall * Precision}{Recall + Precision}, \quad (3.3)$$

A precisão é a proporção da quantidade de exemplos classificados como uma classe específica que realmente são desta classe, por enquanto que a revocação é a proporção da quantidade de exemplos que eram de uma classe e conseguiram ser classificados como ela. Já a medida F é a média harmônica dessas duas medidas, para conseguir uma medida que represente um valor balanceado entre essas duas outras medidas.

4 RESULTADOS

Nesta seção serão descritos os modelos utilizando os algoritmos previamente citados na metodologia, analisando os resultados das classificações e seus desempenhos através de matrizes de confusão e medidas de acurácia.

4.1 Modelos Aprendidos

Utilizando os três algoritmos previamente descritos, modelos contendo regras que descrevem os dados foram gerados. Por se utilizar o método de validação cruzada com 5 partições, foram gerados no total 15 modelos, 5 para cada um dos algoritmos. Com o objetivo de simplificar a apresentação, foram gerados 3 modelos utilizando todos os dados. Primeiramente, foi gerado o modelo pelo algoritmo C4.5 de árvores de decisão, que pode ser visto na Figura 4.1.

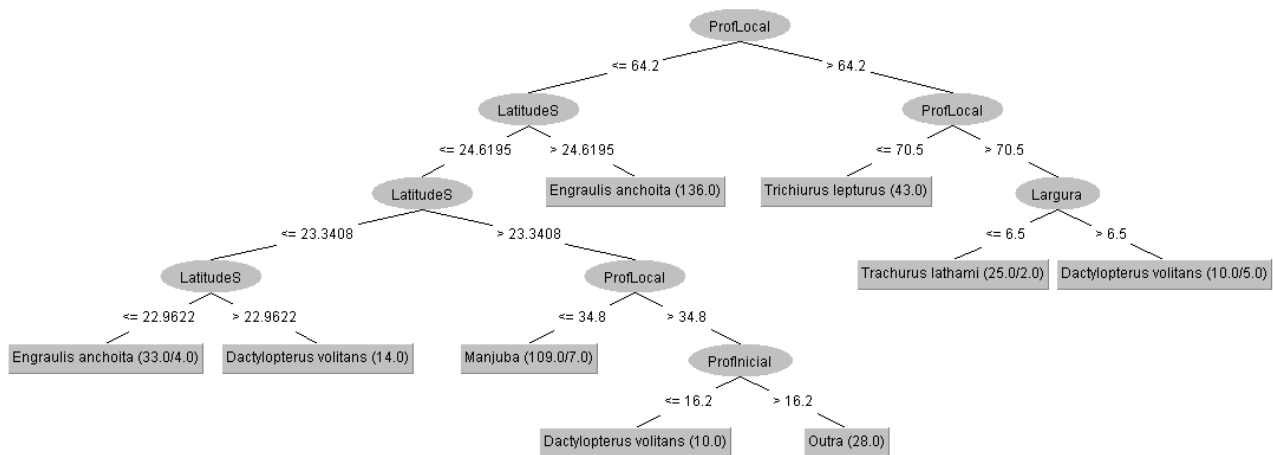


Figura 4.1: Modelo gerado pelo Weka utilizando o algoritmo de árvores de decisão C4.5.

A cada nó é verificado se o atributo é maior ou menor que o valor mostrado nas arestas. O número entre parenteses no lado da classe representa o total de instâncias/erros. Pode se observar que nele os atributos *ProfLocal* (que representa a profundidade do local onde o cardume foi encontrado) e *LatitudeS* (latitude ao sul da linha do equador) foram os principais utilizados para se diferenciar os cardumes, com um única aparição do atributo *Largura* (que refere-se a largura do cardume). Com esse modelo, a acurácia, que é o percentual de dados de teste que foram corretamente classificados, foi de 92.1569%.

Para o FARC-HD foram geradas as seguintes regras:

1. LatitudeS IS L2(6) AND ProfLocal IS L2(6) AND ProfInicial IS L1(6): *Dactylopterus volitans* CF: 0.9363

2. LatitudeS IS L_0(6) AND ProfFinal IS L_1(6): *Dactylopterus_volitans* CF: 0.4459
3. LongitudeO IS L_2(6) AND Largura IS L_2(6): *Dactylopterus_volitans* CF: 1.0
4. LongitudeO IS L_2(6) AND Altura IS L_3(6): *Dactylopterus_volitans* CF: 0.8822
5. ProfLocal IS L_2(6) AND ProfInicial IS L_0(6): *Dactylopterus_volitans* CF: 0.8561
6. LongitudeO IS L_1(6): *Engraulis_anchota* CF: 0.8348
7. LatitudeS IS L_3(6): *Engraulis_anchota* CF: 0.9011
8. LongitudeO IS L_5(6) AND ProfFinal IS L_1(6): *Engraulis_anchota* CF: 0.9995
9. LongitudeO IS L_2(6) AND ProfLocal IS L_4(6): *Trachurus_lathami* CF: 0.8616
10. LatitudeS IS L_1(6) AND LongitudeO IS L_4(6): *Manjuba* CF: 0.8722
11. LongitudeO IS L_3(6) AND ProfLocal IS L_1(6) AND ProfInicial IS L_1(6): *Manjuba* CF: 0.7430
12. LatitudeS IS L_2(6) AND LongitudeO IS L_3(6) AND ProfLocal IS L_5(6): *Outra* CF: 0.9466
13. LatitudeS IS L_1(6) AND ProfLocal IS L_2(6) AND ProfInicial IS L_2(6): *Outra* CF: 0.9637
14. LongitudeO IS L_3(6) AND ProfInicial IS L_3(6): *Outra* CF: 0.9852
15. ProfInicial IS L_5(6): *Trichiurus_lepturus* CF: 0.9449
16. LatitudeS IS L_5(6) AND ProfLocal IS L_4(6): *Trichiurus_lepturus* CF: 1.0

Nessas regras, as variáveis L_N , aonde N varia de 0 a 5, são as funções de pertinência triangulares das classes que têm seus respectivos valores nominais representados na Tabela 4. O valor entre parenteses é número total de funções triangulares para a classe. O valor de CF representa o nível de confiança da regra, ou seja, o quanto ela acerta. Pode se observar que todos os atributos foram utilizados. Os principais atributos que aparecem nas regras foram *LatitudeS*, *LongitudeO* (longitude ao oeste do meridiano de Greenwich) e *ProfLocal*, todos relacionados a posição geográfica do cardume, já que a profundidade do local esta diretamente relacionada com a distância relativa da costa. O acerto com essas regras foi de 96,0704%.

	L.0	L.1	L.2	L.3	L.4	L.5
LatitudeS	Muito Mais ao Norte	Mais ao Norte	Pouco Mais ao Norte	Pouco Mais ao Sul	Mais ao Sul	Muito Mais ao Sul
LongitudeO	Muito Mais ao Leste	Mais ao Leste	Pouco Mais ao Leste	Pouco Mais ao Oeste	Mais ao Oeste	Muito Mais ao Oeste
ProfLocal	Muito Raso	Pouco Raso	Meio p/ Raso	Meio p/ Fundo	Pouco Fundo	Muito Fundo
ProfInicial	Muito Raso	Pouco Raso	Meio p/ Raso	Meio p/ Fundo	Pouco Fundo	Muito Fundo
ProfFinal	Muito Raso	Pouco Raso	Meio p/ Raso	Meio p/ Fundo	Pouco Fundo	Muito Fundo
Altura	Muito Baixo	Pouco Baixo	Médio p/ Baixo	Médio p/ Alto	Pouco Alto	Muito Alto
Largura	Muito Estreito	Pouco Estreito	Médio p/ Estreito	Médio p/ Largo	Pouco Largo	Muito Largo

Tabela 4: Proposta de valores para as variáveis linguísticas referentes as funções de pertinência geradas pelo algoritmo FARC-HD.

Alterando os valores do modelo gerado automaticamente pelo algoritmo pelos valores propostos na Tabela 4, são obtidas as seguintes regras:

1. LatitudeS IS Pouco Mais ao Norte AND ProfLocal IS Meio para Raso AND ProfInicial IS Pouco Raso: *Dactylopterus_volitans* CF: 0.9363
2. LatitudeS IS Muito Mais ao Norte AND ProfFinal IS Pouco Raso: *Dactylopterus_volitans* CF: 0.4459
3. LongitudeO IS Pouco Mais ao Leste AND Largura IS Médio para Baixo: *Dactylopterus_volitans* CF: 1.0
4. LongitudeO IS Pouco Mais ao Leste AND Altura IS Médio para Alto: *Dactylopterus_volitans* CF: 0.8822
5. ProfLocal IS Meio para Raso AND ProfInicial IS Muito Raso: *Dactylopterus_volitans* CF: 0.8561
6. LongitudeO IS Mais ao Leste: *Engraulis_anchota* CF: 0.8348
7. LatitudeS IS Pouco Mais ao Sul: *Engraulis_anchota* CF: 0.9011

