

# AVALIAÇÃO DE COMBINAÇÕES DE CLASSIFICADORES FUZZY

Elísia dos Santos Prucole

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA CIVIL.

Aprovada por:

---

Prof. Alexandre Gonçalves Evsukoff, D.Sc.

---

Prof. Luiz Landau, D.Sc.

---

Prof. Beatriz de Souza Leite Pires de Lima, D.Sc.

---

Prof. Gerson Zaverucha, D.Sc.

RIO DE JANEIRO, RJ - BRASIL

DEZEMBRO DE 2006

PRUCOLE, ELÍSIA DOS SANTOS

Avaliação de Combinações de Classificadores Fuzzy [Rio de Janeiro] 2006

XI, 68 p. 29,7 cm (COPPE/UFRJ, M.Sc., Engenharia Civil, 2006)

Dissertação - Universidade Federal do Rio de Janeiro, COPPE

1. Classificação de dados
2. Lógica Fuzzy

I. COPPE/UFRJ II. Título ( série )

Aos meus pais, José Elias e Maria da Penha

## **AGRADECIMENTOS**

A Deus, por tudo o que tenho e sou e por mais um objetivo alcançado em minha vida;

Ao meu orientador Alexandre Evsukoff pela orientação, paciência, boa-vontade, atenção, ensinamentos e amizade;

Aos meus pais pelo amor, pela dedicação, por tudo que eles me ensinaram e pelo apoio incondicional e essencial em todos os momentos desta caminhada;

Ao professor Luiz Landau pela participação na banca, pelo apoio e incentivo e por conceder-me a bolsa de estudos;

Aos Professores Beatriz de Souza Leite Pires de Lima e Gerson Zaverucha por aceitarem participar da banca de avaliação desta dissertação;

À Agência Nacional do Petróleo por ter financiado esta pesquisa;

Ao Programa de Engenharia Civil por fornecer ótimas condições de pesquisa a seu corpo discente;

À competente equipe do laboratório de informática pelo zelo e pelo auxílio dado aos usuários em muitas situações;

A Marcelo, por seu fundamental companheirismo, sempre mostrando sua preocupação e incentivo para a conclusão deste trabalho;

A todas as pessoas que de uma forma direta ou indireta colaboraram para o desenvolvimento deste trabalho;

A todos muitíssimo obrigada.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

## AVALIAÇÃO DE COMBINAÇÕES DE CLASSIFICADORES FUZZY

Elisia dos Santos Prucole

Dezembro/2006

Orientador: Alexandre Gonçalves Evsukoff

Programa: Engenharia Civil

A classificação de dados encontra grande aplicação em diversas áreas da indústria do petróleo, tais como a classificação de óleos, de rochas e de imagens, determinação de limites de reservatórios, dentre outras. Este trabalho tem por objetivo desenvolver um sistema inteligente para a classificação de dados. Algoritmos de classificação *fuzzy* e baseados em regras de decisão foram implementados e tiveram seus desempenhos avaliados isoladamente. A estratégia de *ensembles* para combinar os resultados de classificadores foi empregada, observando-se uma melhora significativa nos resultados com a utilização de classificadores combinados. Diversos problemas *benchmark* foram empregados para avaliar o desempenho dos algoritmos implementados em função de várias métricas de desempenho. Uma base de dados de classificação de óleos também foi utilizada nos testes, mostrando a aplicabilidade das técnicas estudadas à indústria do petróleo.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

## EVALUATION OF FUZZY CLASSIFIERS ENSEMBLES

Elisia dos Santos Prucole

December/2006

Advisor: Alexandre Gonçalves Evsukoff

Department: Civil Engineering

Data classification finds large application in several areas of petroleum industry, such as the classification of oils, of rocks and images, determination of reservoirs limits, among others. This work intends to develop an intelligent system to perform data classification. Fuzzy classification algorithms and decision rules based algorithms were implemented, and they had their performances evaluated separately. The ensembles strategy to combine the results of classifiers was used, being observed a significant improvement in the results with the use of combined classifiers. Several benchmark problems were used in order to evaluate the performance of the implemented algorithms as function of several performance metrics. A database of oil classification was also used in the tests, showing the applicability of the techniques studied to the petroleum industry.

# Índice

---

1. Introdução.....	1
1.1. <i>Data Mining</i> .....	2
1.2. A engenharia de petróleo e a lógica <i>fuzzy</i> .....	6
1.3. Organização do Trabalho.....	9
2. Algoritmos de Classificação <i>Fuzzy</i> .....	10
2.1. Introdução.....	10
2.2. A teoria dos conjuntos <i>fuzzy</i> .....	10
2.3. Classificação <i>fuzzy</i> .....	11
2.4. <i>Fuzzy Pattern Matching</i> .....	13
2.4.1 Método Clássico.....	13
2.4.2 Estimativa das distribuições de possibilidade.....	13
2.4.3 Classificação.....	16
2.4.4 Limites do método clássico.....	16
2.5. <i>Fuzzy Pattern Matching</i> com Função Exponencial.....	17
2.5.1 Divisão de cada classe em $s_i$ subclasses.....	17
2.5.2 Geração de uma função exponencial para cada subclasse.....	18
2.5.3 Fusão das funções exponenciais.....	19
2.5.4 Agregação entre a função global e os valores de possibilidade.....	19
2.6. <i>Fuzzy Pattern Matching</i> Multidensidade.....	20
2.6.1 Divisão de cada classe em $s_i$ subclasses.....	20
2.6.2 Agregação dos valores de possibilidade.....	20
2.6.3 Fusão dos graus de pertinência.....	21
2.7. Conclusão.....	21
3. Algoritmos de Regras de Decisão.....	22
3.1. Introdução.....	22

3.2. <i>DataSqueezer</i> .....	22
3.3. Algoritmo Proposto.....	31
3.4. Conclusão.....	33
4. <i>Ensemble</i> .....	34
4.1. Introdução.....	34
4.2. Vantagens.....	35
4.3. <i>Bagging</i> .....	36
4.4. <i>Boosting</i> .....	37
4.5. Outros métodos de construção de <i>ensembles</i> .....	38
4.5.1 Votação com peso.....	38
4.5.2 Votação sem peso.....	38
4.6. Conclusão.....	38
5. Resultados Obtidos.....	39
5.1. Problemas Estudados.....	39
5.2. Avaliação dos Resultados.....	42
5.3. Resultados Obtidos.....	45
5.3.1 Algoritmos de Classificação <i>Fuzzy</i> .....	45
5.3.2. <i>Ensembles</i> .....	49
6. Conclusão.....	55
Referências Bibliográficas.....	56
Anexo.....	61

# Índice de tabelas

---

Tabela 3.1. Exemplo de um problema de classificação.....	23
Tabela 3.2. <i>POS</i> - Registros positivos.....	24
Tabela 3.3. <i>NEG</i> - Registros negativos.....	24
Tabela 3.4. $G_{POS}$ .....	26
Tabela 3.5. $G_{NEG}$ .....	27
Tabela 3.6. Exemplo de um problema de classificação.....	32
Tabela 5.1. Algoritmos de classificação <i>fuzzy</i> .....	39
Tabela 5.2. Problemas estudados.....	40
Tabela 5.3. Propriedades das bases de dados.....	40
Tabela 5.4. Percentual de Classificações Corretas – Conjunto de Treinamento.....	45
Tabela 5.5. Percentual de Classificações Corretas – Conjunto de Teste.....	46
Tabela 5.6. Percentual de Classificações Corretas - Algoritmos de Regras.....	46
Tabela 5.7. Classificadores utilizados.....	50
Tabela 5.8. Cenários utilizados para realização dos experimentos.....	50
Tabela 5.9. Percentual de Classificações Corretas – Classificadores Individuais.....	51
Tabela 5.10. Percentual de Classificações Corretas – Conjunto de Treinamento.....	51
Tabela 5.11. Percentual de Classificações Corretas – Conjunto de Teste.....	51
Tabela A.1. Sensitividade – Conjunto de Treinamento.....	61
Tabela A.2. Sensitividade – Conjunto de Teste.....	61
Tabela A.3. Especificidade – Conjunto de Treinamento.....	61
Tabela A.4. Especificidade – Conjunto de Teste.....	61
Tabela A.5. Precisão – Conjunto de Treinamento.....	62
Tabela A.6. Precisão – Conjunto de Teste.....	62
Tabela A.7. Medida F – Conjunto de Treinamento.....	62
Tabela A.8. Medida F – Conjunto de Teste.....	62

Tabela A.9. Média GSP – Conjunto de Treinamento.....	63
Tabela A.10. Média GSP – Conjunto de Teste.....	63
Tabela A.11. Média GSE – Conjunto de Treinamento.....	63
Tabela A.12. Média GSE – Conjunto de Teste.....	63
Tabela A.13. Sensitividade - Algoritmo de Regras.....	64
Tabela A.14. Especificidade - Algoritmo de Regras.....	64
Tabela A.15. Precisão - Algoritmo de Regras.....	64
Tabela A.16. Medida F - Algoritmo de Regras.....	64
Tabela A.17. Média GSP - Algoritmo de Regras.....	65
Tabela A.18. Média GSE – Algoritmo de Regras.....	65
Tabela A.19. Sensitividade – Conjunto de Treinamento.....	65
Tabela A.20. Sensitividade – Conjunto de Teste.....	65
Tabela A.21. Especificidade – Conjunto de Treinamento.....	66
Tabela A.22. Especificidade – Conjunto de Teste.....	66
Tabela A.23. Precisão – Conjunto de Treinamento.....	66
Tabela A.24. Precisão – Conjunto de Teste.....	66
Tabela A.25. Medida F – Conjunto de Treinamento.....	67
Tabela A.26. Medida F – Conjunto de Teste.....	67
Tabela A.27. Média GSP – Conjunto de Treinamento.....	67
Tabela A.28. Média GSP – Conjunto de Teste.....	67
Tabela A.29. Média GSE – Conjunto de Treinamento.....	68
Tabela A.30. Média GSE – Conjunto de Teste.....	68

# Índice de figuras

---

Figura 1.1. Tarefas de Mineração de Dados.....	3
Figura 2.1. Discretização <i>fuzzy</i> para uma variável dividida em cinco conjuntos <i>fuzzy</i> .....	12
Figura 2.2. Valores de possibilidades de uma variável.....	15
Figura 2.3. Distribuição de possibilidades de uma variável.....	15
Figura 2.4. Exemplo de um problema de duas classes não-convexas.....	17
Figura 4.1. <i>Ensemble</i> formado por três classificadores .....	35
Figura 5.1. Conjunto de treinamento – base <i>spir2</i> .....	40
Figura 5.2. Conjunto de treinamento – base <i>spir3</i> .....	41
Figura 5.3. Conjunto de treinamento – base <i>wine</i> .....	41
Figura 5.4. Conjunto de treinamento – base <i>well</i> .....	42
Figura 5.5. Estrutura de uma matriz de confusão.....	43
Figura 5.6. Curva ROC – Base <i>Spir2</i> – Conjunto de Teste.....	47
Figura 5.7. Curva ROC – Base <i>Spir3</i> – Conjunto de Teste.....	47
Figura 5.8. Curva ROC – Base <i>Wine</i> – Conjunto de Teste.....	48
Figura 5.9. Curva ROC – Base <i>Well</i> – Conjunto de Teste.....	48
Figura 5.10. Curva ROC – Base <i>Spir2</i> – Conjunto de Teste.....	52
Figura 5.11. Curva ROC – Base <i>Spir3</i> – Conjunto de Teste.....	52
Figura 5.12. Curva ROC – Base <i>Wine</i> – Conjunto de Teste.....	53
Figura 5.13. Curva ROC – Base <i>Well</i> – Conjunto de Teste.....	53

# 1. Introdução

---

Diante do notável desenvolvimento da indústria do petróleo, é crescente o interesse e a necessidade de buscar ferramentas capazes de auxiliar a análise e interpretação da enorme quantidade de dados que é gerada continuamente. Os projetos de exploração e produção de petróleo têm se tornado cada vez mais sofisticados, necessitando de um espectro de informações cada vez maior para sua elaboração. Essa tendência é em parte fruto das descobertas de campos de petróleo, de grande interesse econômico, em locais com condições naturais tecnicamente complexas, tais como águas profundas da plataforma continental. Tal situação gera uma elevada demanda de recursos na busca de tecnologias apropriadas para o desenvolvimento desses campos, e, conseqüentemente, o projeto dos poços torna-se mais aprimorado e complexo.

Perante esse cenário, nas últimas décadas, a inteligência artificial tem ganhado cada vez mais espaço no setor de petróleo e gás. Atualmente, ela já pode ser considerada uma vantagem estratégica na obtenção de novas soluções em toda a indústria do petróleo. Isto pode ser observado pelo crescente número de publicações (BRAUNSCHWEIG, 1995, MOHAGHEGH, 2000, VELEZ-LANGS, 2005).

Na engenharia de petróleo, as técnicas de inteligência artificial podem ser utilizadas em uma série de aplicações, tais como: análise de incertezas, avaliação de riscos, mineração de dados, análise e interpretação de dados, e descoberta de conhecimento, a partir dos dados disponíveis (dados sísmicos, litológicos, geológicos, de perfilação e de produção).

GARCIA & MOHAGHEGH (2004), por exemplo, fizeram uso de algumas técnicas de inteligência artificial para prognosticar a produção de gás natural; e é possível aplicar tais recursos também no prognóstico da produção de óleo (WEISS *et al.*, 2002).

