# Indução de Classificadores Levando em Conta a Imprecisão dos Dados

Jorge Luis Guevara Díaz

Exame de Qualificação de Doutorado
apresentada ao
Instituto de Matemática e Estatística
da
Universidade de São Paulo

Programa: Ciência da Computação

Orientador: Prof. Dr. Roberto Hirata Jr

Durante o desenvolvimento do trabalho o autor recebeu auxílio financeiro da CAPES

São Paulo, maio de 2012

# Resumo

Um problema intrínseco na área de Aprendizado Estatístico e Computacional é o tratamento de conjuntos de dados com imprecisão. Várias podem ser as fontes de problema nos dados e pelo menos o erro de medida estará presente num conjunto de dados real. Até o presente momento, até onde conhecemos, poucos esforços foram feitos para tratar este problema no ámbito dos algoritmos e métodos de aprendizado. Em geral, o problema é tratado na fase de pré-processamento dos dados através, por exemplo, da filtragem de ruídos e, ou, preenchimento de valores faltantes, antes da indução do classificador. Neste trabalho, tratamos o problema durante a indução do classificador usando a teoria dos conjuntos difusos e métodos kernel. Primeiramente, definiu-se uma forma de agregar a informação de imprecisão através da fuzificação dos dados usando números difusos. Depois, foi definido um kernel usando esses números difusos e provou-se que ele é um kernel de Mercer, ou seja, pode ser usado nos métodos de kernel de aprendizado. Finalmente, foram feitos diversos testes para comparar a acurácia dos novos classificadores com os classificadores da literatura e os resultados são promissores.

**Palavras-chave:** conjunto difuso, métodos kernel, imprecisão.

# Sumário

# Capítulo 1

# Introdução

Neste capítulo, são apresentados a motivação da pesquisa e a definição do problema. Este capítulo também descreve o cronograma do doutorado.

## 1.1 Motivação

Medidas tomadas do mundo real não são perfeitas. Essa imperfeição é devida a *incerteza*, *inconsistência* ou *imprecisão*. A *incerteza* acontece quando não é possível determinar a verdade o falsidade da propriedade de um objeto. A medida difusa de Sugeno é usada para tratar a incerteza [MS89], tendo como casos específicos a medida de probabilidade, a medida da possibilidade[Zad99, DP88] e a medida de crença [Ban81]. Por outro lado, a *inconsistência* também é uma fonte de imperfeição nos dados, a qual é dada por dados que apresentam conflito de informação.

Uma fonte importante de imperfeição nos dados é a *imprecisão*, a qual é definida como a falta de precisão nos valores dos dados [Sme91]. A imprecisão pode ser de vários tipos, tal como descreve a Tabela 1.1. O caso de dados incompletos é um caso extremo de imprecisão, isto é, a falta total de precisão.

| Tipo | Descrição | Exemplo |
|---|---|---|
| Incompleta. | Os dados possuem valores faltantes. | Idade=' '. |
| Vaga. | Os dados possuem valores não bem definidos. | João é alto,$x \in [-3, 3]$. |
| Aproximada. | Os valores dos dados são bem definidos. e próximos do valor real. | Idade = anos 30. $x \in [-3, -2.5]$. |
| Ambígua. | A informação contida nos dados possui significados diferentes. | Comida=quente. |
| Com erro. | Dados possuem erro devido a procedimentos de adquisição, medida e preprocessamento. | Dados de experimentos microarray. |
| Inválida e sem sentido | Ruído nos dados. | Idade=245. |

**Tabela 1.1:** *Tipos de imprecisão nos dados. Tabela baseada em [Sme96]*

A modelagem da imprecisão nos dados é feita mediante intervalos ou usando conjuntos difusos, como por exemplo usando números difusos [AFG+00, VOOS10, CS06b, SCC06, PAF12]. Nesse sentido, a modelagem de problemas do mundo real, tem que ter em conta a imperfeição dos dados, em particular a imprecisão, para poder obter modelos confiáveis e precisos.

## 1.2 Definição do problema

Um problema intrínseco à área de Aprendizado Estatístico e Computacional é o tratamento de conjuntos de dados com imprecisão. Nesse sentido, várias podem ser as fontes de problemas

nos dados como mostra a Tabela 1.1. Em geral, é difícil saber se um conjunto de dados é livre de imprecisão no mundo real. Até onde conhecemos, poucos esforços foram feitos na comunidade de Aprendizado Estatístico e Computacional para tratar esse problema no âmbito de algoritmos e métodos. Usualmente o problema é tratado como um problema de pré-processamento dos dados, isto é, fazendo uma filtragem de ruídos, ou via preenchimento de valores faltantes.

O objectivo deste trabalho é a construção de classificadores supervisionados que levem em conta a imprecisão dos dados, em outras palavras:

***Como considerar a imprecisão dos dados no desenho de classificadores supervisionados?***

Dado que a teoria difusa permite tratar a imprecisão, usaremos ela para modelar a imprecisão nos dados de entrada, no contexto de classificadores supervisionados.

Por outro lado, os métodos kernel exploram *padrões* complexos nos dados, ou seja, relações, estruturas ou regularidades nos dados [STC04]. Eles possuem duas componentes: uma *função kernel* que faz o mapeamento dos dados num espaço de dimensionalidade maior chamado o *espaço de características*. A outra componente é um algoritmo de análise de padrões que trabalha no espaço de características. Exemplos de métodos kernel são: as máquinas de vetores de suporte, o discriminante de Fisher baseado em kernel, a análise de componentes principais baseadas em kernel entre outros.

As vantagens de usar os métodos kernel são as seguintes:

- *podem tratar problemas não lineares no espaço de entrada.* Os dados são mapeados num espaço vetorial de dimensionalidade maior chamado de espaço de características. Um algoritmo procura padrões nesse espaço, onde somente precisa-se conhecer o produto interno das imagens dos dados e não o sistema de coordenadas. A função kernel permite computar eficientemente esses produtos internos;

- *modularidade.* A função kernel é especifica dos dados e pode-se combinar com diferentes algoritmos. Da mesma maneira, o algoritmo de análise de padrões pode-se combinar com diferentes kernels;

- *o uso de kernels habilita a aplicação de algoritmos a dados não vetoriais.* Os métodos kernel não requerem que os dados de entrada sejam vetores. Os dados de entrada podem ser de diferentes tipos, por exemplo: cadeias, estruturas discretas, imagens, séries temporais, entre outros.

Este trabalho pretende usar tanto a teoria difusa para modelar a imprecisão, quanto os métodos kernel para a construção de classificadores supervisionados. Através deste estudo pretendemos responder às questões:

- RQ1: Que relação existe entre alguns dos resultados da teoria difusa e métodos kernel? Por exemplo que relação existe entre o classificador difuso baseado em sistemas de lógica difusa e as máquinas de vetores de suporte?

- RQ2: É possível construir kernel positivos definidos cujo domínio seja o espaço de números difusos e usar esses kernels eficientemente?

- RQ3: Que estrategias usar para fuzzificar os dados?

## 1.3 Cronograma de atividades

O cronograma do doutorado apresentado na tabela 1.2 corresponde aos anos 2010-2011. A lista completa de disciplinas cursadas são:

- MAC5714-4/3 Programação Orientada a Objetos.

- MAC5768-4/2 Visão e Processamento de Imagens - Parte I.
- MAC5770-4/5 Introdução à Teoria dos Grafos.
- MAC5920-1/1 Algoritmos para Processamento de Áudio, Imagem e Vídeo.
- MAC5711-10/2 Análise de Algoritmos
- IBI5031-3/1 Reconhecimento de Padrões I (Curso Interunidades: Bioinformática - Universidade de São Paulo).

Também foi realizado um Estágio Supervisionado do Programa de Aperfeiçoamento do Ensino - PAE. Foi realizada também uma monitoria na disciplina de Visão e Processamento de Imagens - Parte I.

| Atividades do doutorado 2010-2011 | | | | |
|---|---|---|---|---|
| | 2010 | | 2011 | |
| Tarefas | 03-06 | 08-12 | 03-06 | 08-12 |
| Disciplinas | x | x | | |
| PAE | x | x | | |
| Monitoria | | | x | |
| Revisão bibliográfica | | | x | x |
| Experimentos | | | | x |

**Tabela 1.2:** *Cronograma geral 2010-2011.*

O cronograma geral do doutorado a partir de janeiro de 2012 até a provável data de defesa da tese é apresentada na Tabela 1.3.

| Atividades do doutorado a partir de janeiro de 2012 | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | 2012 | | | | | | 2013 | | 2014 |
| Tarefas | 01 | 02 | 03 | 04 | 05-09 | 10-12 | 01-09 | 10-12 | 01-08 |
| Estudo e implementações da metodologia proposta | x | x | | | x | | | | |
| Escrita do texto de qualificação | | x | x | x | | | | | |
| Exame de qualificação | | | | | x | | | | |
| Estágio no exterior | | | | | | x | x | | |
| Atividades pós estágio até a defesa da tese | | | | | | | | x | x |

**Tabela 1.3:** *Cronograma geral a partir de janeiro de 2012.*

O cronograma inclui um estágio de um ano no Laboratório de LITIS, no departamento de Architecture des Systèmes d'Information ASI, INSA de Rouen, orientado pelo professor Dr. Stéphane Canu.

Após o retorno ao Brasil, ainda restará cerca de um ano para a conclusão do doutorado. Este tempo será utilizado para dar continuidade ao projeto depois do estágio, para a redação da tese, conclusão de artigos e para a defesa.

Na Tabela 1.4 é apresentado o cronograma específico para as atividades durante o estágio.

| Atividades do estágio no exterior | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 2012 | | | | 2013 | | | | | | | |
| Tarefas | 10 | 11 | 12 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 |
| Adaptação com ambiente de trabalho e grupo de pesquisa | x | | | | | | | | | | | |
| Implementação da proposta | x | x | x | x | x | x | x | x | x | x | | |
| Etapa de testes | | | | | x | x | x | x | x | x | | |
| Validação de resultados | | | | | | | x | x | x | x | x | |
| Produção de artigo(s) | | | | | | | | | | | x | x |

**Tabela 1.4:** *Cronograma relativo ao período de estágio no exterior.*

## 1.4 Estrutura do texto

Esta qualificação tem duas partes:

**Parte 1** contém três capítulos, os quais apresentam uma introdução nos fundamentos teóricos usados nesta pesquisa e o levantamento bibliográfico feito até agora. O Capítulo 2 apresenta uma introdução na área da teoria difusa. Os métodos kernel são apresentados no Capitulo 3. O levantamento bibliográfico em quanto a classificadores para dados com imprecisão usando teoria difusa e métodos kernel é feito no Capitulo 4.

**Parte 2** contém um capítulo, o capítulo 5 que contém os resultados preliminares do projeto.

# Capítulo 2

# Teoria difusa

> *Fuzzy logic is not fuzzy. Fuzzy logic is precise. Basically, fuzzy logic is a precise logic of imprecision. Zadeh. [Zad10]*

Neste capítulo apresentamos diversos conceitos da teoria difusa como conjunto e número difuso, normas triangulares e inferência difusa. Finalmente, é descrito o sistema de lógica difusa tipo-1 nonsingleton.

## 2.1 Introdução

Os conjuntos difusos foram introduzidos em 1965 por Lotfi A. Zadeh no artigo " *fuzzy sets*" [Zad65] como uma extensão da teoria clássica de conjuntos. Além das operações básicas entre conjuntos difusos como a união e intersecção, foram introduzidos os conceitos de relação e composição difusa.

Trabalhos posteriores, como a sequência de artigos [Zad75a, Zad75c, Zad75b], apresentaram conceitos como variável linguística, conjuntos difusos tipo-n, princípio de extensão, lógica difusa e raciocínio aproximado. Em [LZ71], foram apresentadas a relação difusa de similaridade e ordem difusa. Atualmente, a teoria difusa é uma área muito ampla, sendo difícil fazer uma relação completa de áreas de conhecimento que foram beneficiadas por ela. As aplicações, tanto em matemática e computação quanto na medicina e na engenheira são prova desse fato.

No que segue, revisamos os conceitos relacionados a nossa pesquisa. Para uma completa revisão do tema existem diversos periódicos especializados na área, entre os mais importantes temos: o periódico *IEEE transanctions on fuzzy systems* e o periódico *Fuzzy sets and Systems* que compilam os avanços feitos tanto na teoria quanto nas aplicações práticas. Entre alguns textos de consulta, referenciamos os livros [PG07, MC05].

### 2.1.1 Conjuntos difusos

Seja $U$ um conjunto universal. Um conjunto difuso $F \subset U$, é caracterizado pela função:

$$\begin{aligned} \mu_F : U &\rightarrow [0,1] \\ x &\mapsto \mu_F(x), \end{aligned} \tag{2.1}$$

chamada de *função de pertinência*. Um conjunto difuso $F$ é do tipo *singleton* se:

$$\mu_F(x) = \begin{cases} 1 & \text{se } x = x', \\ 0 & \text{caso contrario,} \end{cases} \quad x' \in U. \tag{2.2}$$

Um conjunto difuso $F$ é *normal* se existe pelo menos um ponto $x \in U$ tais que $\mu_F(x) = 1$, e é *convexo* se para qualquer $x_1, x_2 \in U$ e $\lambda \in [0,1]$ tem-se: $\mu_F(\lambda x_1 + (1-\lambda)(x_2)) \leq \min(\mu_F(x_1), \mu_F(x_2))$.

### 2.1.2 Número difuso

Um número difuso $F$ é um conjunto difuso definido em $\mathbb{R}$[1]. A função de pertinência $\mu_F$ de um número difuso satisfaz [DP78, APM08]:

- $\mu_F$ é normal;
- $\mu_F$ é convexo;
- $\mu_F$ é semi-contínua superiormente, isto é, $\forall x_0 \in \mathbb{R}$ e $\forall \epsilon > 0$ existe a vizinhança $V(x_0)$ tal que $\mu_F(x) \leq \mu_F(x_0) + \epsilon$, $\forall x \in V(x_0)$;
- o fecho do suporte definido como: $F_0 = \{x \in U : \mu_F(x) > 0\}$ é compacto.

Essas propriedades implicam que para cada $0 < \alpha \leq 1$, o conjunto $\alpha$-*corte* de $F$ definido por $F_\alpha = [F_L(\alpha), F_U(\alpha)]$ onde:

$$F_L(\alpha) = \inf\{x \in \mathbb{R} : \mu_F(x) \geq \alpha\}$$
$$F_U(\alpha) = \sup\{x \in \mathbb{R} : \mu_F(x) \geq \alpha\},$$

é um intervalo fechado (não vazio) em $\mathbb{R}$, assim como também é o suporte $F_0$.

É possível representar a translação [BC11] de um número difuso $F$ por um valor real $z$ usando o conjunto $\alpha$-*corte*:

$$(F + z)_\alpha = [F_L(\alpha) + z, F_U(\alpha) + z], \quad \alpha \in [0, 1]. \tag{2.3}$$

Uma representação alternativa dos números difusos é a *representação L-R* [NP08].

$$F = \bigcup_{\alpha \in [0,1]} (\alpha, F_\alpha) = \bigcup_{\alpha \in [0,1]} (\alpha, [F_L(\alpha), F_U(\alpha)]. \tag{2.4}$$

Finalmente denotaremos como $E$ ao *espaço de números difusos*.

Existem na literatura vários trabalhos sobre números difusos. Alguns estudos relacionados podem ser encontrados em [BC11, WW99, DP78, Lee04]. As operações nos números difusos são feitas usualmente utilizando as normas triangulares, conceito que revisamos a seguir.

### 2.1.3 Normas triangulares

Menger [Men42] introduziu as normas e co-normas triangulares no contexto da teoria de probabilidades. Estas operações generalizam as operações de intersecção e união de conjuntos difusos. Para um tratamento detalhado do tema, o livro [KMP00] é um texto de referência no assunto.

A *norma triangular*, chamada comumente de T-norma, oferece uma classe geral de operadores que generaliza a operação de intersecção de conjuntos difusos. A T-norma é uma operação binária $T : [0, 1] \times [0, 1] \to [0, 1]$, que satisfaz os seguintes axiomas [YZ08, KMP00]:

**(i)** Comutatividade, $T(x, y) = T(y, x), \quad \forall x, y \in [0, 1]$;

**(ii)** Associatividade $T(x, T(y, z)) = T(T(x, y), z) \quad \forall x, y, z \in [0, 1]$;

**(iii)** Monoticidade $x_1 \leqslant x_2, y_1 \leqslant y_2 \Rightarrow T(x_1, y_1) \leqslant T(x_2, y_2), \quad \forall x_1, x_2, y_1, y_2 \in [0, 1]$;

**(iv)** Condições Limite $T(0, x) = 0, \quad T(1, x) = x, \quad \forall x \in [0, 1]$.

A notação infixa usada para a T-norma é o simbolo $\star$. Alguns exemplos de operadores T-norma são a função mínimo: $\min(x, y)$, e o produto algébrico: $xy$.

Por outro lado, a *co-norma triangular* chamada comumente de *S-norma* também oferece uma classe geral de operadores que generaliza a operação de união de conjuntos difusos. A S-norma é uma

[1] Neste caso $U = \mathbb{R}$.

operação binária $\perp : [0,1]\times[0,1] \to [0,1]$ que satisfaz os axiomas de comutatividade, associatividade e monoticidade anteriores, e possui a condição limite: $\perp(0,x) = x, \perp(1,x) = 1, \forall x \in [0,1]$.

A notação infixa usada para a S-norma é o simbolo $\oplus$. Alguns exemplos de operadores S-norma são a função máximo: $\max(x,y)$, e a soma algébrica: $x + y - xy$.

As normas triangulares são usadas de maneira ampla na teoria difusa. Em particular, no conceito de inferência difusa que revisamos a seguir.

### 2.1.4 Inferência difusa

Muitos problemas referentes a aplicações práticas podem ser modeladas com regras do tipo

$$\textbf{If } u_1 \text{ is } F_1 \text{ and} \ldots \text{and } \textbf{If } u_p \text{ is } F_p \textbf{ Then } v \text{ is } G, \tag{2.5}$$

conhecidas como regras *If-Then*. A teoria difusa fornece uma maneira qualitativa e quantitativa de manipular essas expressões, usando os conceitos de variável linguística [Zad75c] e relação difusa [LZ71]. Uma variável linguística é uma variável cujos valores são conjuntos difusos, assim, a proposição:

$$u_j \text{ is } F_j, \quad \text{para } j = 1, \ldots, p,$$

é interpretada como a variável linguística $u_j$ cujo valor é o conjunto difuso $F_j \subset U_j$. Da mesma forma que, $v$ é a variável linguística cujo valor é o conjunto difuso $G \subset V$.

A regra *If-Then* é interpretada como a *implicação difusa*: $U \Rightarrow V$ ($U$ então $V$), onde $U = U_1 \times \cdots \times U_p$ é o produto cartesiano dos respetivos universos do discurso. A implicação difusa é representada usualmente usando o conceito de relação difusa, por exemplo a relação difusa $R$ com função de pertinência:

$$\begin{aligned} \mu_R : U \times V &\to [0,1] \\ (\mathbf{x}, y) &\mapsto \mu_R(\mathbf{x}, y), \end{aligned} \tag{2.6}$$

onde os pontos $\mathbf{x} = (x_1, \ldots, x_p)$ e $y$, pertencem aos universos do discurso $U$ e $V$ das variáveis linguísticas $\mathbf{u} = (u_1, \ldots, u_p)$ e $v$. O valor de $\mu_R$ é calculado usando algum operador de implicação difusa [Zad73, CF91, RK93] como a T-norma.

$$\mu_R(\mathbf{x}, y) = \mu_{F_1}(x_1) \star \cdots \star \mu_{F_p}(x_p) \star \mu_G(y). \tag{2.7}$$

A inferência difusa, chamada também de *raciocínio difuso* ou *raciocínio aproximado*, pode ser vista como um *modus ponens* generalizado e possui o seguinte esquema:

$$\begin{array}{rl} \text{Implicação difusa:} & \textbf{If } u_1 \text{ is } F_1 \text{ and} \ldots \text{and } \textbf{If } u_p \text{ is } F_p \textbf{ Then } v \text{ is } G \\ \text{Premissa difusa:} & u_1 \text{ is } X_1 \text{ and} \ldots \text{and } u_p \text{ is } X_p \\ \hline \text{Conclusão:} & v \text{ is } Y. \end{array} \tag{2.8}$$

Se a implicação difusa é representada pela Equação 2.6, e a premissa difusa pela relação difusa $S$ com função de pertinência $\mu_S(\mathbf{x}) = \mu_{X_1}(x_1) \star \cdots \star \mu_{X_p}(x_p)$, para $X_j \subset U_j$, então o conjunto difuso $Y \subset V$ da parte da conclusão é obtido pela *composição difusa* [Zad73]:

$$Y = R \circ S, \tag{2.9}$$

com função de pertinência:

$$\begin{aligned}\mu_Y(y) &= \sup_{(x_1,\ldots,x_p)\in U_1\times\cdots\times U_p}\{\mu_S(x_1,\ldots,x_p)\star\mu_R(x_1,\ldots,x_p,y)\}\\ &= \sup_{\mathbf{x}\in U}\{\mu_S(\mathbf{x})\star\mu_R(\mathbf{x},y)\}.\end{aligned} \tag{2.10}$$

## 2.2 Sistemas de lógica difusa

Os sistemas de lógica difusa (SLD) [Zad74] chamados também de sistemas de inferência difusa, sistemas difusos baseados em regras, sistemas espertos difusos ou sistemas difusos, são aproximadores universais de funções [Kos94] e tem quatro elementos principais:

1. *Fuzzificador:* transforma as entradas do sistema em conjuntos difusos. O fuzzificador é chamado de *fuzzificador singleton*, se o resultado da fuzzificação produz conjuntos difusos singleton. Por outro lado, se o resultado do mapeamento são números difusos ou intervalos difusos [MM97b], o fuzzificador é chamado de *fuzzificador nonsingleton*.