8. LongitudeO IS Muito Mais ao Oeste AND ProfFinal IS Pouco Raso: *Engraulis anchoita* CF: 0.9995
9. LongitudeO IS Pouco Mais ao Leste AND ProfLocal IS Pouco Fundo: *Trachurus lathami* CF: 0.8616
10. LatitudeS IS Mais ao Norte AND LongitudeO IS Mais ao Oeste: *Manjuba* CF: 0.8722
11. LongitudeO IS Pouco Mais ao Oeste AND ProfLocal IS Pouco Raso AND ProfInicial IS Pouco Raso: *Manjuba* CF: 0.7430
12. LatitudeS IS Pouco Mais ao Norte AND LongitudeO IS Pouco Mais ao Oeste AND ProfLocal IS Muito Fundo: *Outra* CF: 0.9466
13. LatitudeS IS Mais ao Norte AND ProfLocal IS Meio para Raso AND ProfInicial IS Meio para Raso: *Outra* CF: 0.9637
14. LongitudeO IS Pouco Mais ao Oeste AND ProfInicial IS Meio para Fundo: *Outra* CF: 0.9852
15. ProfInicial IS Muito Fundo: *Trichiurus lepturus* CF: 0.9449
16. LatitudeS IS Muito Mais ao Sul AND ProfLocal IS Pouco Fundo: *Trichiurus lepturus* CF: 1.0

Pode-se observar que, quando se tem os valores linguísticos definidos, as regras geradas pelos algoritmos *fuzzy* são de fácil entendimento do usuário, possibilitando uma melhor análise da veracidade dessas regras.

Por fim, foram geradas as seguintes regras com o algoritmo FURIA:

1. (ProfInicial in $[-inf, -inf, 15, 15.6]$) and (ProfLocal in $[34.4, 37, inf, inf]$) and (ProfLocal in $[-inf, -inf, 50.2, 52.2]$) and (LatitudeS in $[22.5692, 22.668, inf, inf]$) $\Rightarrow$ Especie Predominante = *Dactylopterus volitans* (CF = 0.93)
2. (LatitudeS in $[-inf, -inf, 23.4812, 23.5356]$) and (LatitudeS in $[22.9622, 23.3408, inf, inf]$) $\Rightarrow$ Especie Predominante = *Dactylopterus volitans* (CF = 0.9)
3. (LatitudeS in $[-inf, -inf, 22.6673, 22.9538]$) and (LatitudeS in $[22.5748, 22.6606, inf, inf]$) $\Rightarrow$ Especie Predominante = *Dactylopterus volitans* (CF = 0.69)
4. (LongitudeO in $[46.7905, 47.2304, inf, inf]$) and (ProfLocal in $[-inf, -inf, 64.2, 64.8]$) $\Rightarrow$ Especie Predominante = *Engraulis anchoita* (CF = 0.99)
5. (LatitudeS in $[-inf, -inf, 22.9622, 23.2709]$) and (LatitudeS in $[22.668, 22.9538, inf, inf]$) $\Rightarrow$ Especie Predominante = *Engraulis anchoita* (CF = 0.95)

6. (LatitudeS in [-inf, -inf, 22.5748, 22.6606]) => Especie Predominante = Engraulis anchoita (CF = 0.87)
7. (ProfLocal in [70.5, 75.4, inf, inf]) and (ProfLocal in [-inf, -inf, 79.4, 84]) and (LatitudeS in [23.4812, 23.7938, inf, inf]) => Especie Predominante = Trachurus lathami (CF = 0.93)
8. (ProfLocal in [-inf, -inf, 33.5, 47.2]) and (LatitudeS in [-inf, -inf, 24.6195, 25.0546]) and (LongitudeO in [46.3589, 46.3751, inf, inf]) => Especie Predominante = Manjuba (CF = 0.98)
9. (ProfLocal in [-inf, -inf, 21.6, 22]) => Especie Predominante = Manjuba (CF = 0.93)
10. (LongitudeO in [-inf, -inf, 46.3589, 46.3751]) and (ProfInicial in [20, 24.7, inf, inf]) and (LatitudeS in [22.9622, 23.5441, inf, inf]) => Especie Predominante = Outra (CF = 0.94)
11. (LongitudeO in [-inf, -inf, 46.3589, 46.3751]) and (LatitudeS in [24.1989, 24.2114, inf, inf]) and (LatitudeS in [-inf, -inf, 24.2315, 24.5747]) => Especie Predominante = Outra (CF = 0.8)
12. (LatitudeS in [-inf, -inf, 23.5595, 23.7938]) and (LatitudeS in [23.4812, 23.5356, inf, inf]) => Especie Predominante = Outra (CF = 0.86)
13. (ProfFinal in [27.3, 51.8, inf, inf]) and (LatitudeS in [26.1526, 26.9094, inf, inf]) => Especie Predominante = Trichiurus lepturus (CF = 0.96)
14. (LatitudeS in [27.7117, 28.1857, inf, inf]) => Especie Predominante = Trichiurus lepturus (CF = 0.7)

