Universidade Federal de Pernambuco
Centro de Informática

Graduação em Ciências da Computação

Classificadores para dados simbólicos do tipo intervalo baseados em modelos geométricos

Diogo Rodrigues dos Santos Salazar

Trabalho de Graduação

Recife
17 de novembro de 2009

Universidade Federal de Pernambuco
Centro de Informática

Diogo Rodrigues dos Santos Salazar

Classificadores para dados simbólicos do tipo intervalo baseados em modelos geométricos

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

Orientadora: *Renata Maria Cardoso Rodrigues de Souza*

Recife
17 de novembro de 2009

Este trabalho é dedicado a Ulliane por todo o apoio dado e a meus pais, pois sem eles, este trabalho seria impossível.

Agradecimentos

Primeiramente, agradeço a professora Renata, por todo o apoio dado durante estes 4 anos de trabalho conjunto.

A Nina por ter cuidado de mim quando eu era criança.

Aos amigos do colégio por sempre tentarem me lembrar que existe todo um mundo fora da faculdade.

Aos amigos da faculdade, por sofrerem junto comigo, miséria adora companhia.

E aos meus familiares (que eu não vou perder meu tempo citando todos, vocês são muitos e daria para encher 10 páginas só falando o nome de cada um) por todo apoio e incentivo.

E por último, e não menos importante, a Ulli por ter criado a imagem 3.1 no CAD em cima da hora.

Alguns livros são para serem saboreados, outros para serem engolidos e alguns para serem mastigados e digeridos.

—SIR FRANCIS BACON (cientista, filósofo, jurista e político)

Resumo

Esse trabalho introduz diferentes classificadores de padrões para dados intervalares baseados em uma abordagem geométrica. Dois classificadores foram idealizados. Esses classificadores diferem na maneira de avaliar a dissimilaridade entre um novo elemento e cada classe. O primeiro classificador utiliza uma abordagem de lógica difusa para calcular a dissimilaridade. Enquanto o segundo utiliza uma abordagem de lógica clássica. Além disso, novas funções de dissimilaridade são apresentados neste trabalho. Experimentos com conjuntos de dados sintéticos e uma aplicação com um conjunto de dados intervalares real demonstram a funcionalidade e eficiência desses classificadores.

Palavras-chave: Análise de Dados Simbólicos, Aprendizagem de Máquina, Classificação, Dados Simbólicos Intervalares

Abstract

This work introduces different pattern classifiers for interval data based on a geometric approach. Two classifiers are considered. These classifiers differ in the way of evaluating de dissimilarity between a new element and each class. The first classifier uses a fuzzy logic approach. While the second one uses a classical logic approach. Besides, new dissimilarity functions are presented in this thesis. Experiments with synthetic data sets and an application with a real interval data set demonstrate the usefulness of these classifiers.

Keywords: Symbolic Data Analysis, Machine Learning, Classification, Interval Symbolic Data

Sumário

Lista de Figuras

Lista de Tabelas

CAPÍTULO 1

Introdução

Nunca confie em um computador que você não consiga jogar pela janela.
—STEVE WOZNIAK (co-fundador da Apple)

Com o crescente avanço da tecnologia de banco de dados e sistemas distribuídos, tem-se obtido bases de dados cada vez maiores, o que torna a extração de conhecimento um processo extremamente custoso. A descoberta de conhecimento em bases de dados (*Knowledge Discovery in Databases -KDD*) é uma área de pesquisa de aprendizagem de máquina de extrema importância, para que possa automatizar e agilizar a extração do conhecimento em bases de dados. Existem diversas técnicas estatísticas para analisar conjuntos de dados, mas, na maioria das vezes, estas técnicas não conseguem lidar com dados mais complexos ou a informação obtida é genérica demais. Neste sentido, a utilização de dados intervalares com a agregação de dados de vários indivíduos tem se tornado uma prática comum. Este tipo de dado tem sido bastante utilizado em Análise de Dados Simbólicos (SDA - *Symbolic Data Analysis*) [1], que é um domínio na área de obtenção de conhecimento e administração de dados. Dados simbólicos permitem múltiplos valores para cada variável e de diferentes tipos (intervalar, categórico multivalorado, modal).

A sumarização de grandes bases de dados em dados simbólicos levam a criação de tabelas de dados simbólicos. A diferença de uma tabela de dados simbólicos para uma tabela de dados é que em uma célula de uma tabela de dados simbólicos não necessariamente contém um único valor categórico ou quantitativo, mas diversos valores (tais como um intervalo ou distribuição) que podem ser pesados e conectados por regras lógicas e taxonomias. Portanto, uma evolução da análise de dados clássicos e métodos estatísticos para tais dados se tornam imprescindíveis. O trabalho proposto apresenta uma nova abordagem na classificação de dados simbólicos ao expandir a idéia sugerida por Ichino *et al.* [2] ao já utilizar dados simbólicos como entrada do classificador ao invés de dados clássicos e criando um mapa de regiões [3]. Para contornar o problema de sobreposição de regiões (*overlapping*) um classificador difuso (*fuzzy*) conforme apresentado por Keller *et al.* [4] também é avaliado. Para demonstrar a utilidade do modelo proposto foram realizados testes com dois conjuntos de dados sintéticos de baixa e alta sobreposição e um conjunto de dados reais onde é comparado a abordagem não-difusa (*crisp*) e difusa.

No capítulo 2 será apresentado uma visão geral sobre classificação supervisionada e uma apresentação sobre dados e tabelas de dados simbólicos. O capítulo 3 detalha os conceitos básicos por trás dos algoritmos utilizados, assim como apresenta as funções de dissimilaridade

usadas. Os resultados obtidos nas simulações são apresentados no capítulo 4. E, por fim, o capítulo 5 apresenta as conclusões e sugestões de trabalho futuro.

CAPÍTULO 2

Aprendizagem Supervisionada e Dados Simbólicos

O verdadeiro problema não é se as máquinas são capazes de pensar, mas se os homens são.

—B. F. SKINNER (escritor, inventor, poeta e psicólogo)

Aprendizagem de máquina (*Machine Learning*)é uma área da inteligência artificial que busca desenvolver algoritmos e modelos que torne possibilite sistemas adquirirem conhecimento a partir de dados. Uma das principais sub-áreas da aprendizagem de máquina é aprender a reconhecer padrões complexos e auxiliar na tomada de decisões. Neste capítulo será realizada uma revisão dos principais conceitos sobre Aprendizagem Supervisionada e Análise de Dados Simbólicos.

2.1 Aprendizagem Supervisionada

O ser humano sempre teve a necessidade de encontrar ordem na natureza. Desde separar o grão de feijão bom do grão de feijão ruim a um médico diagnosticar a doença de uma pessoa olhando o resultado de seus exames. Em ambos os casos, o indivíduo utiliza o conhecimento adquirido no passado para classificar um novo objeto a um grupo. A aprendizagem de máquina é uma área da inteligência artificial que investiga sistemas capazes de adquirir conhecimento a partir de dados para auxiliar em uma tomada de decisão.

Em aprendizagem de máquina, existem dois paradigmas: a aprendizagem supervisionada e aprendizagem não-supervisionada. De fato, a aprendizagem supervisionada é utilizada em problemas de regressão e classificação enquanto a aprendizagem não-supervisionada é utilizada em geração de regras de associação e problemas de agrupamento.

Os problemas de classificação seguem duas etapas: a etapa de treinamento ou aprendizagem e a etapa de alocação ou classificação, como mostra a figura 2.1.

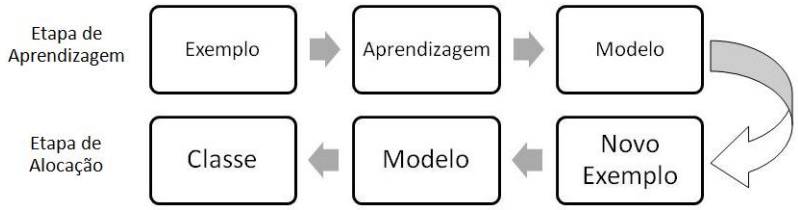

Figura 2.1 Etapas do processo de classificação

2.1.1 Etapa de Aprendizagem

A etapa de aprendizagem começa com o classificador recebendo vários exemplos de objetos para poder iniciar o seu aprendizado. Algum pré-processamento pode ser feito com estes exemplos antes do classificador iniciar o seu treinamento. Depois disso, o classificador começará a aprendizagem. Esta sub-etapa pode ser realizada de várias maneiras, dependendo do tipo de classificador usado.

2.1.2 Etapa de Classificação

Após terminada a etapa de aprendizado, os novos exemplos são aplicados ao modelo obtido através de regras, tais regras podem ser baseadas em medidas de proximidade (similaridade, dissimilaridade), escores ou outros critérios.

2.2 Análise de Dados Simbólicos

Dados simbólicos podem surgir de diferentes maneiras. Considere que a variável de interesse é X =Língua Oficial e a população de interesse como ω =Países Europeus. Tomando como exemplos, Bélgica e Suíça, onde Bélgica,Suíça $\in \omega$ eles assumiriam os valores X(Bélgica)=Francês, Holandês, Alemão e X(Suíça)=Alemão, Italiano, Francês, Romanche. Em outro exemplo, você pode obter a altura média de cada turma de uma escola, mas desta forma, você estaria perdendo a informação de que cada turma possue diferentes variações, sendo assim, você poderia guardar a informação de uma turma como [1.61,1.81] ao invés de 1.71 [1].

Dados simbólicos são capazes de reduzir grandes bases de dados clássicos em novos conjuntos de dados simbólicos de tamanho menor, tornando mais fácil sua análise. Os dados presentes numa base simbólica retêm as informações contidas numa base de dados clássica apresentado-as em conjunto menor e, na grande maioria das vezes, sem perda de informação [1].

Uma tabela de dados simbólicos pode conter em uma única célula, informações expressas por intervalos, distribuições de probabilidade, etc. enquanto que uma tabela de dados clássicos cada célula apresenta um único valor.