2. *Regras:* os sistemas de lógica difusa tem um conjunto de regras *if-then* interpretadas como implicações difusas e representadas por relações difusas [WM92]. As regras fornecem a *base do conhecimento* do sistema e é com elas que o SLD pode aproximar funções.

3. *Algoritmo de inferência difusa:* é um algoritmo baseado em inferência difusa que utiliza as regras para fazer o mapeamento dos conjuntos difusos de entrada para conjuntos difusos de saída.

4. *Defuzzificador:* é uma parte opcional encarregada de converter os conjuntos difusos em saídas do sistema. A literatura reporta vários tipos de defuzzificadores [LK99]. A escolha de algum deles é baseada na simplicidade computacional e no tipo de aplicação [Men95].

   Entre os defuzzificadores mas conhecidos temos: o defuzzicador centro de gravidade (COG), chamado também de centro de área ou defuzzificador centroide [Men01] , o defuzzificador média de máximos e o defuzzificador centro de somas.

   Um defuzzificador usado amplamente é o defuzzificador média de centros ou defuzzificador de altura [DHR96], por sua simplicidade computacional.

Existem dois tipos de SLD bem conhecidos: o sistema de lógica difusa de Mamdani [Mam74] e o sistema de lógica difusa de Takagi-Sugeno-Kang (TSK) [TS85]. A diferença entre eles é a forma dos consequentes das regras ***If-Then***. O consequente do sistema difuso Mamdani é um conjunto difuso, enquanto que o consequente do sistema difuso TSK é uma função.

Mendel [Men01] divide os SLD's de Mandami como segue:

- *SLD tipo-1 singleton*: é o SLD mais conhecido, não leva em conta a imprecisão dos dados de entrada e faz uso do fuzzificador singleton.

- *SLD tipo-1 nonsingleton*: chamado também de SLD nonsingleton, leva em conta a imprecisão dos dados de entrada e faz uso do fuzzificador nonsingleton modelando as entradas como números difusos ou intervalos difusos.

- *SLD tipo-2 singleton*: leva em conta imprecisão associada aos antecedentes e consequentes das regras usando conjuntos difusos do tipo-2 [Zad75a]. Não tem em conta a imprecisão associada as entradas do sistema.

- *SLD tipo-1 nonsingleton tipo-2* leva em conta a imprecisão associada aos antecedentes e consequentes das regras usando conjuntos difusos do tipo-2. Faz uso do fuzzificador nonsingleton modelando a imprecisão das entradas com números difusos.

- *SLD tipo-2 nonsingleton tipo-2* leva em conta a imprecisão associada aos antecedentes e consequentes das regras usando conjuntos difusos do tipo-2. Modela as imprecisões das entradas como números difusos do tipo-2 [Men01].

Os SLD's são aproximadores universais de funções, isto é, dada uma função arbitrária $f$ e um número real $\epsilon > 0$, o SLD pode aproximar $f$ com um fator de aproximação $\epsilon$. Um revisão detalhada sobre o tema pode ser encontrada em [KM96].

A seguir apresentamos o SLD tipo-1 nonsingleton. Para uma tratamento detalhado do tema referenciamos o livro [Men01] e o artigo [MM97b].

### 2.2.1 Sistema de lógica difusa tipo-1 nonsingleton

Este tipo de SLD foi proposto em [MM97b]. Sua característica principal é o uso do fuzzificador nonsingleton modelando as imprecisões associadas as entradas do sistema com números difusos ou com intervalos difusos. Em [SF07], ele foi aplicado para atenuar as perturbações de objetos ferromagnéticos de levitação, levando em conta a imprecisão do erro de entrada. Em [Hak05], ele foi o encarregado de controlar a velocidade de um motor DC, onde as entradas tinham imprecisão devido à perturbações externas. Diversas outras aplicações foram feitas como a modelagem e predição de séries temporais caóticas de observações com ruido [MM97b], classificadores de arritmias [CT11] e modelagem de dados com ruido de series temporais caóticas [KHP04].

Em geral, os SLD's tipo-1 nonsingleton mostram melhores resultados na prática em comparação aos os SLD's tradicionais ou SLD's tipo-1 singleton, tal como mostra o estudo feito em [CRP+11]. Uma variante desse tipo de SLD foi apresentado em [MM97a].

A estrutura de um SLD tipo-1 nonsingleton é composta por $M$ regras dadas pela Equação 2.5, representado por $l$ relações difusas $R^l (l = 1, 2, \ldots, M)$, com funções de pertinência

$$\mu_{R^l}(\mathbf{x}, y) = \mu_{F_1^l}(x_1) \star \cdots \star \mu_{F_p^l}(x_p) \star \mu_{G^l}(y), \tag{2.11}$$

onde é considerado a T-norma como operador de implicação difusa. Usando o fuzzificador nonsingleton, a entrada do sistema é modelada pela relação difusa $S$ com função de pertinência:

$$\mu_S(\mathbf{x}) = \mu_{X_1}(x_1) \star \cdots \star \mu_{X_p}(x_p). \tag{2.12}$$

A inferência difusa é feita primeiramente calculando a composição difusa de $S$ com cada regra $R^l$ para obter os conjuntos difusos $Y^l$ com função de pertinência

$$\begin{aligned}
\mu_{Y^l}(y) &= \mu_{R^l \circ S}(y) && (2.13)\\
&= \sup_{\mathbf{x} \in U} \{\mu_S(\mathbf{x}) \star \mu_{R^l}(\mathbf{x}, y)\} && (2.14)\\
&= \sup_{\mathbf{x} \in U} \{\mu_S(x_1, \ldots, x_p) \star \mu_R(x_1, \ldots, x_p, y)\} \\
&= \sup_{\mathbf{x} \in U} \{\mu_{X_1}(x_1) \star \cdots \star \mu_{X_p}(x_p) \star \mu_{F_1^l}(x_1) \star \cdots \star \mu_{F_p^l}(x_p) \star \mu_{G^l}(y)\} \\
&= \mu_{G^l}(y) \star \sup_{x_1 \in U_1} \{\mu_{X_1}(x_1) \star \mu_{F_1^l}(x_1)\} \star \cdots \star \sup_{x_p \in U_p} \{\mu_{X_p}(x_p) \star \mu_{F_p^l}(x_p)\},
\end{aligned}$$

para então, calcular a saída como:

$$Y = \bigcup_{l=1}^{M} Y^l, \tag{2.15}$$

a função de pertinência é obtida através da S-norma

$$\mu_Y = \oplus_{l=1}^{M} \mu_{Y^l}. \tag{2.16}$$

Finalmente, dependendo da aplicação a saída do sistema pode ser o conjunto difuso $Y$, ou pode ser usado qualquer método de defuzzificação. Em particular se for usado o defuzzicador média de

centros ou defuzzificador de altura, o SLD tipo-1 nonsingleton consiste de funções da forma

$$f(\mathbf{x}) = \frac{\sum_{l=1}^{M} \bar{y}^l \mu_{G^l}(\bar{y}^l) \mathcal{T}_{j=1}^{p} \sup_{x_j \in U_j} \{\mu_{X_j}(x_j) \star \mu_{F_j^l}(x_j)\}}{\sum_{l=1}^{M} \mu_{G^l}(\bar{y}^l) \mathcal{T}_{j=1}^{p} \sup_{x_j \in U_j} \{\mu_{X_j}(x_j) \star \mu_{F_j^l}(x_j)\}}, \tag{2.17}$$

onde $\bar{y}^l$ é o ponto que possui o máximo valor na função de pertinência do conjunto $Y^l$ e, portanto, o valor máximo em $\mu_{G^l}(y)$. Então, considerando que a função de pertinência $\mu_{G^l}(\bar{y}^l) = 1$, a função da Equação 2.17 pode ser escrita como

$$f(\mathbf{x}) = \frac{\sum_{l=1}^{M} \bar{y}^l \mathcal{T}_{j=1}^{p} \sup_{x_j \in U_j} \{\mu_{X_j}(x_j) \star \mu_{F_j^l}(x_j)\}}{\sum_{l=1}^{M} \mathcal{T}_{j=1}^{p} \sup_{x_j \in U_j} \{\mu_{X_j}(x_j) \star \mu_{F_j^l}(x_j)\}}. \tag{2.18}$$

### 2.2.2 Aprendizagem de parâmetros de um SLD

A fim de que o SLD possa aproximar a função desejada, tem-se que estabelecer de maneira adequada seus parâmetros como o número de regras e as funções de pertinência. Diversas abordagens são usadas com essa finalidade, desde a definição das regras *if-then* por parte de especialistas até métodos de aprendizagem baseados nos dados de treinamento disponíveis. É amplamente usado na literatura [Men01] o termo *sistema neuro-difuso* como referência ao processo de ajuste de parâmetros de um SLD.

Entre alguns dos métodos mais conhecidos baseados nos dados de treinamento temos o método de Wang-Mendel [Men01] que é um método bem simples, mais pode levar para um SLD com muitas regras sem dar informação das funções de pertinência das regras. Métodos baseados em mínimos quadrados [Wan94] também são bastante conhecidos, e são usados para ajustar os consequentes das regras, mais tem-se que especificar o número de regras e as funções de pertinência dos antecedentes.

Um dos métodos mas conhecido, é a aprendizagem usando gradiente descendente, o qual pode ajustar tanto as funções de pertinência dos antecedentes e os consequentes, mais não o número de regras. O método apresentado em [MM96b], é um método que ajusta o número de regras mediante descomposição em valores singulares SVD-QR , mas precisa de que as funções de pertinência dos antecedentes das regras sejam conhecidas a priori.

Métodos baseados em algoritmos de agrupamento [DK96, Set99] dividem os dados de treinamento em grupos. Cada grupo define uma regra do SLD e ajuda a estabelecer as funções de pertinência dos antecedentes de cada regra. A maioria desses métodos precisam ter informação a priori do número de grupos que irão definir as regras do SLD. Em [Set99] é usado um algoritmo de redução de regras baseado em mínimos quadrados ortogonais para determinar o número ótimo de regras.

# Capítulo 3

# Métodos kernel

*Simplicity is the ultimate sophistication.*
*Leornado Da Vinci*

Neste capítulo apresentamos alguns conceitos da teoria dos métodos kernel.

## 3.1 Introdução

Os métodos kernel são uma classe de algoritmos de aprendizagem que detectam e exploram padrões complexos nos dados, sendo usados em métodos de agrupamento, classificação, ranking, limpeza de dados (cleaning), entre outros [STC04, SS02, RBCG07, MMR+01, JK03, CST00, CS06a, CMR09, CLR+11, Can10]. Uma característica dos métodos kernel é que podem ser usados como medida de similaridade entre padrões complexos como, por exemplo, em problemas da bioinformática onde alguns tipos de padrões consistem de sequências de caracteres. As máquinas de vetores de suporte (SVM), o discriminante de Fisher baseado em kernels (KDF)[1], a análise de componentes principais baseado em kernels (KPCA)[2], entre outros, são métodos kernel.

Qualquer algoritmo de aprendizagem basado em métodos kernel possui duas componentes principais: uma *função kernel* que faz o mapeamento dos dados para o *espaço de características*, que é um espaço vetorial de dimensionalidade maior; e um algoritmo de aprendizagem que trabalha neste espaço.

A seguir descrevemos as características e propriedades das funções kernel.

### 3.1.1 Kernels positivos definidos

Seja $\mathcal{X}$ um conjunto não vazio. A função kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{K}$, é chamada de positiva definida se e somente se:

$$\sum_{i,j=1}^{n} c_i \bar{c_j} k(\mathbf{x}_i, \mathbf{x}_j) \geq 0, \tag{3.1}$$

onde $\mathbb{K} = \mathbb{R}$ ou $\mathbb{K} = \mathbb{C}$, $n \in \mathbb{N}$, $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{X}$, $c_i, \bar{c_j} \in \mathbb{K}$ e $\bar{c_j}$ é o complexo conjugado de $c_j$.

Note que se $k$ é positivo definido, então $k(\mathbf{x}, \mathbf{x}) \geq 0$ para todo $\mathbf{x} \in \mathcal{X}$, e $k$ satisfaz a desigualdade de Cauchy-Schwartz:

$$|k(\mathbf{x}_i, \mathbf{x}_j)|^2 \leq k(\mathbf{x}_i, \mathbf{x}_i) k(\mathbf{x}_j, \mathbf{x}_j). \tag{3.2}$$

As seguintes duas proposições descrevem duas importantes propriedades dos kernels positivos definidos.

**Proposição 3.1.1.** *[BCR84] O kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ é positivo definido se e somente se é simétrico.*

[1] kernel Fisher discriminat
[2] kernel principal component analysis

**Proposição 3.1.2.** *[SS02] O kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ é positivo definido se e somente se para todo $n \in \mathbb{N}$, $\mathbf{x}_i \in \mathcal{X}$ e para qualquer número $c_i \in \mathbb{R}$ gera a matriz simétrica $A = (a_{ij})$ de tamanho $n \times n$, isto é, $k(\mathbf{x_i}, \mathbf{x_j}) = a_{ij}$ que satisfaz*

$$\sum_{i,j=1}^{n} c_i c_j a_{ij} = c^T A c \geq 0 \quad \textit{onde } c = [c_1, \ldots, c_n]. \tag{3.3}$$

A seguir descrevemos como pode ser construído um espaço vetorial de funções de Hilbert chamado Espaço de Hilbert com kernel reproduzível, onde as funções kernels são equivalentes ao produto interno de duas funções.

### 3.1.2 Espaço de Hilbert com kernel reproduzível

Um kernel positivo definido pode ser representado como um produto interno num espaço vetorial de funções. Estes espaços são conhecidos como espaços pré-Hilbert, pois são facilmente convertidos em espaços de Hilbert [3].

Seja o conjunto $\mathbb{R}^{\mathcal{X}}$ o qual contém todos os possíveis mapeamentos de $\mathcal{X}$ para $\mathbb{R}$. Seja $\mathcal{X}$ um conjunto não vazio, e seja $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ um kernel positivo definido. Considere o subespaço linear $\mathcal{F}_0 \subseteq \mathbb{R}^{\mathcal{X}}$ gerado pelas funções $k_{\mathbf{x}} : \mathcal{X} \to \mathbb{R}$, onde $k_{\mathbf{x}}(\mathbf{y}) = k(\mathbf{x}, \mathbf{y})$ para todo $\mathbf{x}, \mathbf{y} \in \mathcal{X}$.

O mapeamento no espaço de características é definido como:

$$\begin{aligned} \Phi : \mathcal{X} &\to \mathcal{F}_0 \\ \mathbf{x} &\mapsto (\Phi(\mathbf{x}) = k_{\mathbf{x}}). \end{aligned} \tag{3.4}$$

Logo, considerando as funções $f(\mathbf{x}) = \sum_i c_i k_{\mathbf{x}}(\mathbf{x}_i)$ e $g(\mathbf{x}) = \sum_j c_j k_{\mathbf{x}}(\mathbf{y}_j)$ definidas em $\mathcal{F}_0$, é possível estabelecer o produto interno entre elas como:

$$\langle f, g \rangle = \sum_i c_i g(\mathbf{x}_i) \tag{3.5}$$

$$= \sum_j c_j f(\mathbf{y}_j) \tag{3.6}$$

$$= \sum_{i,j} c_i c_j k(\mathbf{x}_i, \mathbf{y}_j). \tag{3.7}$$

Vale a pena destacar que o produto interno não depende das representações de $f$ e $g$. Por outro lado, o produto interno $\langle ., . \rangle$ é bilinear, simétrico e positivo definido: $\langle f, f \rangle = \sum c_i c_j k(\mathbf{x}_i \mathbf{x}_j) \geq 0$. Em particular, $\langle ., . \rangle$ é um kernel positivo definido em $\mathcal{F}_0$:

$$\sum_{i,j=1}^{n} \gamma_i \gamma_j \langle f_i, f_j \rangle = \left\langle \sum_i \gamma_i f_i, \sum_i \gamma_i f_i \right\rangle \geq 0. \tag{3.8}$$

Uma consequência importante é a propriedade de *reprodução do kernel* [Aro50]:

$$\langle f, k_{\mathbf{x}} \rangle = \sum_i c_i k(\mathbf{x}, \mathbf{x}_i) = f(\mathbf{x}) \quad \text{para todo } f \in \mathcal{F}_0 \text{ e } \mathbf{x} \in \mathcal{X}. \tag{3.9}$$

É por isso que os kernels positivos definidos são conhecidos como *kernels reproduzíveis*, permitindo escrever um kernel positivo definido como o produto interno entre funções no espaço $\mathcal{F}_0$:

$$\langle k_{\mathbf{x}}, k_{\mathbf{y}} \rangle = \langle k(\mathbf{x}, .), k(\mathbf{y}, .) \rangle = k(\mathbf{x}, \mathbf{y}). \tag{3.10}$$

[3] Um espaço de Hilbert $\mathcal{F}$ é um espaço de produto interno que é *completo*. A completitude se refere a que qualquer sequencia de Cauchy em $\mathcal{F}$ converge em $\mathcal{F}$.

Usado $\Phi(\mathbf{x}) = k_{\mathbf{x}}$, podemos escrever a função kernel como o produto interno das imagens de $\Phi$:

$$\langle \Phi(\mathbf{x}), \Phi(\mathbf{y}) \rangle = k(\mathbf{x}, \mathbf{y}). \tag{3.11}$$

Por outro lado, usando a Equação 3.2, temos

$$|f(\mathbf{x})|^2 \leq \langle f, f \rangle .k(\mathbf{x}, \mathbf{x}), \tag{3.12}$$

que implica que $\langle f, f \rangle = 0$ se e somente se $f = 0$.

O espaço $\mathcal{F}_0$ é convertido num espaço de Hilbert $\mathcal{F}$ agregando todos os pontos limites das sequências de Cauchy de $\mathcal{F}_0$ [4]. Assim, $\mathcal{F}_0$ é um subespaço *denso*, ou seja, cada elemento de $\mathcal{F}$ é o limite de uma sequência de Cauchy em $\mathcal{F}_0$. Este espaço de funções de Hilbert $\mathcal{F}$ é chamado de *espaço de funções com kernel reprodutíveis* (RKHS). Por tanto, um kernel positivo definido pode ser visto como um produto interno num RKHS[5].

A seguir mostramos como o kernel pode ser visto como um produto interno de dois vetores.

### 3.1.3 Kernel de Mercer

Um teorema que fornece conhecimento da geometria dos espaços de características é o teorema de Mercer [Mer09, WSS01, SS02]. O teorema de Mercer define o espaço de características em termos de vetores explícitos no lugar de um espaço de funções como o RKHS. O teorema de Mercer permite verificar se um kernel pode ser representado como um produto interno no espaço de características e tem sido amplamente usado no estudo das máquinas de vetores de suporte.