Neste modelo também teve uma grande quantidade de regras que utilizaram *Latitude* como atributo principal para o diferenciamento dos cardumes mas outros atributos também foram bastante utilizados, com exceção de *Altura* e *Largura*. O acerto deste modelo nos testes foi de 97,7898%.

4.2 Avaliação dos Modelos

Após tratar os dados e gerar os modelos dos classificadores utilizando os três algoritmos citados anteriormente, foram feitas análises sobre os dados obtidos com o Weka e o KEEL. As Tabelas 5, 6 e 7 mostram as matrizes de confusão dos modelos gerados pelos algoritmos C4.5, FURIA e FARC-HD, respectivamente.

Espécie classificada-> Espécie real	Dactylopterus volitans	Engraulis anchoita	Trachurus lathami	Manjuba	Outra	Trichiurus lepturus	FN
Dactylopterus volitans	22	4	5	0	2	0	11
Engraulis anchoita	1	163	0	0	0	1	2
Trachurus lathami	1	0	25	0	0	0	1
Manjuba	0	0	0	101	1	0	1
Outra	5	0	4	7	23	0	18
Trichiurus lepturus	0	0	1	0	0	42	1
FP	7	4	10	7	7	3	

Tabela 5: Matriz de Confusão do C4.5.

Espécie classificada-> Espécie real	Dactylopterus volitans	Engraulis anchoita	Trachurus lathami	Manjuba	Outra	Trichiurus lepturus	FN
Dactylopterus volitans	29	2	0	0	2	0	4
Engraulis anchoita	1	164	0	0	0	0	1
Trachurus lathami	0	0	26	0	0	0	0
Manjuba	0	0	0	98	4	0	4
Outra	0	0	0	0	38	1	1
Trichiurus lepturus	0	1	0	0	0	42	1
FP	1	3	0	0	6	1	

Tabela 6: Matriz de Confusão do FURIA.

Espécie classificada-> Espécie real	Dactylopterus volitans	Engraulis anchoita	Trachurus lathami	Manjuba	Outra	Trichiurus lepturus	FN
Dactylopterus volitans	28	3	1	0	1	0	5
Engraulis anchoita	3	162	0	0	0	0	3
Trachurus lathami	0	0	26	0	0	0	0
Manjuba	0	0	0	101	1	0	1
Outra	0	0	0	5	34	0	5
Trichiurus lepturus	0	1	0	0	0	42	1
FP	3	4	1	5	2	0	

Tabela 7: Matriz de Confusão do FARC-HD.

Essas matrizes possibilitam que, por exemplo, a visualização de quantos cardumes A foram classificados corretamente, quantos foram classificados como se fossem de outra classe B e quantos de outra classe B foram classificados como se fossem dessa classe A . Este tipo de informação é aplicada nas medidas de acurácia de classe. Na Tabela 8 pode se ver algumas medidas de acurácia para cada uma das classes dos modelos dos três algoritmos.

Algoritmos->	C4.5			FURIA			FARC-HD		
Classes	P	R	F	P	R	F	P	R	F
Dactylopterus volitans	0.759	0.667	0.710	0.967	0.879	0.921	0.848	0.903	0.875
Engraulis anchoita	0.976	0.988	0.982	0.982	0.994	0.988	0.982	0.976	0.979
Trachurus lathami	0.714	0.962	0.820	1.000	1.000	1.000	1.000	0.963	0.981
Manjuba	0.935	0.990	0.962	1.000	0.961	0.980	0.99	0.953	0.971
Outra	0.885	0.590	0.708	0.864	0.974	0.916	0.872	0.944	0.907
Trichiurus lepturus	0.977	0.977	0.977	0.977	0.977	0.977	0.977	1.000	0.988

Tabela 8: Tabela com algumas medidas de acurácia por classe para cada um dos algoritmos. “R” se refere a Revocação, “P” para a Precisão e “F” para a Medida F.