A Análise de Dados Simbólicos surgiu da influência simultânea de 3 áreas:

1. Da Análise de Dados Exploratória (*EDA- Exploratory Analysis Data*) onde a análise é

feita de forma individual. Neste contexto, a abordagem simbólica extende os métodos clássicos para descrições mais complexas e fornece novos resultados baseados em objetos simbólicos [5].

2. Da Inteligência Artificial onde o desenvolvimento de novas linguagens capazes de representar dados mais complexos tem sido bastante pesquisado. Neste sentido, em *SDA* estamos mais preocupados com a sua modelagem matemática, na sua análise, representação gráfica, etc.

3. Da Taxonomia Numérica na biologia, onde os conceitos de hierarquia de ordenamento de espécies de tem sido bastante utilizado e investigado [6].

Em todas as 3 áreas a seguinte questão surgiu: "Como se consegue obter classes e suas descrições?". Do ponto de vista histórico, podemos falar de 3 modelos principais [1]:

1. O primeiro, proposto por A.L. Jussieu [1] é inspirado no modelo aristotélico. Onde classes são definidas 'de cima para baixo' (*top-down*) da mais geral para as mais específicas escolhendo as propriedades dos seus atributos que melhor caracterizem-nas. Ao final, é obtida uma árvore de decisão onde cada nó é caracterizado por um conjunto de características.

2. O segundo modelo, sugerido por Adanson [1] que propôs o primeiro algoritmo de Agrupamento Hierárquico Aglomerativo Seqüencial (*Sequential Agglomerative Hierarchical Clustering - SAHC*). Este conhecido método 'de baixo para cima' (*bottom up*) começa com classes reduzidas a indivíduos, onde elas vão se reunindo iterativamente. As classes obtidas desta maneira contém objetos similares sendo possível descrvê-las como uma disjunção de conjunções de características.

3. O terceiro modelo consiste em buscar diretamente por classes e suas características. O método de agrupamento dinâmico (*Dynamic Clustering Method*) [7] por exemplo, fornece uma estrutura geral para descobrir classes e suas propriedades de maneira simultânea de tal forma que elas se encaixem de forma ótima.

2.2.1 Tabelas de Dados Simbólicos

Conforme mostrado anteriormente, os dados simbólicos são capazes de descrever indíviduos sendo capaz de considerar imprecisão ou incerteza, ou podem descrever itens mais complexos, tais como grupos de indivíduos. Tais dados são apresentados em tabelas de dados simbólicos.

Em tabelas de dados simbólicos, as linhas correspondem aos indivíduos ou grupos de indivíduos e as colunas são as variáveis simbólicas que os descrevem. Um exemplo de tabela de dados simbólicos é apresentado abaixo. Neste exemplo as linhas são grupos de indivíduos e nas colunas temos três variáveis simbólicas: PIB (representado por um intervalo), nome do país (representado por um conjunto de categorias) e proporção da população que fala mais de uma língua (representado por uma distribuição de pesos).

ID	PIB (em trilhões de dólares)	Nome do país	Bilíngüe+
1	[0.31, 4.28]	{Bélgica, Japão, Suíça }	{(3/4) Sim, (1/4) Não}
2	[0.37, 1.98]	{Argentina, Brasil, Uruguai}	{(1/6) Sim, (5/6) Não}
3	[0.16, 14.02]	{EUA, França, Chile}	{(2/5) Sim, (3/5) Não}

Tabela 2.1 Exemplo de uma tabela de dados simbólicos.

2.2.2 Técnicas de aprendizagem supervisionada para dados simbólicos

Vários métodos de aprendizagem supervisionada foram estendidas para lidar com dados simbólicos: um desses métodos, proposto por Diday, Ichino *et al.* [2] serve como base para este trabalho de graduação. Este método, utilizando uma abordagem baseada em regiões para dados multi-valorados. Nessa abordagem os exemplos de classes são descritos por uma região ou conjunto de regiões obtida através da utilização de uma aproximação de um grafo de vizinhança mútua (*Mutual Neighbourhood Graph - MNG*) e um operador de junção simbólico, mais sobre isso será explicado no capítulo seguinte. Ciampi, Diday, *et al.* [**?**] introduziram uma generalização das árvores de decisão binárias para predizer o conjunto dos membros de classe de dados simbólicos. Silva e Brito [8] propuseram três abordagens para a análise multivariada de dados intervalares, tendo como foco a análise do discriminante linear.

CAPÍTULO 3

Classificadores Propostos

Não há riqueza como o conhecimento, nenhuma pobreza como a ignorância.

—ALI IBN ABI-TALIB (Quarto Califa)

Nesse capítulo serão apresentados os dois classificadores propostos para dados de tipo intervalo. Duas etapas principais estão envolvidas na construção desses classificadores: a etapa de aprendizado, onde um mapa de junção será construído com um conjunto de dados simbólicos de entrada, e a etapa de alocação ou classificação, onde novos elementos são classificados como pertencentes à determinada classe com a menor distância para os elementos de treinamento, no caso do classificador não-difuso (*crisp*) ou com base na sua maior similaridade, no caso do classificador difuso (*fuzzy*).

3.1 Conceitos Básicos

3.1.1 Operador de Junção

A junção [3], denotada por $s_1 \oplus s_2$, resulta no objeto $s_3 = \overset{p}{\wedge}_{i=1} [y_1 \in d_{3i}]$, onde $d_{3i} = d_{1i} \oplus d_{2i}$, $\forall i = 1,...,p$, é obtido da seguinte forma:

•se a variável é quantitativa, têm-se que $d_{1i} = [d_{1iL}, d_{1iU}]$ e $d_{2i} = [d_{2iL}, d_{2iU}]$. Então $d_{3i} = [\min(d_{1iL}, d_{2iL}), \max(d_{1iU}, d_{2iU})]$, onde d_{1iL} e d_{2iL}) são, respectivamente, os limites inferiores dos intervalos d_{1i} e d_{2i}) e d_{1iU} e d_{2iU}) são, respectivamente, os limites superiores dos mesmos intervalos.

•se a variável y_i é qualitativa, $d_{3i} = d_{1i} \cup d_{2i}$.

•$d_{3i} = \emptyset \Leftrightarrow d_{1i} = d_{2i} = \emptyset$, independente do tipo da variável y_i.
•$d_{3i} = d_{1i} \Leftrightarrow d_{2i} = \emptyset$, independente do tipo da variável y_i.
•$d_{3i} = d_{2i} \Leftrightarrow d_{ii} = \emptyset$, independente do tipo da variável y_i.

A junção satifaz às seguintes propriedades:
Sejam n objetos simbólicos $s_1,..., s_m$, onde $s_i = \wedge_{i=1}^{p} [y_1 \in d_{3i}]$.

1) $\forall s_1, s_2, s_1 \oplus s_2 = s_2 \oplus s_1$

2) $\forall s_1, s_2, s_3, (s_1 \oplus s_2) \oplus s_3 = s_1 \oplus (s_2) \oplus s_3)$

3) $\forall s_1, s_1 \oplus s_1 = s_1$

4) $\forall s_i, i \in 1,2,\ldots,n, s_1 \wedge (s_2 \oplus \ldots \oplus s_n) = (s_1 \wedge s_2) \oplus \ldots \oplus (s_1 \wedge s_n)$

Exemplo: Considere dois objetos simbólicos $s_1 = [y_1 \in [10,20]] \wedge [y_2 \in [30,60]]$ e $s_2 = [y_1 \in [25,40]] \wedge [y_2 \in [10,30]]$. Então $s_1 \oplus s_2 = [y_1 \in [10,40]] \wedge [y_2 \in [10,60]]$. A figura 3.1 ilustra essa operação.

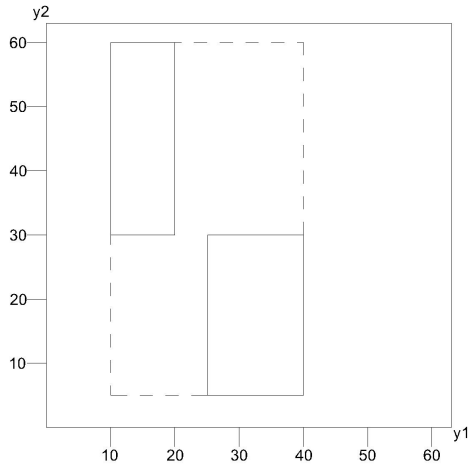

Figura 3.1 Representação da junção

3.1.2 Região

Uma *região* [9] $R(C_k)$ associada a classe C_k é uma região em $\Re^p$ cuja descrição é dada ao juntar todos os pares de objetos para obter um vetor de intervalos $\mathbf{s}(C_k) = (s_{k1}, \ldots, s_{kp})$ onde $s_{kj} = [\alpha_{kj}, \beta_{kj}]$ é um intervalo com $\alpha_{kj} = min\{a_{k1j}, \ldots, a_{kn_kj}\}$ e $\beta_{kj} = max\{b_{k1j}, \ldots, b_{kn_kj}\}$, $(j = 1, \ldots, p)$.

3.1.3 Grafo de Vizinhança Mútua

O grafo de vizinhança mútua (*mutual neighborhood graph - MNG)* carrega informações sobre a estrutura inter-classes. Os objetos pertencentes a classe C_k são cada um **vizinhos mútuos**[10] se $\forall \omega_{k'i} \in C_{k'}$ $(k' \in \{1,\ldots,m\}, k' \neq k), \mathbf{x}_{k'i} \cap \mathbf{g}_k \notin R(C_k)$ $(i = 1, \ldots, n_{k'})$. Neste caso, o *MNG* of C_k contra $\overline{C_k} = \cup_{\substack{k'=1 \\ k' \neq k}}^{m} C_{k'}$, o qual é construido ao juntar todos os pares de objetos que são vizinhos mútuos, é um grafo completo. Se os objetos pertencentes a classe C_k não são vizinhos

mútuos, nós procuramos por todos os subconjuntos de C_k cujos elementos são cada um vizinhos mútuos entre si e que sejam um **clique máximo** no *MNG*, que, neste caso, não é um grafo completo. Para cada um destes subconjuntos de C_k nós podemos associar uma **região**. Esta etapa exige alguns cuidados. Quando o *MNG* de uma classe C_k não é um grafo completo, torna-se necessário construir uma aproximação do *MNG* porque a complexidade computacional em tempo para encontrar todos os cliques máximos em um grafo é de ordem exponencial. A figura a seguir mostra um exemplo de construção de *MNG* para um conjunto de dados com duas classes, onde as linhas pontilhadas mostram a criação de uma região de vizinhança mútua.