Dado um conjunto $\mathcal{X}$ equipado com uma $\sigma$-álgebra, y uma medida $\mu(\mathcal{X}) < \infty$, o teorema de Mercer é escrito como [SS02]:

**Teorema 3.1.3** (Teorema de Mercer ). *Seja $(\mathcal{X}, \mu)$ um espaço finito de medida. Suponha que $k \in L_\infty(\mathcal{X} \times \mathcal{X})$ seja um kernel simétrico tal que o operador $T_k : L_2(\mathcal{X}) \to L_2(\mathcal{X})$ definido por:*

$$(T_k f)(\mathbf{x}) = \int_{\mathcal{X}} k(\mathbf{x}, \mathbf{x}') f(\mathbf{x}') d\mu(\mathbf{x}'), \tag{3.13}$$

*seja positivo definido, isto é, para todo $f \in L_2(\mathcal{X})$, temos*

$$\int_{\mathcal{X} \times \mathcal{X}} k(\mathbf{x}, \mathbf{x}') f(\mathbf{x}) f(\mathbf{x}') d\mu(\mathbf{x}) d\mu(\mathbf{x}') \geq 0. \tag{3.14}$$

*Então é possível expandir $k$ em uma serie convergente e uniforme*

$$k(\mathbf{x}, \mathbf{x}') = \sum_{j=1}^{N_{\mathcal{H}}} \lambda_j \psi_j(\mathbf{x}) \psi_j(\mathbf{x}'), \tag{3.15}$$

*onde $\mathcal{N}_{\mathcal{H}} \in \mathbb{N}$, ou $\mathcal{N}_{\mathcal{H}} = \infty$ e as funções $\psi_j \in L_2(\mathcal{X})$ são as autofunções ortogonais normalizadas de $T_k$ associadas aos autovalores positivos $\lambda_j \in l_1$ ordenados em forma decrescente.* [6]

De (3.15) segue que $k(\mathbf{x}, \mathbf{x}')$ é um produto interno $\langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle$, no espaço $l_2^{\mathcal{N}_{\mathcal{H}}}$, onde o mapeamento $\Phi$ tem a forma:

$$\Phi : \mathcal{X} \to l_2^{\mathcal{N}_{\mathcal{H}}} \tag{3.16}$$

$$\mathbf{x} \mapsto (\sqrt{\lambda_j} \psi_j(\mathbf{x}))_{j=1,\ldots \mathcal{N}_{\mathcal{H}}}. \tag{3.17}$$

[4] Uma sequencia $(x_i) i \in \mathbb{N}$ num espaço normado é chamada de Cauchy, se para cada $\varepsilon > 0$ existe um $n \in \mathbb{N}$, tal que, para todo $n', n'' > n$, $||x_{n'} - x_{n''}|| < \varepsilon$.

[5] Do inglês: reproducing kernel Hilbert space.

[6] $l_p$ é o espaço de sequências com p-norma.

### 3.1.4 Caracterização dos kernels

Os kernels podem ser vistos como produtos internos em um espaço de características que pode ser um espaço vetorial de funções como o RKHS ou pode ser o espaço $l_2^{\mathcal{N}_\mathcal{H}}$ descrito no teorema de Mercer. O teorema abaixo descreve a caracterização das funções kernel como um produto interno no espaço de características.

**Teorema 3.1.4** (Caracterização dos kernels [STC04]). *A função $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ pode ser expressa como $\langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle$ num espaço de Hilbert se e somente se $k$ é positivo definido.*

As funções kernels foram amplamente estudadas no contexto da teoria de equações integrais [Mer09]. Os espaços RKHS e a propriedade de kernel reproduzível pode ser encontrado em [Kol41, Aro50]. Um livro de referência no assunto das propriedades teóricas dos kernels é [BCR84]. Na área de aprendizagem computacional existem vários artigos de referência como: [WSS01, RC05, RBCG07, OMCS04, CS06a, CMR09, CLR+11, Can10]. e os livros [SS02, STC04].

Em aprendizagem computacional as funções kernel permitem a construção de classificadores não lineares ao fazer o mapeamento dos dados no espaço de características. solucionando o problema do custo computacional de trabalhar com vetores de dimensionalidade grande.

## 3.2 Máquinas de vetores de suporte

A teoria de aprendizagem estatística [Vap95] cria um laço entre a generalização do classificador, o risco empírico e a complexidade da classe das funções. A máquina de vetores de suporte (SVM) introduzida por Vapnik [BGV92], é um algoritmo que minimiza a capacidade da classe de funções maximizando o margem entre os dados de treinamento e a superfície de decisão.

A superfície de decisão obtida é um hiperplano chamado de *hiperplano ótimo* ou *hiperplano de margem maximal* que depende só de um subconjunto dos dados de treinamento, chamados *vetores de suporte*. A SVM é formulada como um problema de otimização quadrática convexa e portanto, a solução encontra-se em um ótimo global.

O custo computacional de trabalhar em espaços de alta dimensionalidade é solucionado pelo mapeamento implícito da função kernel $k(\mathbf{x}_i, \mathbf{x}_j) = \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}_j) \rangle$ no espaço de características. O Teorema de Cover [Cov65] justifica o uso dos kernels, estabelecendo que dado um conjunto de treinamento não linearmente separável, a projeção dele em um espaço de alta dimensionalidade pode-se transformar com alta probabilidade, num conjunto de treinamento linearmente separável. Um tratamento detalhado em SMV's pode ser obtido em [CST00, STC04, SS02].

### 3.2.1 SVM com margem suave

O algoritmo da SVM procura o hiperplano de margem máxima

$$\{\Phi(\mathbf{x}) \in \mathcal{F} | \langle \mathbf{w}, \Phi(\mathbf{x}) \rangle + b = 0, \mathbf{w} \in \mathcal{F}, b \in \mathbb{R}\}. \tag{3.18}$$

Cortes e Vapnik [CV95] introduziram um algoritmo para o SVM que permite que uma fracão dos dados de entrada tenham margem menor que $1/||\mathbf{w}||$ (erro do margem) introduzindo as *variáveis de folga* $\xi_i$. Na SVM com margem suave, o hiperplano de margem máxima é construído solucionando o seguinte problema de otimização na forma primal:

$$\begin{aligned} \min_{b,\mathbf{w},\xi} \quad & \frac{1}{2} \parallel \mathbf{w} \parallel^2 + C \sum_{i=1}^{n} \xi_i \\ \text{sujeito a} \quad & y_i \left( \langle \mathbf{w}, \Phi(\mathbf{x}_i) \rangle + b \right) \geq 1 - \xi_i \\ & \xi_i \geq 0, \end{aligned} \tag{3.19}$$

onde:

- $\{\mathbf{x}_i, y_i\}_{i=1}^n$ é o conjunto de treinamento, para $\mathbf{x}_i \in \mathbb{R}^p$ e $y_i \in \{-1, 1\}$;
- $\mathcal{F}$ é um espaço de Hilbert (espaço de características);
- $\Phi$ é o mapeamento $\Phi : \mathbf{x}_i \in \mathcal{X} \to \Phi(\mathbf{x}_i) \in \mathcal{F}$;
- $C$ é uma constante positiva que controla o trade-off entre maximizar o margem e minimizar os erros do margem;
- $\xi_i$ são as variáveis de folga que correspondem a erros do margem.

A formulação do problema de otimização na forma dual é:

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j}^{n} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \\ \text{sujeito a} \quad & 0 \leq \alpha_i \leq C \\ & \sum_{i=1}^{n} \alpha_i y_i = 0, \end{aligned} \tag{3.20}$$

nesta formulação, os valores $\mathbf{x}_i$ associados aos multiplicadores de Lagrange $\alpha_i \geq 0$ são chamados de *vetores de suporte*. Usando as condições Karush-Kuhn-Tucker (KKT) [Fle87] os valores de $\mathbf{w}$ e $b$ são calculados usando só o conjunto de vetores de suporte:

$$\mathbf{w} = \sum_{i=0}^{sv} \alpha_i y_i \Phi(\mathbf{x}_i), \tag{3.21}$$

$$b = y_j - \sum_{i=1}^{sv} y_i \alpha_i k(\mathbf{x}_i, \mathbf{x}_j), \quad \alpha_j > 0, \tag{3.22}$$

onde $sv$ é o número de vetores de suporte. A função de decisão da SVM é:

$$\begin{aligned} f_{sv}(\mathbf{x}) &= \operatorname{sign}\left(\langle \mathbf{w}, \Phi(\mathbf{x}) \rangle + b\right) \\ &= \operatorname{sign}\left(\sum_{i=0}^{sv} \alpha_i y_i \langle \Phi(\mathbf{x}_i), \Phi(\mathbf{x}) \rangle + b\right) \\ &= \operatorname{sign}\left(\sum_{i=0}^{sv} y_i \alpha_i k(\mathbf{x}_i, \mathbf{x}) + b\right). \end{aligned} \tag{3.23}$$

# Capítulo 4

# Revisão Bibliográfica

*Exploit the tolerance for imprecision, uncertainty, and partial truth to achieve tractability, robustness, and low solution cost. Zadeh. [Zad94a]*

A maioria dos classificadores difusos usam conjuntos difusos para estabelecer os parâmetros de suas respectivas funções de decisão e para obter uma interpretação linguística do conjunto de regras. Os parâmetros desses classificadores são obtidos desde usando o conhecimento de especialistas, até métodos de aprendizagem baseados nos dados de treinamento disponíveis como algoritmos genéticos [SC07], gradiente descendente [Men01], algoritmos de agrupamento [DK96, Set99], descomposição de valores singulares [MM96b], mínimos quadrados [Wan94, Set99] entre outros [Men01].

A diferença dos classificadores difusos de outros classificadores é a interpretação linguística dada pelos conjuntos difusos. No entanto, *é difícil estabelecer quais os problemas em que o uso dos classificadores difusos é melhor que os outros.*

Dados com imprecisão são atrativos para serem usados por classificadores difusos. Isso constitui uma diferença fundamental dos classificadores difusos para outro tipo de classificadores. Nesse sentido, em [SC07] é proposto testar os classificadores difusos com dados difusos.

Muitos problemas reais possuem dados com imprecisão. Essa imprecisão deve-se ao erro intrínseco dos instrumentos de medição, dados com valores ausentes, dados cujo valor é uma lista dispersa de items como por exemplo os dados obtidos em questionários, dados obtidos por opiniões subjetivas e dados com valores linguísticos.

Um enfoque simples para modelar a imprecisão nos dados são os números difusos. Este enfoque tem sido usado com exito para reduzir e organizar dados geográficos [AFG+00], para calibrar os taxímetros com dados obtidos de um GPS[1] [VOOS10], para modelar variáveis compostas por um conjunto de parâmetros [CS06b, SCC06], para mineração de bases de dados [PAF12], etc.

No que segue, fazemos um levantamento bibliográfico dos trabalhos relacionados que consideram imprecisão nos dados de entrada.

## 4.1 Imprecisão nos dados de entrada em Sistemas de Lógica Difusa

A teoria dos conjuntos difusos [Zad65] tem sido usada para tratar a imprecisão de uma maneira quantitativa e qualitativa [Zad75a, Zad75c, Zad75b]. Aplicações bem sucedidas a diversas áreas da inteligencia artificial [YZ92], processamento de imagens [CYP96], bases de dados, aprendizagem computacional [BB99a, BB99b] e muitas outras áreas, como importantes contribuições teóricas [LZ71, Zad94b, Men42, KMP00, PG07, MC05] foram feitas nos últimos quarenta anos.

Na área de aprendizagem computacional, a teoria dos conjuntos difusos tem sido usada para construir classificadores supervisionados baseados em Sistemas de Lógica Difusa (SLD). Existem

[1] GPS: Global Positioning System

SLD's que tratam a imprecisão associada aos dados de entrada usando números difusos, como o caso do SLD nonsingleton [MM96a, MM97b, MM97a] e usando conjuntos difusos do tipo-2 como algumas variantes dos SLD's do tipo-2 [Men01, MJL06, MJ02, Men07, LM00, KM01a, KML99, KM01b, KM98].

Em [PSC09, SOV06, SCC09] foi apresentada uma abordagem baseada em algoritmos genéticos e SLD, para tratar dados com imprecisão em problemas de classificação supervisionada. Essa abordagem usa uma função fitness difusa para ajustar os parâmetros do classificador difuso [SC07]. Variações dessa abordagem tem sido aplicadas para a classificação de dados de atletismo [PSC11c], diagnóstico de dyslexia [PSC10a], para problemas com custo por classe [PSC11a]. Outras variações do tema incluem o preprocessamento de dados com imprecisão para dados não balanceados [PSC10b] e o uso do técnicas baseadas em boosting [PSC11b].

Estes trabalhos compartilham uma representação possibilística [Zad99, DP88] da imprecisão nos dados. Se os dados imprecisos são fornecidos como intervalos, então a saída é um conjunto de possíveis saídas para cada valor que pertença ao intervalo. Se os dados são dados difusos, a saída é um conjunto difuso. Cada alfa corte de um número difuso é um conjunto aleatório que contém o valor exato e desconhecido da variável com uma probabilidade de pelo menos um menos o valor do alfa corte.

Os algoritmos genéticos usados em aprendizagem de regras fuzzy, produzem modelos linguísticos compressíveis, porém, o custo computacional é muito maior que outros modelos que reportam acurácia semelhante [SOV06]. Luciano Sánchez et al., [SC07], faz menção de que uma contribuição precisa dos sistemas genéticos difusos é quando tem-se dados com imprecisão. O artigo propõe testar os classificadores difusos em geral nesse tipo de dados.

## 4.2 Abordagens para tratar imprecisão nos dados de entrada usando SVM e teoria difusa

No contexto da teoria difusa e as SVM's, tem sido reportados alguns trabalhos para tratar dados com imprecisão, principalmente para problemas de regressão, considerando diferentes combinações de dados difusos de entrada e saída [HH03, YX07, WL10, Wu10]. Em geral, a imprecisão é modelada usando números difusos triangulares, alguns deles, constroem funções de aproximação difusas [HH03, WL10], e outros usam diferentes tipos de SVM como $\nu$-SVM [YX07] e *one-class* SVM [Hao08]. Outros usam a medida de possibilidade [DP88] para definir as restrições da SVM [JPQ10, FYE12]. Todos esses trabalhos compartilham uma característica comum: são redefinidas as restrições da função objetivo da SVM. Este enfoque tem três desvantagens: (1) a formulação é tão complicada que o problema de otimização e solucionado usando métodos heurísticos como algoritmos genéticos [WL10] e otimização por enxame de partículas [Wu10], (2) os cômputos são custosos por isso são reportados experimentos com poucos dados, (3) os argumentos da função kernel são números reais que não modelam a imprecisão nos dados de entrada.

Um enfoque geométrico para tratar imprecisão nos dados de entrada foi apresentado na tese de doutorado [JoNYaBIE08]. O problema de procurar o hiperplano de margem máxima da SVM é transformado em procurar os pontos mais próximos de dois *fechos convexos* usando números difusos. A limitação da proposta é que só considera o caso linear.

Exitem outras abordagens que usam as SVM e a teoria difusa que não consideram a modelagem da imprecisão nos dados de entrada. Como por exemplo, com a finalidade de tratar amostras que não podem ser atribuídas com certeza a algumas das classes, e para diminuir a influência de dados com ruido na construção da superfície de decisão da SVM, foi apresentada a *fuzzy SVM* [LW02, HL02b]. Esta formulação usa funções de pertinência difusa por classe, e reformula o problema de otimização quadrática, onde cada dado de entrada contribui de maneira diferente na aprendizagem da superfície de decisão. Uma formulação similar para regressão foi apresentada em [SS03]. Em [HC07] foi apresentada uma SVM para o caso de regressão que procura a construção de um hiperplano difuso [Tan82]. Além disso, a teoria difusa tem sido usada no contexto das SVM's para solucionar problemas de multi-classificação [IA01, TA03].

## 4.3 Aprendizagem usando máquinas de vetores de suporte

Afim de estabelecer os parâmetros de um classificadores difuso, técnicas como agrupamento de dados e gradiente descente são extensamente usadas. O problema com essa abordagem é que o aprendizado de parâmetros é baseado no princípio de minimização do risco empírico que, por conta de haver poucos dados, nem sempre garante o bom desempenho do classificador na etapa de teste super-ajustando a hipótese e causando overfitting.

Para superar esse problema, alguns trabalhos foram apresentados para o aprendizado de parâmetros mediante SVM's. Esse método aproxima o erro teórico de generalização para hiperplanos separadores mediante o esquema de minimização do risco estrutural [JL99, SD01, CW03, CH04, WM92]. Uma consequência importante é que o SLD com aprendizado SVM induze funções kernels como mostra a literatura [JL99, LYL+06, JCC07].

O aprendizagem via SVM foi introduzido no contexto da teoria difusa em [JL99], o qual usava uma SVM para calcular as regras e os pesos de uma rede neural de um sistema neuro-difuso. Também foi usada em [SD01] para reduzir o número de amostras de treinamento num mapa auto-organizativo com funções difusas de pertinência por classe.

Em [CW03], é reportado o aprendizagem SVM de um classificador difuso, estabelecendo uma conexão entre as regras de um SLD, os vetores de suporte da SVM e as funções kernel. O classificador difuso define de maneira implícita um kernel invariante a translação, com a suposição de que todas as funções de pertinência são obtidas usando translações de funções de referência. consequentemente, é associado um vetor de suporte para cada regra *if-then* do classificador difuso. De acordo com [CH04], o aprendizagem SVM de um FLS estabelece uma conexão entre a função kernel do SVM e as funções de base difusa do FLS [WM92].

Com a finalidade de decrementar o número de vetores de suporte e em consequente o número de regras dos classificadores difusos, foram apresentados diversos estudos. Em [LYL+06] as regras e as funções de pertinência são inicialmente determinadas usando agrupamento. Posteriormente, é construída uma rede neural de quatro camadas usando as regras obtidas: a camada um e dois representam os antecedentes das regras, a camada três representa o consequente e a camada quatro combina todas as saídas das regras. Logo, é aplicado o aprendizagem SVM, nas camadas três e quatro da rede, com um kernel chamado de kernel difuso adaptativo construído a partir das regras iniciais, com a finalidade de otimizar as funções de pertinência e os pesos de conexão da rede. Finalmente, são removidas as regras irrelevantes usando um método de redução de regras.

Em [JCC07] o classificador é construído a partir de um SLD de Takagi-Sugeno-Kang (TSK) As regras iniciais são obtidas usando uma versão do método de agrupamento descrita em [JL98], logo é aplicado aprendizagem SVM linear nos consequentes das regras. Em [WW08] é usado o agrupamento c-médias e algoritmos genéticos para ajustar os parâmetros dos antecedentes das regras e aprendizagem SVM para ajustar os consequentes das regras.

O artigo [ZG07] apresenta um método de aprendizagem SVM, usando um L2-SVM para determinar as regras do classificador difuso. O trabalho propõe também o uso de dois índices para ponderar as regras: o índice $\alpha$ baseado nos multiplicadores de Lagrange e o índice $\omega$ obtido a partir da estrutura das regras.

Em [CJ11] foi apresentado um classificador baseado em SLD TSK com aprendizagem SVM incremental onde todos os parâmetros são treinados usando diferentes subconjuntos do conjunto de treinamento de maneira incremental no tempo, optimizando assim o uso de memória.

As vantagens de usar o aprendizagem SVM para ajustar os parâmetros de um SLD são:

- A SVM é um método baseado no princípio de minimização do risco estrutural. A SVM intenta de minimizar o erro de generalização, minimizando o erro de treinamento e a camadacidade da classe de funções onde a hipótese pertence, evitando o problema de overfitting.

- O número de regras obtidas é igual ao número de vetores de suporte das SVM. Consequentemente, esse valor não tem relação com a dimensão dos dados de entrada, evitando assim a maldição da dimensionalidade.

- A minimização da função objetivo das SVM é formulado como um problema de otimização convexa quadrática. Assim, a solução é um minimo global, evitando o problema do mínimos locais de outros métodos.
- A SVM é um método kernel, as funções kernel permitem representar os dados em espaços de alta dimensionalidade, transformando problemas de classificação não lineares no espaço de entrada, em problemas lineares no espaço de características.

# Capítulo 5

# Resultados preliminares

*Every little bit helps.*
*Proverb*

Este capítulo apresenta os resultados preliminares teóricos e experimentais do trabalho. Primeiramente, apresentamos a definição do kernel difuso nonsigleton e do classificador difuso nonsigleton, que foram formulados estudando-se as regras entre os sistemas de lógica difusa nonsigleton e as máquinas de vetores de suporte. Depois, apresentamos alguns resultados experimentais que até agora são bastante animadores.

## 5.1 Kernel difuso nonsingleton

Nesta seção, definimos um kernel com domínio nos números difusos

**Definição 5.1.1** (Kernel Difuso Nonsingleton)**.** Sejam $X$ e $Z$ dois números difusos com funções de pertinência $\mu_X$ e $\mu_Z$ respectivamente. Seja $E$ o espaço de números difusos. O Kernel Difuso Nonsingleton (KDN) é o mapeamento $k_{ns} : E \times E \to \mathbb{R}$ definido por:

$$k_{ns}(X, Z) = \sup\{\mu_X(x) \star \mu_Z(x) : x \in \mathbb{R}\}, \tag{5.1}$$

onde $\star$ é o operador T-norma.