Os algoritmos C4.5 e FARC-HD foram os que geraram, respectivamente, o menor e maior número de regras para classificação dos ecotipos. Na Tabela 9 observamos que os atributos “Latitude”, “ProfLocal” e “Longitude” (com exceção no caso do C4.5, aonde “Longitude” não é utilizado) foram os que mais influenciaram na decisão dos classificadores.

	C4.5	FARC-HD	FURIA
LatitudeS	6	7	12
LongitudeO	0	8	4
ProfLocal	9	7	5
ProfInicial	2	5	2
ProfFinal	0	2	1
Altura	0	1	0
Largura	2	1	0
Total Regras	9	16	14

Tabela 9: Tabela com a contagem de regras em que um atributo está presente nos modelos gerados pelos algoritmos.

4.3 Interpretação dos Resultados

4.3.1 Classificação de ecotipos de cardumes pesqueiros baseada em regras fuzzy

O emprego de regras *fuzzy* sobre atributos de registros hidroacústicos de lances pesqueiros, definidos por especialista, mostrou ser capaz de classificar eficientemente ecotipos pesqueiros. O algoritmo FURIA, por exemplo, mostrou uma precisão global maior do que 95% na classificação dos ecotipos. Outros autores consideraram como satisfatórios valores de precisão menores do que o nosso, aplicando regras *fuzzy* para classificar solos (Ribeiro et al. (2014); 83-84%) e expressões faciais de emoções (Bahreini et al. (2019); 83.2%).

Os algoritmos *fuzzy*, particularmente o FURIA, apresentaram melhor desempenho do que o algoritmo clássico de árvore de decisão (C4.5) na distinção de ecotipos (classes) com menores números de instâncias, em relação a todos os parâmetros da acurácia (precisão, a revocação e a medida F). Hühn and Hüllermeier (2009) coletaram 45 diferentes bancos de dados (entrevistas, tipos de bactérias, carros, meteorológicos, etc.) e compararam suas classificações pelo algoritmos C4.5, FURIA e outros com lógica *fuzzy*. Estes pesquisadores também concluíram que o FURIA teve desempenho um melhor do que o C4.5, em termos de precisão, para a maioria dos bancos de dados. Mais recentemente, Palacios et al. (2014) sugeriram que o melhor desempenho do FURIA em relação a algoritmos clássicos e a outros *fuzzy* pode ocorrer particularmente para bancos de dados desbalanceados, isto é, onde algumas classes apresentam poucas instâncias e outras uma grande quantidade delas. Segundo estes autores a capacidade do FURIA de lidar melhor com classes “raras” resulta principalmente da redução dos falsos negativos (a não inclusão nas classes de instâncias que são realmente delas). Esta interpretação é compatível com os nossos resultados, onde a classe com menor número de instâncias (“*Trachurus lathami*”) apresentou uma maior precisão e um menor número de falsos negativos (maior revocação) com o FURIA. Palacios et al. (2014) afirmam que isto ocorre porque a acurácia do FURIA não é restrita pela escolha de uma partição linguística, e os antecedentes das regras trocam dinamicamente quando aparece uma instância que não é coberta pela regra (i.e. , substituição de todas as regras por suas “generalizações mínimas”, como introduzido por (Eineborg and Boström (2001); veja seção 2.4.4).

4.3.2 Importância do número de instâncias e dos diferentes atributos na classificação dos ecotipos de cardumes pesqueiros

No caso dos ecotipos que tiveram as melhores classificações quanto aos índices de acurácia podemos observar dois casos. Primeiramente as duas classes que possuíam mais exemplos, “*Engraulis anchoita*” e “Manjuba”, tiveram índices de acerto alto, já que com uma maior quantidade de dados foi possível criar um modelo com melhor credibilidade. Já no caso da classe “*Trachurus lathami*”, mesmo tendo a menor quantidade de instancias, ela teve a melhor classificação entre todas as classes. Isso significa que os exemplos obtidos possuíam características

que possibilitaram diferenciar ela das demais. O xixarro “*Trachurus lathami*”, é um peixe de hábito demersal-pelágico que atinge 30-60 cm de comprimento total (Rossi-Wongtschowski et al., 2014). Ele forma grandes cardumes sobre a plataforma continental, em profundidades menores do que 150 m (Costa et al., 2015), mas no caso das instâncias coletadas os cardumes estavam concentrados em profundidades bem específicas entre 70-80 m.