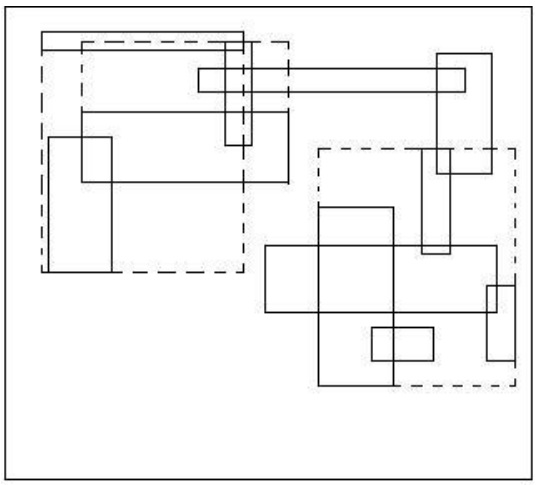

Figura 3.2 Construção de um grafo de vizinhança mútua

3.1.4 Descrição potencial

Fujnções de dissimilaridade baseadas na noção de descrição potencial são apresentados neste trabalho, utilizando duas abordagens. Na primeira, o volume do hiper-cubo definido pela região $R_j(C_k)$, $\pi(R_j(C_k))$ corresponde a descrição potencial. E a segunda que utiliza o somatória da diferença entre o ponto superior e inferior do hiper-cubo (o *range*) para cada dimensão p, $\sum_{j=1}^{p} \mu^p(Ck)$.

3.2 Algoritmos

Tanto para o classificador difuso quanto para o classificador não-difuso, o algoritmo de aprendizado é o mesmo. Apenas durante a etapa de alocação que cada um dos classificadores utiliza uma abordagem diferente.

3.2.1 Etapa de Aprendizado

O objetivo da etapa de aprendizado é construir um conjunto de regiões definidas por hiper-cubos que formam um grafo de vizinhança mútua.

Algorithm 1 Etapa de treinamento

Para cada classe: $k = 1, \ldots, m$ faça

1: **Encontre** a região $R(C_k)$ (de acordo com a *definição 3.1.2*) associada a classe C_k
Verifique se os objetos pertencentes a esta classe são, cada um, vizinhos mútuos de acordo com a *definição 3.3*
2: **Se** isso ocorrer, construa o *MNG* (que é um grafo completo) e pare;
3: **Se** não ocorrer, do:
3.1: **Escolha** um objeto de C_k como uma semente de acordo com a ordem lexicográfica destes objetos em C_k;
Faça $t = 1$ e coloque a semente no conjunto C_k^t;
Remova a semente de C_k
3.2: **Adicione** o próximo objeto de C_k (de acordo com a ordem lexicográfica) a C_k^t
Se todos os objetos agora pertencents a C_k^t permanecem vizinhos mútuos de acordo com a *definição 3.1.3*, remova este objeto de C_k
3.3: **Repita** o passo 3.2) para todos os objetos restantes de C_k
3.4: **Encontre** a região $R(C_k^t)$ (de acordo com a *definição 3.1.2*) associada a C_k^t
3.5: **Se** $C_k \neq \emptyset$,
Faça $t = t + 1$
Repita passos 3.1) a 3.4) até $C_k = \emptyset$
3.6: **Construa** o *MNG* (que não é um grafo completo) e pare

3.2.2 Etapa de Alocação

O objetivo da etapa de alocação é associar um novo elemento a uma classe baseado em uma função de dissimilaridade que compara a descrição da classe (uma região ou conjunto de regiões) com a descrição do elemento (um vetor intervalar).

Seja ω um novo elemento, que é candidato a ser associado a uma classe $C_k(k = 1, \ldots, m)$, e sua descrição correspondente dada por um vetor intervalar contínuo $\mathbf{x}_\omega = (x_{\omega 1}, \ldots, x_{\omega p})$ onde $x_{\omega j} = [a_{\omega j}, b_{\omega j}]$ é um intervalo. Lembrando que da etapa de aprendizado é obtido os subconjuntos $C_k^1, \ldots, C_k^{v_k}$ de C_k.

Algorithm 2 Etapa de alocação não difusa

Para um novo elemento ω faça:

1: Para cada classe $k = 1, \ldots, m$ calcule
$\delta(\omega, C_k) = min\{d(\omega, C_k^1), \ldots, d(\omega, C_k^{v_k})\}$ onde

$$d(\omega, C_k^s)$$

é uma das funções de dissimilaridade apresentadas abaixo.
$s = 1, \ldots, v_k$
2: Aloque ω a classe C_k

$$\delta(\omega, C_k) \leq \delta(\omega, C_h), \forall h \in \{1, \ldots, m\}$$

Algorithm 3 Etapa de alocação difusa

Para um novo elemento ω faça:

1: Para cada classe $k = 1, \ldots, m$ calcule

$$u_k(\omega) = \frac{\sum_{t=1}^{n} u_{ik}(1/d(\omega, C^t)^{2/m-1})}{\sum_{t=1}^{n}(1/d(\omega, C^t)^{2/m-1})}$$

onde t é o número de hiper-cubos da saída do treinamento e a variável u_{ik} determina o grau de associação (que varia entre 0 e 1) que o objeto C^t apresenta à classe C_k. Como é conhecido *a priori* a classe ao qual o objeto C^t pertence, o seu grau de associação é sempre 1 para a própria classe e 0 para qualquer outra classe.

2: Aloque ω a classe C_k

$$u_k(\omega) \geq u_h(\omega), \forall h \in \{1, \ldots, m\}$$

onde $u_k(\omega)$ é o grau de associação de ω em C_k.

A variável m [4] determina quão fortemente a distância é ponderada quando se calcula a contribuição que cada vizinho dá ao grau de assossiação da classe. Quanto maior o m os vizinhos são ponderados de forma mais uniforme e as distâncias relativas entre o elemento a ser classificado passam a influenciar menos. Quando o m se aproxima de 1, o peso dos vizinhos mais próximos é muito maior do que os mais distantes. Neste trabalho m teve os seguintes valores: 1.5, 2 e 3.

3.3 Funções de dissimilaridade

Algumas dessas funções foram introduzidas em outros trabalhos por De Carvalho [11], Palumbo [12] e Souza [3], estas funções serão respectivamente apresentadas como DC, P e S. As novas funções introduzidas neste trabalho serão apresentadas como SS (Salazar-Souza).

3.3.1 Funções de De Carvalho

a) **DC1**

$$d(\omega, C_k^t) = \frac{\pi(R(C_k^t \oplus \omega)) - \pi(R(C_k^t))}{\pi(R(C_k^t \oplus \omega))} \tag{3.1}$$

b) **DC2**

$$d(\omega, C_k^t) = \sum_{j=1}^{p} \frac{\mu^p(C_k^t \oplus \omega) - \mu^p(R(C_k^t)}{\mu^p(R(C_k^t \oplus \ \omega)} \tag{3.2}$$

onde p é o número de dimensões.

3.3.2 Funções de Palumbo

c) **P1**

$$d(\omega, C_k^t) = \frac{\pi(R(C_k^t \oplus \omega)) - \pi(R(C_k^t))}{\pi(R(C_k^t))} \tag{3.3}$$

d) **P2**

$$d(\omega, C_k^t) = \frac{\pi(R(C_k^t \oplus \omega)) - \pi(R(\omega))}{\pi(R(\omega))} \tag{3.4}$$

3.3.3 Funções de Souza

e) **S1**

$$d(\omega, C_k^t) = \sum_{j=1}^{p} \frac{\mu^p(C_k^t \oplus \omega)) - \mu^p(C_k^t)}{\mu^p(C_k^t)} \tag{3.5}$$

onde p é o número de dimensões.

f) **S2**

$$d(\omega, C_k^t) = \sum_{j=1}^{p} \frac{\mu^p(C_k^t \oplus \omega) - \mu^p(\omega)}{\mu^p(\omega)} \tag{3.6}$$

onde p é o número de dimensões.

3.3.4 Funções de Salazar-Souza

g) **SS1**

$$d(\omega, C_k^t) = \pi(R(C_k^t \oplus \omega)) - \pi(R(C_k^t)) \tag{3.7}$$

h) **SS2**

$$d(\omega, C_k^t) = \pi(R(C_k^t \oplus \omega)) - \pi(R(\omega)) \tag{3.8}$$

i) **SS3**

$$d(\omega, C_k^t) = \sum_{j=1}^{p} \mu^p(C_k^t \oplus \omega) - \mu^p(C_k^t) \tag{3.9}$$

onde p é o número de dimensões.

j) **SS4**

$$d(\omega, C_k^t) = \sum_{j=1}^{p} \mu^p(C_k^t \oplus \omega) - \mu^p(\omega) \tag{3.10}$$

onde p é o número de dimensões.

l) **SS5**

$$d(\omega, C_k^t) = \frac{\pi(R(C_k^t \oplus \omega)) - \pi(R(C_k^t))}{\pi(R(\omega))} \tag{3.11}$$

m) **SS6**

$$d(\omega, C_k^t) = \sum_{j=1}^{p} \frac{\mu^p(C_k^t \oplus \omega)) - \mu^p(C_k^t)}{\mu^p(\omega)} \tag{3.12}$$

onde p é o número de dimensões.

n) **SS7**

$$d(\omega, C_k^t) = \frac{\pi(R(C_k^t \oplus \omega)) - \pi(R(\omega))}{\pi(R(C_k^t))} \tag{3.13}$$

o) **SS8**

$$d(\omega, C_k^t) = \sum_{j=1}^{p} \frac{\mu^p(C_k^t \oplus \omega) - \mu^p(\omega)}{\mu^p(C_k^t)} \tag{3.14}$$

onde p é o número de dimensões.