O KND pode ser visto como uma medida da interseção de dois números difusos

$$k(X, Z) = \sup(X \cap Z), \tag{5.2}$$

onde o operador T-norma implementa a interseção dos números difusos $X$ e $Z$.

### 5.1.1 Propriedades

**Proposição 5.1.1.** *O kernel nonsingleton difuso é simétrico, i.e., $k_{ns}(X, Z) = k_{ns}(Z, X)$, para qualquer número difuso $X$, $Z \in E$.*

*Demonstração.* Usando a propriedade comutativa da T-norma temos:

$$\begin{aligned} k_{ns}(X, Z) &= \sup\{\mu_X(x) \star \mu_Z(x) : x \in \mathbb{R}\} \\ &= \sup\{\mu_Z(x) \star \mu_X(x) : x \in \mathbb{R}\} \\ &= k_{ns}(Z, X). \end{aligned}$$

□

**Proposição 5.1.2.** *O kernel nonsingleton difuso é invariante a traslação, i.e., $k_{ns}(X,Z) = k_{ns}(X+c, Z+c)$, para $c \in \mathbb{R}$ e para quaisquer números difusosx $X$, $Z \in E$.*

*Demonstração.* Suponha um valor qualquer $c \in \mathbb{R}$ e os números difusos $X = \bigcup_{\alpha\in[0,1]}(\alpha, X_\alpha)$ e $Z = \bigcup_{\alpha\in[0,1]}(\alpha, Z_\alpha)$. Sejam $X' = X + c$ e $Z' = Z + c$, os números difusos obtidos depois da translação de $X$ e $Z$ pelo valor $c$. Sejam $\mu_{X'}$ e $\mu_{Z'}$ suas respectivas funções de pertinência. Usando a representação $L$-$R$ para a traslação de números difusos temos:

$$\begin{aligned} X' &= \bigcup_{\alpha\in[0,1]}(\alpha, X'_\alpha) = \bigcup_{\alpha\in[0,1]}(\alpha, [X_L(\alpha)+c, X_U(\alpha)+c] \\ Z' &= \bigcup_{\alpha\in[0,1]}(\alpha, Z'_\alpha) = \bigcup_{\alpha\in[0,1]}(\alpha, [Z_L(\alpha)+c, Z_U(\alpha)+c]. \end{aligned}$$

Note que $\mu_{X'}(x) = \alpha$ para $x = X_L(\alpha) + c$ ou $x = X_U(\alpha) + c$. Mas por definição $X$ possui função de pertinência $\mu_x(X_L(\alpha)) = \mu_x(x - c) = \alpha$. Então $\mu_{X'}(x) = \mu_X(x - c)$. Portanto, o valor $\sup_{x\in\mathbb{R}}\{\mu_{X'}(x) \star \mu_{Z'}(x)\} = \sup_{x\in\mathbb{R}}\{\mu_X(x-c) \star \mu_Z(x-c)\}$. Logo, $k_{ns}(X,Z) = k_{ns}(X+c, Z+c)$ □

**Proposição 5.1.3.** *O kernel nonsingleton difuso satisfaz: $k_{ns}(X,Z) \in [0,1]$, para qualquer número difuso $X$, $Z \in E$.*

*Demonstração.* Pela monoticidade da T-norma temos que $\mu_X(x) \star \mu_Z(x) \in [0,1]$. Logo por definição do *sup*, temos: $\sup\{\mu_X(x) \star \mu_Z(x) : x \in \mathbb{R}\} \in [0,1]$. □

### 5.1.2 Representação no espaço de características

Mostraremos a seguir que o kernel nonsingleton difuso é um kernel de Mercer, isto é, admite a representação de produto interno em um espaço de características (um espaço de Hilbert de alta dimensionalidade).

**Teorema 5.1.4.** *O kernel nonsingleton difuso é um kernel de Mercer, isto é, admite uma representação da forma $k_{ns}(X,Z) = \langle\Phi(X), \Phi(Z)\rangle$, para qualquer $X$, $Y \in E$.*

*Demonstração.* Sejam $\{X_1, \ldots, X_n\}$ pertencentes a $E$, e sejam $\{c_1, \ldots, c_n\}$ pertencentes a $\mathbb{R}$. Seja a matriz $A = (a_{ij})$ de temanho $n \times n$ tal que $a_{ij} = k_{ns}(X_i, X_j)$. A matriz $A$ é simétrica por que $k_{ns}$ é simétrico. Como $k_{ns} \in [0,1]$, então $A$ satisfaz

$$\sum_{i,j=1}^{n} c_i c_j a_{ij} = c^T A c \geq 0.$$

Pelas Proposiçoes 3.1.1 e 3.1.2 $k_{ns}$ é positivo definido. Finalmente, pelo Teorema 3.1.4, concluímos que $k_{ns}(X,Z) = \langle\Phi(X), \Phi(Z)\rangle$. □

### 5.1.3 Extensão para vetores de números difusos

**Definição 5.1.2** (Kernel difuso nonsingleton para vetores de números difusos)**.** Sejam os vetores difusos $\mathbf{X} = [X_1, \ldots, X_p]^T$ e $\mathbf{Z} = [Z_1, \ldots, Z_p]^T$ com funções de pertinência $\mu_{X_1}, \ldots, \mu_{X_p}$ e $\mu_{Z_1}, \ldots, \mu_{Z_p}$ respetivamente. Seja $E$ o espaço de números difusos. O kernel difuso nonsingleton para vetores difusos é o mapeamento $k_{nsv} : E^p \times E^p \to \mathbb{R}$ definido como:

$$k_{nsv}(\mathbf{X}, \mathbf{Z}) = \mathcal{T}_{j=1}^{p} \sup\{\mu_{X_j}(x) \star \mu_{Z_j}(x) : x \in \mathbb{R}\}, \tag{5.3}$$

onde $\star$ é o operador T-norma e $\mathcal{T}_{j=1}^{p}$ é a sequência de $p-1$ operações T-norma.

É fácil provar que esse kernel é simétrico, invariante a traslação e com intervalo em $[0,1]$, usando o mesmo raciocínio das provas do kernel para números difusos.

Se interpretamos cada vetor de números difusos como relações difusas, então o kernel difuso nonsingleton para vetores de números difusos pode ser visto como uma medida da interseção de duas relações difusas

$$k(\mathbf{X}, \mathbf{Z}) = \sup(\mathbf{X} \cap \mathbf{Z}), \tag{5.4}$$

### 5.1.4 Alguns exemplos

Os cálculos feitos em [MM97b] no contexto de um sistema de lógica difusa tipo-1 nonsingleton, podem ser usados para calcular o valor de $k_{nsv}(\mathbf{X}, \mathbf{Z})$. Estes cálculos são mostrados no seguinte exemplo.

**Exemplo 5.1.1** (Função de pertinência gaussiana e produto algébrico para a T-norma)**.** Sejam os vetores de números difusos $\mathbf{X} = [X_1, \ldots, X_p]$ e $\mathbf{Z} = [Z_1, \ldots, Z_p]$ com funções de pertinência:

$$\begin{aligned} \mu_{X_j}(x) &= \exp\left(-\frac{1}{2}\frac{x^2 - m_{X_j}}{\sigma^2_{X_j}}\right), \quad 1 \leq j \leq p \\ \mu_{Z_j}(x) &= \exp\left(-\frac{1}{2}\frac{x^2 - m_{Z_j}}{\sigma^2_{Z_j}}\right), \quad 1 \leq j \leq p, \end{aligned} \tag{5.5}$$

onde $\sigma^2_{X_j}$, $\sigma^2_{Z_j}$, $m_{X_j}$ e $m_{Z_j}$ são as variançcas e as médias de cada número difuso $X_j$ e $Z_j$ pertencente a os vetores difusos $\mathbf{X}$ e $\mathbf{Z}$ respectivamente, e $x \in \mathbb{R}$. Logo, o Assim, pode-se calcular o sup das funções de pertinência:

$$\sup\{\mu_{X_j}(x) \star \mu_{Z_j}(x) : x \in \mathbb{R}\} = \exp\left(-\frac{1}{2}\frac{(m_{X_j} - m_{Z_j})^2}{\sigma^2_{X_j} + \sigma^2_{Z_j}}\right), \tag{5.6}$$

que acontece no ponto:

$$x = \frac{\sigma^2_{X_j} m_{Z_j} + \sigma^2_{Z_j} m_{X_j}}{\sigma^2_{X_j} + \sigma^2_{Z_j}}. \tag{5.7}$$

Logo, o kernel difuso nonsingleton para vetores difusos com funções de pertinências gaussiana é dado por:

$$k_{nsv}(\mathbf{X}, \mathbf{Z}) = \prod_{j=1}^{p} \exp\left(-\frac{1}{2}\frac{(m_{X_j} - m_{Z_j})^2}{\sigma^2_{X_j} + \sigma^2_{Z_j}}\right). \tag{5.8}$$

**Exemplo 5.1.2** (Funcao de pertinência gaussiana e função minimo para a T-norma)**.** Neste caso o valor do kernel difuso nonsingleton para vetores difusos é dado por:

$$k_{nsv}(\mathbf{X}, \mathbf{Z}) = \prod_{j=1}^{p} \mu_{X_j}(x_{mas}) = \prod_{j=1}^{p} \mu_{Z_j}(x_{mas}), \tag{5.9}$$

que acontece no ponto:

$$x_{mas} = \frac{\sigma_{X_j} m_{Z_j} + \sigma_{Z_j} m_{X_j}}{\sigma_{X_j} + \sigma_{Z_j}}. \tag{5.10}$$

## 5.2 Classificador difuso nonsingleton com aprendizado SVM

Usando a mesma notação que o Capitulo 3 anterior, o NFLS usa uma base de conhecimento (KB) composta pelas regras $\{R^l\}_{l=1}^{M}$, onde cada regra é uma implicação difusa representada usualmente com o conceito de relação difusa com função de pertinência: $\mu^l_R : \mathbb{R}^p \times \mathbb{R} \to [0, 1]$. Usando o operador

T-norma como operador de implicação difusa temos: $\mu_R^l(\mathbf{x}, y) = \mu_{F_1^l}(x_1) \star \cdots \star \mu_{F_p^l}(x_p) \star \mu_{G^l}(y)$, onde:

- $\{\mu_{F_j^l}\}_{j=1}^p$ são funções de pertinência dos conjuntos difusos $\{F_j^l\}_{j=1}^p$ do antecedente da regra $l$,
- $\mu_{G^l}$ é a função de pertinência do conjunto difuso $G^l$ do consequente da regra $l$,
- $\mathbf{x} = [x_1, \ldots, x_p] \in \mathbb{R}^p$, $y \in \mathbb{R}$ e $\star$ é o operador T-norma.

A diferença do NFLS de outros SLDs é o processo fuzzificação nonsigleton que transforma a entrada $\mathbf{x} \in \mathbb{R}^p$ num vetor de números difusos $\mathbf{X} \in E^p$. Cada vetor de números difusos pode ser visto como uma relação difusa com função de pertinência $\mu_{X_1}(x_1) \star \cdots \star \mu_{X_p}(x_p)$.

Usando inferência baseada na composição, defuzificador de altura e operador T;norma [MM97b], o NFLS consiste de funções da forma:

$$f(\mathbf{x}) = \frac{\sum_{l=1}^M \bar{y}^l \mathcal{T}_{j=1}^p \sup_{x_j \in U_j} \{\mu_{X_j}(x_j) \star \mu_{F_j^l}(x_j)\}}{\sum_{l=1}^M \mathcal{T}_{j=1}^p \sup_{x_j \in U_j} \{\mu_{X_j}(x_j) \star \mu_{F_j^l}(x_j)\}}. \tag{5.11}$$

No entanto, $f$ não está bem definida em $\mathbb{R}^p$ se o denominador é zero. Para solucionar esse problema uma prática comum [CW03, LYL+06, JCC07, CH04, ZG07] é agregar uma regra adicional ao SLD:

$$R^0 : \textbf{If } x_1 \text{ is } F_1^0 \text{ AND} \ldots \text{AND } \textbf{If } x_p \text{ is } F_p^0, \textbf{ Then } b \ , \tag{5.12}$$

onde $b \in \mathbb{R}$ e $\mu_{F_j^0}(x_j) = 1$ para $j = 1, \ldots, p$. Então o SLD pode pode ser escrito como:

$$f(\mathbf{x}) = \frac{\sum_{l=1}^M \bar{y}^l \mathcal{T}_{j=1}^p \sup_{x_j \in U_j} \{\mu_{X_j}(x_j) \star \mu_{F_j^l}(x_j)\} + b}{\sum_{l=1}^M \mathcal{T}_{j=1}^p \sup_{x_j \in U_j} \{\mu_{X_j}(x_j) \star \mu_{F_j^l}(x_j)\} + 1}. \tag{5.13}$$

### 5.2.1 Classificador difuso nonsingleton

A função de decisão do classificador difuso nonsingleton (NBFC)[1] pode ser definida como:

**Definição 5.2.1** (Função de decisão do NBFC)**.** Dado o conjunto de treinamento difuso

$$S_f = \{(\mathbf{X}_i, y_i)\}_{i=1}^n, \quad \mathbf{X}_i \in E^p \quad y_i \in \{-1, 1\}, \tag{5.14}$$

onde $E$ é o espaço de números difusos, e $p$ é a dimensionalidade dos dados. A função de decisão do NBFC é o mapeamento $f_{nd} : E^p \to \{-1, 1\}$ definido por:

$$f_{nd}(\mathbf{X}) = \text{sign}\left( \frac{\sum_{l=1}^M \bar{y}^l \mathcal{T}_{j=1}^p \sup_{x_j \in U_j} \{\mu_{X_j}(x_j) \star \mu_{F_j^l}(x_j)\} + b}{\sum_{l=1}^M \mathcal{T}_{j=1}^p \sup_{x_j \in U_j} \{\mu_{X_j}(x_j) \star \mu_{F_j^l}(x_j)\} + 1} \right). \tag{5.15}$$

Onde: $\bar{y}^l \in \mathbb{R}$ é o valor no domínio da função de pertinência do conjunto difuso do consequente na regra $l$, tal que, $\mu_{G^l}$ é máximo e $\mathcal{T}_{j=1}^p$ é uma sequência de $p-1$ operações T-norma.

A seguir é definida a base de conhecimento do NBFC.

**Definição 5.2.2** (Base do conhecimento do NBFC)**.** Seja $f_{nd}$ a função de decisão do classificador difuso nonsigleton. Seja o conjunto de vetores de números difusos $\{\mathbf{F}^l\}_{l=1}^M$ obtido a partir de $f_{nd}$, tal que $\mathbf{F}^l = [F_1^l, \ldots, F_p^l] \in E^p$ possui as funções de pertinência $\mu_{F_1^l}, \ldots, \mu_{F_p^l}$. Seja o conjunto de valores $\{\bar{y}^l\}_{l=1}^M$, $\bar{y}^l \in \mathbb{R}$ e o valor $b \in \mathbb{R}$ obtidos a partir de $f_{nd}$. A base de conhecimento do classificador difuso nonsingleton é representada dado pelo conjunto:

$$KB = \{(\mathbf{F}^1, \bar{y}^1), \ldots (\mathbf{F}^M, \bar{y}^M), (b, 1)\}. \tag{5.16}$$

[1] Nonsingleton binary fuzzy classifier

Finalmente, o NBFC é defindo como:

**Definição 5.2.3** (Classificador difuso nonsigleton)**.** O classificador nonsigleton difuso é um par $(KB, f_{nd})$, onde $KB$ é a base do conhecimento do classificador e $f_{nd}$ é a função de decisão do classificador.

### 5.2.2 Aprendizagem SVM do classificador difuso nonsingleton

É possível obter a base de conhecimento do classificador difuso nonsigleton a partir de uma SVM, tal como estabelece o seguinte teorema.

**Teorema 5.2.1.** *Seja $(KB, f_{nd})$ um classificador nonsigleton difuso. Seja $f_{svm}$ a função de decisão da SVM. Se $f_{nd} = f_{svm}$, então é possível calcular $KB$.*

*Demonstração.* Note que:

$$\begin{aligned}
f_{nd}(\mathbf{X}) &= \operatorname{sign}\left(\frac{\sum_{l=1}^{M} \bar{y}^l \mathcal{T}_{j=1}^{p} \sup_{x_j \in U_j}\{\mu_{X_j}(x_j) \star \mu_{F_j^l}(x_j)\} + b}{\sum_{l=1}^{M} \mathcal{T}_{j=1}^{p} \sup_{x_j \in U_j}\{\mu_{X_j}(x_j) \star \mu_{F_j^l}(x_j)\} + 1}\right). \\
&\quad \text{usando a Equação 5.3 temos} \\
&= \operatorname{sign}\left(\frac{\sum_{l=1}^{M} \bar{y}^l k_{nsv}(\mathbf{X}, \mathbf{F}_l) + b}{\sum_{l=1}^{M} k_{nsv}(\mathbf{X}, \mathbf{F}_l) + 1}\right). \\
&\quad \text{eliminando o denominador} \\
&= \operatorname{sign}\left(\sum_{l=1}^{M} \bar{y}^l k_{nsv}(\mathbf{X}, \mathbf{F}_l) + b\right). \\
&= \operatorname{sign}\left(\sum_{l=1}^{M} y_l \alpha_l \langle \Phi(\mathbf{X}), \Phi(\mathbf{F}_l)\rangle + b\right). \\
&= f_{svm}(\mathbf{X}),
\end{aligned}$$

onde a função de decisão $f_{svm}$ da SVM considera números difusos como argumento e usa o kernel difuso nonsingleton para vetores de números difusos. O conjunto de vetores de suporte desta SVM é $\{\Phi(\mathbf{F}_l)\}_{l=1}^{M}$. Logo o conjunto $KB$ é construído:

$$\begin{aligned}
KB &= \{(\Phi^{-1}(\mathbf{F}_l), y_l \alpha_l)\}_{l=1}^{M} \cup \{(b, 1)\}. \\
&= \{(\mathbf{F}^l, \bar{y}^l)\}_{l=1}^{M} \cup \{(b, 1)\}.
\end{aligned}$$

□

Como resultado do Teorema 5.2.1 temos:

1. *O classificador difuso nonsingleton pode ser treinado usando uma SVM.*
2. *O classificador difuso nonsingleton é um método kernel* que permite estender o conhecimento das relações das SVM e os classificadores difusos.

### 5.2.3 Características do classificador nonsingleton difuso

O classificador difuso nonsingleton tem as seguintes características:

- as entradas são vetores de números difusos que representam a imprecisão nos dados;
- a base de conhecimento e aprendida usando aprendizagem SVM, consequentemente, o classificador tem os benefícios do enfoque de aprendizagem SVM;

- o kernel difuso nonsingleton usado no aprendizagem SVM possui números difusos como argumentos e permite tratar problemas não lineares no espaço de entrada;
- é possível obter uma representação linguística do classificador. Cada elemento do conjunto $KB$ é representado por a regra *if-then*:

$$\textbf{If } u_1 \text{ is } F_1 \text{ and} \ldots \text{and } \textbf{If } u_p \text{ is } F_p \textbf{ Then } v \text{ is } G, \tag{5.17}$$

onde $\{u_j\}_{j=1}^{p}$ são variáveis linguísticas e $\{F_j\}_{j=1}^{p}$ são conjuntos difusos. O conjunto difuso $G$ tem valor máximo no ponto $\bar{y}$.

### 5.2.4 Algoritmo de aprendizado SVM

O Algoritmo 1 descreve o processo de aprendizagem SVM do classificador. A entrada é o conjunto de treinamento $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}, \quad \mathbf{x}_i \in \mathbb{R}^p, \quad y_i \in \{-1, 1\}$ Depois do processo de fuzzificação, cada valor crisp $\mathbf{x}_i$ é transformado em um número difuso $\mathbf{X}_i$, consequentemente, é obtido o conjunto $S_f = \{\mathbf{X}_i, y_i\}, \quad \mathbf{X}_i \in E^p, \quad y_i \in \{-1, 1\}$. Posteriormente, o SVM procura o hiperplano de margem maximal no espaço de características gerado pelo kernel difuso nonsigleton. Como resultado, o algoritmo construí a base de conhecimento do classificador, gerando uma regra fuzzy por cada vetor de números difusos com multiplicador de Lagrange $\alpha > 0$ e o consequente $\bar{y}^l$.