Observando as tabelas mostradas anteriormente, pode se ver que as Classes “Manjuba” e “Outra” foram as mais difíceis de separar com o algoritmo FURIA. A comparação dos valores dos atributos dessas duas classes (teste K-W e comparações múltiplas) indicou que elas se diferenciaram apenas quanto a Profundidade Local (menor em “Manjuba”) e a Longitude (menor em “Outra”). Provavelmente a principal causa disso é a composição de ambas as classes de ecotipos, que são formadas por cardumes mistos (com mais de uma espécie), cujas necessidades biológicas e interações ecológicas podem gerar maior variabilidade em suas distribuições espaço-temporal e na morfologia dos cardumes na coluna d’água. Logo, as duas possuem uma grande variação nos valores de seus atributos, tornando-as difíceis de discriminar uma da outra. “Manjuba”, por exemplo, foi formada por cardumes da família Engraulidae, pequenos peixes pelágicos que juntos habitam estuários e áreas mais rasas da plataforma (Rossi-Wongtschowski et al., 2014).

Apesar dos algoritmos *fuzzy* gerarem 50-70% mais regras para classificação dos ecotipos do que o C4.5, todos algoritmos utilizaram majoritariamente os atributos “Latitude” e “ProfLocal” para compor suas regras. Este resultado está relacionado ao fato destes atributos estarem diretamente relacionados com outros fatores abióticos (e.g., temperatura e a salinidade da água) e bióticos (disponibilidade de alimento) que influenciam os locais em que estes seres marinhos vivem (Lalli and Parsons, 1997) (Costa et al., 2015). Segundo Costa et al. (2015), mundialmente tem sido confirmado que a profundidade é o principal fator estruturador das comunidades de megafauna marinha (animais macroscópicos), sendo que na região de estudo onde foram obtidos os dados hidroacústicos, diferentes massas d’água com características distintas de temperatura, salinidade e nutrientes dissolvidos se distribuem em profundidades de 50 a 1200 m.

Os atributos “ProfInicial” e “ProfFinal”, que tiveram menor influência na classificação, também estão relacionados aos fatores ambientais descritos acima. Como os dados utilizados foram coletados em sua maioria durante o dia, o comportamento dos cardumes é afetado pela intensidade de luz do ambiente (i.e., na distribuição de seus recursos alimentares e de seus predadores), o que determina qual profundidade os cardumes das diferentes espécies irão estar (Soares et al., 2005).

Os atributos morfológicos “Altura” e “Largura” do eco-registro não tiveram muita influência na criação das regras, o que pode representar que a maneira em que os cardumes de peixe são formados nesta região não varia muito entre as espécies amostradas.

5 CONCLUSÃO

A partir dos dados de ecossondas e capturas de cruzeiros de pesquisa pesqueira, foi possível desenvolver modelos *fuzzy* (com os algoritmos FURIA e FARC-HD) e de árvore de decisão (C4.5), que classificaram corretamente mais de 90% dos eco-registros em relação às espécies de peixes capturados.

Quanto as acurácias das classificações dos eco-registros os algoritmos *fuzzy* (particularmente o FURIA) mostraram maiores valores de precisão (*precision*) e revocação (*recall*) do que o C4.5. Isto foi mais evidente para classes com menores números de registros. Um melhor recall do FURIA sugere que este algoritmo seria a melhor do operador da ecossonda para reduzir o erro de não lançar a rede quando deveria ter lançado.

Em relação aos critérios utilizados pelo especialista na identificação dos ecotipos, a latitude e a profundidade do local de registro mostraram maior importância na classificação do que aspectos morfológicos dos eco-registros (e.g. altura e largura).

Foi possível obter e tratar os dados das ecossondas utilizados nos cruzeiros pesqueiros, entretanto o processo foi predominantemente manual, sendo necessária a visualização de imagem e digitação dos dados observados. Para praticidade e agilidade deste processo seria necessário o desenvolvimento de ferramentas para extração da informação. Por exemplo, solicitar ao fabricante uma maneira de disponibilizar os dados dos eco-registros em forma de tabelas.

A utilização do modelo gerado pelos algoritmos *fuzzy* (em especial o FURIA) a bordo de cruzeiros de pesca como ferramenta de auxílio a decisão do lance de rede pode gerar uma maior eficiência das capturas, permitindo o aumento de lucro e a redução da pesca desnecessária de peixes não são os alvos do lance.