CAPÍTULO 4

Experimentos e Resultados

Eu não me sinto obrigado a acreditar que o mesmo Deus que nos presenteou com senso, razão e intelecto, intencionava que nós não os usássemos.

—GALILEO GALILEI (astrônomo, filósofo, físico e matemático)

Para avaliar o desempenho dos classificadores propostos, foram realizados experimentos com dois conjuntos de dados simbólicos intervalares sintéticos, tais conjuntos foram gerados com baixo e moderado graus de sobreposição. Também foram realizados experimentos com um conjunto de dados simbólicos intervalares real.

A precisão dos classificadores para os conjuntos de dados sintéticos é aferida através da taxa de erro de classificação, que é estimada na estrutura de uma simulação de Monte Carlo onde o conjunto de teste e de treinamento são selecionados aleatóriamente de cada conjunto de dados sintético. O conjunto de treinamento corresponde a 50% do conjunto de dados original e o conjunto de teste corresponde aos 50% restantes do conjunto de dados original. A precisão dos classificadores para o conjunto de dados real CAR [8][13][14][15] [16] [17] também é aferida pela taxa de erro de classificação, mas por se tratar de um conjunto de dados pequenos, a sua simulação é feita pelo método *leave one out*, onde o primeiro elemento é removido do conjunto para ser classificado e o treinamento é feito com os restantes, em seguida, o primeiro elemento é retornado ao conjunto e o segundo é removido para ser classificado e assim em diante.

4.1 Experimentos com conjuntos de dados sintéticos

Inicialmente, dois conjuntos de dados clássicos quantitativos no $\Re^2$ são gerados (Figuras 4.1 e 4.2). Esses conjuntos de dados têm 500 pontos dispersos entre três classes de tamanhos diferentes: duas classes com forma de elipse e tamanhos 140 e 160 respectivamente e uma classe de formato esférico de tamanho 200.

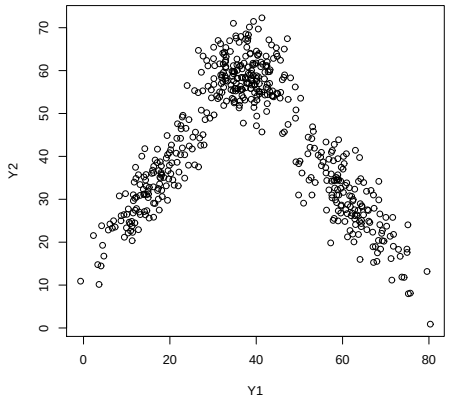

Figura 4.1 Conjunto de dados clássicos quantitativos 1 no $\Re^2$

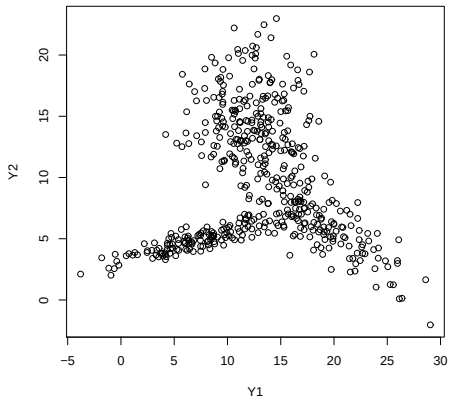

Figura 4.2 Conjunto de dados clássicos quantitativos 2 no $\Re^2$

Cada classe nesses conjuntos de dados intervalares sintéticos foi gerada através de duas distribuições normais independentes. Cada ponto (z_1, z_2) de cada um desses conjuntos de dados clássicos quantitativos é utilizado como "semente"para um vetor de intervalos (retângulo), como definido a seguir:

$$([z_1 - \gamma_1/2, z_1 + \gamma_1/2], [z_2 - \gamma_2/2, z_2 + \gamma_2/2]) \tag{4.1}$$

onde os parâmetros $\gamma_1 e \gamma_2$ são selecionados aleatoriamente de um mesmo intervalo pré-definido.

O conjunto de dados simbólicos intervalares sintéticos 1 foi construído através do conjunto de dados clássicos quantitativos 1, de acordo com os seguintes parâmetros (doravante chama-

dos *configuração 1*):

a) Classe 1: $\mu_1 = 17$, $\mu_2 = 34$, $\sigma_1^2 = 36$, $\sigma_2^2 = 64$ e $\rho_{12} = 0.85$;

b) Classe 2: $\mu_1 = 37$, $\mu_2 = 59$, $\sigma_1^2 = 25$, $\sigma_2^2 = 25$ e $\rho_{12} = 0.0$;

c) Classe 3: $\mu_1 = 61$, $\mu_2 = 31$, $\sigma_1^2 = 49$, $\sigma_2^2 = 100$ e $\rho_{12} = -0.85$;

A figura 4.3 mostra o conjunto de dados intervalares sintéticos 1, no qual γ_1 e γ_2 foram selecionados aleatoriamente do intervalo $[1, 10]$. Esse conjunto de dados intervalar mostra baixo grau de sobreposição.

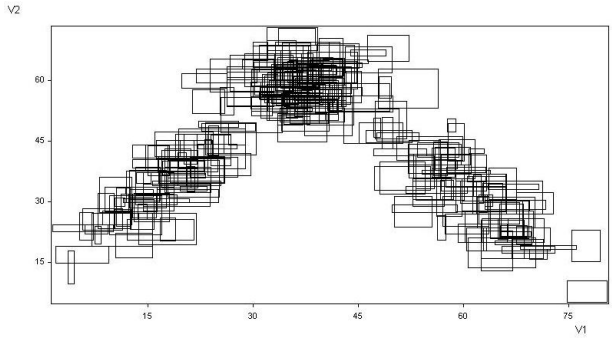

Figura 4.3 Conjunto de dados simbólicos intervalares 1

O conjunto de dados simbólicos intervalares sintéticos 2 foi construído a partir do conjunto de dados clássicos quantitativos 2 de acordo com os seguintes parâmetros (doravante chamados *configuração 2*):

a) Classe 1: $\mu_1 = 8$, $\mu_2 = 5$, $\sigma_1^2 = 16$,$\sigma_2^2 = 1$ e $\rho_{12} = 0.85$;

b) Classe 2: $\mu_1 = 12$, $\mu_2 = 15$, $\sigma_1^2 = 9$, $\sigma_2^2 = 9$ e $\rho_{12} = 0.0$;

c) Classe 3: $\mu_1 = 18$, $\mu_2 = 7$, $\sigma_1^2 = 16$, $\sigma_2^2 = 9$ e $\rho_{12} = -0.85$;

A figura 4.4 mostra o conjunto de dados intervalares sintéticos 2, no qual γ_1 e γ_2 foram selecionados aleatoriamente do mesmo intervalo $[1, 10]$. Esse conjunto de dados intervalar mostra um grau elevado de sobreposição.

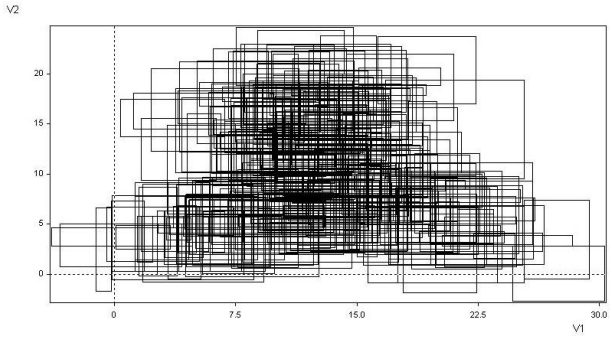

Figura 4.4 Conjunto de dados simbólicos intervalares 2

Na estrutura da simulação de Monte Carlo, 100 iterações desse processo de geração são realizados. Desta forma, evita-se que haja um *overfitting* do aprendizado. Os parâmetros γ_1 e γ_2 são escolhidos aleatoriamente 100 vezes de cada um dos intervalos: $[1,10]$, $[1,20]$, $[1,30]$ e $[1,40]$.

A figura 4.5 mostra o conjunto de dados aleatoriamente selecionado dos elementos apresentados na figura 3 como entrada do algoritmo de aprendizado.

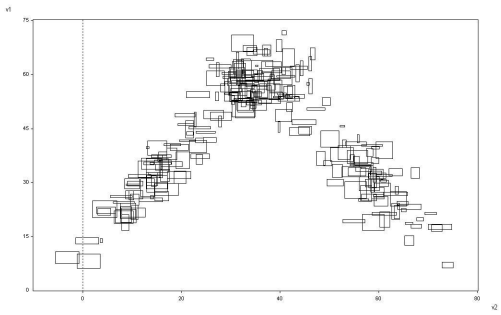

Figura 4.5 Conjunto de entrada do aprendizado 1

A figura 4.6 mostra o conjunto de dados aleatoriamente selecionado dos elementos apresentados na figura 4 como entrada do algoritmo de aprendizado.

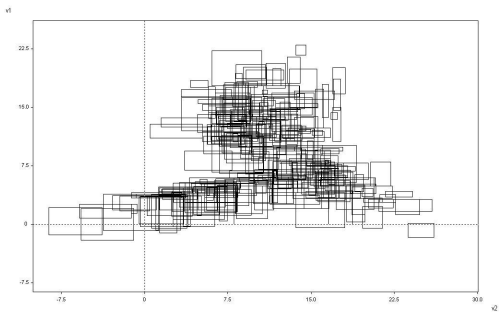

Figura 4.6 Conjunto de entrada do aprendizado 2

Após o treinamento ser concluído, o conjunto de saída do aprendizado apresentará uma quantidade de elementos menor ou igual ao conjunto de entrada. Sendo apresentado abaixo os respectivos conjuntos de saída para os conjuntos de entrada anteriormente apresentados.