NIKRAVESH & AMINZADEH (2001b) apresentaram uma série de aplicações de diversas técnicas de mineração de dados, tais como Algoritmos Genéticos, Redes Neurais Artificiais e Lógica *Fuzzy*, na caracterização de reservatórios.

Sendo assim, a inteligência artificial constitui um instrumento importante à medida em que pode contribuir para a redução de custos e riscos de exploração, e ainda tornar mais eficiente e econômica a produção e recuperação de petróleo.

Este trabalho tem por objetivo desenvolver um sistema inteligente para a classificação de dados, utilizando para este fim uma técnica de *Data Mining*, que é a lógica *fuzzy*. Uma série de algoritmos de classificação será estudada e implementada, incluindo algoritmos de classificação *fuzzy*, algoritmos de regras de decisão, sendo discutido ainda o conceito de *ensemble* de classificadores.

## 1.1. *Data Mining*

A descoberta de conhecimento em bases de dados, também chamada de KDD (*Knowledge Discovery in Databases*) pode ser definida como o processo de identificação de padrões embutidos nos dados. Além disso, os padrões identificados devem ser válidos, novos, potencialmente úteis e compreensíveis (FAYYAD *et al.*, 1996).

As pesquisas relativas a este processo ganharam rápido crescimento a partir de 1990, motivadas pela evolução da tecnologia que vem permitindo a coleta, o armazenamento e o gerenciamento de quantidades cada vez maiores de dados.

O processo de descoberta de conhecimento em bases de dados envolve diversas etapas, destacando-se a seguinte seqüência:

1. Consolidação de dados: onde os dados são obtidos a partir de diferentes fontes (arquivos texto, planilhas ou bases de dados) e consolidados numa única fonte.

2. Seleção e pré-processamento: diversas transformações podem ser aplicadas sobre os dados de forma a obter, no final, um conjunto de dados preparados para utilização dos algoritmos de mineração.

3. Mineração de dados (*Data Mining*): é a etapa de extração de padrões propriamente dita, onde, primeiramente, é feita a escolha da tarefa de mineração conforme os objetivos desejáveis para a solução procurada, isto é, conforme o tipo de conhecimento que se espera extrair dos dados. Em seguida, é escolhido o algoritmo que atenda a tarefa de mineração eleita e que possa representar satisfatoriamente os padrões a serem encontrados. Os algoritmos de mineração mais comuns são: Algoritmos Estatísticos, Algoritmos Genéticos, Árvores de Decisão, Regras de Decisão, Redes Neurais Artificiais, Algoritmos de Agrupamento e Lógica *Fuzzy*.

4. Pós-processamento: nesta etapa os conhecimentos extraídos pelos algoritmos de *Data Mining* devem ser analisados, avaliados e validados junto ao especialista para verificar se a descoberta é interessante ou não aos objetivos previamente definidos.

Dentro deste contexto, a tarefa de extração do conhecimento pode ser vista sob duas vertentes:

1. Atividades Preditivas: o objetivo é prever o valor de uma variável alvo, a partir do conhecimento adquirido de um conjunto de dados nos quais o valor desta variável é conhecido.
2. Atividades Descritivas: busca-se a identificação de padrões de comportamento comuns em um conjunto de dados, não há variável alvo.

A Figura 1.1 ilustra as tarefas de mineração de dados organizadas em atividades preditivas e descritivas.



Figura 1.1. Tarefas de Mineração de Dados (REZENDE *et al.*, 2003)

A classificação é uma importante tarefa de mineração de dados. Seu objetivo é fazer previsões ou tomar decisões baseando-se na informação disponível sobre um problema (MICHIE *et al.*, 1994). Existem diversas situações em que a tarefa de classificação se faz importante, tais como diagnósticos de doenças, diagnósticos de

falhas em peças mecânicas e sistemas, reconhecimento de imagens e análises de riscos de investimentos.

O profissional da indústria do petróleo, em seu cotidiano, é levado a se defrontar com uma série de problemas de grande complexidade, além de ter que tomar decisões importantes (FLETCHER & DAVIS, 2002).

Diante deste contexto, o problema de classificação de dados encontra grande aplicação em diversas áreas da engenharia de petróleo, tais como a classificação de óleos, de rochas e de imagens, determinação de limites de reservatórios, dentre outras. Como exemplos dessas aplicações, podem ser citadas as seguintes atividades (ESPÍNDOLA, 2004):

- a descoberta das condições de prisão de uma coluna de perfuração a partir de uma extensa base de dados de histórico de perfuração;
- a detecção de derramamentos de óleos na superfície marinha por meio de imagens de satélites;
- a identificação das características que levam um projeto de pesquisa e desenvolvimento ser bem sucedido ou não;
- a classificação de óleos a partir de dados de cromatografia gasosa;
- a identificação de litofácies de poços de petróleo através de dados sísmicos;
- a determinação da localização ótima de um novo poço através de dados de sísmica tridimensional;
- a determinação dos limites de um reservatório de gás natural por meio de dados geoquímicos de superfície;
- estudos de tarifação de gás natural para oferecer preço compatível com o setor industrial, levando em consideração o combustível a ser substituído, características de localização e políticas, dentre outros fatores;
- a avaliação da viabilidade econômica de um reservatório de gás natural.

O processo de construção do classificador inicia-se com a elaboração de um modelo. Essa construção é feita analisando as amostras de uma base de dados, onde as amostras são descritas por atributos e cada uma delas pertence a uma classe predefinida, identificada por um dos atributos, chamado atributo rótulo da classe ou, simplesmente, classe. O conjunto de amostras usadas neste passo é o conjunto de treinamento, dados de treinamento ou amostras de treinamento.

As formas mais comuns de representar o conhecimento (ou padrões) aprendido na fase de treinamento são regras de classificação, árvores de decisão ou formulações

matemáticas. Este conhecimento pode ser usado para prever as classes de amostras desconhecidas futuras, bem como pode permitir um melhor entendimento dos conteúdos da base de dados.

O modelo construído deve ser testado, isto é, o modelo é usado para a classificação de um novo conjunto de amostras, independentes daquelas usadas na fase de treinamento. Este novo conjunto é chamado conjunto de teste, dados de teste ou amostras de teste. Como este conjunto também possui as classes conhecidas, após a classificação, pode-se calcular o percentual de acertos, comparando as classes previstas pelo modelo com as classes esperadas (ou conhecidas). Este percentual é conhecido como acurácia ou precisão do modelo para o conjunto de teste em questão.

Se a acurácia for considerada aceitável, o modelo pode ser usado na classificação de amostras desconhecidas futuras, ou seja, amostras cuja classe não é conhecida. A partir da identificação da necessidade de resolver um problema de classificação, deve-se escolher um dos diversos métodos existentes. Para isso, pode-se comparar esses métodos conforme os seguintes critérios (HAN & KAMBER, 2001):

- Acurácia de Predição: é a habilidade do modelo prever corretamente a classe de amostras desconhecidas.
- Desempenho: critério relativo aos custos computacionais envolvidos na geração e na utilização do modelo.
- Robustez: é a habilidade do modelo fazer previsões corretas em amostras com atributos faltando ou com ruídos.
- Escalabilidade: é a habilidade de construir um modelo eficiente a partir de grandes quantidades de dados.
- Interpretabilidade: é a habilidade de tornar compreensível o conhecimento gerado pelo modelo.

Visando melhorar a acurácia, o desempenho e a escalabilidade do modelo, é importante executar um pré-processamento sobre os dados, de forma a prepará-los para a classificação. Essa preparação envolve as seguintes tarefas:

1. Limpeza: são técnicas que devem ser usadas para garantir a qualidade dos dados. As mais comuns são eliminação de erros gerados no processo de coleta (erros de

digitação ou de leitura por sensores), tratamento de atributos faltando e eliminação ou redução de ruídos.

2. Análise de relevância: é uma análise realizada sobre os atributos das amostras de treinamento para identificar e excluir atributos irrelevantes ou redundantes, que em nada contribuem no processo de classificação. A diminuição do tamanho das amostras com essa exclusão concorre para melhorar o desempenho e a escalabilidade do modelo.

3. Transformação: as transformações mais comuns aplicáveis aos dados de treinamento são: resumo, onde um conjunto de atributos é agrupado para formar resumos; discretização, onde dados contínuos são transformados em discretos da forma baixo, médio e alto, por exemplo; transformação de tipo, para que o dado fique numa forma mais apropriada para classificação, e normalização, aplicada sobre dados contínuos para colocá-los em intervalos determinados de valores, por exemplo, entre -1 e 1.

Diversas técnicas computacionais podem ser empregadas para a execução da classificação, tais como os sistemas baseados em regras (LIU *et al.*, 1998), as árvores de decisão (QUINLAN, 1993), as redes neurais artificiais (HAYKIN, 1999) e os métodos estatísticos (HOLMSTRÖM *et al.*, 1996). Estas são, provavelmente, as mais comuns; entretanto, outras técnicas também têm sido bastante utilizadas e estão ocupando papéis de destaque na produção de sistemas classificadores, como por exemplo a computação evolucionária (GOLDBERG, 1989) e a lógica *fuzzy* (ZADEH, 1965).

## **1.2. A engenharia de petróleo e a lógica *fuzzy***

A natureza imprecisa dos dados disponíveis na indústria de petróleo faz da teoria de conjuntos *fuzzy* uma ferramenta apropriada para a interpretação destes dados.

A lógica *fuzzy* pode ser usada para extrair dimensões de corpos geológicos, e o geólogo pode usar esta técnica para a caracterização de reservatórios de maneira prática, deixando para trás procedimentos antigos e tediosos.

CHAPPAZ (1977) e BOIS (1983, 1984) propuseram o uso da teoria de conjuntos *fuzzy* para a interpretação de seções sísmicas. BOIS utilizou a lógica *fuzzy*

como ferramenta de reconhecimento de padrões para interpretação sísmica e análise de reservatórios. Ele concluiu que a teoria de conjuntos *fuzzy*, em particular, pode ser usada para interpretar dados sísmicos, que são imprecisos e contêm erro humano, sendo possível assim determinar a informação geológica e por conseguinte prognosticar os limites do reservatório que contém hidrocarbonetos.

No campo da geofísica, CHEN *et al.* (1995) fizeram uso da teoria de conjuntos *fuzzy* na extração de parâmetros para a equação da Lei de Archie.

Na análise geoestatística, vale destacar o trabalho de BEZDEK *et al.* (1981), que contém uma série de aplicações da teoria de conjuntos *fuzzy*.

A análise *fuzzy* de perfis de poços tem sido aplicada extensivamente em muitos estudos de caracterização de reservatórios. Por exemplo, FUNG *et al.* (1997) desenvolveram um sistema de inferência *fuzzy* para a predição de propriedades petrofísicas a partir de dados de perfilagem de poços, enquanto que HUANG *et al.* (1999) apresentaram um interpolador *fuzzy* para a predição de permeabilidade baseando-se em perfis de poços do noroeste da Austrália.

A lógica *fuzzy* tem sido também utilizada para determinar formações de hidrocarbonetos a partir de perfis de poços localizados na região sul do Mar do Norte (CUDDY, 2000).

As técnicas de análise *fuzzy* foram ainda aplicadas na identificação de aquíferos no Taiwan (HSIEH *et al.*, 2005).

MARTINEZ-TORRES (2002) desenvolveu um sistema *fuzzy* para a identificação e caracterização de reservatórios fraturados naturalmente a partir da perfilagem de poços. Perfis de poços convencionais constituem a maior fonte disponível de informação para o estudo de poços, e uma vez que todos os perfis de poços são afetados de alguma forma pela presença de fraturas, um sistema de inferência *fuzzy* foi implementado para obter um índice de fraturação a partir de perfis de poços convencionais, utilizando assim os perfis para a análise quantitativa de reservatórios fraturados naturalmente. O método proposto foi testado utilizando dados disponíveis do poço Mills McGee, localizado no Texas; e representa uma importante contribuição para a caracterização de reservatórios, uma vez que ele permite a identificação de fraturas no poço utilizando dados de perfilagem.

Há uma grande variedade de aplicações possíveis da lógica *fuzzy* na exploração de petróleo. Uma série destas aplicações é discutida no trabalho de AMINZADEH (1994).

HONGJIE & HOLDRTICH (1994) fizeram uso desta técnica para selecionar o método ótimo de estimulação de poços, incluindo barreiras potenciais para tratamentos de fratura. HONGJIE (1995) incluiu neste método de seleção um modelo, também utilizando a lógica *fuzzy*, para diagnosticar mecanismos de formação de danos de formação, associando a tipos de estimulação e seleção de fluidos.

A lógica *fuzzy* pode ainda ser usada para resolver problemas tais como: fraturamento hidráulico (RIVERA, 1994) e recuperação de óleo (CHUNG *et al.*, 1995).

GARROUCH & LABABIDI (2001) desenvolveram um sistema de inferência *fuzzy* para examinar poços em possível condição de desbalanceamento e para fazer uma seleção preliminar do fluido de perfuração apropriado, para uma variedade de condições de reservatório e poço.

Na área de caracterização de reservatórios, NIKRAVESH *et al.* (2001), apresentaram uma metodologia integrada para identificar relações não-lineares e mapeamento entre dados de sísmica 3D e dados de perfilagem para selecionar a localização de um novo poço.

FINOL *et al.* (2001) fizeram a implementação de um modelo baseado em regras *fuzzy* para estimar parâmetros petrofísicos, tais como porosidade e permeabilidade das rochas, utilizando dados de perfilagem e análise de testemunhos.

No campo da geoquímica, KRAMAR (1995) fez uma modificação no algoritmo de agrupamento *fuzzy c-means* e aplicou sua implementação na determinação de regiões anômalas (com ocorrência de hidrocarbonetos).