A principal contribuição desta dissertação foi desenvolver e analisar métodos explicáveis para classificação de eco-registros de organismos marinhos, obtidos em cruzeiros pesqueiros, com ecossondas, que seria o primeiro passo no sentido de possibilitar a automatização do processo. O diferencial na escolha destes métodos não está somente na qualidade dos resultados, mas também na interação com o especialista que foi possibilitada pela fácil interpretação dos modelos, o que permitiu sua intervenção em todas as fases do processo.

5.1 Trabalhos Futuros

A partir dos resultados obtidos nesta dissertação, pode-se prosseguir o trabalho de diversas maneiras. Uma possibilidade é a coleta de mais dados de ecogramas para se expandir o banco de dados, assim possibilitando um maior alcance de classificação, adicionando mais espécies,

diferentes localizações, outros períodos do ano, mais informações sobre as espécies que já foram encontradas, entre outros fenômenos descritos pelos registros acústicos.

Poderia ser feito também o desenvolvimento de uma ferramenta para a extração das informações presentes nos ecogramas, extraindo elas diretamente dos dados brutos ou através das imagens obtidas, para poder facilitar o trabalho que neste trabalho foi feito manualmente. Um outro método que poderia ser utilizado seria classificar diretamente as imagens dos ecogramas, estudando como esse tipo de ferramenta de classificação funciona e se é viável utilizá-la neste tipo de situação.

6 REFERÊNCIAS

- Adadi, A. and Berrada, M. (2018). Peeking inside the black-box: A survey on explainable artificial intelligence (xai). *IEEE Access*, 6:52138–52160.
- Alcalá, R., Alcalá-Fdez, J., and Herrera, F. (2007). A proposal for the genetic lateral tuning of linguistic fuzzy systems and its interaction with rule selection. *IEEE Transactions on Fuzzy Systems*, 15(4):616–635.
- Alcala-Fdez, J., Alcalá, R., and Herrera, F. (2011). A fuzzy association rule-based classification model for high-dimensional problems with genetic rule selection and lateral tuning. *IEEE Transactions on Fuzzy Systems*, 19(5):857–872.
- Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., and Herrera, F. (2011). Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic & Soft Computing*, 17.
- Aljawarneh, S., Aldwairi, M., and Yassein, M. B. (2018). Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. *Journal of Computational Science*, 25:152–160.
- Ashqar, B. A., Abu-Nasser, B. S., and Abu-Naser, S. S. (2019). Plant seedlings classification using deep learning.
- Bahreini, K., van der Vegt, W., and Westera, W. (2019). A fuzzy logic approach to reliable real-time recognition of facial emotions. *Multimedia Tools and Applications*, pages 1–24.
- Beliakov, G., Pradera, A., Calvo, T., et al. (2007). *Aggregation functions: a guide for practitioners*, volume 221. Springer.
- Bonini, J. A. (2016). Aplicação de algoritmos de árvore de decisão sobre uma base de dados de câncer de mama. *Revista ComInG-Communications and Innovations Gazette*, 1(1):57–67.
- Boström, H. (2004). Pruning and exclusion criteria for unordered incremental reduced error pruning.
- Calazans, D. and Griep, G. H. (2015). Introdução às ciências do mar. *Editora Textos, Pelotas*, pages 541–592.
- Cohen, W. W. (1995). Fast effective rule induction. In *Machine learning proceedings 1995*, pages 115–123. Elsevier.
- Cordón, O., del Jesus, M. J., and Herrera, F. (1999). A proposal on reasoning methods in fuzzy rule-based classification systems. *International Journal of Approximate Reasoning*, 20(1):21–45.