A figura 4.7 mostra o conjunto de saída do aprendizado para a configuração 1.

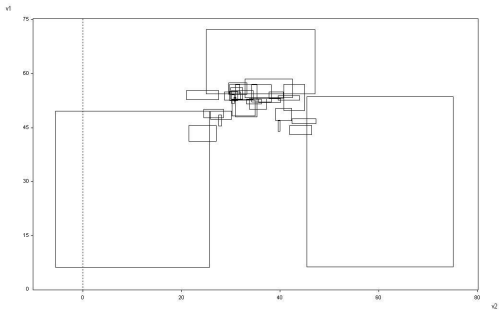

Figura 4.7 Conjunto de saída do aprendizado 1

A figura 4.8 mostra o conjunto de saída do aprendizado para a configuração 2.

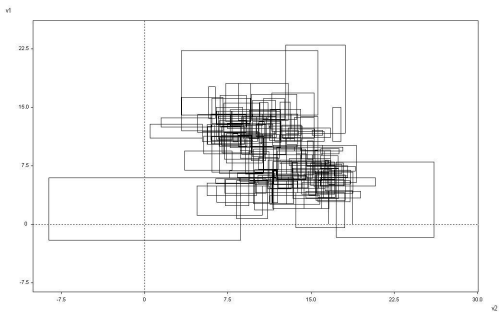

Figura 4.8 Conjunto de saída do aprendizado 2

Após a etapa de treinamento, os classificadores difuso e não-difuso são aplicados aos conjuntos de dados simbólicos e cada uma das funções de distância utilizadas para calcular a similaridade. A taxa de erro de classificação é calculada para o conjunto de teste e a taxa de erro estimada de classificação corresponde à média das taxas de erro encontradas nas 100 iterações do conjunto de teste para cada intervalo de γ_1 e γ_2. Além disso, é calculado o valor médio e os desvio-padrão da quantidade de hiper-cubos formados nas 100 iterações para avaliar a capacidade de compactação do algoritmo de aprendizado.

4.1.1 Resultados do conjunto de dados 1

As tabelas 4.1 a 4.14 mostram os valores (em porcentagem) das médias e desvios padrão (em parênteses) da taxa de erro de classificação obtidas com esses classificadores para o conjunto de dados simbólicos intervalares sintético 1 para cada γ_1 e γ_2 escolhidos dos intervalos $[1,10]$, $[1,20]$, $[1,30]$ e $[1,40]$, abaixo de cada intervalo é mostrado o os valores das médias e desvios padrão da quantidade de hiper-cubos formados na saída do treinamento. Os dados intervalares nessa configuração mostram baixo grau de dificuldade.

Tabela 4.1 Conjunto 1: Distância DC1

	crisp	*fuzzy* m=1.5	*fuzzy* m=2	*fuzzy* m=3
[1:10] Média: 20.04 h Desvio: 10.7786 h	0.28324 (0.194168)	0.01836 (0.00887414)	0.02324 (0.011767)	0.1246 (0.0705756)
[1:20] Média: 49.72 h Desvio: 16.4128 h	0.07592 (0.0681645)	0.01944 (0.00827565)	0.0336 (0.0125984)	0.4228 (0.163689)
[1:30] Média: 83.14 h Desvio: 19.0074 h	0.04996 (0.0326509)	0.0264 (0.0108959)	0.0506 (0.023175)	0.46124 (0.167415)
[1:40] Média: 122.23 h Desvio: 17.3999 h	0.04872 (0.0202574)	0.03292 (0.0133009)	0.05352 (0.0204727)	0.3406 (0.160177)

Tabela 4.2 Conjunto 1: Distância DC2

	crisp	*fuzzy* m=1.5	*fuzzy* m=2	*fuzzy* m=3
[1:10] Média: 20.04 h Desvio: 10.7786 h	0.32616 (0.167493)	0.02464 (0.0129209)	0.42152 (0.223077)	0.55896 (0.0959577)
[1:20] Média: 49.72 h Desvio: 16.4128 h	0.12768 (0.084722)	0.15188 (0.169911)	0.5796 (0.0899858)	0.59048 (0.0464785)
[1:30] Média: 83.14 h Desvio: 19.0074 h	0.0864 (0.0439818)	0.24756 (0.176574)	0.58668 (0.0420706)	0.596 (0.0294265)
[1:40] Média: 122.23 h Desvio: 17.3999 h	0.07736 (0.0322997)	0.15188 (0.086473)	0.54812 (0.0779075)	0.59408 (0.031468)

Tabela 4.3 Conjunto 1: Distância P1

	crisp	*fuzzy* m=1.5	*fuzzy* m=2	*fuzzy* m=3
[1:10] Média: 20.04 h Desvio: 10.7786 h	0.28324 (0.194168)	0.89296 (0.0438219)	0.89088 (0.0455862)	0.88668 (0.0492473)
[1:20] Média: 49.72 h Desvio: 16.4128 h	0.07592 (0.0681645)	0.80872 (0.0445004)	0.80372 (0.0456044)	0.80308 (0.0462579)
[1:30] Média: 83.14 h Desvio: 19.0074 h	0.04996 (0.0326509)	0.73524 (0.047651)	0.73332 (0.0474065)	0.743 (0.0439322)
[1:40] Média: 122.23 h Desvio: 17.3999 h	0.04872 (0.0202574)	0.68408 (0.0392448)	0.683 (0.0400934)	0.69348 (0.0409318)

Tabela 4.4 Conjunto 1: Distância P2

	crisp	*fuzzy* m=1.5	*fuzzy* m=2	*fuzzy* m=3
[1:10] Média: 20.04 h Desvio: 10.7786 h	0.02352 (0.0129402)	0.34888 (0.192852)	0.45296 (0.155663)	0.53264 (0.118842)
[1:20] Média: 49.72 h Desvio: 16.4128 h	0.03068 (0.0146866)	0.16296 (0.118624)	0.3712 (0.169242)	0.5682 (0.114698)
[1:30] Média: 83.14 h Desvio: 19.0074 h	0.03328 (0.0178696)	0.145 (0.0529192)	0.26612 (0.133943)	0.52316 (0.146257)
[1:40] Média: 122.23 h Desvio: 17.3999 h	0.044 (0.0207461)	0.22132 (0.0465585)	0.25188 (0.0709534)	0.46024 (0.111835)

Ambas as funções de Palumbo apresentaram resultados insatisfatórios com a abordagem difusa.

Tabela 4.5 Conjunto 1: Distância S1

	crisp	*fuzzy* m=1.5	*fuzzy* m=2	*fuzzy* m=3
[1:10] Média: 20.04 h Desvio: 10.7786 h	0.2802 (0.184409)	0.7346 (0.135548)	0.69972 (0.123416)	0.65536 (0.0949033)
[1:20] Média: 49.72 h Desvio: 16.4128 h	0.0882 (0.0699951)	0.75604 (0.13289)	0.71268 (0.131148)	0.64 (0.099952)
[1:30] Média: 83.14 h Desvio: 19.0074 h	0.0584 (0.0327854)	0.81244 (0.130437)	0.79432 (0.140707)	0.68024 (0.129544)
[1:40] Média: 122.23 h Desvio: 17.3999 h	0.05456 (0.0222173)	0.86476 (0.104124)	0.84476 (0.114976)	0.731 (0.13728)

Tabela 4.6 Conjunto 1: Distância S2

	crisp	*fuzzy* m=1.5	*fuzzy* m=2	*fuzzy* m=3
[1:10] Média: 20.04 h Desvio: 10.7786 h	0.02376 (0.0130208)	0.98064 (0.0200935)	0.97324 (0.0360564)	0.7156 (0.163639)
[1:20] Média: 49.72 h Desvio: 16.4128 h	0.03104 (0.0148391)	0.9898 (0.0107536)	0.83404 (0.126327)	0.6224 (0.083511)
[1:30] Média: 83.14 h Desvio: 19.0074 h	0.03324 (0.0182489)	0.9946 (0.00607618)	0.8508 (0.123984)	0.62236 (0.0691158)
[1:40] Média: 122.23 h Desvio: 17.3999 h	0.04376 (0.0206287)	0.99608 (0.00479933)	0.94616 (0.0588079)	0.68628 (0.140343)

Assim como as funções de Palumbo, as funções de Souza apresentaram resultados insatisfatórios com a abordagem difusa.