ISAKSEN & KIM (1997) desenvolveram uma metodologia, utilizando a lógica *fuzzy*, para a interpretação de dados geoquímicos.

EVSUKOFF *et al.* (2004) implementaram uma metodologia para determinar regiões anômalas através da combinação de um algoritmo de agrupamento (*fuzzy c-means*), que executa de forma integrada o processamento dos dados de geoquímica de superfície, e um classificador *fuzzy*, que por sua vez faz o mapeamento dos grupos gerados no sistema de coordenadas geográficas.

Modelos híbridos também são muito utilizados na engenharia de petróleo. ESPÍNDOLA & EBECKEN (2002) desenvolveram um sistema *fuzzy*-genético para classificar dados da indústria de petróleo. NIKRAVESH & AMINZADEH (2001a) implementaram um sistema neuro-*fuzzy* para mineração e fusão de dados sísmicos, litológicos e de perfilagem.

### 1.3. Organização do Trabalho

Este trabalho tem como objetivo o desenvolvimento e implementação de uma metodologia eficiente para a classificação de dados, e para efetuar esta proposta, está dividido em seis capítulos.

O Capítulo 2 discorre a respeito de algoritmos de classificação *fuzzy*. Inicialmente, são descritos alguns conceitos teóricos, sobre a teoria dos conjuntos *fuzzy* e classificação *fuzzy*, em seguida é feita uma descrição de uma série de algoritmos estudados e implementados, propostos na literatura e relacionados a este trabalho.

O Capítulo 3 trata de algoritmos de regras de decisão, discutindo detalhes a respeito de sua implementação, tanto do algoritmo *DataSqueezer*, como também do algoritmo de regras de decisão *fuzzy* proposto neste trabalho.

O Capítulo 4 discute a técnica de agrupar classificadores, ou *ensemble*; são discutidos alguns conceitos que motivam a construção de *ensembles* de classificadores, e também alguns métodos de construção de *ensembles* encontrados na literatura.

O Capítulo 5 apresenta a análise dos resultados obtidos, detalhando os experimentos realizados para avaliar o comportamento dos algoritmos implementados.

O Capítulo 6 traz as considerações finais inerentes ao trabalho realizado e as propostas de pesquisas futuras.

## 2. Algoritmos de Classificação *Fuzzy*

---

### 2.1. Introdução

Os problemas reais são caracterizados pela necessidade de serem capazes de processar informação vaga, incompleta, imprecisa, enfim, informação que contém incerteza associada. Esta incerteza pode ser tratada através da teoria dos conjuntos *fuzzy*.

Na engenharia de petróleo, há um vasto campo de aplicações de sistemas *fuzzy* para classificação de dados, dentre as quais pode-se citar:

- Estimativa de reservas;
- Controle e otimização de processos;
- Caracterização de reservatórios integrando dados de perfilagem de poços, análise de testemunhos e informação geológica. Neste caso, a lógica *fuzzy* pode desempenhar um importante papel, à medida em que é capaz de executar um tratamento das incertezas contidas nos dados disponíveis.
- Interpretação de seções sísmicas;
- Interpretação de dados geoquímicos;
- Determinação de regiões anômalas (com ocorrência de hidrocarbonetos);

### 2.2. A teoria dos conjuntos *fuzzy*

A teoria dos conjuntos *fuzzy* foi desenvolvida a partir de 1965 com os trabalhos de Lotfi Zadeh, professor da Universidade da Califórnia em Berkeley.

Formalmente, um conjunto *fuzzy*  $A$  do universo de discurso  $\Omega$  é definido por uma função de pertinência  $\mu_A : \Omega \rightarrow [0,1]$ . Essa função associa a cada elemento  $x$  de  $\Omega$  o grau  $\mu_A(x)$ , com o qual  $x$  pertence a  $A$ . A função de pertinência  $\mu_A(x)$  indica o grau de compatibilidade entre  $x$  e o conceito expresso por  $A$ :

- $\mu_A(x) = 1$  indica que  $x$  é completamente compatível com  $A$ ;
- $\mu_A(x) = 0$  indica que  $x$  é completamente incompatível com  $A$ ;

- $0 < \mu_A(x) < 1$  indica que  $x$  é parcialmente compatível com  $A$ , com grau  $\mu_A(x)$ .

Um conjunto  $A$  da teoria dos conjuntos clássica pode ser visto como um conjunto *fuzzy* específico, denominado usualmente de “*crisp*”, para o qual  $\mu_A : \Omega \rightarrow \{0,1\}$ , ou seja, a pertinência é do tipo “tudo ou nada”, “sim ou não”, e não gradual como para os conjuntos *fuzzy*.

Os conjuntos *fuzzy* permitem que seus elementos possuam um certo grau de pertinência associado, sendo esta propriedade conhecida como multivalência. Isto permite a aproximação com o mundo real que não é bivalente, mas é na realidade multivalente com um vasto número de opções ao invés de somente duas. A lógica *fuzzy*, então, permite trabalhar com tais incertezas de fenômenos naturais de forma rigorosa e sistemática.

### 2.3. Classificação *fuzzy*

Suponha um problema de classificação com  $n_c$  classes, onde  $\omega_1, \omega_2, \dots, \omega_{n_c}$  representam as classes do problema. A solução do problema de classificação de dados através de métodos baseados na teoria de conjuntos *fuzzy* representa cada classe do problema por um conjunto *fuzzy*:

$$\omega_1 = \{(x, \mu_{\omega_1}(x)), x \in \Omega\} \quad (2.1)$$

$$\omega_2 = \{(x, \mu_{\omega_2}(x)), x \in \Omega\} \quad (2.2)$$

⋮

$$\omega_{n_c} = \{(x, \mu_{\omega_{n_c}}(x)), x \in \Omega\} \quad (2.3)$$

A abordagem do problema de classificação *fuzzy* consiste em calcular a função de pertinência  $\mu_{\omega_j}(x)$ ,  $1 \leq j \leq n_c$ , partindo da definição de uma discretização (partição) *fuzzy* sobre o universo de cada atributo.

A representação matemática de um conjunto ordenado de conceitos da linguagem natural através de conjuntos *fuzzy* pode ser feita por uma discretização *fuzzy*  $\{A_1, A_2, \dots, A_n\}$  do universo  $\Omega$  tal que:

$$\forall x \in \Omega, \exists A_i, \mu_{A_i}(x) \neq 0$$

A Figura 2.1 apresenta um exemplo de discretização *fuzzy* para o caso de uma variável dividida em cinco conjuntos *fuzzy*:

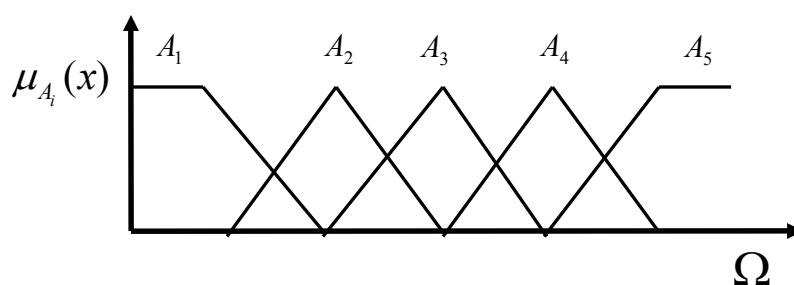


Figura 2.1. Discretização *fuzzy* para uma variável dividida em cinco conjuntos *fuzzy*

Diferentes discretizações do universo permitem a representação da informação em diferentes níveis de generalização; quanto maior for o número de conjuntos *fuzzy*, maior será a precisão obtida.

A *fuzzificação* é responsável pelo mapeamento das entradas numéricas em conjuntos *fuzzy*, variáveis lingüísticas. Na *fuzzificação*, o vetor de pertinências de entrada é calculado a partir do valor numérico de entrada e da discretização *fuzzy* de entrada.

A inferência é realizada mapeando-se valores lingüísticos de entrada em valores lingüísticos de saída com o uso da base de regras.

A base de regras é implementada de acordo com o conhecimento do especialista, e é formada por estruturas do tipo *se <premissa> então <conclusão>*. É importante que existam tantas regras quantas forem necessárias para mapear totalmente as combinações dos termos das variáveis, isto é, que a base seja completa, garantindo que exista sempre ao menos uma regra a ser disparada para qualquer entrada. Também são essenciais a consistência, onde procura-se evitar a possibilidade de contradições, e a interação entre as regras, gerenciada pela função de implicação de modo a contornar as situações de

ciclo. As premissas são relacionadas pelos conectivos lógicos, dados pelo operador de conjunção (e) e o operador de disjunção (ou).

Para cada variável, uma conclusão parcial é calculada a partir de uma base de regras. A conclusão final é obtida pela agregação de todas as conclusões parciais.

## **2.4. Fuzzy Pattern Matching**

### **2.4.1 Método Clássico**

Introduzido em 1988, este é um algoritmo de classificação *fuzzy* supervisionada que utiliza a definição de  $c$  distribuições de possibilidade descritas por  $\alpha$  atributos (DUBOIS *et al.*, 1988).

Cada atributo é representado por uma série de conjuntos *fuzzy* expressando o conjunto de valores típicos deste atributo para as classes de treinamento.

Estes conjuntos *fuzzy* são representados pelas distribuições de possibilidade denotadas  $\pi_i^k$  para a classe  $C_i$  e o atributo  $k$ .

### **2.4.2 Estimativa das distribuições de possibilidade**

As formas das classes são as distribuições de possibilidades estimadas para cada classe e cada atributo (ZADEH, 1978). O cálculo é baseado na teoria das possibilidades. No algoritmo *Fuzzy Pattern Matching*, o aprendizado consiste em construir estas distribuições, para cada classe e cada atributo, a partir das amostras.

As distribuições de possibilidade podem ser calculadas a partir das distribuições de probabilidade, que são estimadas a partir dos histogramas.

Os histogramas dos dados são estabelecidos a partir do conjunto de treinamento. Os suportes dos histogramas de uma classe  $C_i$  de acordo com um atributo são geralmente definidos pelos valores máximos e mínimos dos componentes dos pontos de treinamento dentro da classe  $C_i$ . Portanto, estes suportes podem ser determinados pelo usuário.

O parâmetro  $h$  é o número de colunas dos histogramas. A transformação dos histogramas em distribuições de possibilidade requer duas operações. Cada coluna é

representada pelo centro do intervalo  $y_i$ . A razão entre a altura da coluna e o número total de pontos de treinamento, chamada  $p(y_i)$ , determina a probabilidade associada ao centro do intervalo.

A distribuição de possibilidade  $\{\pi(y_i) | i = 1, \dots, h\}$  é deduzida a partir da distribuição de probabilidade pela transformação bijetiva de Dubois e Prade (DUBOIS & PRADE, 1987). Uma vez que os valores de probabilidade  $\{p(y_i) | i = 1, \dots, h\}$  dos centros dos intervalos  $y_i$  são arranjados em ordem decrescente  $p(y_1) \geq p(y_2) \geq p(y_3) \geq p(y_4) \geq \dots \geq p(y_h)$ , a distribuição é calculada através da seguinte expressão:

$$\pi(y_i) = \sum_{j=1}^h \min [p(y_i), p(y_j)] = i \cdot p(y_i) + \sum_{j=i+1}^h p(y_j) \quad (2.4)$$

A expressão satisfaz a condição de normalização da teoria das possibilidades:

$$\pi(y_1) = 1 \quad (2.5)$$

A transformação em distribuição de possibilidades é feita através da interpolação linear da distribuição discreta, o que é exemplificado nas Figuras 2.2 e 2.3.

A Figura 2.2 apresenta, para um problema de duas classes, os valores de possibilidade de uma variável. Como o número de colunas dos histogramas escolhido pelo usuário foi 10, há 10 valores de possibilidade para cada classe.

A Figura 2.3 mostra a distribuição de possibilidades para a mesma variável, através da interpolação linear da distribuição discreta.