**Algorithm 1** Aprendizagem SVM para o classificador nonsingleton difuso

**Require:** $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}, \quad \mathbf{x}_i \in \mathbb{R}^p, \quad y_i \in \{-1, 1\}$

NONSINGLETON FUZZIFICATION
$S_f \leftarrow \phi$
**for** $i = 0$ **to** $n$ **do**
  $\mathbf{X}_i \leftarrow nonsingletonFuzzification(\mathbf{x}_i)$
  $S_f \leftarrow S_f \cup \{\mathbf{X}_i, y_i\}$
**end for**

SVM LEARNING
using $S_f$ solve
  $\max_\alpha \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j}^{n} \alpha_i \alpha_j y_i y_j k_{nsv}(\mathbf{X}_i, \mathbf{X}_j)$
  subject to
    $0 \leq \alpha_i \leq C$
    $\sum_{i=i}^{n} \alpha_i y_i = 0$

$l \leftarrow 0$, $b = 0$, $KB = \phi$
**for** $\alpha_j > 0$ **do**
  $\bar{y}^l \leftarrow \alpha_j y_j$
  $KB \leftarrow KB \cup \{(\mathbf{X}, \bar{y}^l)\}$
  $b \leftarrow b + y_j - \sum_{i=1}^{sv} y_i \alpha_i k_{nsv}(\mathbf{X}_i, \mathbf{X}_j)$
  $l \leftarrow l + 1$
**end for**
$M \leftarrow l$
$b \leftarrow b/M$
$KB \leftarrow KB \cup \{(b, 1)\}$
**return** $KB$

## 5.3 Experimentos em conjunto de dados tipo crisp

Nesta seção apresentamos os experimentos feitos usando o classificador difuso nonsigleton em conjuntos de dados tipo crisp.

### 5.3.1 Dados e implementação

O experimento foi feito considerando que cada atributo $x$ pertencente ao conjunto de dados tem a forma:

$$x = x_t + u_t \tag{5.18}$$

onde $x_t$ é o valor "verdadeiro", $u_t$ é um erro associado ao valor observado. Em outras palavras, $x_t$ não pode ser observada diretamente devido a diferentes fontes que podem causar imprecisão como, por exxemplo, erro de medida, imprecisão devido ao processamento dos dados ou imprecisão adicionada de outras formas. Essa imprecisão será modelada usando números difusos.

Neste experimento usamos os conjuntos de dados Iris, Wine, Glass, Ecoli, Vowel, Breast, Australian, Vehicle e Segment, do repositório *UCI machine learning repository* [FA10]. A Tabela 5.1 contém o resumo dos conjuntos de dados.

**Tabela 5.1:** *Sumario dos conjuntos de dados tipo crisp*

| Dataset | Amostras | Classes | Atributos |
|---|---|---|---|
| Iris | 150 | 3 | 4 |
| Wine | 178 | 3 | 13 |
| Glass | 214 | 6 | 9 |
| Ecoli | 336 | 8 | 7 |
| Vowel | 528 | 10 | 11 |
| Breast | 683 | 2 | 10 |
| Australian | 690 | 2 | 14 |
| Vehicle | 946 | 4 | 18 |

### 5.3.2 Fuzzificação

Cada atributo de cada um dos conjuntos de dados foi fuzzificado usando números difusos com função de pertinência gaussiana. Como resultado, a média da função de pertinência gaussiana de cada número difuso foi estabelecida como sendo o valor do atributo, e o desvio padrão foi obtido como o máximo de dois números escolhidos aleatoriamente no intervalo:

$$[\eta \times \sigma, \sigma], \tag{5.19}$$

onde $\eta$ é um fator de escala[2].

Nesta fuzzificação, a largura dos números difusos e controlada para não ser um valor perto de zero ou um valor tão aleatório. Isso permite a construção de números difusos com diferentes larguras para o mesmo conjunto de dados, diferentemente do que os enfoques clássicos. A figura 5.1 ilustra essa ideia.

### 5.3.3 Seleção de modelo

A etapa de seleção de modelos busca selecionar o melhor modelo em termos de alguma medida de desempenho variando-se o número de parâmetros. Como medida de performance foi usada a acurácia no conjunto de teste, definida como a soma dos verdadeiros positivos e verdadeiros negativos,

[2] o parâmetro $\sigma$ é descrito na seção seguinte

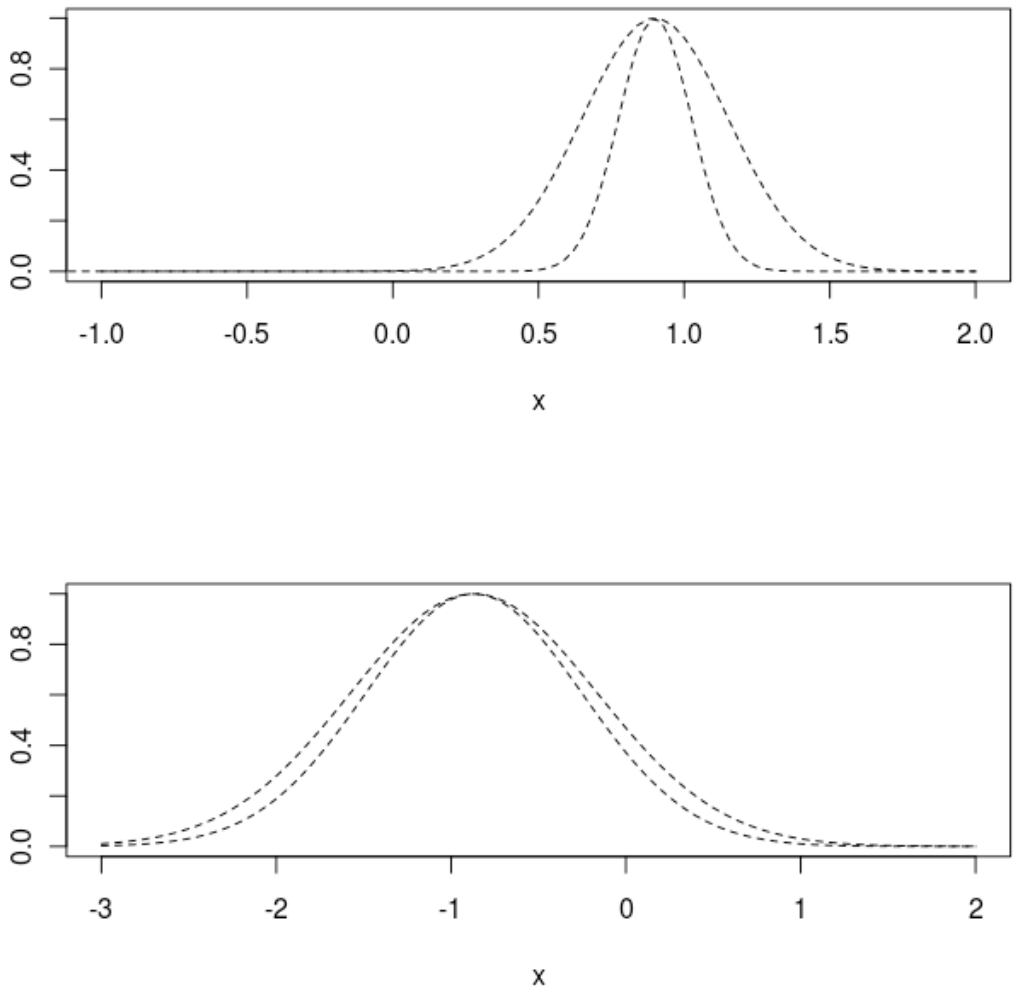



**Figura 5.1:** *Todos os números difusos construídos para os experimentos usam função de pertinência gaussiana com desvio padrão no intervalo* $[\eta \times \sigma, \sigma]$ *(linhas tracejadas). Imagem superior. Números difusos construídos para o valor* $x = 0.9$*, com* $\eta = 0.5$ *e* $\sigma = 0.25$*. As larguras obtidas estão no intervalo* $[0.125, 0.25]$*. Imagem inferior. Números difusos construídos para o valor* $x = -0.9$*, com* $\eta = 0.75$ *e* $\sigma = 0.70711$*. As larguras obtidas estão no intervalo* $[0.66291, 0.70711]$*.*

.

dividido pelo total de dados de teste. Em outras palavras o porcentagem de acertos do classificador. Todas as acurácias foram obtidas usando seleção de modelo para cada um dos conjuntos de teste.

Foram usados três parâmetros: $\gamma, C$ e $\eta$ na etapa de seleção de modelos.

- $\gamma \in \{2^4, 2^3, \ldots, 2^{-10}\}$, relacionado as funções de pertinência gaussiana dos números difusos como:
$$\sigma = \sqrt{\frac{1}{2\gamma}}.$$
- $C \in \{2^{-1}, 2^0, \ldots, 2^{14}\}$, permite o balance entre o margem maximal e erro do margem da SVM.
- $\eta \in \{1 - 2^0, 1 - 2^{-1}, \ldots, 1 - 2^{-4}\} = \{0, 0.5, 0.75, 0.88, 0.94\}$, esse parâmetro foi usado para calcular o desvio padrão das funções de pertinência gaussianas dos números difusos dadas pela Equação 5.19.

Com o propósito de fazer um estudo comparativo, foi implementado o classificador positivo definido (PDFC) com função de pertinência gaussiana [CW03], e foram usados os resultados reportados em [HL02a] para uma SVM com kernel gaussiano. O PDFC, o SVM e o classificador difuso nonsigleton usam os parâmetros $\gamma$ e $C$. O PDFC usa o valor $\gamma$ para estabelecer a largura das funções de pertinência de sua base de regras, esse valor é constante para todo o conjunto de dados. Similarmente, o SVM com kernel gaussiano usa o valor $\gamma$ também constante para todo o conjunto de dados. Por outro lado, o classificador difuso nonsigleton usa o valor $\gamma$ conjuntamente com o valor $\eta$ para calcular a largura dos números difusos gaussianos no intervalo dado pela Equação 5.19. Isso tem duas consequências: 1) Este valor não é um valor constante para todo o conjunto de dados, permitindo modelar a imprecisão nos dados de maneira flexível, e 2) Isso permite a construção da base de regras com diferentes larguras para as funções de pertinência gaussiana.

Para cada conjunto de teste e para cada um das $15 \times 16$ combinações de parâmetros $\gamma$ e $C$ e mantendo fixo o valor $\eta$, foram calculadas as acurácias usando validação cruzada 10-fold. Para esto, os dados foram particionados aleatoriamente em dez partes. Usando todas as possíveis combinações para nove partes como conjunto de treinamento e uma parte como conjunto de teste, foi calculada a média da acurácia para os conjuntos de teste. Finalmente, foi reportado o par $(C, \gamma)$ com melhor acurácia nas $15 \times 16$ combinações para cada parâmetro $\eta$. Isso é, no total, para cada conjunto de dato, foram usadas $15 \times 16 \times 5$ combinações diferentes.

### 5.3.4 Resultados e discussão para experimentos em dados tipo crisp

A Tabela 5.2 mostra as acurácias do classificador difuso nonsigleton (NBFC) no conjunto de teste, variando o valor de $\eta$. Os resultados sugerem boa capacidade de generalização para o NBFC. Os melhores resultados são atingidos no valor $\eta = 0.75$ para os conjuntos de dados Wine, Glass, Vowel e Australian, $\eta = 0.88$ para os conjunto de dados Ecoli e Breast, $\eta = 0.5$ para o conjunto de dados Iris e $\eta = 0.94$ para o conjunto de dados Vehicle.

### 5.3.5 Resultados comparativos

Com a finalidade de comparar o NBFC com outros métodos, foi implementado o *classificador difuso positivo definido* PDFC com função de pertinência gaussiana [CW03]. O PDFC é um classificador baseado em sistema de lógica difusa e aprendizagem SVM, mas os conjuntos difusos são tomados em consideração somente na construção das regras. Além disso, nos reportamos os resultados obtidos em [HL02a] para SVM multi-classe com kernel gaussiano e com o método *one-against-one*.

No experimento foi escolhido o classificador difuso nonsigleton com melhor acurácia na etapa de seleção de modelo, esse classificador foi denotado como NBFC. Também foi escolhido o classificador difuso nonsigleton como menor número de regras na etapa de seleção de modelo, essa escolha foi denotada como NBFCLR.

**Tabela 5.2:** *Resultados para os conjuntos de dados tipo crisp*

| | $\eta$ | | | | |
|---|---|---|---|---|---|
| Dataset | 0 | 0.5 | 0.75 | 0.88 | 0.94 |
| Iris | 96.93 | **98.37** | 97.95 | 97.95 | 97.95 |
| Wine | 99.17 | 99.00 | **99.50** | 98.96 | 98.96 |
| Glass | 68.92 | 72.00 | **74.23** | 73.79 | 73.96 |
| Ecoli | 86.28 | 86.91 | 87.69 | **87.71** | 87.63 |
| Vowel | 98.55 | 99.43 | **99.60** | 99.43 | 99.43 |
| Breast | 96.93 | 97.27 | 97.36 | **97.49** | 97.35 |
| Australian | 85.93 | 86.24 | **86.85** | 85.80 | 86.74 |
| Vehicle | 73.04 | 78.23 | 82.61 | 84.54 | **86.05** |

A Tabela 5.3 mostra os resultados de comparação entre NBFC, NFBCLR, PDFC e SVM em termos da acurácia. Os resultados sugerem que o método proposto tem melhor desempenho que os outros métodos em todos os conjuntos de dados, com exceção do conjunto de dados Vehicle. Também, os resultados sugerem que o NBFCLR tem desempenho melhor ou igual que o PDFC e a SVM.

A Tabela 5.4 contém os parâmetros obtidos na etapa de seleção de modelo. O número de regras e o número de vetores de suporte são valores reais pois é reportado a média do procedimento k-fold.

A Tabela 5.5 mostra o número de regras/vetores de suporte (sv) e as acurácias associadas dos classificadores NBFCLR, PDFC e SVM. Os resultados mostram que o NBFCLR tem menor número de regras e melhor acurácia que os outros métodos. com exceção dos conjuntos de dados Glass e Vehicle.

**Tabela 5.3:** *Resultados de comparação entre diferentes classificadores para conjuntos de dados tipo crisp, onde NA significa, "não disponível"*

| Dataset | NBFC | NBFCLR | PDFC | SVM[HL02a] |
|---|---|---|---|---|
| Iris | **98.37** | 97.95 | 97.47 | 97.33 |
| Wine | **99.50** | 99.00 | 98.96 | 99.43 |
| Glass | **74.23** | 73.96 | 74.11 | 71.495 |
| Ecoli | **87.71** | **87.71** | 87.43 | NA |
| Vowel | **99.60** | 99.43 | 99.25 | 99.05 |
| Breast | **97.49** | 97.35 | 97.23 | NA |
| Australian | **86.85** | 86.74 | 85.59 | NA |
| Vehicle | 86,05 | 86,05 | **86,77** | 86,64 |

## 5.4 Experimentos em conjunto de dados com ruído

Nesta seção apresentamos os experimentos feitos usando o classificador difuso nonsigleton em conjuntos de dados com ruído nos atributos.

### 5.4.1 Dados e implementação

Muitos problemas de classificação consideram dados com ruído, usualmente adicionado pelos próprios instrumentos de medida, ou em alguma etapa de preprocessamento dos dados, causando

**Tabela 5.4:** *Parâmetros dos classificadores, onde NA significa, "não disponível"*

| | NBFC | | | | NBFCLR | | | | PDFC | | | SVM[HL02a] | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Dataset | C | $\gamma$ | $\eta$ | regras | C | $\gamma$ | $\eta$ | regras | C | $\gamma$ | regras | C | $\gamma$ | s |
| Iris | $2^8$ | $2^{-9}$ | 0.5 | 34.3 | $2^{14}$ | $2^{-7}$ | 0.94 | 13.1 | $2^{11}$ | $2^{-6}$ | 14.5 | $2^{11}$ | $2^{-6}$ | 16. |
| Wine | $2^6$ | $2^{-8}$ | 0.75 | 42.2 | $2^8$ | $2^{-5}$ | 0.5 | 24.5 | $2^3$ | $2^{-4}$ | 39.6 | $2^7$ | $2^{-10}$ | 56. |
| Glass | $2^3$ | $2^2$ | 0.75 | 151.9 | $2^4$ | $2^1$ | 0.94 | 141.2 | $2^3$ | $2^2$ | 150.6 | $2^{11}$ | $2^{-2}$ | 112. |
| Ecoli | $2^{11}$ | $2^{-7}$ | 0.88 | 113.5 | $2^{11}$ | $2^{-7}$ | 0.88 | 113.5 | $2^0$ | $2^4$ | 156.5 | NA | NA | N |
| Vowel | $2^2$ | $2^1$ | 0.75 | 359.2 | $2^6$ | $2^{-1}$ | 0.88 | 330.7 | $2^5$ | $2^0$ | 345.2 | $2^4$ | $2^0$ | 345. |
| Breast | $2^7$ | $2^{-6}$ | 0.88 | 45.5 | $2^{12}$ | $2^{-9}$ | 0.94 | 39.1 | $2^2$ | $2^{-3}$ | 59.8 | NA | NA | N |
| Austral. | $2^4$ | $2^{-6}$ | 0.75 | 187.9 | $2^8$ | $2^{-7}$ | 0.94 | 181.9 | $2^{-1}$ | $2^{-3}$ | 236.7 | NA | NA | N |
| Vehicle | $2^7$ | $2^{-2}$ | 0.94 | 335.5 | $2^7$ | $2^{-2}$ | 0.94 | 335.5 | $2^{12}$ | $2^{-5}$ | 277.4 | $2^9$ | $2^{-3}$ | 302. |

**Tabela 5.5:** *Comparação do número de regras para os conjuntos de dados tipo crisp*

| | NBFCLR | | PDFC | | SVM[HL02a] | |
|---|---|---|---|---|---|---|
| Dataset | regras | acc. | regras | acc. | sv | acc. |
| Iris | **13.1** | 97.95 | 14.5 | 97.47 | 16.9 | 97.33 |
| Wine | **24.5** | 99.00 | 39.6 | 98.95 | 56.3 | 99.43 |
| Glass | 141.2 | 73.96 | 150.6 | 74.11 | **112.5** | 71.495 |
| Ecoli | **113.5** | 87.71 | 156.5 | 87.43 | NA | NA |
| Vowel | **330.7** | 99.43 | 345.2 | 99.25 | 345.3 | 99.05 |
| Breast | **39.1** | 97.34 | 59.8 | 97.23 | NA | NA |
| Australian | **181.9** | 86.74 | 236.7 | 85.59 | NA | NA |
| Vehicle | 335 | 86.05 | **277.4** | 86.77 | 302.4 | 86.64 |

imprecisão nos dados. O objetivo do experimento é avaliar o desempenho do classificador difuso nonsingleton em conjunto de dados com diferentes níveis de ruído nos atributos. O ruído nos atributos usualmente degrada o desempenho do classificador e o problema é tratado como uma tarefa de preprocessamento dos dados.

**Conjunto de dados com ruido nos atributos** Neste experimento usamos o esquema de adição de ruído proposto em [ZWY04], e usado em vários trabalhos [ZW04, JAS12]. Este esquema escolhe uma porcentagem e, então, para algum atributo determinado, é gerado um valor aleatório uniforme entre o máximo e mínimo valor no domínio daquele atributo. Neste esquema, o nível de ruído gerado e menor que o nível do ruído teórico, porque, o valor escolhido aleatoriamente pode ser o próprio. Uma característica desse esquema é que o ruído introduzido em um atributo tem pouca relação com o ruído introduzido no resto dos dados.

Usamos os seguintes conjuntos de dados com ruído nos atributos: Iris, Wine e Ecoli cada um com 0%, 5%, 10%, 15% e 20% níveis de ruído gerados de acordo com [ZWY04], Todos os quinze conjuntos de dados foram obtidos do repositório *KEEL* [AFFL+11]. Vale a pena destacar que não foi usado nenhum algoritmo de preprocessamento de ruído. A Tabela 5.6 contém o sumario dos conjuntos de dados usados.