Tabela 4.7 Conjunto 1: Distância SS1

	crisp	*fuzzy* m=1.5	*fuzzy* m=2	*fuzzy* m=3
[1:10] Média: 20.04 h Desvio: 10.7786 h	0.10292 (0.114901)	0.34504 (0.19219)	0.45132 (0.152789)	0.5306 (0.11616)
[1:20] Média: 49.72 h Desvio: 16.4128 h	0.02884 (0.0264986)	0.131 (0.116497)	0.36896 (0.16539)	0.555 (0.0933816)
[1:30] Média: 83.14 h Desvio: 19.0074 h	0.03176 (0.0163151)	0.04184 (0.031922)	0.21092 (0.132277)	0.49004 (0.130668)
[1:40] Média: 122.23 h Desvio: 17.3999 h	0.04308 (0.0191654)	0.03128 (0.0143486)	0.09444 (0.0797534)	0.38456 (0.121034)

Tabela 4.8 Conjunto 1: Distância SS2

	crisp	*fuzzy* m=1.5	*fuzzy* m=2	*fuzzy* m=3
[1:10] Média: 20.04 h Desvio: 10.7786 h	0.02576 (0.0118297)	0.43356 (0.163118)	0.51976 (0.123428)	0.56648 (0.0806886)
[1:20] Média: 49.72 h Desvio: 16.4128 h	0.02536 (0.0123964)	0.28052 (0.15115)	0.48904 (0.12052)	0.57724 (0.061177)
[1:30] Média: 83.14 h Desvio: 19.0074 h	0.03112 (0.0161055)	0.1074 (0.0787835)	0.33064 (0.123733)	0.52232 (0.085938)
[1:40] Média: 122.23 h Desvio: 17.3999 h	0.04136 (0.0191685)	0.04672 (0.025056)	0.17792 (0.0862637)	0.4148 (0.095466)

Tabela 4.9 Conjunto 1: Distância SS3

	crisp	*fuzzy* m=1.5	*fuzzy* m=2	*fuzzy* m=3
[1:10] Média: 20.04 h Desvio: 10.7786 h	0.25044 (0.204072)	0.01828 (0.00985097)	0.01752 (0.00844332)	0.03088 (0.022479)
[1:20] Média: 49.72 h Desvio: 16.4128 h	0.06516 (0.0730289)	0.02196 (0.00942329)	0.02192 (0.00816784)	0.12364 (0.125391)
[1:30] Média: 83.14 h Desvio: 19.0074 h	0.0434 (0.0351107)	0.02632 (0.0114986)	0.03028 (0.0133297)	0.14364 (0.116915)
[1:40] Média: 122.23 h Desvio: 17.3999 h	0.04164 (0.0191424)	0.03052 (0.0122331)	0.03412 (0.0142936)	0.0908 (0.0331662)

Tabela 4.10 Conjunto 1: Distância SS4

	crisp	*fuzzy* m=1.5	*fuzzy* m=2	*fuzzy* m=3
[1:10] Média: 20.04 h Desvio: 10.7786 h	0.02004 (0.0100159)	0.4332 (0.164835)	0.52108 (0.127718)	0.56836 (0.0844258)
[1:20] Média: 49.72 h Desvio: 16.4128 h	0.02476 (0.0129947)	0.30552 (0.158764)	0.51404 (0.133496)	0.60572 (0.0748829)
[1:30] Média: 83.14 h Desvio: 19.0074 h	0.03208 (0.0171206)	0.20044 (0.0998141)	0.41664 (0.143422)	0.60808 (0.105592)
[1:40] Média: 122.23 h Desvio: 17.3999 h	0.04256 (0.0194362)	0.22308 (0.0493538)	0.34844 (0.0930581)	0.5768 (0.100267)

Tabela 4.11 Conjunto 1: Distância SS5

	crisp	*fuzzy* m=1.5	*fuzzy* m=2	*fuzzy* m=3
[1:10] Média: 20.04 h Desvio: 10.7786 h	0.02396 (0.0119532)	0.0228 (0.010016)	0.02192 (0.00962464)	0.03292 (0.0178872)
[1:20] Média: 49.72 h Desvio: 16.4128 h	0.02728 (0.0110328)	0.02136 (0.00905486)	0.02952 (0.0107149)	0.20636 (0.113004)
[1:30] Média: 83.14 h Desvio: 19.0074 h	0.03132 (0.0134944)	0.02464 (0.00954518)	0.04168 (0.0173821)	0.28896 (0.121432)
[1:40] Média: 122.23 h Desvio: 17.3999 h	0.03476 (0.0125053)	0.0288 (0.0123548)	0.04408 (0.0182952)	0.24008 (0.0836736)

Esta função de distância apresentou o melhor resultado geral, tanto para o classificador não-difuso quanto para o classificador difuso.

Tabela 4.12 Conjunto 1: Distância SS6

	crisp	*fuzzy* m=1.5	*fuzzy* m=2	*fuzzy* m=3
[1:10] Média: 20.04 h Desvio: 10.7786 h	0.02844 (0.0139587)	0.01724 (0.00846064)	0.01904 (0.00844976)	0.99796 (0.00409859)
[1:20] Média: 49.72 h Desvio: 16.4128 h	0.03068 (0.0132188)	0.02152 (0.00881417)	0.04964 (0.0279147)	0.99056 (0.0250217)
[1:30] Média: 83.14 h Desvio: 19.0074 h	0.03284 (0.014884)	0.03024 (0.01232)	0.0956 (0.0564)	0.96164 (0.0681366)
[1:40] Média: 122.23 h Desvio: 17.3999 h	0.038 (0.013464)	0.03692 (0.0132164)	0.0886 (0.0383307)	0.94484 (0.0744973)

Esta função de distância segue o mesmo modelo da função 3.11, mas utilizando a abordagem de range, da mesma forma que a função anterior, esta função apresentou excelentes resultados.

Tabela 4.13 Conjunto 1: Distância SS7

	crisp	*fuzzy* m=1.5	*fuzzy* m=2	*fuzzy* m=3
[1:10] Média: 20.04 h Desvio: 10.7786 h	0.02576 (0.0118297)	0.02396 (0.0114612)	0.02884 (0.0130635)	0.1268 (0.0732481)
[1:20] Média: 49.72 h Desvio: 16.4128 h	0.02536 (0.0123964)	0.05712 (0.0279146)	0.06492 (0.0292074)	0.4356 (0.184143)
[1:30] Média: 83.14 h Desvio: 19.0074 h	0.03112 (0.0161055)	0.12688 (0.0457299)	0.13776 (0.0490855)	0.48168 (0.190043)
[1:40] Média: 122.23 h Desvio: 17.3999 h	0.04136 (0.0191685)	0.21388 (0.0469296)	0.22176 (0.046263)	0.4032 (0.126697)

Tabela 4.14 Conjunto 1: Distância SS8

	crisp	*fuzzy* m=1.5	*fuzzy* m=2	*fuzzy* m=3
[1:10] Média: 20.04 h Desvio: 10.7786 h	0.02084 (0.00981908)	0.04056 (0.0195552)	0.26296 (0.141771)	0.55828 (0.0963581)
[1:20] Média: 49.72 h Desvio: 16.4128 h	0.02532 (0.0129003)	0.04636 (0.0314914)	0.51292 (0.144101)	0.58308 (0.0700693)
[1:30] Média: 83.14 h Desvio: 19.0074 h	0.0336 (0.0174172)	0.0424 (0.0382539)	0.43172 (0.140309)	0.58308 (0.0441084)
[1:40] Média: 122.23 h Desvio: 17.3999 h	0.0444 (0.0197707)	0.03296 (0.0141887)	0.24172 (0.126797)	0.5226 (0.0941461)

No geral, as funções de distância Salazar-Souza apresentaram resultados excelentes, em ambas as abordagens difusa e não-difusa.

4.1.2 Resultados do conjunto de dados 2

As tabelas 4.15 a 4.28 mostram os valores (em porcentagem) das médias e desvios padrão (em parênteses) da taxa de erro de classificação obtidas com esses classificadores para o conjunto de dados simbólicos intervalares sintético 2 para cada γ_1 e γ_2 escolhidos dos intervalos $[1,10]$, $[1,20]$, $[1,30]$ e $[1,40]$, abaixo de cada intervalo é mostrado o os valores das médias e desvios padrão da quantidade de hiper-cubos formados na saída do treinamento.. Dados intervalares

nessa configuração mostram alto grau de dificuldade.

Tabela 4.15 Conjunto 2: Distância DC1

	crisp	*fuzzy* m=1.5	*fuzzy* m=2	*fuzzy* m=3
[1:10] Média: 145.12 h Desvio: 18.1225 h	0.15004 (0.036479)	0.08732 (0.0185994)	0.09232 (0.020315)	0.18456 (0.106272)
[1:20] Média: 218.97 h Desvio: 8.12952 h	0.1914 (0.046714)	0.11224 (0.0233979)	0.11592 (0.0230337)	0.1286 (0.0289351)
[1:30] Média: 233.14 h Desvio: 6.15633 h	0.24384 (0.038991)	0.13912 (0.0279747)	0.1476 (0.0305863)	0.18292 (0.0422217)
[1:40] Média: 238.87 h Desvio: 5.12183 h	0.26188 (0.0464188)	0.1688 (0.0363846)	0.18128 (0.0413023)	0.244 (0.0716949)

Tabela 4.16 Conjunto 2: Distância DC2

	crisp	*fuzzy* m=1.5	*fuzzy* m=2	*fuzzy* m=3
[1:10] Média: 145.12 h Desvio: 18.1225 h	0.19608 (0.0476621)	0.12844 (0.0640922)	0.33248 (0.102893)	0.46 (0.120967)
[1:20] Média: 218.97 h Desvio: 8.12952 h	0.19868 (0.0479868)	0.11964 (0.0245673)	0.1624 (0.0462584)	0.35504 (0.14313)
[1:30] Média: 233.14 h Desvio: 6.15633 h	0.2464 (0.0394482)	0.15344 (0.0301225)	0.22832 (0.0553929)	0.46048 (0.106367)
[1:40] Média: 238.87 h Desvio: 5.12183 h	0.26416 (0.0473588)	0.1872 (0.040044)	0.29708 (0.0756917)	0.51532 (0.0720401)

Tabela 4.17 Conjunto 2: Distância P1

	crisp	*fuzzy* m=1.5	*fuzzy* m=2	*fuzzy* m=3
[1:10] Média: 145.12 h Desvio: 18.1225 h	0.15004 (0.036479)	0.60268 (0.0575362)	0.59644 (0.0578547)	0.59308 (0.0603824)
[1:20] Média: 218.97 h Desvio: 8.12952 h	0.1914 (0.046714)	0.55164 (0.0414385)	0.54588 (0.0423337)	0.54136 (0.0424235)
[1:30] Média: 233.14 h Desvio: 6.15633 h	0.24384 (0.038991)	0.54908 (0.0415533)	0.54548 (0.0407476)	0.54368 (0.0403888)
[1:40] Média: 238.87 h Desvio: 5.12183 h	0.26188 (0.0464188)	0.5556 (0.0416442)	0.552 (0.0416615)	0.54944 (0.0424359)

Tabela 4.18 Conjunto 2: Distância P2

	crisp	*fuzzy* m=1.5	*fuzzy* m=2	*fuzzy* m=3
[1:10] Média: 145.12 h Desvio: 18.1225 h	0.10928 (0.0235075)	0.2772 (0.0435449)	0.2642 (0.0430158)	0.29844 (0.0673591)
[1:20] Média: 218.97 h Desvio: 8.12952 h	0.16904 (0.0382397)	0.49696 (0.0417387)	0.48292 (0.039089)	0.47136 (0.0384405)
[1:30] Média: 233.14 h Desvio: 6.15633 h	0.2246 (0.0340323)	0.60212 (0.0426088)	0.5908 (0.0427663)	0.578 (0.0414439)
[1:40] Média: 238.87 h Desvio: 5.12183 h	0.27168 (0.035071)	0.63356 (0.0414763)	0.62504 (0.0402139)	0.61416 (0.0394093)

Assim como na configuração 1, as funções de Palumbo não serviram para a abordagem difusa.