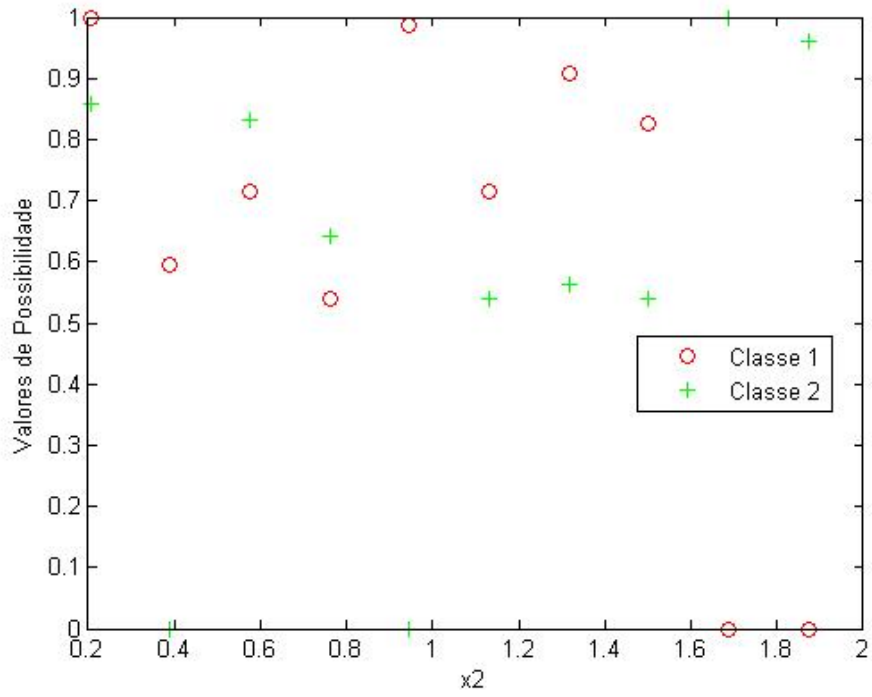


Figura 2.2. Valores de possibilidades de uma variável

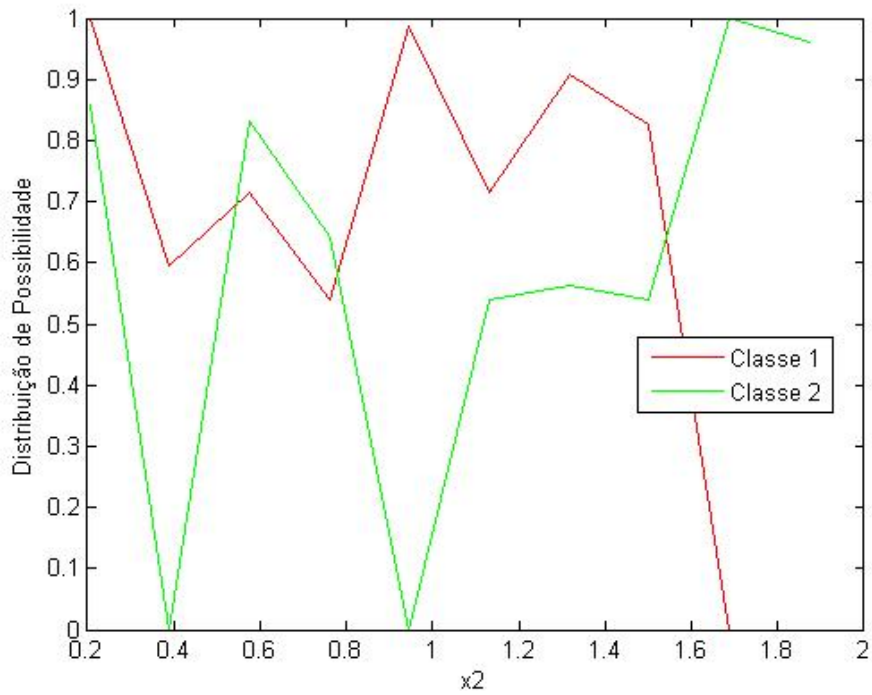


Figura 2.3. Distribuição de possibilidades de uma variável

### 2.4.3 Classificação

A classificação de uma nova amostra  $\vec{x}$ , onde os valores dos diferentes atributos são  $(x^1, \dots, x^\alpha)$  é feita em duas etapas:

- determinação do grau de similaridade entre a amostra  $\vec{x}$  e as diferentes distribuições de possibilidade. O grau de similaridade  $\pi_i^k(x^k)$  entre o ponto e a representação da distribuição de possibilidade no atributo  $k$  é obtido pela correspondência entre o valor  $x^k$  e a distribuição.

- fusão de todos os graus de similaridade  $\{\pi_i^1(x^1), \dots, \pi_i^\alpha(x^\alpha)\}$  para cada classe através de um operador de agregação  $H$ :

$$u_i(\vec{x}) = H[\pi_i^1(x^1), \dots, \pi_i^\alpha(x^\alpha)] \quad (2.6)$$

O resultado  $u_i(\vec{x})$  desta fusão representa o grau de correspondência entre a nova amostra e a classe  $C_i$ .

No método *Fuzzy Pattern Matching*, este valor é considerado ser o grau de pertinência do registro  $\vec{x}$  à classe  $C_i$ . O operador de fusão pode ser de multiplicação, mínimo, média ou integral *fuzzy*. Aqui será utilizado o operador mínimo. O registro  $\vec{x}$  é finalmente atribuído à classe de máxima pertinência.

### 2.4.4 Limites do método clássico

Este método não pode ser usado para separar classes de formato não-convexo, uma vez que nele as classes são representadas por uma distribuição de possibilidade para cada atributo. Assim, o aprendizado não integra nenhuma informação sobre o formato das classes.

A Figura 2.4 apresenta um exemplo de um problema de duas classes de formato não-convexo.

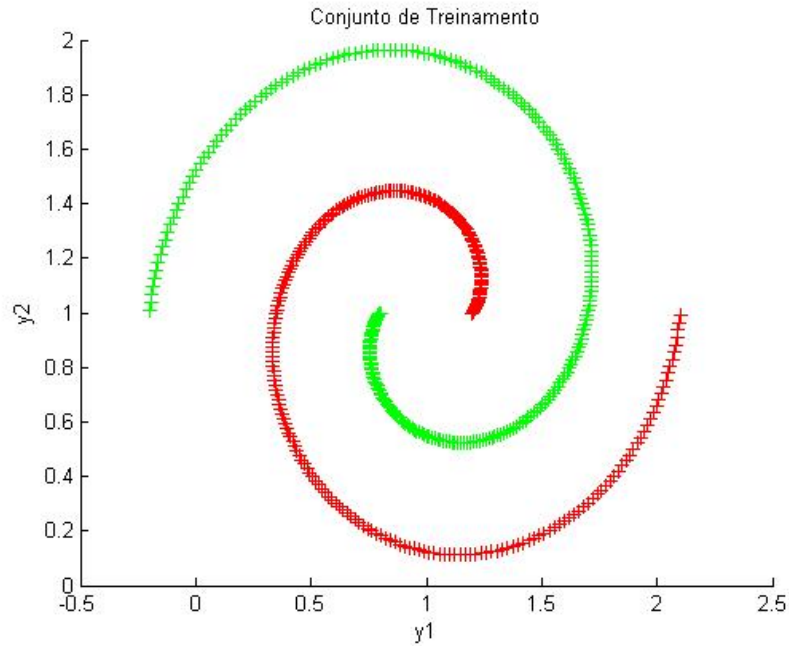


Figura 2.4. Exemplo de um problema de duas classes não-convexas

Portanto, serão estudados também dois métodos de classificação supervisionada que são aproximações do método *Fuzzy Pattern Matching* (FPM): *Fuzzy Pattern Matching* com Função Exponencial (FPME) e *Fuzzy Pattern Matching* Multidensidade (FPMM).

## 2.5. *Fuzzy Pattern Matching* com Função Exponencial

### 2.5.1 Divisão de cada classe em $s_i$ subclasses

Este é um algoritmo de classificação *fuzzy* supervisionada para classes não-convexas (DEVILLEZ, 2004). Assim, cada classe  $C_i$ , incluindo  $n_i$  pontos de treinamento, é dividida em  $s_i$  subclasses  $C_{ij}$  pelo algoritmo *fuzzy c-means*.

Não há método ou critério para determinar um número ótimo de subclasses em cada classe. O número de subclasses na classe  $C_i$  é  $s_i$ , e deve ter seu valor contido no intervalo entre 1 e  $n_i$ .

Cada subclasse é representada por um centróide  $\bar{g}_{ij}$ , o que implica que uma classe  $C_i$  é representada por  $s_i$  centróides ou protótipos.

Para cada ponto de treinamento pertencendo à classe  $C_i$ , o algoritmo *fuzzy c-means* calcula um grau de pertinência para cada subclasse.

Estes graus formam a matriz de pertinência  $W_i$ , cujo tamanho é  $s_i \times n_i$  e onde o elemento  $W_{i_{jk}}$  representa o grau de pertinência do ponto  $\bar{x}_k$  da classe  $C_i$  à subclasse  $C_{ij}$ .

## 2.5.2 Geração de uma função exponencial para cada subclasse

Para cada novo registro a ser classificado, uma função exponencial é calculada para cada subclasse:

$$\mu_{ij}(\bar{x}) = \exp \left[ -\lambda_{ij} \cdot d(\bar{x}, \bar{g}_{ij}) \right] \quad (2.7)$$

onde  $\lambda_{ij}$  é uma constante associada à subclasse  $C_{ij}$  e  $d(\bar{x}, \bar{g}_{ij})$  representa a distância *fuzzy* Mahalanobis entre a amostra  $\bar{x}$  e o centróide  $\bar{g}_{ij}$ . Ela usa uma matriz de covariância e é definida pela seguinte expressão:

$$d(\bar{x}, \bar{g}_{ij}) = (\bar{x} - \bar{g}_{ij})^t \cdot \Sigma_{ij}^{-1} \cdot (\bar{x} - \bar{g}_{ij}) \quad (2.8)$$

O conhecimento dos graus de pertinência dos pontos de treinamento permite calcular a matriz de covariância *fuzzy*  $\Sigma_{ij}$  de cada subclasse  $C_{ij}$ :

$$\Sigma_{ij} = \frac{S_{ij}}{\sum_{k=1}^{n_i} (w_{i_{jk}})^2} \quad (2.9)$$

$S_{ij}$  é a matriz de dispersão *fuzzy* definida por:

$$S_{ij} = \sum_{k=1}^{n_i} (w_{i_{jk}})^2 \cdot (\bar{x}_k - \bar{g}_{ij}) \cdot (\bar{x}_k - \bar{g}_{ij})^t \quad (2.10)$$

A introdução desta matriz no cálculo de distâncias nos permite respeitar o formato elipsóide das subclasses, respeitando a distribuição dos níveis de pertinência.

### 2.5.3 Fusão das funções exponenciais

Para cada classe  $C_i$ , uma função de pertinência global é determinada pela fusão  $F$  das  $s_i$  funções exponenciais definidas para as  $s_i$  subclasses  $(C_{i1}, \dots, C_{is_i})$ :

$$\mu_i(\vec{x}) = F(\mu_{i1}, \dots, \mu_{is_i}) = \min \left[ 1, \sum_{j=1}^{s_i} \mu_{ij}(\vec{x}) \right] \quad (2.11)$$

É possível usar outras  $t$ -conormas para o operador  $F$ , como o operador máximo. Uma boa opção é o operador soma limitada, uma vez que esse operador garante um nível mínimo de pertinência. Ele não causa um decréscimo da pertinência dos pontos localizados nas interfaces das subclasses calculadas pelo algoritmo *fuzzy c-means* e permite construir as funções de pertinência, sem ter que atingir o limite superior de definição do intervalo. Os vales de baixos níveis de pertinência que existem no caso do operador máximo são preenchidos no caso do operador soma limitada. Conseqüentemente, este operador tem um interessante poder de “suavização” na superfície de resposta do classificador e será utilizado na implementação.

### 2.5.4 Agregação entre a função global e os valores de possibilidade

O grau de pertinência  $u_i(\vec{x})$  da amostra  $\vec{x}$  à classe  $C_i$  é obtido pela agregação entre o valor da função de pertinência global e os valores de possibilidade do ponto em cada atributo. Esta agregação é feita pelo operador mínimo, e os valores de possibilidade  $\pi_i^j, j = 1, \dots, \alpha$ , são calculados conforme a expressão (2.4):

$$u_i(\vec{x}) = \min[\pi_i^1(x^1), \pi_i^2(x^2), \pi_i^3(x^3), \pi_i^4(x^4), \dots, \pi_i^\alpha(x^\alpha), \mu_i(\vec{x})] \quad (2.12)$$

O interesse por esta operação de agregação se justifica pela sua capacidade de diminuir os valores de alta pertinência, que causam a aparência de alguns picos de

pertinência na superfície de resposta do classificador. Além disso, quando os pontos estão situados na área central das classes, a função exponencial global é maior do que os valores de possibilidade em cada atributo. A superfície de resposta é também atenuada. Entretanto, se o ponto está situado na periferia das classes, a função exponencial global é menor do que as possibilidades. O operador mínimo favorece a função exponencial de pertinência que naturalmente respeita a forma das classes.

## **2.6. Fuzzy Pattern Matching Multidensidade**

### **2.6.1 Divisão de cada classe em $s_i$ subclasses**

Este é um algoritmo de classificação *fuzzy* supervisionada para classes não-convexas (DEVILLEZ, 2004). Cada classe  $C_i$  é dividida em um número  $s_i$  de subclasses  $C_{ij}$  pelo algoritmo *fuzzy c-means*, como no método anterior.

A determinação do número de subclasses apresenta a mesma dificuldade vista no método FPME.

Cada ponto do conjunto de treinamento é atribuído à subclasse de máxima pertinência. O aprendizado consiste em calcular as distribuições de possibilidade para cada subclasse e cada atributo.

Este método é chamado Multidensidade porque cada classe é representada por  $s_i$  densidades de possibilidade em cada atributo.

### **2.6.2 Agregação dos valores de possibilidade**

Quando um ponto aparece no espaço de classificação, um valor de pertinência é calculado para cada atributo e para cada subclasse  $C_{ij}$ . Esta etapa é similar ao método clássico.

Ao invés de calcular valores de pertinência para a classe, é calculado um valor para cada subclasse pertencente a cada classe  $C_i$ .

A agregação destes valores pelo operador mínimo gera uma pertinência  $\eta_{ij}$  para cada subclasse da classe  $C_i$ :

$$\eta_{ij}(\vec{x}) = \min[\pi_{ij}^1(x^1), \pi_{ij}^2(x^2), \pi_{ij}^3(x^3), \pi_{ij}^4(x^4), \dots, \pi_{ij}^\alpha(x^\alpha)] \quad (2.13)$$

A primeira etapa deste algoritmo é similar ao algoritmo clássico. Para cada classe  $C_i$ , o resultado gerado é um número  $s_i$  de graus de pertinência às subclasses que devem ser agregados.