**Tabela 5.6:** *Sumario dos conjuntos de dados com ruído*

| Dataset | Amostras | Classes | Atributos | %Ruido |
|---|---|---|---|---|
| Iris | 150 | 3 | 4 | 0% |
| Iris5 | 150 | 3 | 4 | 5% |
| Iris10 | 150 | 3 | 4 | 10 % |
| Iris15 | 150 | 3 | 4 | 15 % |
| Iris20 | 150 | 3 | 4 | 20 % |
| Wine | 178 | 3 | 13 | 0% |
| Wine5 | 178 | 3 | 13 | 5% |
| Wine10 | 178 | 3 | 13 | 10% |
| Wine15 | 178 | 3 | 13 | 15% |
| Wine20 | 178 | 3 | 13 | 20% |
| Ecoli | 336 | 8 | 7 | 0% |
| Ecoli5 | 336 | 8 | 7 | 5% |
| Ecoli10 | 336 | 8 | 7 | 10% |
| Ecoli15 | 336 | 8 | 7 | 15% |
| Ecoli20 | 336 | 8 | 7 | 20% |

### 5.4.2 Definição de parâmetros, fuzzificação e seleção de modelo

Para a definição dos parâmetros foram considerados:

- o escalamento dos dados no intervalo $[-1, 1]$;
- os mesmos parâmetros $\gamma$, $C$ e $\eta$ usados para os conjuntos de datos tipo crisp;
- o mesmo esquema de fuzzificação usado para o conjunto de dados tipo crisp;
- as acurácias foram obtidas usando a seleção de modelo descrita para o conjunto de dados tipo crisp.

### 5.4.3 Resultados e discussão para experimentos em conjuntos de dados com ruído

A Tabela 5.7 mostra a acurácia do classificador difuso nonsingleton (NBFC) para diferentes valores de $\eta$. Os resultados sugerem que o NBFC tem boa acurácia no conjunto de teste, consequentemente, boa capacidade de generalização. Os melhores resultados são atingidos para $\eta \in \{0.5, 0.75, 0.88, 0.94\}$ nos quinze conjuntos de dados.

**Tabela 5.7:** *Resultados para conjunto de dados com ruído*

| | $\eta$ | | | | |
|---|---|---|---|---|---|
| Dataset | 0 | 0.5 | 0.75 | 0.88 | 0.94 |
| Iris | 96.93 | **98.37** | 97.95 | 97.95 | 97.95 |
| Iris5 | 90,12 | 92.41 | **93.03** | 92.79 | 92.74 |
| Iris10 | 87.45 | 89.29 | 90.24 | **90.96** | 90.89 |
| Iris15 | 89.48 | **90.71** | 90.23 | 90.23 | 90.47 |
| Iris20 | 88.08 | 88.61 | 89.64 | 89.64 | **89.64** |
| Wine | 99.17 | 99.00 | **99.50** | 98.96 | 98.96 |
| Wine5 | 96.00 | 97.04 | 97.29 | **98.13** | 97.92 |
| Wine10 | 93.52 | 95.46 | **96.71** | 95.74 | 95.53 |
| Wine15 | 90.76 | 90.77 | 92.00 | **92.06** | 91.64 |
| Wine20 | 88.46 | **90.84** | 89.90 | 89.90 | 89.06 |
| Ecoli | 86.28 | 86.91 | 87.69 | **87.71** | 87.63 |
| Ecoli5 | 77.62 | 79.71 | **79.84** | 79.78 | 79.35 |
| Ecoli10 | 68.93 | 68.80 | 70.27 | 70.23 | **71.33** |
| Ecoli15 | 65.03 | 69.85 | 69.73 | 69.42 | **70.14** |
| Ecoli20 | 61.58 | **62.67** | 62.37 | 62.53 | 62.09 |

### 5.4.4 Resultados comparativos

Com a finalidade de comparar o NBFC contra um método robusto para dados com ruído nos atributos, foi implementado o PDFC com função de pertinência gaussiana. Este classificador é baseado em FLS é usa aprendizagem SVM. O PDFC usa os conjuntos difusos para a modelagem das regras e não para modelar a imprecisão nos dados de entrada. A escolha do PDFC foi feita pois esse classificador difuso é estatisticamente mais notável em termos de acurácias em dados com ruído nos atributos. Um trabalho comparativo do PDFC em conjuntos de dados com ruído contra classificadores crisp e classificadores difusos é apresentado em [SLH12]..

Neste experimento foi escolhido o NBFC com maior acurácia para algum valor de $\eta$. Também foi escolhido o classificador difuso nonsigleton com menor número de regras para algum valor de $\eta$. Esta escolha foi chamada de NBFCFR.

A Tabela 5.8 mostra o resultado da comparação das acurácias entre NBFC. NBFCLR e PDFC. Os resultados sugerem que o desempenho do NBFC é melhor em todos os conjuntos de dados com exceção do conjunto de dados Ecoli10. Quanto é incrementado o nível de ruído nos atributos, as acurácias do classificador diminuem. Por isso, usamos a medida de *perda relativa de acurácia* (RLA)[3] definida em [SLH12], para medir a variação em termos de acurácia do classificador em conjuntos de dados com ruído. O RLA é definido como:

$$RLA_{x\%} = \frac{Acc_{0\%} - Acc_{x\%}}{Acc_{0\%}}, \tag{5.20}$$

[3]relative loss of accuracy

onde $Acc_{x\%}$ é a acurácia no nível de ruído $x\%$.

A perda relativa de acurácia para os conjuntos de dados Wine e Ecoli é menor para o NBFC que para o PDFC para todos os níveis de ruído com a única exceção do conjunto de dados Ecoli10. No caso do conjunto de dados Iris, o RLA é menor somente nos níveis de ruído 5% e 10% para o NBFC.

**Tabela 5.8:** *Resultados comparativos*

| | NBFC | | NBFCLR | | PDFC | |
|---|---|---|---|---|---|---|
| Dataset | Acc | RLA % | Acc | RLA % | Acc | RLA% |
| Iris | **98.37** | 0.00 | 97.95 | 0.00 | 97.47 | 0.00 |
| Iris5 | **93.03** | 5.43 | 92.74 | 5.32 | 92.27 | 5.33 |
| Iris10 | **90.96** | 7.53 | 90.89 | 7.21 | 89.33 | 8.35 |
| Iris15 | **90.71** | 7.79 | 90.23 | 7.88 | 90.23 | 7.43 |
| Iris20 | 89.64 | 8.87 | 89.64 | 8.48 | 89.64 | 8.03 |
| Wine | **99.50** | 0.00 | 99.00 | 0.00 | 98.96 | 0.00 |
| Wine5 | **98.13** | 1.38 | 97.29 | 2.22 | 97.50 | 1.48 |
| Wine10 | **96.71** | 2.80 | 95.74 | 3.78 | 95.82 | 3.17 |
| Wine15 | **92.06** | 7.48 | 92.00 | 7.54 | 91.22 | 7.82 |
| Wine20 | **90.84** | 8.70 | 89.90 | 9.65 | 89.30 | 9.76 |
| Ecoli | **87.71** | 0.00 | **87.7**1 | 0.00 | 87.43 | 0.00 |
| Ecoli5 | **79.84** | 8.97 | **79.84** | 8.97 | 79.35 | 9.24 |
| Ecoli10 | 71.33 | 18.68 | 70.23 | 19.93 | **71.35** | 18.39 |
| Ecoli15 | **70.14** | 20.03 | **70.14** | 20.03 | 69.72 | 20.26 |
| Ecoli20 | **62.67** | 28.55 | 62.36 | 28.90 | 62.13 | 28.94 |

A Tabela 5.10 mostra o número de regras é as acurácias entre os métodos NBFCLR e PDFC. Os resultados sugerem que o NBFCLR tem menor número de regras é melhores acurácias na maioria dos casos. Nos outros casos, o NBFCLR tem similar ou melhores acurácias que o PDFC, com exceção do conjunto de dados Ecoli10 Os parâmetros associados ao experimento estão descritos na Tabela 5.9.

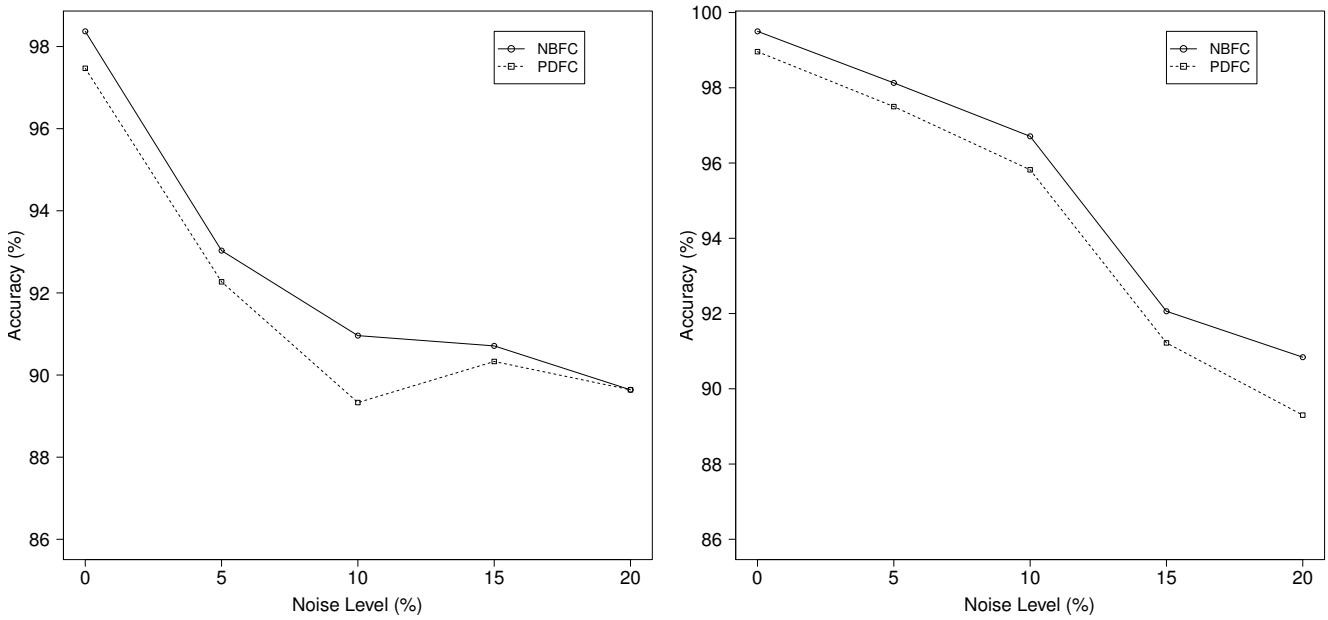
A Figura 5.2 mostra a comparação entre o NBFC e PDFC nos conjuntos de dados Iris e Wine.

**Tabela 5.9:** *Parâmetros dos classificadores*

| | NBFC | | | | NBFCLR | | | | PDFC | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Dataset | C | $\gamma$ | $\eta$ | regras | C | $\gamma$ | $\eta$ | regras | C | $\gamma$ | regras |
| Iris | $2^{8}$ | $2^{-9}$ | 0.5 | 34.3 | $2^{14}$ | $2^{-7}$ | 0.94 | **13.1** | $2^{11}$ | $2^{-6}$ | 14.5 |
| Iris5 | $2^{6}$ | $2^{-5}$ | 0.75 | 36.4 | $2^{11}$ | $2^{-7}$ | 0.94 | **28.8** | $2^{5}$ | $2^{-4}$ | 38.7 |
| Iris10 | $2^{7}$ | $2^{-4}$ | 0.88 | 38.5 | $2^{10}$ | $2^{-6}$ | 0.94 | **36.4** | $2^{4}$ | $2^{-4}$ | 52 |
| Iris15 | $2^{2}$ | $2^{-6}$ | 0.5 | 88.3 | $2^{-1}$ | $2^{1}$ | 0.88 | 79.9 | $2^{-1}$ | $2^{1}$ | **79.1** |
| Iris20 | $2^{1}$ | $2^{0}$ | 0.94 | 66.1 | $2^{1}$ | $2^{0}$ | 0.94 | 66.1 | $2^{1}$ | $2^{0}$ | **65.9** |
| Wine | $2^{6}$ | $2^{-8}$ | 0.75 | 42.2 | $2^{8}$ | $2^{-5}$ | 0.5 | **24.5** | $2^{3}$ | $2^{-4}$ | 39.6 |
| Wine5 | $2^{4}$ | $2^{-5}$ | 0.88 | 44.8 | $2^{12}$ | $2^{-4}$ | 0.75 | **34.7** | $2^{7}$ | $2^{-8}$ | 43.9 |
| Wine10 | $2^{2}$ | $2^{-3}$ | 0.75 | 64 | $2^{3}$ | $2^{-4}$ | 0.88 | **57.9** | $2^{-1}$ | $2^{-4}$ | 118.1 |
| Wine15 | $2^{5}$ | $2^{-9}$ | 0.88 | 99.5 | $2^{9}$ | $2^{-2}$ | 0.75 | **78.5** | $2^{4}$ | $2^{-8}$ | 101.6 |
| Wine20 | $2^{14}$ | $2^{-2}$ | 0.5 | 94.6 | $2^{8}$ | $2^{-2}$ | 0.75 | 91.9 | $2^{5}$ | $2^{-5}$ | **63.4** |
| Ecoli | $2^{11}$ | $2^{-7}$ | 0.88 | 113.5 | $2^{11}$ | $2^{-7}$ | 0.88 | **113.5** | $2^{0}$ | $2^{4}$ | 156.5 |
| Ecoli5 | $2^{2}$ | $2^{-2}$ | 0.75 | 179.9 | $2^{2}$ | $2^{-2}$ | 0.75 | **179.9** | $2^{0}$ | $2^{1}$ | 210.1 |
| Ecoli10 | $2^{3}$ | $2^{-4}$ | 0.94 | 200.8 | $2^{3}$ | $2^{-4}$ | 0.88 | 200.4 | $2^{5}$ | $2^{-5}$ | **196.4** |
| Ecoli15 | $2^{7}$ | $2^{-4}$ | 0.94 | 195.7 | $2^{7}$ | $2^{-4}$ | 0.94 | **195.7** | $2^{1}$ | $2^{-1}$ | 222.2 |
| Ecoli20 | $2^{0}$ | $2^{-0}$ | 0.5 | 254.8 | $2^{7}$ | $2^{-9}$ | 0.75 | **226.6** | $2^{1}$ | $2^{-1}$ | 262.7 |

**Tabela 5.10:** *Comparação do número de regras para os conjuntos de dados com ruído*

| | NBFCLR | | PDFC | |
|---|---|---|---|---|
| Dataset | regras | acc. | regras | acc. |
| Iris5 | **28.8** | 92.74 | 38.7 | 92.27 |
| Iris10 | **36.4** | 90.89 | 52 | 89.33 |
| Iris15 | 79.9 | 90.23 | **79.1** | 90.23 |
| Iris20 | 66.1 | 89.64 | **65.9** | 89.64 |
| Wine5 | **34.7** | 97.29 | 43.9 | 97.50 |
| Wine10 | **57.9** | 95.74 | 118.1 | 95.82 |
| Wine15 | **78.5** | 92.00 | 101.6 | 91.22 |
| Wine20 | 91.9 | 89.90 | **63.4** | 89.30 |
| Ecoli5 | **179.9** | 79.84 | 210.1 | 79.35 |
| Ecoli10 | 200.4 | 70.23 | **196.4** | 71.35 |
| Ecoli15 | **195.7** | 70.14 | 222.2 | 69.72 |
| Ecoli20 | **226.6** | 72.36 | 262.7 | 62.13 |




**(a)** **(b)**

**Figura 5.2:** *Acuracia do NBFC e PDFC para diferentes níveis de ruído. O eixo-x mostra o nível de ruído, e o eixo-y mostra a acurácia de ambos classificadores. O caso de* 0% *nível de ruído corresponde aos conjuntos de dados originais. e os casos de* $5\%, 10\%, 15\%$ *e* 20% *correspondem a diferentes níveis de ruído nos conjuntos de dados A figura mostra que o desempenho do NBFC é melhor que o PFCD em termos de acurácia. a) Conjunto de dado Iris. b) Conjunto de dado Wine.*

## 5.5 Experimentos em conjuntos de dados de baixa qualidade

Nesta seção apresentamos os experimentos feitos usando o classificador difuso nonsigleton em conjuntos de dados de baixa qualidade.

### 5.5.1 Dados e Implementação

Dados de baixa qualidade, são dados com algum grau de ignorância referente ao verdadeiro valor de um atributo. Nesta categoria estão dados obtidos com erro de medida, dados com valores ausentes, dados fornecidos como intervalos, dados fuzzy, dados cujos atributos tem valores maiores ou menores a um limiar e atributos cujos valores são uma lista dispersa de valores.

Neste experimento usamos quatro conjuntos de dados de baixa qualidade do repositório [AFFL+11]. A Tabela 5.11 contém o resumo dos conjuntos de dados.

**Tabela 5.11:** *Resumo dos conjuntos de dados de baixa qualidade*

| Dataset | Amostras | Classes | Atributos | Valores Ausentes |
|---|---|---|---|---|
| Long-4 | 25 | 2 | 4 | No |
| 100mlI-4 | 52 | 2 | 4 | No |
| 100mlP-4 | 52 | 2 | 4 | No |
| Dyslexic-12-4 | 65 | 4 | 12 | Sim |

Descrevemos brevemente a seguir esses conjuntos de dados. Uma descrição detalhada pode ser encontrada em [PSC09, PSC11c].

#### Conjunto de dados de desempenho de atletismo

- *Dataset Long-4*. Este conjunto de dados é usado para classificar se um atleta consegue melhorar um limiar em salto em distância. O conjunto de dados tem 4 atributos: a proporção entre peso e altura, a máxima velocidade aos 40 metros , o número de flexões de abdominais por minuto e uma prova de alongamento. Todas as características são representadas por intervalos. O treinador determina as primeiras duas na forma de valores linguísticos, intervalos ou números. As outras duas são medidas trés vezes, produzindo informação imprecisa representada como intervalos. Também, é permitido ao treinador introduzir sua experiencia pessoal nos atributos. A Tabela 5.12 contém duas amostras do conjunto de dados *Long-4* onde cada característica é um intervalo. Note que a segunda amostra pertence a duas classes.

- *Dataset 100mlI-4*. os atributos desse conjunto de dados são intervalos que contém informação de: a proporção entre peso e altura, o tempo de reação, a velocidade de inicio ou velocidade aos 20 metros e a velocidade máxima ou velocidade aos 40 metros. As medidas foram obtidas por trés diferentes observadores. Este conjunto de dados é usado para classificar se uma determinada marca no 100 metros de corrida é atingida.

- *Dataset 100mlP-4*. Tem as mesmas características que o conjunto de dados 100mlI-4, com a diferença que as medidas foram obtidas pela opinião de um treinador em termos de valores linguísticos como " o tempo de reação é lento".

- *Dataset Dyslexic-12-4* Tem doze atributos e quatro classes: { dyslexia, no dyslexia, controle, outros problemas}. O conjunto de dados contém valores ausentes.

Em cada conjunto de dados de baixa qualidade, cada atributo é um intervalo na reta real. Consequentemente, cada amostra do conjunto de dados tem vários intervalos. Outra característica desse conjunto de dados é que cada dado pode pertencer a duas classes no mesmo tempo.

**Tabela 5.12:** *Dos amostras do conjunto de dados de baixa qualidade* Long-4

| $x_1$ | $x_2$ | $x_3$ | $x_4$ | $y$ |
|---|---|---|---|---|
| $[8.7, 10.1]$ | $[45, 47]$ | $[2, 2.15]$ | $[5, 5.1]$ | $\{1\}$ |
| $[9.5, 10]$ | $[60, 64]$ | $[2.21, 2.23]$ | $[5.33, 5.4]$ | $\{0, 1\}$ |

### 5.5.2 Escalamento dos dados

Antes de escalar os dados, cada amostra que pertence a duas classes diferentes, foi eliminada, pois vamos nos restringir apenas à imprecisão nas entradas. Além disso, cada valor ausente foi substituído por um intervalo, cujos valores extremos são o minimo e o máximo valor no domínio do atributo.

A seguir, escalamos os dados da seguinte maneira: Seja $S$ uma matriz $n \times p$ de atributos, $S = (x_{ij})$, com $1 \leq i \leq n$ e $1 \leq j \leq p$. Cada fila $\mathbf{x_i} = [x_{i1}, \ldots x_{ip}]$ representa uma amostra onde cada atributo $x_{ij}$ é representado pelo intervalo $[l_{ij}, r_{ij}]$. As matrizes:

$$S^l = (l_{ij}) \quad 1 \leq i \leq n, \quad 1 \leq j \leq p \tag{5.21}$$
$$S^r = (r_{ij}) \quad 1 \leq i \leq n, \quad 1 \leq j \leq p, \tag{5.22}$$

contém as partes esquerdas e diretas dos intervalos, logo escalou-se cada coluna da matriz:

$$S^{lr} = \left| \begin{matrix} S^l \\ S^r \end{matrix} \right|,$$

no intervalo $[-1, 1]$, i.e., $-1 \leq l_{ij} \leq 1$ e $-1 \leq r_{ij} \leq 1$, com $l_{ij} \leq r_j$, aplicando uma transformação linear. Finalmente, atualizou-se a matriz original $S$ com os valores escalados $S^{lr}$.