Tabela 4.19 Conjunto 2: Distância S1

	crisp	*fuzzy* m=1.5	*fuzzy* m=2	*fuzzy* m=3
[1:10] Média: 145.12 h Desvio: 18.1225 h	0.15308 (0.0361236)	0.91416 (0.0689589)	0.92976 (0.0420222)	0.90356 (0.0691014)
[1:20] Média: 218.97 h Desvio: 8.12952 h	0.1914 (0.0459508)	0.91296 (0.0607477)	0.93964 (0.0425586)	0.94128 (0.0356332)
[1:30] Média: 233.14 h Desvio: 6.15633 h	0.24432 (0.0391129)	0.86952 (0.065787)	0.90272 (0.058869)	0.91268 (0.0604121)
[1:40] Média: 238.87 h Desvio: 5.12183 h	0.2636 (0.0449052)	0.80216 (0.0780716)	0.84696 (0.0841534)	0.86196 (0.0858529)

Tabela 4.20 Conjunto 2: Distância S2

	crisp	*fuzzy* m=1.5	*fuzzy* m=2	*fuzzy* m=3
[1:10] Média: 145.12 h Desvio: 18.1225 h	0.1094 (0.0238554)	0.9478 (0.0185073)	0.92516 (0.0297955)	0.88612 (0.0662346)
[1:20] Média: 218.97 h Desvio: 8.12952 h	0.1684 (0.0376149)	0.96344 (0.0126367)	0.96828 (0.0116104)	0.95544 (0.0249673)
[1:30] Média: 233.14 h Desvio: 6.15633 h	0.2242 (0.0337775)	0.95476 (0.0168256)	0.9598 (0.0145588)	0.93372 (0.030949)
[1:40] Média: 238.87 h Desvio: 5.12183 h	0.27152 (0.0352331)	0.9476 (0.0182822)	0.94952 (0.0201338)	0.88724 (0.0502528)

Novamente, assim como na configuração 1, fica claro que as funções de Souza não funcionam apresentam resultados insatisfatórios com uma abordagem difusa.

Tabela 4.21 Conjunto 2: Distância SS1

	crisp	*fuzzy* m=1.5	*fuzzy* m=2	*fuzzy* m=3
[1:10] Média: 145.12 h Desvio: 18.1225 h	0.11104 (0.0222216)	0.1064 (0.0228526)	0.10364 (0.0359782)	0.18308 (0.100049)
[1:20] Média: 218.97 h Desvio: 8.12952 h	0.16936 (0.0373619)	0.1226 (0.0231128)	0.10408 (0.01944)	0.1044 (0.0216998)
[1:30] Média: 233.14 h Desvio: 6.15633 h	0.23692 (0.0397225)	0.14952 (0.0313982)	0.12032 (0.0242367)	0.12384 (0.0250179)
[1:40] Média: 238.87 h Desvio: 5.12183 h	0.28772 (0.0409732)	0.16428 (0.0337627)	0.12876 (0.0263739)	0.14204 (0.0351249)

Tabela 4.22 Conjunto 2: Distância SS2

	crisp	*fuzzy* m=1.5	*fuzzy* m=2	*fuzzy* m=3
[1:10] Média: 145.12 h Desvio: 18.1225 h	0.1078 (0.0220499)	0.10072 (0.0219336)	0.11588 (0.0486669)	0.2128 (0.0929791)
[1:20] Média: 218.97 h Desvio: 8.12952 h	0.1646 (0.0374182)	0.14092 (0.0286879)	0.1136 (0.019992)	0.1084 (0.0198273)
[1:30] Média: 233.14 h Desvio: 6.15633 h	0.21988 (0.0330161)	0.20368 (0.0355648)	0.16488 (0.0319979)	0.13844 (0.0270401)
[1:40] Média: 238.87 h Desvio: 5.12183 h	0.26836 (0.0339274)	0.2484 (0.0477711)	0.20988 (0.0406289)	0.17152 (0.0263046)

É interessante notar que ao contrário da configuração 1, na configuração 2, a função 3.8 apresentou resultado geral melhor quando $m = 3$, ao contrário de todas as outras funções em que um m maior implica em uma maior homogeneidade entre os hipercubos.

Tabela 4.23 Conjunto 2: Distância SS3

	crisp	*fuzzy* m=1.5	*fuzzy* m=2	*fuzzy* m=3
[1:10] Média: 145.12 h Desvio: 18.1225 h	0.1316 (0.032929)	0.08864 (0.0188613)	0.08468 (0.0162289)	0.16056 (0.088381)
[1:20] Média: 218.97 h Desvio: 8.12952 h	0.18392 (0.0468461)	0.11744 (0.0283606)	0.10732 (0.0222211)	0.11044 (0.0217322)
[1:30] Média: 233.14 h Desvio: 6.15633 h	0.23796 (0.0422772)	0.14896 (0.0320356)	0.13568 (0.0279524)	0.13952 (0.0259247)
[1:40] Média: 238.87 h Desvio: 5.12183 h	0.25568 (0.0481985)	0.18496 (0.032324)	0.16904 (0.03334)	0.17028 (0.0315532)

Tabela 4.24 Conjunto 2: Distância SS4

	crisp	*fuzzy* m=1.5	*fuzzy* m=2	*fuzzy* m=3
[1:10] Média: 145.12 h Desvio: 18.1225 h	0.109 (0.0229809)	0.26224 (0.0418314)	0.27416 (0.0468373)	0.34148 (0.0735122)
[1:20] Média: 218.97 h Desvio: 8.12952 h	0.16524 (0.0378193)	0.48836 (0.0410545)	0.47652 (0.0400666)	0.4706 (0.0394132)
[1:30] Média: 233.14 h Desvio: 6.15633 h	0.22164 (0.0336433)	0.59552 (0.0422877)	0.58464 (0.0416911)	0.57508 (0.0421421)
[1:40] Média: 238.87 h Desvio: 5.12183 h	0.2694 (0.0348546)	0.62808 (0.0425882)	0.61824 (0.0398128)	0.61476 (0.0397097)

Tabela 4.25 Conjunto 2: Distância SS5

	crisp	*fuzzy* m=1.5	*fuzzy* m=2	*fuzzy* m=3
[1:10] Média: 145.12 h Desvio: 18.1225 h	0.11388 (0.0234228)	0.08556 (0.0167131)	0.08732 (0.0165704)	0.1424 (0.0650268)
[1:20] Média: 218.97 h Desvio: 8.12952 h	0.16532 (0.0333517)	0.10872 (0.0222522)	0.10584 (0.0202419)	0.11352 (0.0223913)
[1:30] Média: 233.14 h Desvio: 6.15633 h	0.2156 (0.035582)	0.12804 (0.0243606)	0.12668 (0.0228809)	0.14488 (0.0263975)
[1:40] Média: 238.87 h Desvio: 5.12183 h	0.23956 (0.0396016)	0.15196 (0.027291)	0.14612 (0.0268713)	0.18844 (0.0371716)

Tabela 4.26 Conjunto 2: Distância SS6

	crisp	*fuzzy* m=1.5	*fuzzy* m=2	*fuzzy* m=3
[1:10] Média: 145.12 h Desvio: 18.1225 h	0.11896 (0.0243286)	0.0872 (0.0170552)	0.09996 (0.0258828)	0.91428 (0.076548)
[1:20] Média: 218.97 h Desvio: 8.12952 h	0.1744 (0.0373695)	0.1148 (0.0202188)	0.1196 (0.0211547)	0.9638 (0.0147445)
[1:30] Média: 233.14 h Desvio: 6.15633 h	0.22808 (0.0354445)	0.14364 (0.0273881)	0.15188 (0.0272556)	0.94412 (0.0160445)
[1:40] Média: 238.87 h Desvio: 5.12183 h	0.25316 (0.0415473)	0.17472 (0.0309471)	0.1826 (0.0320967)	0.9266 (0.0200908)

Tabela 4.27 Conjunto 2: Distância SS7

	crisp	fuzzy m=1.5	fuzzy m=2	fuzzy m=3
[1:10] Média: 145.12 h Desvio: 18.1225 h	0.1078 (0.0220499)	0.24576 (0.0415551)	0.24216 (0.0404986)	0.2944 (0.0771222)
[1:20] Média: 218.97 h Desvio: 8.12952 h	0.1646 (0.0374182)	0.4712 (0.0386285)	0.47096 (0.0387014)	0.47336 (0.0384758)
[1:30] Média: 233.14 h Desvio: 6.15633 h	0.21988 (0.0330161)	0.57712 (0.0422718)	0.57768 (0.0413112)	0.584 (0.0402433)
[1:40] Média: 238.87 h Desvio: 5.12183 h	0.26836 (0.0339274)	0.6164 (0.0387629)	0.61796 (0.0388577)	0.62704 (0.0385378)

Tabela 4.28 Conjunto 2: Distância SS8

	crisp	*fuzzy* m=1.5	*fuzzy* m=2	*fuzzy* m=3
[1:10] Média: 145.12 h Desvio: 18.1225 h	0.10928 (0.0233983)	0.08576 (0.0170957)	0.13368 (0.0706183)	0.2582 (0.100985)
[1:20] Média: 218.97 h Desvio: 8.12952 h	0.1684 (0.0383145)	0.10924 (0.0179216)	0.11064 (0.0180508)	0.12312 (0.0235598)
[1:30] Média: 233.14 h Desvio: 6.15633 h	0.22348 (0.033369)	0.13988 (0.0241161)	0.13896 (0.0233143)	0.15492 (0.0253163)
[1:40] Média: 238.87 h Desvio: 5.12183 h	0.2732 (0.0343884)	0.16656 (0.0252889)	0.1614 (0.0234154)	0.19 (0.0355325)

Assim como na configuração 1, as funções de distância Salazar-Souza apresentaram resultados geral satisfatórios, em ambas as abordagens difusa e não-difusa.