### 2.6.3 Fusão dos graus de pertinência

Para unir os graus de pertinência calculados para cada subclasse, recomenda-se o uso do operador soma limitada.

Quando um número  $s_i$  de subclasses  $C_{ij}$  existe em uma classe  $C_i$ , a função de pertinência para esta classe é definida por:

$$u_i(\vec{x}) = F[\eta_{i1}(\vec{x}), \dots, \eta_{is_i}(\vec{x})] = \min\left[1, \sum_{j=1}^{s_i} \eta_{ij}(\vec{x})\right] \quad (2.14)$$

A amostra  $\vec{x}$  é atribuída à classe para a qual apresenta o máximo grau de pertinência.

## 2.7. Conclusão

Diante dos algoritmos apresentados, vale ressaltar as vantagens dos algoritmos *Fuzzy Pattern Matching* com Função Exponencial e *Fuzzy Pattern Matching* Multidensidade frente ao algoritmo tradicional *Fuzzy Pattern Matching*, visto que os dois primeiros oferecem ao usuário a opção de escolher o número de protótipos para representar cada classe do problema, enquanto que, pelo método tradicional, cada classe é representada por somente um protótipo.

A divisão de cada classe em várias subclasses faz com que o aprendizado respeite o formato das classes do problema, sendo portanto uma alternativa de grande valia para o caso de classes de formato não-convexo.

## 3. Algoritmos de Regras de Decisão

---

### 3.1. Introdução

Uma regra de decisão é uma implicação lógica, apresentada no formato “SE <condição> ENTÃO <classe>”; onde a <condição>, também chamada de antecedente, é formada por argumentos agrupados por operadores lógicos (*e*, *ou*, *não* etc.), e a <classe>, também chamada de conseqüente, é a instância da classe rotulada, ou seja, realiza a conclusão da informação.

A qualidade interpretativa de uma base de regras pode ser avaliada mediante uma série de fatores, tais como: a quantidade de condições e conclusões presentes na estrutura de representação (quanto mais elementos uma regra possuir, mais difícil é o seu entendimento), a quantidade de regras da base, a quantidade de atributos utilizados pelas regras e a presença de qualificadores lingüísticos para descrevê-los (ESPÍNDOLA, 2004).

A presença de qualificadores lingüísticos é um fator de grande importância em termos de facilitar a compreensão e interpretação de uma base de regras. Afinal, uma regra que utiliza elementos lingüísticos é claramente mais inteligível do que outra que utiliza elementos numéricos, uma vez que grande parte do conhecimento humano está armazenada em forma lingüística.

As técnicas originadas da teoria dos conjuntos *fuzzy* permitem a transformação dos elementos numéricos das bases de dados em elementos lingüísticos, dando origem aos sistemas baseados em regras *fuzzy*. Esta transformação se faz extremamente importante, pois a maior parte dos problemas contém informações imprecisas e ambíguas, pelo simples motivo de não existirem medidas apropriadas.

### 3.2. *DataSqueezer*

O algoritmo proposto neste trabalho é uma versão do algoritmo *DataSqueezer* (KURGAN *et al.*, 2006), sendo adicionada uma etapa inicial de *fuzzificação* do conjunto de registros. Uma descrição do algoritmo original é mostrada a seguir:

Seja  $D$  o conjunto de dados de treinamento, que contém  $N_{TOT}$  registros e  $\alpha$  atributos. Seja  $D_P$  o conjunto de registros positivos e  $D_N$  o conjunto de registros negativos.  $D_P$  e  $D_N$  devem satisfazer as seguintes propriedades:

$$D_P \cup D_N = D$$

$$D_P \cap D_N = \emptyset$$

$$D_P \neq \emptyset$$

$$D_N \neq \emptyset$$

Após a separação de classes, são geradas duas tabelas,  $POS$  e  $NEG$ , contendo  $N_{POS}$  registros positivos e  $N_{NEG}$  registros negativos, respectivamente; onde  $N_{POS} + N_{NEG} = N_{TOT}$ .

O algoritmo então executa uma redução de dados a fim de generalizar a informação contida nos dados das tabelas  $POS$  e  $NEG$ , sendo geradas nesta etapa as tabelas  $G_{POS}$  e  $G_{NEG}$ .

As tabelas  $G_{POS}$  e  $G_{NEG}$  têm uma coluna adicional  $\alpha + 1$ , onde está inserido o número de registros das tabelas  $POS$  e  $NEG$  que são descritos pelas respectivas linhas de  $G_{POS}$  e  $G_{NEG}$ , ou seja, dividindo-se a coluna  $\alpha + 1$  pelo número de registros tem-se o suporte de cada regra contida em  $G_{POS}$  e  $G_{NEG}$ .

Suponha, por exemplo, um conjunto de treinamento cujas variáveis são medidas de condições de tempo e a classe é a decisão sobre viajar ou não viajar.

A Tabela 3.1 apresenta o conjunto de treinamento do exemplo proposto.

Tabela 3.1. Exemplo de um problema de classificação

<b>Decisão</b>	<b>Temperatura</b>	<b>Vento</b>	<b>Umidade</b>
Viajar	alta	médio	baixa
Viajar	alta	médio	média
Viajar	média	médio	média
Viajar	média	fraco	baixa
Viajar	média	fraco	muito baixa
Ficar em Casa	alta	forte	alta
Ficar em Casa	alta	fraco	alta
Ficar em Casa	baixa	médio	alta

Após a separação de classes, o conjunto de treinamento é dividido em duas tabelas, *POS* e *NEG*, contendo os registros positivos e os registros negativos, respectivamente:

Tabela 3.2. *POS* - Registros positivos

<b>Temperatura</b>	<b>Vento</b>	<b>Umidade</b>
alta	médio	baixa
alta	médio	média
média	médio	média
média	fraco	baixa
média	fraco	muito baixa

Tabela 3.3. *NEG* - Registros negativos

<b>Temperatura</b>	<b>Vento</b>	<b>Umidade</b>
alta	forte	alta
alta	fraco	alta
baixa	médio	alta

Na etapa de redução de dados, são geradas as tabelas  $G_{POS}$  e  $G_{NEG}$ .

As tabelas  $G_{POS}$  e  $G_{NEG}$  têm uma coluna adicional  $\alpha + 1$ , e esta coluna adicional é inicializada armazenando o valor 1 (um).

Para melhor exemplificar o mecanismo da rotina de redução de dados, vamos começar pela tabela *POS*. Tomando sua primeira linha, [alta médio baixa], o algoritmo compara esta linha com a linha seguinte, [alta médio média], e armazena em uma variável temporária, *temp*, as variáveis comuns, substituindo as variáveis diferentes por um asterisco. Neste caso,  $temp = [alta \quad médio \quad *]$ .

Em seguida, o algoritmo faz uma contagem do número de variáveis não nulas (que não estejam representadas pelo asterisco) da variável temporária *temp*. No caso, há duas variáveis não nulas armazenadas em *temp*: alta e médio.

Quando o número de variáveis não nulas é igual ou superior a dois, o algoritmo armazena, na linha da tabela que está sendo criada,  $G_{POS}$ , toda a informação contida na variável  $temp$ , adicionando uma unidade na coluna adicional  $\alpha + 1$ .

Quando o número de variáveis não nulas é inferior a dois, o algoritmo passa para a próxima linha da tabela  $G_{POS}$ , armazenando nesta linha, e também na variável temporária  $temp$ , toda a informação contida na linha referente da tabela  $POS$ , e mantém o valor 1 na coluna adicional  $\alpha + 1$ .

Então, sendo  $g_1$  a primeira linha da tabela  $G_{POS}$ ,  $g_2$  a segunda linha, e assim por diante, neste exemplo tem-se  $g_1 = [\text{alta} \quad \text{médio} \quad * \quad 2]$ .

O algoritmo continua a redução de dados, comparando a variável temporária  $temp$  com a terceira linha da tabela  $POS$  como na etapa anterior, substituindo as variáveis diferentes por um asterisco. Da etapa anterior,  $temp = [\text{alta} \quad \text{médio} \quad *]$ , e da tabela  $POS$ , a terceira linha é  $[\text{média} \quad \text{médio} \quad \text{média}]$ , então, a variável temporária  $temp$  passa a armazenar  $[* \quad \text{médio} \quad *]$ . Contando as variáveis não nulas, desta vez só há uma variável (médio). Sendo assim, o algoritmo passa para a segunda linha de  $G_{POS}$ , armazenando nesta linha, e também na variável temporária  $temp$ , toda a informação contida na terceira linha da tabela  $POS$ , e mantém o valor 1 na coluna adicional  $\alpha + 1$ . Então,  $g_2 = [\text{média} \quad \text{médio} \quad \text{média} \quad 1]$  e  $temp = [\text{média} \quad \text{médio} \quad \text{média}]$ .

Agora a variável temporária  $temp$  será comparada com a quarta linha da tabela  $POS$ , substituindo as variáveis diferentes por um asterisco. Da etapa anterior,  $temp = [\text{média} \quad \text{médio} \quad \text{média}]$ , e da tabela  $POS$ , a quarta linha é  $[\text{média} \quad \text{fraco} \quad \text{baixa}]$ , então, a variável temporária  $temp$  passa a armazenar  $[\text{média} \quad * \quad *]$ . Como só há uma variável não nula (média), o algoritmo passa para a terceira linha de  $G_{POS}$ , armazenando nesta linha, e também na variável temporária  $temp$ , toda a informação contida na quarta linha da tabela  $POS$ , e mantém o valor 1 na coluna adicional  $\alpha + 1$ . Então,  $g_3 = [\text{média} \quad \text{fraco} \quad \text{baixa} \quad 1]$  e  $temp = [\text{média} \quad \text{fraco} \quad \text{baixa}]$ .

Finalmente, a variável temporária  $temp$  será comparada com a quinta e última linha da tabela  $POS$ , substituindo as variáveis diferentes por um asterisco. Da etapa anterior,  $temp = [\text{média} \quad \text{fraco} \quad \text{baixa}]$ , e da tabela  $POS$ , a quinta linha é  $[\text{média} \quad \text{fraco} \quad \text{muito baixa}]$ , então, a variável temporária  $temp$  passa a armazenar  $[\text{média} \quad \text{fraco}$

\* ]. Agora há duas variáveis não nulas (média e fraco), então o algoritmo permanece na terceira linha de  $G_{POS}$ , armazenando nesta linha toda a informação contida na variável  $temp$  e adicionando uma unidade na coluna adicional  $\alpha + 1$ . Então,  $g_3 = [média \quad fraco \quad * \quad 2]$ .

A Tabela 3.4 apresenta a tabela  $G_{POS}$  do exemplo proposto.

Tabela 3.4.  $G_{POS}$

alta	médio	*	2
média	médio	média	1
média	fraco	*	2

Pode-se observar que, na coluna adicional  $\alpha + 1$  de  $G_{POS}$ , está inserido o número de registros da tabela  $POS$  que são descritos pela respectiva linha de  $G_{POS}$ , ou seja, se efetuarmos uma soma dos valores contidos na coluna  $\alpha + 1$  de  $G_{POS}$ , deve-se encontrar o número total de registros da tabela  $POS$ . No caso:  $2 + 1 + 2 = 5$ , que é o número de linhas da tabela  $POS$ .

Agora, com este mesmo mecanismo de redução de dados, partindo da tabela  $NEG$ , o algoritmo gera a tabela  $G_{NEG}$ . Tomando a primeira linha de  $NEG$ , [alta forte alta], o algoritmo compara esta linha com a linha seguinte, [alta fraco alta], e armazena na variável temporária  $temp$ , as variáveis comuns, substituindo as variáveis diferentes por um asterisco. Neste caso,  $temp = [alta \quad * \quad alta]$ .

No caso, há duas variáveis não nulas armazenadas em  $temp$ , visto que a variação “alta” aparece duas vezes. Então, o algoritmo armazena, na primeira linha de  $G_{NEG}$ , toda a informação contida na variável  $temp$ , adicionando uma unidade na coluna adicional  $\alpha + 1$ . Sendo  $g_1$  a primeira linha da tabela  $G_{NEG}$ ,  $g_2$  a segunda linha, e assim por diante, neste exemplo tem-se  $g_1 = [alta \quad * \quad alta \quad 2]$ .

O algoritmo continua a redução de dados, comparando a variável temporária  $temp$  com a terceira e última linha da tabela  $NEG$ , substituindo as variáveis diferentes por um asterisco. Da etapa anterior,  $temp = [alta \quad * \quad alta]$ , e da tabela  $NEG$ , a terceira linha é [baixa médio alta], então, a variável temporária  $temp$  passa a armazenar [ $* \quad * \quad alta$ ]. Contando as variáveis não nulas, desta vez só há uma variável (alta). Sendo

assim, o algoritmo passa para a segunda linha de  $G_{NEG}$ , armazenando nesta linha, e também na variável temporária  $temp$ , toda a informação contida na terceira linha da tabela  $NEG$ , e mantém o valor 1 na coluna adicional  $\alpha + 1$ . Então,  $g_2 = [baixa \quad médio \quad alta \quad 1]$ .

A Tabela 3.5 apresenta a tabela  $G_{NEG}$  do exemplo proposto.

Tabela 3.5.  $G_{NEG}$

alta	*	alta	2
baixa	médio	alta	1

Como no caso de  $G_{POS}$ , na coluna adicional  $\alpha + 1$  de  $G_{NEG}$ , está inserido o número de registros da tabela  $NEG$  que são descritos pela respectiva linha de  $G_{NEG}$ , ou seja, somando os valores contidos na coluna  $\alpha + 1$  de  $G_{NEG}$ , tem-se:  $2 + 1 = 3$ , que é o número de linhas da tabela  $NEG$ .