### 5.5.3 Definição de parâmetros e seleção de modelo

Para a definição dos parâmetros foram considerados:

- os mesmos parâmetros $\gamma$, $C$ e $\eta$ usados para os conjuntos de datos tipo crisp;
- as acurácias foram obtidas usando o procedimento de seleção de modelo descrito para o conjunto de dados tipo crisp, só que esta vez foram usadas as partições 10-fold obtidas do repositório *KEEL* [AFFL+11].

### 5.5.4 Fuzzificação

Foram usados números difusos com função de pertinência gaussiana para fuzzificar cada atributo $x_{ij} = [l_{ij}, r_{ij}]$, onde

- $1 \leq i \leq n$ e $1 \leq j \leq p$, $n$ é a quantidade de amostras do conjunto de dados e $p$ é a dimensionalidade dos dados,
- $l_{ij}$ é $r_{ij}$ são as partes esquerda e direita do intervalo, respectivamente.

A média da função de pertinência gaussiana de cada número difuso foi obtida a partir dos intervalos como $(l_{ij} + r_j)/2$. Para o cálculo do desvio padrão, levou-se em conta a largura do intervalo $\delta_{ij} = |l_{ij} - r_{ij}|$, considerando o máximo valor de dois números gerados aleatoriamente de uma distribuição uniforme no intervalo:

$$[\eta \times \delta_{ij} \times \sigma, \delta_{ij} \times \sigma], \tag{5.23}$$

No caso de que algum atributo $x_{ij}$ possuisse $\delta_{ij} = 0$, então, o desvio padrão foi obtido como o máximo valor de dois números gerados aleatoriamente de uma distribuição uniforme no intervalo:

$$[\eta \times \sigma, \sigma]. \tag{5.24}$$

### 5.5.5 Resultados gerais

A Tabela 5.13 descreve as acurácias para o NBFC variando o valor de $\eta$. Os resultados mostram que o método proposta tem boa acurácia no conjunto de teste, isto é, bom poder de generalização. Os melhores resultados são atingidos para $\eta = \{0.5, 0.75, 0.88\}$ nos quatro conjuntos de dados. Note, que quando o valor $\eta$ aproxima-se ao um, os números difusos obtidos são mas similares entre sim. Contrariamente, quando o valor $\eta$ aproxima-se ao zero, eles vão ser menos similares.

**Tabela 5.13:** *Resultados para conjuntos de dados de baixa qualidade*

| | $\eta$ | | | | |
|---|---|---|---|---|---|
| Dataset | 0 | 0.5 | 0.75 | 0.88 | 0.94 |
| Long-4 | 68.30 | 65.00 | 67.67 | **70.00** | 68.33 |
| 100mlI-4 | 98.00 | 98.00 | **98.00** | 98.00 | 98.00 |
| 100mlP-4 | 88.00 | 88.00 | **90.50** | 88.00 | 88.00 |
| Dyslexic-12-4 | 43.83 | **47.17** | 41.10 | 40.50 | 41.33 |

### 5.5.6 Resultados comparativos

Aqui também foi usado o classificador PDFC [CW03][4] com funções de pertinência gaussinas.

[4] Positive definite fuzzy classifier

Neste experimento, o classificador difuso nonsingleton com melhor desempenho na etapa de seleção de modelo (Tabela 5.13) foi escolhido e denotado como NBFC. Também foi escolhido o classificador difuso nonsigleton com menor número de regras e denotado como NBFCFR

A Tabela 5.14 mostra os resultados da comparação entre: NBFC, NBFCFR e PDFC. Os parâmetros obtidos na seleção de modelo são mostrados na Tabela 5.15.

A Tabela 5.16 mostra o número de regras e as acurácias entre o NBFCFR e o PDFC.

Os resultados sugerem que o método proposto tem melhor desempenho em termos de acurácia e número de regras que os outros métodos para esses quatro conjuntos de dados.

**Tabela 5.14:** *Resultados comparativos*

| Dataset | NBFC | NBFCLR | PDFC |
|---|---|---|---|
| Long-4 | 70.00 | 68.33 | **73.33** |
| 100mlI-4 | **98.00** | 98.00 | 96.00 |
| 100mlP-4 | **90.50** | 90.50 | 88.00 |
| Dyslexic-12-4 | **47.17** | 41.101 | 36.00 |

**Tabela 5.15:** *Parâmetros dos classificadores*

| | NBFC | | | | NBFCLR | | | | PDFC | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Dataset | C | $\gamma$ | $\eta$ | regras | C | $\gamma$ | $\eta$ | regras | C | $\gamma$ | regras |
| Long-4 | $2^4$ | $2^9$ | 0.875 | 9.9 | $2^5$ | $2^{-8}$ | 0.94 | **9.4** | $2^0$ | $2^{-3}$ | 15.1 |
| 100mlI-4 | $2^{13}$ | $2^{-2}$ | 0.75 | 21.3 | $2^{13}$ | $2^{-2}$ | 0.75 | 21.3 | $2^{14}$ | $2^{-4}$ | **12.1** |
| 100mlP-4 | $2^{11}$ | $2^{-2}$ | 0.75 | 26.1 | $2^{11}$ | $2^{-2}$ | 0.75 | **26.1** | $2^3$ | $2^2$ | 28.6 |
| Dyslexic-12-4 | $2^7$ | $2^{-7}$ | 0.5 | 25.7 | $2^{13}$ | $2^{-9}$ | 0.75 | **22.5** | $2^5$ | $2^{-8}$ | 33 |

**Tabela 5.16:** *Comparação do número de regras*

| | NBFCLR | | PDFC | |
|---|---|---|---|---|
| Dataset | regras | acc. | regras | acc. |
| Long-4 | 9.4 | 68.33 | 15.1 | 73.33 |
| 100mlI-4 | 21.33 | 98.00 | 12.1 | 96.00 |
| 100mlP-4 | **26.1** | 90.5 | 28.6 | 88.00 |
| Dyslexic-12-4 | **22.5** | 41.10 | 33.00 | 36.00 |

### 5.5.7 Testes usando a informação dos intervalos

Neste experimento o treinamento do classificador difuso nonsingleton foi feito com o procedimento mostrado nas Seções 5.5.2, 5.5.3 e 5.5.4. Porém, os testes foram feitos da seguinte maneira: para cada cada atributo $x_{ij} = [l_{ij}, r_{ij}]$ do conjunto de teste, foram construídos números difusos gaussianos $X_{ij}$. A média da função de pertinência de cada número difuso $X_{ij}$ foi estabelecido como sendo o valor $[l_{ij}, r_{ij}]/2$, e foram testados três diferentes valores para o desvio padrão $\sigma_{ij}$ usando a informação da largura do intervalo $\delta_{ij} = |l_{ij} - r_{ij}|$. Estes valores foram $\sigma_{ij} = \{\delta_{ij} * 0,5, \delta_{ij} * 1.0, \delta_{ij} * 2.0\}$

A Tabela 5.17 reporta as acurácias deste experimento. Os resultados sugerem que a melhor acurácia é obtida para o conjunto difuso de teste construído com $\sigma_{ij} = \delta_{ij} * 0.5$. Os resultados mostram

**Tabela 5.17:** *Resultados para conjuntos de dados de baixa qualidade*

| | | $\eta$ | | | | |
|---|---|---|---|---|---|---|
| Dataset | $\sigma_{ij}$ | 0 | 0.5 | 0.75 | 0.88 | 0.94 |
| Long-4 | $\delta_{ij} * 0.5$ | 68.33 | 63.33 | 68.33 | 63.33 | **68.33** |
| | $\delta_{ij} * 1.0$ | 58.33 | 53.33 | **61.67** | 58.33 | 58.33 |
| | $\delta_{ij} * 2.0$ | 58.33 | **58.33** | 58.33 | 58.33 | 58.33 |
| 100mlI-4 | $\delta_{ij} * 0.5$ | **98.00** | 98.00 | 98.00 | 98.00 | 98.00 |
| | $\delta_{ij} * 1.0$ | 98.00 | **98.00** | 98.00 | 98.00 | 98.00 |
| | $\delta_{ij} * 2.0$ | 96.00 | 96.33 | 96.00 | **98.00** | 96.33 |
| 100mlP-4 | $\delta_{ij} * 0.5$ | 88.00 | **88.00** | 88.00 | 88.00 | 88.00 |
| | $\delta_{ij} * 1.0$ | 88.00 | 88.00 | 88.00 | **88.00** | 88.00 |
| | $\delta_{ij} * 2.0$ | 84.33 | **86.00** | 86.00 | 86.00 | 84.33 |
| Dyslexic-12-4 | $\delta_{ij} * 0.5$ | 41.29 | **49.76** | 46.10 | 45.50 | 48.10 |
| | $\delta_{ij} * 1.0$ | 41.59 | 45.60 | 45.76 | **48.00** | 45.10 |
| | $\delta_{ij} * 2.0$ | 40.43 | 39.93 | 44.69 | 40.52 | **45.02** |

que incrementando a largura do desvio padrão, decrementa a acurácia, nos quatro conjuntos de teste, com exceção do conjunto de teste *100mlI-4* cuja acurácia permanece constante.

A Tabela 5.18 mostra os resultados comparativos entre o NBFC, o NBFCLR e o PDFC. Os resultados mostram que com exceção do conjunto *Long-4*, os resultados são melhores usando números difusos gaussianos com $\sigma_{ij} = \{\delta_{ij} * 0,5, \delta_{ij} * 1.0\}$.

**Tabela 5.18:** *Resultados comparativos*

| Dataset | $\sigma_{ij}$ | NBFC | NBFCLR | PDFC |
|---|---|---|---|---|
| Long-4 | $\delta_{ij} * 0.5$ | 68.33 | 68.33 | |
| | $\delta_{ij} * 1.0$ | **61.67** | 58.33 | **73.33** |
| | $\delta_{ij} * 2.0$ | **58.33** | **53.33** | |
| 100mlI-4 | $\delta_{ij} * 0.5$ | **98.00** | **98.00** | |
| | $\delta_{ij} * 1.0$ | **98.00** | **98.00** | 96.00 |
| | $\delta_{ij} * 2.0$ | **98.00** | 96.33 | |
| 100mlP-4 | $\delta_{ij} * 0.5$ | **88.00** | **88.00** | |
| | $\delta_{ij} * 1.0$ | **88.00** | **88.00** | 88.00 |
| | $\delta_{ij} * 2.0$ | **86.00** | 84.33 | |
| Dyslexic-12-4 | $\delta_{ij} * 0.5$ | **49.76** | 41.29 | |
| | $\delta_{ij} * 1.0$ | **48.00** | 45.76 | 36.00 |
| | $\delta_{ij} * 2.0$ | **45.02** | 44.69 | |

Em quanto ao número de regras a Tabela 5.19 mostra que para os conjuntos de dados *100mlP-4* e *Dyslexic-12-4* existe melhor performance com menor número de regras para o NBFC que para o PDFC

**Tabela 5.19:** *Parâmetros dos classificadores*

| | | NBFC | | | | NBFCLR | | | | PDFC | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Dataset | $\sigma_{ij}$ | C | $\gamma$ | $\eta$ | regras | C | $\gamma$ | $\eta$ | regras | C | $\gamma$ | regras |
| Long-4 | $\delta_{ij} * 0.5$ | $2^7$ | $2^{-5}$ | 0.94 | 8.3 | $2^7$ | $2^{-5}$ | 0.94 | **8.3** | $2^0$ | $2^{-3}$ | 15.1 |
| | $\delta_{ij} * 1.0$ | $2^1$ | $2^{-6}$ | 0.75 | 16.1 | $2^{12}$ | $2^{-5}$ | 0.875 | 8.5 | | | |
| | $\delta_{ij} * 2.0$ | $2^2$ | $2^{-3}$ | 0.5 | 15.2 | $2^2$ | $2^{-3}$ | 0.5 | 15.2 | | | |
| 100mlI-4 | $\delta_{ij} * 0.5$ | $2^9$ | $2^{-2}$ | 0.00 | 24 | $2^9$ | $2^{-2}$ | 0.00 | 24 | $2^{14}$ | $2^{-4}$ | **12.1** |
| | $\delta_{ij} * 1.0$ | $2^9$ | $2^{-2}$ | 0.5 | 20.6 | $2^9$ | $2^{-2}$ | 0.5 | 20.6 | | | |
| | $\delta_{ij} * 2.0$ | $2^{12}$ | $2^{-2}$ | 0.875 | 22.3 | $2^{11}$ | $2^{-3}$ | 0.5 | **16.9** | | | |
| 100mlP-4 | $\delta_{ij} * 0.5$ | $2^2$ | $2^{-2}$ | 0.5 | 26.5 | $2^2$ | $2^{-2}$ | 0.5 | 26.5 | $2^3$ | $2^2$ | 28.6 |
| | $\delta_{ij} * 1.0$ | $2^3$ | $2^{-2}$ | 0.875 | 25.5 | $2^3$ | $2^{-2}$ | 0.875 | 25.5 | | | |
| | $\delta_{ij} * 2.0$ | $2^2$ | $2^{-2}$ | 0.5 | 26.5 | $2^2$ | $2^{-4}$ | 0.94 | **20.00** | | | |
| Dyslexic-12-4 | $\delta_{ij} * 0.5$ | $2^{12}$ | $2^{-4}$ | 0.5 | 39.1 | $2^6$ | $2^{-6}$ | 0.0 | 27.4 | $2^5$ | $2^{-8}$ | 33 |
| | $\delta_{ij} * 1.0$ | $2^4$ | $2^{-4}$ | 0.875 | 37.1 | $2^7$ | $2^{-7}$ | 0.75 | **25.2** | | | |
| | $\delta_{ij} * 2.0$ | $2^{14}$ | $2^{-6}$ | 0.94 | 26.00 | $2^7$ | $2^{-7}$ | 0.75 | **25.2** | | | |

# Capítulo 6

# Trabalhos Futuros

*The future depends on what you do today.*
*Mahatma Ghandi*

Alguns trabalhos futuros a considerar no contexto de dados com imprecisão são:

- Estudar a relação dos sistemas Takagi Sugeno Kang e as SVM.
- Experimentar com $\nu$-SVM. O parâmetro $\nu$ controla o número de vetores de suporte e porém o número de regras.
- Estudar a relação dos sistemas de lógica difusa tipo-2 e os métodos kernel.
- Explorar outras funções kernel difusas.
- Usar fuzzy kernels em algoritmos como regressão logística e PCA.
- Estudar modelos de fuzzificação. Considerando interpretação possibilística, variáveis aleatórias difusas ou outros.

# Referências Bibliográficas


[AFFL+11] JesÃºs AlcalÃ¡-Fdez, Alberto FernÃ¡ndez, JuliÃ¡n Luengo, JoaquÃn Derrac e Salvador GarcÃa. Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Multiple-Valued Logic and Soft Computing*, 17(2-3):255–287, 2011. 33, 37, 39

[AFG+00] A.M Anile, B Falcidieno, G Gallo, M Spagnuolo e S Spinello. Modeling undertain data with fuzzy b-splines. *Fuzzy Sets and Systems*, 113(3):397 – 410, 2000. 1, 17

[APM08] Salih Aytar, Serpil Pehlivan e Musa A. Mammadov. The core of a sequence of fuzzy numbers. *Fuzzy Sets and Systems*, 159(24):3369 – 3379, 2008. <ce:title>Theme: Fuzzy Intervals and Optimisation</ce:title>. 6

[Aro50] N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68, 1950. 12, 14

[Ban81] G. Banon. Distinction between several subsets of fuzzy measures. *Fuzzy Sets and Systems*, 5(3):291 – 305, 1981. 1

[BB99a] A. Baraldi e P. Blonda. A survey of fuzzy clustering algorithms for pattern recognition. i. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 29(6):778 –785, dec 1999. 17

[BB99b] A. Baraldi e P. Blonda. A survey of fuzzy clustering algorithms for pattern recognition. ii. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 29(6):786 –801, Dezembro 1999. 17

[BC11] Adrian I. Ban e Lucian C. Coroianu. Translation invariance and scale invariance of approximations of fuzzy numbers. *EUSFLAT*, 1:742 – 748, 2011. 6

[BCR84] C. Berg, J. P. R. Christensen e P. Ressel. *Harmonic Analysis on Semigroups*. Springer, Berlin, 1984. 11, 14

[BGV92] Bernhard E. Boser, Isabelle M. Guyon e Vladimir N. Vapnik. A training algorithm for optimal margin classifiers. Em *Proceedings of the fifth annual workshop on Computational learning theory*, COLT '92, páginas 144–152, New York, NY, USA, 1992. ACM. 14

[BSt98] Peter Bartlett e John Shawe-taylor. Generalization performance of support vector machines and other pattern classifiers, 1998.

[Can10] S. Canu. Recent advances in kernel machines. Em *Proceedings of the 15th Iberoamerican congress conference on Progress in pattern recognition, image analysis, computer vision, and applications*, páginas 1–1. Springer-Verlag, 2010. 11, 14

[CF91] JÃ¡nos C. e Fodor. On fuzzy implication operators. *Fuzzy Sets and Systems*, 42(3):293 – 300, 1991. 7

[CH04] Jung-Hsien Chiang e Pei-Yi Hao. Support vector learning mechanism for fuzzy rule-based modeling: a new approach. *IEEE Transactions on Fuzzy Systems*, 12(1):1 – 12, feb. 2004. 19, 24

[CJ11] Wei-Yuan Cheng e Chia-Feng Juang. An incremental support vector machine-trained ts-type fuzzy system for online classification problems. *Fuzzy Sets Syst.*, 163:24–44, January 2011. 19

[CLR+11] S. Canu, G. Loosli, A. Rakotomamonjy et al. Svm and kernel machines. 2011. 11, 14

[CMR09] S. Canu, X. Mary e A. Rakotomamonjy. Functional learning through kernels. *Arxiv preprint arXiv:0910.1013*, 2009. 11, 14

[Cov65] T. M. Cover. Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition. *Electronic Computers, IEEE Transactions on*, EC-14(3):326–334, 1965. 14

[CRP+11] A.B. Cara, I. Rojas, H. Pomares, C. Wagner e H. Hagras. On comparing non-singleton type-1 and singleton type-2 fuzzy controllers for a nonlinear servo system. Em *Advances in Type-2 Fuzzy Logic Systems (T2FUZZ), 2011 IEEE Symposium on*, páginas 126 –133, april 2011. 9

[CS06a] S. Canu e A. Smola. Kernel methods and the exponential family. *Neurocomputing*, 69(7-9):714–720, 2006. 11, 14

[CS06b] J. Casillas e L. Sánchez. Knowledge extraction from fuzzy data for estimating consumer behavior models. Em *IEEE International Conference on Fuzzy Systems(FUZZ-IEEE 2006)*, páginas 572–578, 2006. 1, 17

[CST00] Nello Cristianini e John Shawe-Taylor. *An introduction to support Vector Machines: and other kernel-based learning methods.* Cambridge University Press, New York, NY, USA, 2000. 11, 14

[CT11] Teck Wee Chua e Woei Wan Tan. Non-singleton genetic fuzzy logic system for arrhythmias classification. *Engineering Applications of Artificial Intelligence*, 24(2):251–259, 2011. 9

[CV95] Corinna Cortes e Vladimir Vapnik. Support-vector networks. *Mach. Learn.*, 20(3):273–297, Setembro 1995. 14

[CW03] Yixin Chen e J.Z. Wang. Support vector learning for fuzzy rule-based classification systems. *IEEE Transactions on Fuzzy Systems*, 11(6):716 – 728, dec. 2003. 19, 24, 30, 39

[CYP96] Zheru Chi, Hong Yan e Tuan Pham. *Fuzzy Algorithms: With Applications to Image Processing and Pattern Recognition.* World Scientific Publishing Co., Inc., River Edge, NJ, USA, 1996. 17

[DHR96] Dimiter Driankov, Hans Hellendoorn e Michael Reinfrank. *An introduction to fuzzy control (2. Aufl.).* Springer, 1996. 8

[DK96] J.A. Dickerson e B. Kosko. Fuzzy function approximation with ellipsoidal rules. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 26(4):542 –560, aug 1996. 10, 17

[DP78] D. Dubois e H. Prade. Operations on fuzzy numbers. *International Journal of Systems Sciences*, 9:613–626, 1978. 6

[DP88] D. Dubois e H. Prade. *Possibility theory*. Plenum Press, New-York, 1988. 1, 18

[FA10] A. Frank e A. Asuncion. UCI machine learning repository, 2010. 28

[Fle87] R. Fletcher. *Practical methods of optimization; (2nd ed.)*. Wiley-Interscience, New York, NY, USA, 1987. 15