4.2 Experimentos com conjunto de dados real

O conjunto de dados do tipo intervalo CAR tem sido utilizado para validar diversos classificadores supervisionados e não-supervisionados na literatura de dados simbólicos. [8] [16] [17] [13] [14] [15] Este conjunto apresenta 8 atributos e um rótulo, sendo que 4 destes atributos são intervalos e os outros 4 são pontos e o rótulo apresenta 4 classes. O conjunto possui

apenas 33 elementos, por isso, para a simulação foi utilizado o método deixe-um-fora (*leave-one-out*) conforme explicado anteriormente.A tabela 4.29 mostra a taxa de erro de classificação obtidas com os classificadores para o conjunto de dados simbólicos CAR. Todas as saídas de treinamento das 33 simulações sempre geraram 4 hipercubos (um para cada classe). Como mencionado anteriormente, Ichino *et al.* [2] introduziram um classificador onde os objetos de entrada eram pontos, mas que também poderia ser extendido para intervalos, onde, os pontos, nada mais são do que intervalos degenerados (onde o limite inferior é igual ao limite superior). Este trabalho de graduação, extende a sua idéia, mostrando a sua eficácia.

Tabela 4.29 Conjunto de dados CAR

	crisp	*fuzzy* m=1.5	*fuzzy* m=2	*fuzzy* m=3
DC1	0.69697	0.393939	0.393939	0.393939
DC2	0.181818	0.393939	0.393939	0.393939
P1	0.69697	0.393939	0.393939	0.393939
P2	1	1	1	1
S1	0.69697	0.969697	0.969697	0.969697
S2	1	1	1	1
SS1	0.242424	0.363636	0.363636	0.363636
SS2	0.454545	0.606061	0.606061	0.606061
SS3	0.545455	0.393939	0.393939	0.393939
SS4	0.545455	0.606061	0.606061	0.606061
SS5	1	1	1	1
SS6	1	1	1	1
SS7	0.181818	0.393939	0.393939	0.393939
SS8	0.69697	0.424242	0.424242	0.424242

CAPÍTULO 5

Conclusão

Um projetista sabe que atingiu a perfeição, não quando não há mais nada a ser adicionado, mas quando não há mais nada a ser removido.

—ANTOINE DE SAINT-EXUPÉRY (aviador e escritor)

Com os resultados obtidos dos conjuntos de dados sintéticos é possível observar que o comportamento do classificador difuso (com m=1.5) apresenta resultado geral superior ao classificador não-difuso, o que não chega a ser uma surpresa, pois classificadores difusos foram criados para lidar com sobreposições. Como explicado anteriormente, quanto maior o valor de m, mais homogêneo ficam o peso de todos os hiper-cubos, enquanto que com m=1.5, os hiper-cubos mais próximos apresentam um peso maior que os hiper-cubos mais distantes, o que explica a sua maior taxa de erro. Em compensação, as funções de dissimilaridade 3.3, 3.4, 3.5, 3.6, 3.8 (apenas na configuração 1), 3.10 e 3.13 apresentaram péssimos resultados com o classificador difuso enquanto as mesmas funções, aplicadas ao classificador não-difuso apresentaram resultados satsifatórios. É necessário salientar um certo comportamento anômalo detectado em alguns dos classificadores não-difusos (e em alguns poucos casos difusos), onde a taxa de erro da configuração 1 apresenta um comportamento decrescente conforme o intervalo usado aumenta. O esperado seria que com menos sobreposições, a classificação deveria ser mais fácil nestes casos. As seguintes funções de dissimilaridade apresentaram este comportamento: 3.1, 3.2, 3.5, 3.7, 3.8, e 3.9 (não-difusos) e 3.3, 3.4 e 3.10 (difusos). O motivo para este comportamento ainda está sendo investigado. Dentre as funções apresentadas 3.11 e 3.12 apresentaram o melhor resultado geral, tanto para o classificador não-difuso e difuso, quanto para as duas configurações de dados sintéticos.

Os resultados conseguidos do conjunto de dados CAR merece algumas explicações iniciais. Como mencionado anteriormente, 4 dos seus atributos são pontos, o que significa que ao calcular o seu volume em 8 dimensões, os elementos de teste apresentarão volume igual a zero, o que já elimina as funções 3.4 e 3.11 pois irão apresentar divisão por zero. O mesmo ocorre com as funções 3.6 e 3.12 ao calcular o *range* de uma dimensão cujo valor seja um ponto. Além disso, em todas as 33 simulações de treinamento, sempre foram formados 4 hiper-cubos (sem sobreposição) o que explica o desempenho mediano dos classificadores difusos contra os não-difusos. É importante ressaltar que o resultado obtido com as funções 3.2 e 3.13 apresentam resultado melhor do que o apresentado na atual literatura.

Como trabalhos futuros, serão propostos testes para outras bases de dados, tanto com um maior número de elementos como uma quantidade maior de atributos do tipo intervalo, assim como testar outras funções de associação difusa para avaliar o seu desempenho, especialmente

para verificar se as funções de Palumbo e Souza não funcionam com abordagens difusas ou apenas com a função de associação difusa de Keller.

Referências Bibliográficas

[1] BOCK, H.; DIDAY, E. ***Analysis of Symbolic Data***. 1. ed. ed. 2000. 1, 2.2

[2] DIDAY, E.; ICHINO, M.; YAGUCHI, H. **Symbolic pattern classifiers based on the cartesian system model**. *Ordinal and Symbolic Data Analysis*, p. 92–102, 1996. 1, 2.2.2, 4.2

[3] SOUZA, R. D. ***Classificação de Imagens SAR baseadas em uma abordagem simbólica***. Mar. 1999. Dissertação (Mestrado em Ciência da Computação) - Universidade Federal de Pernambuco, Mar. 1999. 1, 3.1.1, 3.3

[4] GIVENS, J.; GRAY, M.; KELLER, J. **A fuzzy K-nearest neighbor algorithm**. *IEEE Trans. Systems Man Cybernetic*, v. 15, p. 580–585, 1985. 1, 3.2.2

[5] TUKEY, J. ***Exploratory Data Analysis***. Reading: Addison Wesley, 1958. 2.2

[6] SNEATH, P.; SOKAL, R. ***Numerical Taxonomy***. San Francisco: Freeman, 1973. 2.2

[7] DIDAY, E. **La méthode des nueés dynamiques**. *Revve de Statist*, v. 19, n. 2, p. 19–34, 1971. 2.2

[8] BRITO, P.; SILVA, A. **Linear Discriminant Analysis for Interval Data**. *Computational Statistics*, v. 21, p. 289–308, 2006. 2.2.2, 4, 4.2

[9] DIDAY, E.; ICHINO, M.; YAGUCHI, H. ***A fuzzy symbolic pattern classifier***. 2.. ed. Springer, 1998. 3.1.2

[10] CARVALHO, F. A. T. D.; FRERY, A. C.; SOUZA, R. D. **Symbolic approach to SAR image classification**. *International Geoscience and Remote Sensing Symposium*, p. 1318–1320, 1999. 3.1.3

[11] CARVALHO, F. A. T. D.; DOLIVEIRA JUNIOR, S. T.; SOUZA, R. M. C. R. D. **A Classifier for Quantitative Feature Values based on a Region Oriented Symbolic Approach**. *Lectures Notes in Artificial Intelligence*, p. 464–473, 2004. 3.3

[12] BENEDETTO, M.; PALUMBO, F. **A generalisation measure for symbolic objects**. *Knowledge Extraction form Statistical Data*, 1998. 3.3

[13] CARVALHO, F. D. **Fuzzy c-means Clustering Methods for Symbolic Interval Data**. *Pattern Recognition Letters*, v. 28, p. 423–437, 2007. 4, 4.2

[14] CARVALHO, F. D.; PIMENTEL, J.; SOUZA, L. E.; SOUZA, R. D. **Clustering symbolic interval data based on a single adaptive Hausdorff distance**. *IEEE International Conference on Systems, Man and Cybernetics*, p. 451–455, 2007. 4, 4.2

[15] CARVALHO, F. D.; PIMENTEL, J.; SOUZA, L. E.; SOUZA, R. D. **A dynamical clustering method for symbolic interval data based on a single adaptive Euclidean distance**. *Proceedings of the Symposium on Artificial Neural Networks*, v. 8, 2008. 4, 4.2

[16] BOCK, H.; BRITO, P.; CARVALHO, F. D. **Dynamic Clustering for Interval Data Based on L2 Distance**. *Computational Statistics*, v. 21, p. 231–250, 2006. 4, 4.2

[17] CARVALHO, F. D.; PIZZATO, D.; SOUZA, R. D. **A Partitioning Method for Mixed Feature-Type Symbolic Data using a Squared Euclidean Distance**. *Lecture Notes on Artificial Intelligence*, , n. 29, p. 260–273, 2006. 4, 4.2

Apêndice A: Assinaturas

_____

Renata Maria Cardoso Rodrigues de Souza
Orientadora

_____

Diogo Rodrigues dos Santos Salazar
Aluno