O próximo passo é a geração de regras. Partindo da tabela  $G_{POS}$ , o algoritmo efetua uma análise detalhada de cada atributo. O primeiro atributo é “temperatura”, que pode ser “baixa”, “média” ou “alta”, assumindo 3 posições diferentes. Um valor é atribuído para cada variação do atributo “temperatura”, utilizando para isso os valores contidos na coluna  $\alpha + 1$  de  $G_{POS}$ . A variação “baixa” não aparece em  $G_{POS}$ , logo terá valor nulo, a variação “alta” aparece na primeira linha e tem valor 2 (valor contido na coluna  $\alpha + 1$  de  $G_{POS}$ ) e a variação “média” aparece na segunda e também na terceira linha e tem valor 3 (soma dos valores contidos na coluna  $\alpha + 1$  de  $G_{POS}$ ,  $1 + 2 = 3$ ). Estes valores são então multiplicados pelo número de posições diferentes que o atributo “temperatura” pode assumir, que é 3.

Então, para o atributo “temperatura”:

$$\text{“baixa”} = 0 \times 3 = 0$$

$$\text{“média”} = 3 \times 3 = 9$$

$$\text{“alta”} = 2 \times 3 = 6$$

O segundo atributo é “vento”, que pode ser “fraco”, “médio” ou “forte”, assumindo 3 posições diferentes. A variação “médio” aparece na primeira e também na segunda linha e tem valor 3 (soma dos valores contidos na coluna  $\alpha + 1$  de  $G_{POS}$ ,  $2 + 1$

= 3), a variação “fraco” aparece na terceira linha e tem valor 2 (valor contido na coluna  $\alpha + 1$  de  $G_{POS}$ ) e a “forte” não aparece em  $G_{POS}$ , tendo valor nulo. Estes valores são então multiplicados pelo número de posições diferentes que o atributo “vento” pode assumir, que é 3.

Então, para o atributo “vento”:

$$\text{“fraco”} = 2 \times 3 = 6$$

$$\text{“médio”} = 3 \times 3 = 9$$

$$\text{“forte”} = 0 \times 3 = 0$$

O terceiro atributo é “umidade”, que pode ser “muito baixa”, “baixa”, “média” ou “alta”, assumindo 4 posições diferentes. Destas, a única variação presente em  $G_{POS}$  é a variação “média”, que aparece na segunda linha e tem valor 1 (valor contido na coluna  $\alpha + 1$  de  $G_{POS}$ ). Este valor é multiplicado pelo número de posições diferentes que o atributo “umidade” pode assumir, que é 4.

Então, para o atributo “umidade”:

$$\text{“muito baixa”} = 0 \times 4 = 0$$

$$\text{“baixa”} = 0 \times 4 = 0$$

$$\text{“média”} = 1 \times 4 = 4$$

$$\text{“alta”} = 0 \times 4 = 0$$

Toma-se então o maior dos valores calculados, dentre todas as variações de todos os atributos, no caso de empate a preferência é do atributo que aparece primeiro na coluna  $G_{POS}$ . O maior valor é 9, para o atributo “temperatura”; “média” =  $3 \times 3 = 9$ . Verifica-se então se esta variação, “temperatura = média”, descreve alguma das linhas da tabela  $G_{NEG}$ . Observando-se a primeira coluna da tabela  $G_{NEG}$ , referente ao atributo “temperatura”, não se encontra a variação “temperatura = média”. Então esta passa a ser a primeira regra gerada: SE “temperatura = média” ENTÃO “decisão = viajar”.

Todas as linhas de  $G_{POS}$  que contêm a variação “temperatura = média” são eliminadas, restando uma única linha [alta médio \* 2]. Novamente, partindo de  $G_{POS}$ , agora contendo somente uma linha, o algoritmo efetua uma análise detalhada de cada atributo.

Então, para o atributo “temperatura”:

$$\text{“baixa”} = 0 \times 3 = 0$$

$$\text{“média”} = 0 \times 3 = 0$$

$$\text{“alta”} = 2 \times 3 = 6$$

Para o atributo “vento”:

$$\text{“fraco”} = 0 \times 3 = 0$$

$$\text{“médio”} = 2 \times 3 = 6$$

$$\text{“forte”} = 0 \times 3 = 0$$

E para o atributo “umidade”:

$$\text{“muito baixa”} = 0 \times 4 = 0$$

$$\text{“baixa”} = 0 \times 4 = 0$$

$$\text{“média”} = 0 \times 4 = 0$$

$$\text{“alta”} = 0 \times 0 = 0$$

Tomando-se agora o maior dos valores calculados, o maior valor é 6, para o atributo “temperatura”; “alta” =  $2 \times 3 = 6$ . Verifica-se então se esta variação, “temperatura = alta”, descreve alguma das linhas da tabela  $G_{NEG}$ . Observando-se a primeira coluna da tabela  $G_{NEG}$ , referente ao atributo “temperatura”, a variação “temperatura = alta” ocorre na primeira linha. Sendo assim, o algoritmo busca, dentre os valores calculados (retirando-se a variação “temperatura = alta”) o valor máximo. Este valor novamente é 6, para o atributo “vento”; “médio” =  $2 \times 3 = 6$ . Verifica-se então se a variação, “temperatura = alta E vento = médio”, descreve alguma das linhas da tabela  $G_{NEG}$ . Observando-se as duas primeiras colunas da tabela  $G_{NEG}$ , referentes aos atributos “temperatura” e “vento”, a variação “temperatura = alta E vento = médio” não ocorre em  $G_{NEG}$ . Então esta passa a ser a segunda regra gerada: SE “temperatura = alta E vento = médio” ENTÃO “decisão = viajar”.

Todas as linhas de  $G_{POS}$  que contêm a variação “temperatura = alta E vento = médio” são eliminadas, eliminando a linha restante de  $G_{POS}$  e encerrando o processo de geração de regras.

Caso o número de classes  $n_c$  seja superior a dois, deve-se executar o algoritmo em  $n_c$  etapas, na primeira etapa considerando a primeira classe positiva e as demais negativas, na segunda etapa considerando a segunda classe positiva e as demais negativas, e assim por diante.

O pseudocódigo do algoritmo *DataSqueezer* é apresentado a seguir:

```

Dados:      POS, NEG, Ntot,  $\alpha$            // Ntot registros e  $\alpha$  atributos
// POS é a tabela de registros positivos e NEG é a tabela de registros negativos
-----Etapa 1-----
1.1         Gpos = ReduçãoDados(POS, $\alpha$ );
1.2         Gneg = ReduçãoDados(NEG, $\alpha$ );
-----Etapa 2-----
2.1         Inicializar REGRAS=[]; i=1;
// REGRAS é a matriz que armazena as regras, regrasi é a i-ésima regra gerada
2.2         LISTA = lista de todas as colunas de Gpos;
2.3         Para cada coluna de Gpos contida em LISTA, para cada
            valor não nulo a da coluna j, calcular a soma Saj dos
            valores da coluna [ $\alpha$ +1] de Gposi para cada linha i
            que contenha o valor a; multiplicar Saj pelo número
            de variações do atributo j;
2.4         Selecionar máximo Saj; remover j de LISTA; adicionar
            "j=a" a regrasi;
2.5         SE regrasi não descreve nenhuma linha de Gneg
            ENTÃO remover todas as linhas de Gpos descritas
            por regrasi; i=i+1;
            SE Gpos  $\neq \emptyset$  retornar para 2.2; SENÃO FIM;
2.6         SENÃO retornar para 2.3;
-----Rotina de Redução de Dados-----
ReduçãoDados(D, $\alpha$ );
3.1         Inicializar G=[]; i=1; temp=d1; g1=d1; g1[ $\alpha$ +1]=1;
            // di[j] refere-se a i-ésima linha e a j-ésima coluna de D
            // gi[j] refere-se a i-ésima linha e a j-ésima coluna de G
3.2         PARA j = 1 até ND           // ND é o número de registros de D
            PARA k = 1 até  $\alpha$ 
                SE ((dj+1[k]  $\neq$  temp[k]) OU (dj+1[k] =  $\emptyset$ ))
                    ENTÃO temp[k] = '*';
                SE (número de valores não nulos de temp  $\geq$  2)
                    ENTÃO gi=temp; gi[ $\alpha$ +1]=gi[ $\alpha$ +1]+1;
                    SENÃO i=i+1; gi=dj+1; gi[ $\alpha$ +1]=1; temp=dj+1;
3.3         RETORNAR G;                 // G é a matriz Gpos ou Gneg

```

Este algoritmo de regras de decisão apresenta uma série de vantagens, tais como: o conhecimento gerado é de fácil interpretação, tem bom desempenho computacional e alto poder de predição, é eficiente e robusto, sendo um modelo eficiente para o caso de bases com dados faltando ou com ruídos, que acontecem em boa parte dos problemas de classificação.

Porém, há uma desvantagem no uso do *DataSqueezer*; que consiste no fato de que ele trabalha com dados discretos, e como atributos contínuos ocorrem com grande frequência, este é um fator limitante na utilização deste algoritmo.

### 3.3. Algoritmo Proposto

O algoritmo apresentado é uma versão do algoritmo *DataSqueezer*, sendo adicionada uma etapa inicial de *fuzzificação* do conjunto de registros.

Como foi dito anteriormente, um fator limitante no uso do *DataSqueezer* é o fato de que ele trabalha com dados discretos; e é justamente neste limite que a lógica *fuzzy* tende a contribuir, de maneira a não mais restringir o uso deste algoritmo, mas sim fazer com que o mesmo seja aplicado a qualquer base de dados, ao mesmo tempo em que torna o algoritmo menos sensível aos limites da discretização.

Seja  $D$  o conjunto de dados de treinamento, que contém  $N_{TOT}$  registros e  $\alpha$  atributos, que agora podem ser contínuos ou discretos. Seja  $D_p$  o conjunto de registros positivos e  $D_N$  o conjunto de registros negativos.  $D_p$  e  $D_N$  devem satisfazer as seguintes propriedades:

$$D_p \cup D_N = D$$

$$D_p \cap D_N = \emptyset$$

$$D_p \neq \emptyset$$

$$D_N \neq \emptyset$$

Após a separação de classes, são geradas duas tabelas, *POS* e *NEG*, contendo  $N_{POS}$  registros positivos e  $N_{NEG}$  registros negativos, respectivamente; onde  $N_{POS} + N_{NEG} = N_{TOT}$ .

A etapa seguinte é a *fuzzificação* dos dados, que é responsável pelo mapeamento das entradas numéricas em conjuntos *fuzzy*, variáveis linguísticas. Nesta etapa é

definido o número de conjuntos *fuzzy* para cada atributo, que é um parâmetro escolhido pelo usuário; a representação da informação pode ser feita em diferentes níveis de generalização; quanto maior for o número de conjuntos *fuzzy*, maior será a precisão obtida.

Em seguida é efetuada uma discretização (partição) *fuzzy* sobre o universo de cada atributo e as funções de pertinência e a matriz de pesos são calculadas.

Suponha, por exemplo, um conjunto de treinamento cujas variáveis são medidas de condições de tempo e a classe é a decisão sobre viajar ou não viajar.

A Tabela 3.6 apresenta o conjunto de treinamento do exemplo proposto.

Tabela 3.6. Exemplo de um problema de classificação

<b>Decisão</b>	<b>Temperatura</b>	<b>Vento</b>	<b>Umidade</b>
Viajar	35	médio	70
Viajar	32.5	médio	76
Viajar	25	médio	75.5
Viajar	28	fraco	68
Viajar	22	fraco	61
Ficar em Casa	34	forte	80
Ficar em Casa	37	fraco	82
Ficar em Casa	16	médio	85

Observando o conjunto de registros do problema, tem-se duas variáveis contínuas, temperatura e umidade. Dividindo-se a variável “temperatura” em três conjuntos *fuzzy*, “baixa”, “média” e “alta”; e a variável “umidade” em quatro conjuntos *fuzzy*, “muito baixa”, “baixa”, “média” e “alta”, este problema se torna o mesmo do exemplo anterior, desta vez sem limitações pelo simples fato de haver variáveis contínuas no problema.

O algoritmo então executa uma redução de dados a fim de generalizar a informação contida nos dados das tabelas *POS* e *NEG*, sendo geradas nesta etapa as

tabelas  $G_{POS}$  e  $G_{NEG}$ . Todo o mecanismo seguinte é análogo ao do algoritmo descrito na seção anterior.

O pseudocódigo do algoritmo *FuzzyDataSqueezer* é apresentado a seguir:

```
Dados:      POS, NEG, Ntot,  $\alpha$  , nfi      // Ntot registros e  $\alpha$  atributos
// POS é a tabela de registros positivos e NEG é a tabela de registros negativos
// nfi é o número de conjuntos fuzzy para a variável xi escolhido pelo usuário

-----Etapa 1-----
1.1      Particionamento fuzzy
1.2      Cálculo das funções de pertinência
1.3      Geração das combinações possíveis
1.4      Cálculo da matriz de pesos
1.5      Atribuição de cada registro ao conjunto fuzzy de
          maior pertinência, para cada atributo.

-----as etapas seguintes são análogas ao algoritmo anterior-----
```

### 3.4. Conclusão

Este capítulo discutiu o uso de algoritmos de regras de decisão. Diante dos algoritmos apresentados, vale ressaltar a contribuição oferecida pela aplicação da lógica *fuzzy* no algoritmo original *DataSqueezer*, eliminando assim o limite que havia na utilização deste algoritmo no caso de bases de dados contendo atributos contínuos, ao mesmo tempo em que executa o tratamento das incertezas contidas nos dados e faz com que o algoritmo seja menos sensível aos limites da discretização.