- Costa, P. A. S. d., Mincarone, M. M., Braga, A. d. C., Martins, A. S., Lavrado, H. P., Haimovici, M., and Falcão, A. P. d. C. (2015). Megafaunal communities along a depth gradient on the tropical brazilian continental margin. *Marine Biology Research*, 11(10):1053–1064.
- Eineborg, M. and Boström, H. (2001). Classifying uncovered examples by rule stretching. In *International Conference on Inductive Logic Programming*, pages 41–50. Springer.
- Fürnkranz, J. and Widmer, G. (1994). Incremental reduced error pruning. In *Machine Learning Proceedings 1994*, pages 70–77. Elsevier.
- Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18.
- Han, S. S., Kim, M. S., Lim, W., Park, G. H., Park, I., and Chang, S. E. (2018). Classification of the clinical images for benign and malignant cutaneous tumors using a deep learning algorithm. *Journal of Investigative Dermatology*, 138(7):1529–1538.
- Herrera, F. and Martínez, L. (2000). A 2-tuple fuzzy linguistic representation model for computing with words. *IEEE Transactions on fuzzy systems*, 8(6):746–752.
- Hühn, J. and Hüllermeier, E. (2009). Furia: an algorithm for unordered fuzzy rule induction. *Data Mining and Knowledge Discovery*, 19(3):293–319.
- Ishibuchi, H. (2009). *Classification and Modeling with Linguistic Information Granules: Advanced Approaches to Linguistic Data Mining*. Springer-Verlag, Berlin, Heidelberg.
- Ishibuchi, H. and Nakashima, T. (2001). Effect of rule weights in fuzzy rule-based classification systems. *IEEE Transactions on Fuzzy Systems*, 9(4):506–515.
- Kavšek, B. and Lavrač, N. (2006). Apriori-sd: Adapting association rule learning to subgroup discovery. *Applied Artificial Intelligence*, 20(7):543–583.
- Lalli, C. and Parsons, T. R. (1997). Biological oceanography: an introduction. pages 22–42.
- Lucca, G. (2018). *Aggregation and pre-aggregation functions in fuzzy rule-based classification systems*. Universidad Pública de Navarra - Departamento de Estadística, Informática y Matemáticas, Pamplona.
- Madureira, L. S. P. (2004). Prospecção de recursos pelágicos por método hidroacústico na plataforma, talude e região oceânica da costa central do brasil. *USP-Instituto Oceanográfico*, page 56p.
- Madureira, L. S. P., Pinho, M. P., Weigert, S. C., Coletto, J. L., Costa, P. L., and Valderrama, R. C. (2015). Fishing up and down the marine food web with hydroacoustics. in: Acoustics in underwater geosciences symposium. *IEEE/OES RIO Acoustics*, 8(3):1 – 5.

- Murphy, D. and Stich, S. (2000). Darwin in the madhouse: Evolutionary psychology and the classification of mental disorders. *Evolution and the human mind: Modularity, language and meta-cognition*, 62.
- Palacios, A., Trawiński, K., Cordon, O., and Sánchez, L. (2014). Cost-sensitive learning of fuzzy rules for imbalanced classification problems using furia. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 22(05):643–675.
- Powers, D. M. (2011). Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation.
- Quinlan, J. R. (1983). Learning efficient classification procedures and their application to chess end games. In *Machine learning*, pages 463–482. Springer.
- Quinlan, J. R. (1995). Mdl and categorical theories (continued). In *Machine Learning Proceedings 1995*, pages 464–470. Elsevier.
- Quinlan, J. R. (2014). *C4. 5: programs for machine learning*. Elsevier.
- Ribeiro, M. V., Cunha, L. M. S., Camargo, H. A., and Rodrigues, L. H. A. (2014). Applying a fuzzy decision tree approach to soil classification. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 87–96. Springer.
- Rossi-Wongtschowski, C. L. D. B., Vaz-dos Santos, A. M., and Siliprandi, C. C. (2014). Checklist of the marine fishes collected during hydroacoustic surveys in the southeastern brazilian bight from 1995 to 2010. *Arquivos de Zoologia*, pages 45: 73–88.
- Ruggieri, S. (2002). Efficient c4. 5 [classification algorithm]. *IEEE transactions on knowledge and data engineering*, 14(2):438–444.
- Siegel, S. and Castellan Jr, N. J. (1975). *Estatística não-paramétrica para ciências do comportamento*. Artmed Editora.
- Soares, C. F., Madureira, L. S. P., Habiaga, R. P., Laurino, L. D., Ferreira, C. S., and Weigert, S. C. (2005). Prospecção de recursos pesqueiros pelágicos na zona econômica exclusiva da região sudeste-sul do brasil: hidroacústica e biomassas. *Documentos REVIZEE - Score Sul*, pages 17–62.
- Srividya, T. and Arulmozhi, V. (2018). Feature selection classification of skin cancer using genetic algorithm. In *2018 3rd International Conference on Communication and Electronics Systems (ICCES)*, pages 412–417. IEEE.
- Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society: Series B (Methodological)*, 36(2):111–133.

- Weigert, S. C. and Madureira, L. S. P. (2011). Registros acústicos biológicos detectados na zona econômica exclusiva da região nordeste do brasil—uma classificação em ecotipos funcionais. pages 33(1) 15–32.
- Zadeh, L. (1965). Fuzzy sets. *Information and Control*, 8(3):338 – 353.
- Zadeh, L. A. (1988). Fuzzy logic. *Computer*, 21(4):83–93.
- Zhang, C. and Zhang, S. (2002). *Association rule mining: models and algorithms*. Springer-Verlag.