[FYE12] Y. Forghani, H. Sadoghi Yazdi e S. Effati. Comment on âsupport vector machine for classification based on fuzzy training dataâ by a.-b. ji, j.-h. pang, h.-j. qiu [expert systems with applications 37 (2010) 3495â3498]. *Expert Systems with Applications*, 39(8):7581 – 7583, 2012. 18

[Hak05] Z. Hakan Akpolat. Non-singleton fuzzy logic control of a dc motor. *Journal of Applied Sciences*, 5:887–891, 2005. 9

[Hao08] Pei-Yi Hao. Fuzzy one-class support vector machines. *Fuzzy Sets and Systems*, 159(18):2317 – 2336, 2008. <ce:title>Theme: Information Processing</ce:title>. 18

[HC07] Pei-Yi Hao e Jung-Hsien Chiang. A Fuzzy Model of Support Vector Regression Machine. *International Journal of Fuzzy Systems*, 9(1):45–50, Março 2007. 18

[HH03] Dug Hun Hong e Changha Hwang. Support vector fuzzy regression machines. *Fuzzy Sets and Systems*, 138(2):271 – 281, 2003. 18

[HL02a] Chih-Wei Hsu e Chih-Jen Lin. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, 2002. 30, 31, 32

[HL02b] H P Huang e Y H Liu. Fuzzy support vector machines for pattern recognition and data mining. *International Journal Of Fuzzy Systems*, 4(3):826–835, 2002. 18

[IA01] T. Inoue e S. Abe. Fuzzy support vector machines for pattern classification. Em *Neural Networks, 2001. Proceedings. IJCNN '01. International Joint Conference on*, volume 2, páginas 1449 –1454 vol.2, 2001. 18

[JAS12] F. Herrera JosÃ© A. SÃ¡ez, J. Luengo. On the suitability of fuzzy rule-based classification systems with noisy data. *IEEE Transactions on Fuzzy Systems*, 2012. (In press). 33

[JCC07] Chia-Feng Juang, Shih-Hsuan Chiu e Shu-Wew Chang. A self-organizing ts-type fuzzy network with support vector learning and its application to classification problems. *IEEE Transactions on Fuzzy Systems*, 15(5):998 –1008, oct. 2007. 19, 24

[JK03] Tony Jebara e Risi Kondor. Bhattacharyya and expected likelihood kernels. Em *In Conference on Learning Theory*. press, 2003. 11

[JL98] Chia-Feng Juang e Chin-Teng Lin. An online self-constructing neural fuzzy inference network and its applications. *IEEE Transactions on Fuzzy Systems*, 6(1):12 –32, feb 1998. 19

[JL99] Jin-Tsong Jeng e Tsu-Tain Lee. Support vector machines for the fuzzy neural networks. Em *Systems, Man, and Cybernetics, 1999. IEEE SMC '99 Conference Proceedings. 1999 IEEE International Conference on*, volume 6, páginas 115 –120 vol.6, 1999. 19

[JoNYaBIE08] A.V. Joshi e State University of New York at Binghamton. Industrial Engineering. *Extension of support vector machines for imprecise data using fuzzy set theory*. State University of New York at Binghamton, 2008. 18

[JPQ10] Ai-bing Ji, Jia-hong Pang e Hong-jie Qiu. Support vector machine for classification based on fuzzy training data. *Expert Syst. Appl.*, 37(4):3495–3498, Abril 2010. 18

[KHP04] Dongwon Kim, Sung-hoe Huh e Gwi-tae Park. Modeling corrupted time series data via nonsingleton fuzzy logic system 2 nonsingleton fuzzy logic system. *Time*, páginas 1298–1303, 2004. 9

[KM96] V. Kreinovich e G. C. Mouzouris. Fuzzy rule based modeling as a universal approximation tool, 1996. 9

[KM98] N.N. Karnik e J.M. Mendel. Introduction to type-2 fuzzy logic systems. Em *Fuzzy Systems Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*, volume 2, páginas 915–920. IEEE, 1998. 18

[KM01a] Nilesh N. Karnik e Jerry M. Mendel. Operations on type-2 fuzzy sets. *Fuzzy Sets and Systems*, 122(2):327–348, 2001. 18

[KM01b] N.N. Karnik e J.M. Mendel. Centroid of a type-2 fuzzy set. *Information Sciences*, 132(1):195–220, 2001. 18

[KML99] N.N. Karnik, J.M. Mendel e Q. Liang. Type-2 fuzzy logic systems. *IEEE Transactions on Fuzzy Systems*, 7(6):643–658, 1999. 18

[KMP00] Erich Peter Klement, Radko Mesiar e Endre Pap. *Triangular Norms*. Springer, 1 edição, 2000. 6, 17

[Kol41] A. N. Kolmogorov. Stationary sequences in Hilbert space. *Bull. Math. Univ. Moscow*, 2(6), 1941. 14

[Kos94] B. Kosko. Fuzzy systems as universal approximators. *Computers, IEEE Transactions on*, 43(11):1329 –1333, nov 1994. 8

[Lee04] K. H. Lee. *First Course On Fuzzy Theory And Applications*. SpringerVerlag, 2004. 6

[LK99] Werner Van Leekwijck e E E Kerre. Defuzzification: criteria and classification. *Fuzzy Sets and Systems*, 108(2):159–178, 1999. 8

[LM00] Q. Liang e J.M. Mendel. Interval type-2 fuzzy logic systems: Theory and design. *IEEE Transactions on Fuzzy Systems*, 8(5):535–550, 2000. 18

[LW02] Chun-Fu Lin e Sheng-De Wang. Fuzzy support vector machines. *IEEE Transactions on Neural Networks*, 13(2):464 –471, mar 2002. 18

[LYL+06] Chin-Teng Lin, Chang-Mao Yeh, Sheng-Fu Liang, Jen-Feng Chung e N. Kumar. Support-vector-based fuzzy neural network for pattern classification. *IEEE Transactions on Fuzzy Systems*, 14(1):31 – 41, feb. 2006. 19, 24

[LZ71] L.A. e Zadeh. Similarity relations and fuzzy orderings. *Information Sciences*, 3(2):177 – 200, 1971. 5, 7, 17

[Mam74] E.H. Mamdani. Application of fuzzy algorithms for control of simple dynamic plant. *Electrical Engineers, Proceedings of the Institution of*, 121(12):1585 –1588, december 1974. 8

[MC05] Patricia Melin e Oscar Castillo. *Hybrid Intelligent Systems for Pattern Recognition Using Soft Computing: An Evolutionary Approach for Neural Networks and Fuzzy Systems (Studies in Fuzziness and Soft Computing)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005. 5, 17

[Men42] K Menger. Statistical metrics. *Proceedings of the National Academy of Sciences of the United States of America*, 28, 1942. 6, 17

[Men95] J.M. Mendel. Fuzzy logic systems for engineering: a tutorial. *Proceedings of the IEEE*, 83(3):345 –377, Março 1995. 8

[Men01] J. M. Mendel. *Uncertain rule-based fuzzy logic system: introduction and new directions*. Prentice–Hall PTR, 2001. 8, 9, 10, 17, 18

[Men07] J.M. Mendel. Advances in type-2 fuzzy sets and systems. *Information Sciences*, 177(1):84–110, 2007. 18

[Mer09] J. Mercer. Functions of positive and negative type, and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 209:pp. 415–446, 1909. 13, 14

[MJ02] J.M. Mendel e R.I.B. John. Type-2 fuzzy sets made simple. *IEEE Transactions on Fuzzy Systems*, 10(2):117–127, 2002. 18

[MJL06] J.M. Mendel, R.I. John e F. Liu. Interval type-2 fuzzy logic systems made simple. *IEEE Transactions on Fuzzy Systems*, 14(6):808–821, 2006. 18

[MM96a] G. C. Mouzouris e J. M. Mendel. Nonlinear predictive modeling using dynamic non-singleton fuzzy logic systems. Em *Proc. Fifth IEEE Int Fuzzy Systems Conf*, volume 2, páginas 1217–1223, 1996. 18

[MM96b] G.C. Mouzouris e J.M. Mendel. Designing fuzzy logic systems for uncertain environments using a singular-value-qr decomposition method. Em *Fuzzy Systems, 1996., Proceedings of the Fifth IEEE International Conference on*, volume 1, páginas 295 –301 vol.1, sep 1996. 10, 17

[MM97a] G.C. Mouzouris e J.M. Mendel. Dynamic non-singleton fuzzy logic systems for nonlinear modeling. *IEEE Transactions on Fuzzy Systems*, 5(2):199 –208, may 1997. 9, 18

[MM97b] G.C. Mouzouris e J.M. Mendel. Nonsingleton fuzzy logic systems: theory and application. *IEEE Transactions on Fuzzy Systems*, 5(1):56 –71, Fevereiro 1997. 8, 9, 18, 23, 24

[MMR+01] K.-R. Muller, S. Mika, G. Ratsch, K. Tsuda e B. Scholkopf. An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, 12(2):181 –201, mar 2001. 11

[MS89] Toshiaki Murofushi e Michio Sugeno. An interpretation of fuzzy measures and the choquet integral as an integral with respect to a fuzzy measure. *Fuzzy Sets and Systems*, 29(2):201 – 227, 1989. 1

[NP08] Efendi N. Nasibov e Sinem Peker. On the nearest parametric approximation of a fuzzy number. *Fuzzy Sets and Systems*, 159(11):1365 – 1375, 2008. 6

[OMCS04] C.S. Ong, X. Mary, S. Canu e A.J. Smola. Learning with non-positive kernels. Em *Proceedings of the twenty-first international conference on Machine learning*, página 81. ACM, 2004. 14

[PAF12] A. Palacios e J. Alcala-Fdez. Mining fuzzy association rules from low quality data. *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, páginas 0–0, 2012. 1, 17

[PG90] R. Poggio e F. Girosi. Regularization algorithms for learning that are equivalent to multilayer networks. *Science*, 247:978–982, 1990.

[PG07] Witold Pedrycz e Fernando A. C. Gomide. *Fuzzy Systems Engineering - Toward Human-Centric Computing*. Wiley, 2007. 5, 17

[PSC09] A.M. Palacios, L. Sánchez e I. Couso. Extending a simple genetic cooperative-competitive learning fuzzy classifier to low quality datasets. *Evolutionary Intelligence*, 2(1-2):73–84, 2009. 18, 37

[PSC10a] A. Palacios, L. Sánchez e I. Couso. Diagnosis of dyslexia with low quality data with genetic fuzzy systems. *International Journal of Approximate Reasoning*, 51(8):993–1009, 2010. 18

[PSC10b] A.M. Palacios, L. Sanchez e I. Couso. Preprocessing vague imbalanced datasets and its use in genetic fuzzy classifiers. Em *Fuzzy Systems (FUZZ), 2010 IEEE International Conference on*, páginas 1 –8, july 2010. 18

[PSC11a] A. Palacios, L. Sánchez e I. Couso. Linguistic cost-sensitive learning of genetic fuzzy classifiers for imprecise data. *International Journal of Approximate Reasoning*, páginas 0–0, 2011. 18

[PSC11b] A.M. Palacios, L. Sanchez e I. Couso. Using the adaboost algorithm for extracting fuzzy rules from low quality data: Some preliminary results. Em *Fuzzy Systems (FUZZ), 2011 IEEE International Conference on*, páginas 1263 –1270, june 2011. 18

[PSC11c] Ana M. Palacios, Luciano Sánchez e Inés Couso. Future performance modeling in athletism with low quality data-based genetic fuzzy systems. *Multiple-Valued Logic and Soft Computing*, 17(2-3):207–228, 2011. 18, 37

[RBCG07] A. Rakotomamonjy, F. Bach, S. Canu e Y. Grandvalet. More efficiency in multiple kernel learning. Em *Proceedings of the 24th international conference on Machine learning*, páginas 775–782. ACM, 2007. 11, 14

[RC05] A. Rakotomamonjy e S. Canu. Frames, reproducing kernels, regularization and learning. *The Journal of Machine Learning Research*, 6:1485–1515, 2005. 14

[RK93] D. Ruan e E.E. Kerre. Fuzzy implication operators and generalized fuzzy method of cases. *Fuzzy Sets and Systems*, 54(1):23 – 37, 1993. 7

[SC07] L. Sanchez e I. Couso. Advocating the use of imprecisely observed data in genetic fuzzy systems. *IEEE Transactions on Fuzzy Systems*, 15(4):551 –562, aug. 2007. 17, 18

[SCC06] L. Sánchez, I. Couso e J. Casillas. A multiobjective genetic fuzzy system with imprecise probability fitness for vague data. Em *2nd International Symposium on Evolving Fuzzy Systems 2006(EFS06)*, páginas 131–137, 2006. 1, 17

[SCC09] L. Sánchez, I. Couso e J. Casillas. Genetic learning of fuzzy rules based on low quality data. *Fuzzy Sets and Systems*, 160(17):2524–2552, 2009. 18

[SD01] S. Sohn e C.H. Dagli. Advantages of using fuzzy class memberships in self-organizing map and support vector machines. Em *International Joint Conference on Neural Networks, 2001. Proceedings. IJCNN '01.*, volume 3, páginas 1886 –1890 vol.3, 2001. 19

[Set99] M. Setnes. Supervised fuzzy clustering for rule extraction. Em *Fuzzy Systems Conference Proceedings, 1999. FUZZ-IEEE '99. 1999 IEEE International*, volume 3, páginas 1270 –1274 vol.3, 1999. 10, 17

[SF07] Andr Simon e George T Flowers. Non-singleton fuzzy sets for disturbance attenuation. *International Journal*, (September):36849–36849, 2007. 9

[SLH12] J.A. Sáez, J. Luengo e F. Herrera. On the suitability of fuzzy rule-based classification systems with noisy data. *IEEE Transactions on Fuzzy Systems*, 2012. 34

[Sme91] Ph. Smets. Varieties of ignorance and the need for well-founded theories. *Information Sciences*, 57:135–144, 1991. 1

[Sme96] Philippe Smets. Imperfect information: Imprecision and uncertainty. Em *Uncertainty Management in Information Systems*, páginas 225–254. 1996. 1

[SOV06] L. Sánchez, J. Otero e J.R. Villar. Boosting of fuzzy models for high-dimensional imprecise datasets. Em *11th International Conference on Information Processing and Management(IPMU2006)*, páginas 1965–1973, 2006. 18

[SS02] Bernhard Schölkopf e Alexander J. Smola. *Learning with kernels : support vector machines, regularization, optimization, and beyond*. Adaptive computation and machine learning. MIT Press, 2002. 11, 12, 13, 14

[SS03] Zonghai Sun e Youxian Sun. Fuzzy support vector machine for regression estimation. Em *IEEE International Conference on Systems, Man and Cybernetics, 2003.*, volume 4, páginas 3336 – 3341 vol.4, oct. 2003. 18

[STBWA98] J. Shawe-Taylor, P.L. Bartlett, R.C. Williamson e M. Anthony. Structural risk minimization over data-dependent hierarchies. *IEEE Transactions on Information Theory*, 44(5):1926 –1940, sep 1998.

[STC04] John Shawe-Taylor e Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, New York, NY, USA, 2004. 2, 11, 14

[StHB+96] John Shawe-taylor, Royal Holloway, Peter L. Bartlett, Systems Engineering Dept, Robert C. Williamson, Engineering Dept e Martin Anthony. A framework for structural risk minimisation, 1996.

[TA03] D. Tsujinishi e S. Abe. Fuzzy least squares support vector machines. Em *Proceedings of the International Joint Conference on Neural Networks, 2003.*, volume 2, páginas 1599 – 1604 vol.2, july 2003. 18

[Tan82] Uejima Satoru Asai Kiyoji Tanaka, Hideo. Linear regression analysis with fuzzy model. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-12(6):903–907, 1982. cited By (since 1996) 460. 18

[TS85] T Takagi e M Sugeno. Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions On Systems Man And Cybernetics*, 15(1):116–132, 1985. 8

[Vap95] Vladimir N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995. 14

[VOOS10] J.R. Villar, A. Otero, J. Otero e L. Sánchez. Taximeter verification with gps and soft computing techniques. *Soft Computing*, 14(4):405–418, 2010. 1, 17

[Wan94] Li-Xin Wang. *Adaptive fuzzy systems and control - design and stability analysis*. Prentice Hall, 1994. 10, 17

[WL10] Qi Wu e Rob Law. Fuzzy support vector regression machine with penalizing gaussian noises on triangular fuzzy number space. *Expert Syst. Appl.*, 37(12):7788–7795, Dezembro 2010. 18

[WM92] L.-X. Wang e J.M. Mendel. Fuzzy basis functions, universal approximation, and orthogonal least-squares learning. *IEEE Transactions on Neural Networks*, 3(5):807 –814, Setembro 1992. 8, 19

[WSS01] R.C. Williamson, A.J. Smola e B. Scholkopf. Generalization performance of regularization networks and support vector machines via entropy numbers of compact operators. *IEEE Transactions on Information Theory*, 47(6):2516 –2532, Setembro 2001. 13, 14

[Wu10] Qi Wu. Regression application based on fuzzy ν-support vector machine in symmetric triangular fuzzy space. *Expert Syst. Appl.*, 37(4):2808–2814, Abril 2010. 18

[WW99] Congxin Wu e Cong Wu. Some notes on the supremum and infimum of the set of fuzzy numbers. *Fuzzy Sets and Systems*, 103(1):183 – 187, 1999. 6

[WW08] Chua Teck Wee e Tan Woei Wan. Efsvm-fcm: Evolutionary fuzzy rule-based support vector machines classifier with fcm clustering. Em *Fuzzy Systems, 2008. FUZZ-IEEE 2008. (IEEE World Congress on Computational Intelligence). IEEE International Conference on*, páginas 606 –612, june 2008. 19

[YX07] Hong-Sen Yan e Duo Xu. An approach to estimating product design time based on fuzzy $\nu$ -support vector machine. *IEEE Transactions on Neural Networks*, 18(3):721 –731, may 2007. 18

[YZ92] R.R. Yager e L.A. Zadeh. *An Introduction to fuzzy logic applications in intelligent systems*. Kluwer international series in engineering and computer science. Kluwer Academic, 1992. 17

[YZ08] Fusheng Yu e Mingxin Zhang. Generalized triangular norms based product and similarity of fuzzy sets. Em *FSKD (1)'08*, páginas 286–290, 2008. 6

[Zad65] Lotfi A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965. 5, 17

[Zad68] L.A. Zadeh. Probability measures of fuzzy events. *Journal of Mathematical Analysis and Applications*, Vol.23, No.2, pp.:421–427., 1968.

[Zad73] Lotfi A. Zadeh. Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man, and Cybernetics*, (1):28–44, 1973. 7

[Zad74] Lotfi A. Zadeh. Fuzzy logic and its application to approximate reasoning. Em *World Computer Congress*, páginas 591–594, 1974. 8

[Zad75a] Lotfi A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning - i. *Inf. Sci.*, 8(3):199–249, 1975. 5, 8, 17

[Zad75b] Lotfi A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning-iii. *Inf. Sci.*, 9(1):43–80, 1975. 5, 17

[Zad75c] Lotfi A. Zadeh. The concept of a linguistic variable and its applicationto approximate reasoning - ii. *Inf. Sci.*, 8(4):301–357, 1975. 5, 7, 17

[Zad94a] L.A. Zadeh. Soft computing and fuzzy logic. *Software, IEEE*, 11(6):48 –56, nov. 1994. 17

[Zad94b] Lotfi A. Zadeh. Fuzzy logic, neural networks, and soft computing. *Commun. ACM*, 37(3):77–84, 1994. 17

[Zad99] L A Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 100(Supplement 1):9–34, 1999. 1, 18

[Zad10] Lotfi A. Zadeh. A summary and update of fuzzy logic. Em *IEEE International Conference on Granular Computing*, páginas 42–44, 2010. 5

[ZG07] Shang-Ming Zhou e J.Q. Gan. Constructing l2-svm-based fuzzy classifiers in high-dimensional space with automatic model selection and fuzzy rule ranking. *IEEE Transactions on Fuzzy Systems*, 15(3):398 –409, june 2007. 19, 24

[ZW04] Xingquan Zhu e Xindong Wu. Class noise vs. attribute noise: a quantitative study of their impacts. *Artif. Intell. Rev.*, 22:177–210, November 2004. 33

[ZWY04] Xingquan Zhu, Xindong Wu e Ying Yang. Error detection and impact-sensitive instance ranking in noisy datasets. Em *Proceedings of the 19th national conference on Artifical intelligence*, AAAI'04, páginas 378–383. AAAI Press, 2004. 33