## 4. Ensemble

---

### 4.1. Introdução

Uma técnica que vem se destacando na produção de sistemas classificadores e portanto representa uma área ativa em aprendizado de máquina consiste na combinação de classificadores, também chamada *ensemble*.

Um *ensemble* é um conjunto de classificadores cujas decisões individuais são combinadas de alguma forma para classificar um conjunto de dados cuja classe seja desconhecida.

Esta combinação de classificadores pode ser mais precisa do que os classificadores individuais que a compõem. Segundo HANSEN & SALAMON (1990), uma condição necessária para que um conjunto de classificadores seja mais preciso do que seus componentes é que os classificadores que compõem este conjunto sejam distintos. Dois classificadores são distintos quando cometem erros diferentes em novos conjuntos de registros.

A Figura 4.1 ilustra um *ensemble* formado por três classificadores  $h_1$ ,  $h_2$  e  $h_3$ ; e um novo registro  $x$ . Este novo registro  $x$  será classificado por cada classificador componente do *ensemble*. Seja  $h_1(x)$  a classificação dada a este novo registro  $x$  pelo classificador  $h_1$ ,  $h_2(x)$  a classificação dada pelo classificador  $h_2$  e  $h_3(x)$  a classificação dada pelo classificador  $h_3$ . Caso os três classificadores sejam idênticos, quando  $h_1(x)$  estiver errada, logo  $h_2(x)$  e  $h_3(x)$  também estarão. Caso os três classificadores sejam distintos, ou seja, se os erros cometidos pelos classificadores não forem correlacionados, quando  $h_1(x)$  estiver errada,  $h_2(x)$  e  $h_3(x)$  podem estar corretas, fazendo com que o voto majoritário possa classificar corretamente o novo registro  $x$ .

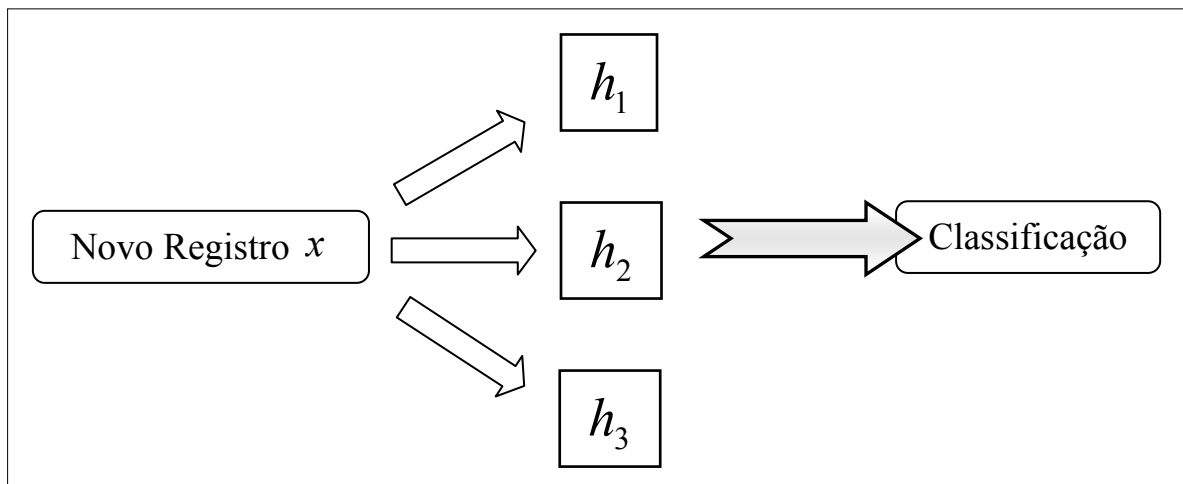


Figura 4.1. *Ensemble* formado por três classificadores (DIETTERICH, 2000)

## 4.2. Vantagens

O objetivo da construção de um conjunto de classificadores, ou seja, de um *ensemble*, é alcançar um algoritmo de classificação com maior poder de predição do que os algoritmos de classificação que o compõem (BERNARDINI, 2006).

Há três motivos principais que tornam vantajosa a construção de um *ensemble* (DIETTERICH, 2000):

1. Estatístico: o problema estatístico surge quando a quantidade de registros de treinamento é muito pequena comparada ao tamanho do espaço de hipóteses. Assim, o algoritmo de classificação pode encontrar muitas hipóteses diferentes, com a mesma precisão sobre os registros de treinamento. Com a construção de um *ensemble*, pode-se calcular a média da precisão das hipóteses, encontrando uma boa aproximação da hipótese verdadeira e reduzindo o erro.

2. Computacional: encontrar a melhor hipótese pode muitas vezes ser uma tarefa difícil para o algoritmo de classificação; através da construção de um *ensemble* é possível obter uma melhor aproximação da hipótese verdadeira, que seja mais precisa do que qualquer um dos classificadores individuais.

3. Representacional: com a união das hipóteses, ou simplesmente aplicando-se pesos a cada uma delas e unindo-as posteriormente, pode-se expandir o espaço das funções representáveis; este é um artifício válido para situações onde, por exemplo, a hipótese verdadeira não se encontra representada pelos classificadores individuais.

### 4.3. *Bagging*

Dentre as diversas técnicas de construção de *ensembles* que têm sido desenvolvidas (DIETTERICH, 2000), destaca-se a técnica *Bagging*, cujo nome é a abreviação de *bootstrap aggregating*.

A técnica *Bagging* apresenta três etapas; a primeira etapa é a construção de  $L$  conjuntos de registros através de replicações *bootstrap* de um conjunto de registros inicial (BREIMAN, 1996).

A partir de um conjunto de registros de treinamento constituído por  $N_{TOT}$  registros, uma seqüência de  $L$  conjuntos de registros é construída, onde cada conjunto deve conter  $N_{TOT}$  registros. Os elementos de cada um destes  $L$  conjuntos são obtidos randomicamente com reposição do conjunto original de registros, podendo assim haver registros repetidos.

Cada um dos  $L$  conjuntos de registros é chamado de replicação *bootstrap* do conjunto de registros original (EFRON & TIBSHIRANI, 1993).

A segunda etapa é a construção de  $L$  classificadores a partir de cada um dos  $L$  conjuntos de registros da seqüência construída na primeira etapa, fazendo uso destes conjuntos para criar  $L$  hipóteses.

A terceira e última etapa da técnica *Bagging* é a classificação. Seja  $h_l(x)$  a hipótese correspondente ao classificador  $h_l$ , onde  $l = 1, \dots, L$ , a classificação de um novo registro  $x$  é dada pela média aritmética das  $L$  hipóteses:

$$h^*(x) = \frac{1}{L} \sum_{l=1}^L h_l(x) \quad (4.1)$$

A equação acima é válida para hipóteses numéricas (contínuas). Para hipóteses discretas, a classificação de um novo registro  $x$  é dada através de votação por maioria entre as  $L$  hipóteses:

$$h^*(x) = \arg \max \sum_{l=1}^L h_l(x) \quad (4.2)$$

A classificação de um novo registro  $x$  será mais precisa à medida em que maior for a divergência entre as hipóteses geradas.

#### 4.4. *Boosting*

Uma outra técnica de construção de *ensembles* que vale ser ressaltada é a técnica *Boosting*, que consiste em gerar hipóteses precisas a partir da combinação de hipóteses menos precisas.

A técnica *Boosting* apresenta três etapas; a primeira etapa é atribuição de um peso  $w_i$  para cada registro (FREUND & SCHAPIRE, 1997).

A partir de um conjunto de registros de treinamento constituído por  $N_{TOT}$  registros, um vetor  $W$  com  $N_{TOT}$  elementos é criado. Este vetor tem a função de armazenar um peso  $w_i$  para cada registro, e é inicializado com todos os registros contendo o mesmo peso  $w_i = 1/N_{TOT}$ , para  $i = 1, \dots, N_{TOT}$ .

A etapa seguinte é a construção de uma hipótese  $h_l$ , onde  $l = 1, \dots, L$  e o cálculo do erro  $\alpha_l$ , que é dado pelo somatório dos pesos dos registros cuja classificação dada por  $h_l$  seja incorreta.

O processo é iterativo, e termina caso  $\alpha_l = 0$  ou  $\alpha_l \geq \frac{1}{2}$ . Para  $0 < \alpha_l < \frac{1}{2}$ , os pesos são recalculados, se  $h_l$  acerta na classificação do registro  $i$ , da seguinte forma:

$$w_{(l+1)} = w_l \frac{\alpha_l}{1 - \alpha_l} \quad (4.5)$$

Assim, conforme o número de iterações aumenta, são reduzidos os pesos dos registros que forem classificados corretamente.

A partir do conjunto de hipóteses  $h_l$ ,  $l = 1, \dots, L$ , é gerada a hipótese final  $h^*$ . Para cada registro  $x$ , é realizada uma verificação das hipóteses de classificação dadas por  $h_l$ ,  $l = 1, \dots, L$ . Cada classe  $C_i$ ,  $i = 1, \dots, n_C$ , onde  $n_C$  é o número de classes, terá um peso de votação, que inicialmente é zerado. Se  $h_l$  classifica  $x$  como sendo pertencente

à classe  $C_i$ , acrescenta-se o valor de  $\log((1-\alpha_i)/\alpha_i)$  ao peso de votação da classe  $C_i$ . A hipótese final  $h^*$  atribui ao registro  $x$  a classe que tiver maior peso na votação.

A classificação de um novo registro  $x$  será mais precisa à medida em que maior for o número de hipóteses geradas.

## 4.5. Outros métodos de construção de *ensembles*

### 4.5.1 Votação com peso

Neste método, o voto de cada classificador é ponderado com algum peso associado a ele. Este peso pode ser calculado por diversas técnicas, uma delas pode ser, por exemplo, utilizando a estimativa da média da taxa de erro associada às hipóteses provenientes dos  $L$  classificadores (FREUND & SCHAPIRE, 1997).

### 4.5.2 Votação sem peso

O método de construção de *ensembles* utilizado neste trabalho será o método de Votação sem peso (*Unweighted Voting – UV*), no qual cada classificador membro do *ensemble* classifica o registro dado, e a classificação do *ensemble* para o exemplo  $x$  é dada pela classe que mais recebeu votos dos  $L$  classificadores.

## 4.6. Conclusão

Neste capítulo, foram apresentados o conceito de *ensemble* de classificadores, alguns métodos de construção de *ensembles* e ainda uma série de motivos pelos quais os *ensembles* funcionam melhor do que um único classificador, o que há de ser comprovado nos capítulos seguintes.

Em relação aos métodos de construção de *ensembles* Votação sem peso e Votação com peso, pode-se imaginar que em ambas situações, com e sem peso, o resultado da classificação é o mesmo. Porém, vale citar que isso não é necessariamente verdade, dependerá da técnica de ponderação utilizada no método de Votação com peso.

## 5. Resultados Obtidos

---

### 5.1. Problemas Estudados

Para avaliar o comportamento dos algoritmos de classificação *fuzzy*, foram estudados quatro problemas de classificação.

Um deles, de identificação de tipos de vinhos, teve sua base de dados obtida no repositório de dados da Universidade da Califórnia (BLAKE & MERZ, 1998).

A base de dados *well* foi utilizada com a finalidade de avaliar a acurácia dos algoritmos na classificação de dados da indústria do petróleo, que é o objetivo do presente trabalho. Trata-se de um problema de classificação de tipos de litologia, no qual os atributos são correspondentes a dados de perfilagem de poços; no caso perfis sônicos.

Os dados dos dois outros problemas são correspondentes a uma função espiral, a fim de avaliar a acurácia dos algoritmos para o caso de classes não-convexas.

A metodologia de teste aplicada foi a divisão randômica de cada base de dados em dois conjuntos, tomando 70% dos registros para o conjunto de treinamento e 30% dos registros para o conjunto de teste. Cada algoritmo foi testado em cinco ciclos, tomando-se a média aritmética dos cinco resultados obtidos.

A Tabela 5.1 abaixo contém o nome do algoritmo *fuzzy* e sua abreviação para futuras referências. A Tabela 5.2 informa o nome do problema, sua abreviação para futuras referências e sua descrição, e a Tabela 5.3 exibe as características estruturais das bases de dados desses problemas.

Tabela 5.1. Algoritmos de classificação *fuzzy*

Abreviação	Algoritmo
FPM	<i>Fuzzy Pattern Matching</i>
FPME	<i>Fuzzy Pattern Matching</i> com Função Exponencial
FPMM	<i>Fuzzy Pattern Matching</i> Multidensidade
REGRAS	Algoritmo de Regras de Decisão <i>Fuzzy</i> ( <i>FuzzyDataSqueezer</i> )

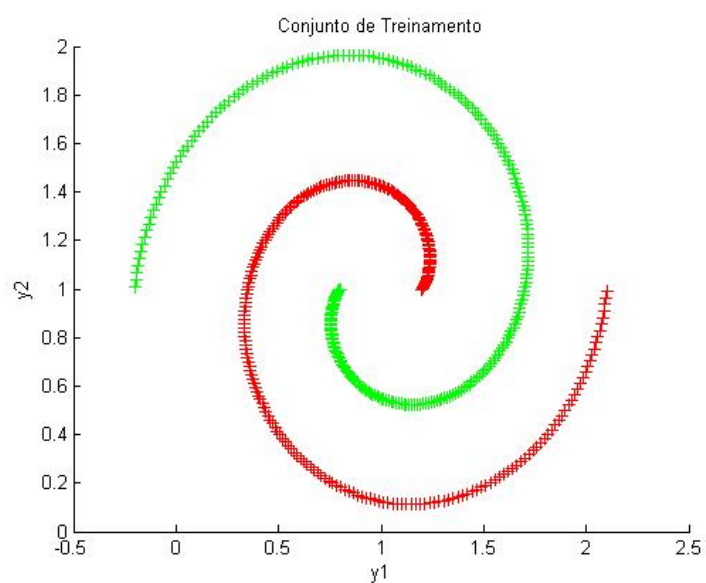
Tabela 5.2. Problemas estudados

Problema	Abreviação	Descrição
espiral 2	<i>spir2</i>	função espiral de comprimento $2\pi$
espiral 3	<i>spir3</i>	função espiral de comprimento $3\pi$
wine	<i>wine</i>	identificação de tipos de vinhos oriundos de uma mesma região na Itália
well	<i>well</i>	classificação de tipos litológicos

Tabela 5.3. Propriedades das bases de dados

Problema	Registros	Atributos	Classes	Distribuição de classes		
<i>spir2</i>	630	2	2	1: 50.0%	2: 50.0%	
<i>spir3</i>	944	2	2	1: 50.0%	2: 50.0%	
<i>wine</i>	178	13	3	1: 33.1%	2: 39.9%	3: 27.0%
<i>well</i>	1506	4	3	1: 44.7%	2: 33.8%	3: 21.5%

As Figuras a seguir ilustram os conjuntos de treinamento dos problemas estudados. Para o caso das bases de dados *wine* e *well*, como o número de atributos é superior a 2, o gráfico é formado pelas duas componentes principais.

Figura 5.1. Conjunto de treinamento – base *spir2*

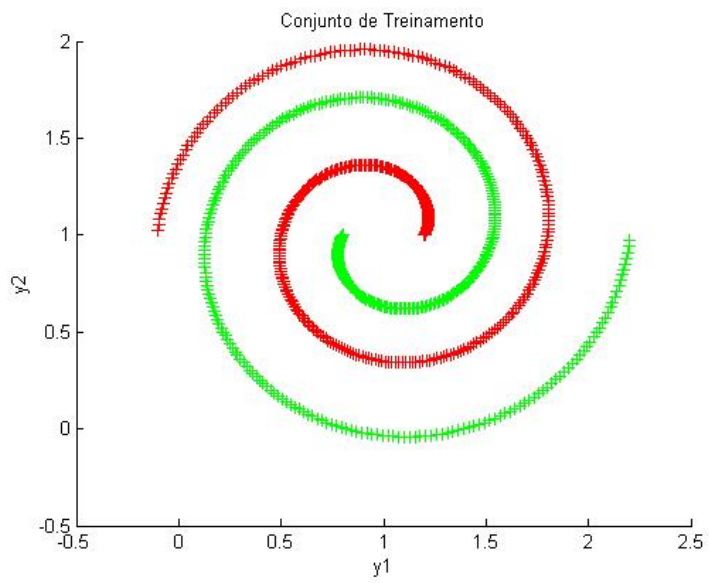


Figura 5.2. Conjunto de treinamento – base *spir3*

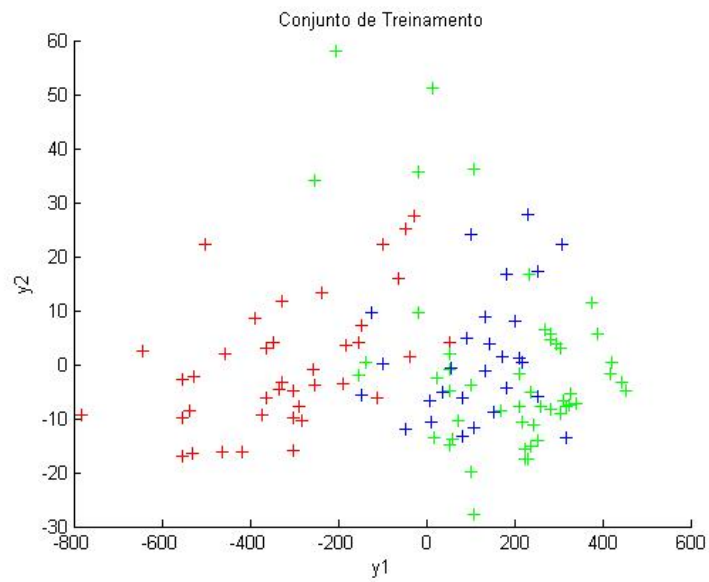


Figura 5.3. Conjunto de treinamento – base *wine*

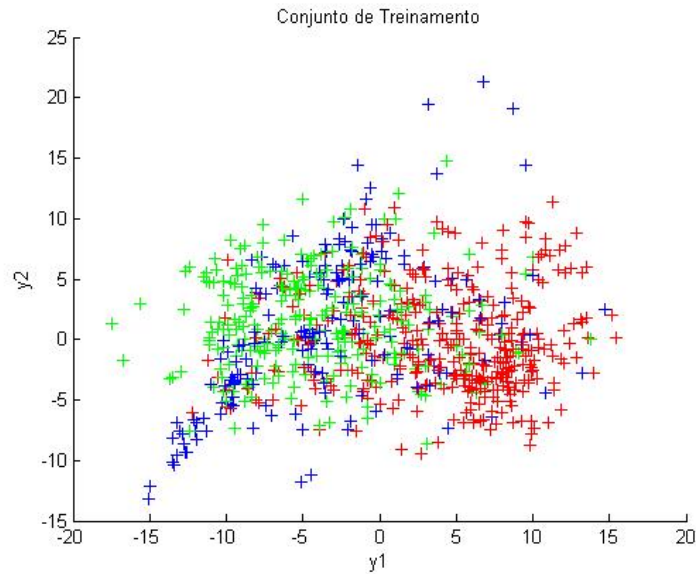


Figura 5.4. Conjunto de treinamento – base *well*

## 5.2. Avaliação dos Resultados

Os algoritmos de classificação podem ser avaliados e comparados utilizando-se diversos critérios (HAN & KAMBER, 2001), já citados anteriormente: acurácia, desempenho, robustez, escalabilidade e interpretabilidade.

As medidas aqui utilizadas são baseadas nas **matrizes de confusão**. A matriz de confusão é uma medida efetiva de avaliação de classificadores, pois fornece os tipos de acertos e erros cometidos durante a classificação, ou seja, o número de classificações corretas contra o número de classificações preditas.

As duas classes de um problema são identificadas como positiva e negativa e a matriz é formada por quatro valores definidos em termos de classes reais e preditas:

-> VP: é a quantidade de elementos da classe positiva que foram preditos como positivos; os elementos são chamados de verdadeiros positivos;

-> FP: é a quantidade de elementos da classe negativa que foram preditos como positivos; os elementos são chamados de falsos positivos;

-> FN: é a quantidade de elementos da classe positiva que foram preditos como negativos; os elementos são chamados de falsos negativos;

-> VN: é a quantidade de elementos da classe negativa que foram preditos como negativos; os elementos são chamados de verdadeiros negativos;

Na matriz de confusão, os valores da diagonal principal representam os acertos e os demais valores os erros. A Figura 5.5 apresenta a estrutura de uma matriz de confusão para um problema com duas classes.

Classes		Predita	
		Classe 1	Classe 2
Real	Classe 1	VP	FN
	Classe 2	FP	VN

Figura 5.5. Estrutura de uma matriz de confusão

Diversas medidas de desempenho podem ser derivadas dos valores da matriz de confusão. Para a presente análise, foram selecionadas as seguintes medidas, e seus resultados estão localizados no Anexo deste trabalho:

1. **sensitividade**: avalia o quanto um classificador pode reconhecer os exemplos positivos e é definida por:

$$sens = \frac{VP}{VP + FN} \quad (5.1)$$

2. **especificidade**: avalia o quanto um classificador pode reconhecer os exemplos negativos e é definida por:

$$espec = \frac{VN}{FP + VN} \quad (5.2)$$

3. **precisão**: é a proporção de elementos classificados como positivos que de fato o são:

$$prec = \frac{VP}{VP + FP} \quad (5.3)$$

4. **medida F**: é a média harmônica da sensibilidade e da precisão:

$$Fmed = \frac{(\beta^2 + 1) \times sens \times prec}{sens + \beta \times prec}, \quad \beta \geq 0 \quad (5.4)$$

Neste estudo,  $\beta = 1$ , isto é, a sensibilidade e a precisão têm a mesma importância. Portanto:

$$Fmed = \frac{2 \times sens \times prec}{sens + prec} \quad (5.5)$$

5. **média GSP**: é a média geométrica entre a sensibilidade e a precisão:

$$GSP = \sqrt{sens \times prec} \quad (5.6)$$

6. **média GSE**: é a média geométrica entre a sensibilidade e a especificidade:

$$GSE = \sqrt{sens \times espec} \quad (5.7)$$

7. **curvas ROC (Receiver Operating Characteristic)**: é um gráfico bidimensional em que o eixo horizontal representa os valores da taxa de falsos positivos ( $1 - Espec$ ) e o eixo vertical os valores de sensibilidade.

No espaço ROC, o ponto (0,0) representa o classificador que nunca acerta a classe positiva, o ponto (1,0) o que erra todos os positivos e negativos, o ponto (1,1) o que sempre classifica como positivo e o ponto (0,1) o classificador perfeito, que acerta todas as classificações. Assim, quanto mais próximo do canto superior esquerdo está um classificador, melhor ele será, pois ele estará acertando mais a classe positiva e errando menos a classe negativa.

Pontos na diagonal representam classificadores com comportamento aleatório. Quanto mais acima está o classificador, mais frequentemente ele decide aleatoriamente pela classe positiva. Como na diagonal as decisões são aleatórias, os classificadores abaixo dela têm desempenhos inferiores aos da adivinhação e não são interessantes. Nestes casos, se suas decisões forem negadas, classificadores localizados no triângulo superior serão produzidos. Quando as decisões de um classificador são negadas, as linhas de sua matriz de confusão são trocadas. A curva ROC é formada ligando-se o ponto do classificador às extremidades da diagonal, ou seja, aos pontos (0,0) e (1,1).

Uma estratégia para abordar esta questão é calcular a área sob a curva ROC identificada pela sigla **AUC** (*Area Under ROC Curve*). Assim, o classificador que tiver maior área é considerado o melhor.

É relevante citar que a área do espaço ROC é unitária e que todos os classificadores melhores que as decisões aleatórias terão áreas maiores que 0.5, já que este valor é área do triângulo inferior à diagonal (LING *et al.*, 2003).

Todos os conceitos aqui apresentados foram definidos para problemas com apenas duas classes. Quando o número de classes é superior a dois, a análise se torna mais complexa e não há consenso entre os pesquisadores sobre qual a melhor forma de agir.

Aqui será feita uma análise ponderada; para tal, cada classe será considerada a classe positiva e todas as demais como a classe negativa. Assim, as medidas de desempenho são calculadas e é realizada a soma ponderada das mesmas, em que os pesos são as freqüências relativas das classes nos dados (HAND & TILL, 2001).

### 5.3. Resultados Obtidos

#### 5.3.1 Algoritmos de Classificação *Fuzzy*

Seguem abaixo os resultados da análise, onde o número entre parênteses indica o número de subclasses em que foi dividida cada classe do problema.

Tabela 5.4. Percentual de Classificações Corretas – Conjunto de Treinamento

Treinamento	Percentual de Classificações Corretas					
	FPM	FPME (3)	FPME (5)	FPMM (3)	FPMM (5)	REGRAS
<i>spir2</i>	55.95	51.79	54.37	71.83	60.52	95.04
<i>spir3</i>	62.20	57.29	54.91	58.75	60.61	82.76
<i>wine</i>	67.80	66.95	67.80	77.12	.....	67.34
<i>well</i>	54.38	60.47	61.99	62.33	64.81	66.79

Tabela 5.5. Percentual de Classificações Corretas – Conjunto de Teste

Treinamento	Percentual de Classificações Corretas					
	FPM	FPME (3)	FPME (5)	FPMM (3)	FPMM (5)	REGRAS
<i>spir2</i>	54.76	50.79	50.79	71.43	59.52	96.03
<i>spir3</i>	63.68	57.37	54.74	60.00	60.53	82.63
<i>wine</i>	65.00	66.67	65.00	76.67	.....	67.22
<i>well</i>	53.66	60.45	60.82	62.04	64.25	66.37

A Tabela 5.6 apresenta uma comparação com os resultados do algoritmo proposto neste trabalho, *FuzzyDataSqueezer* (REGRAS), com o algoritmo original, não *fuzzificado*. Em todas as bases de dados estudadas há uma melhora no percentual de classificações corretas, sendo este aumento um pouco mais significativo no caso das bases *well* e *spir3*.

Tabela 5.6. Percentual de Classificações Corretas – Conjunto de Teste

Treinamento	Percentual de Classificações Corretas	
	<i>DataSqueezer</i>	<i>FuzzyDataSqueezer</i>
<i>spir2</i>	95.17	96.03
<i>spir3</i>	71.32	82.63
<i>wine</i>	65.14	67.22
<i>well</i>	60.53	66.37

Ao analisar a curva ROC da primeira base, *spir2*, pode-se observar que o algoritmo de regras de decisão *fuzzy* apresenta o melhor desempenho na classificação dos dados, o que também pode ser verificado nos valores de medida F obtidos. O algoritmo FPMM com 3 subclasses também gerou um resultado razoável na classificação dos dados.

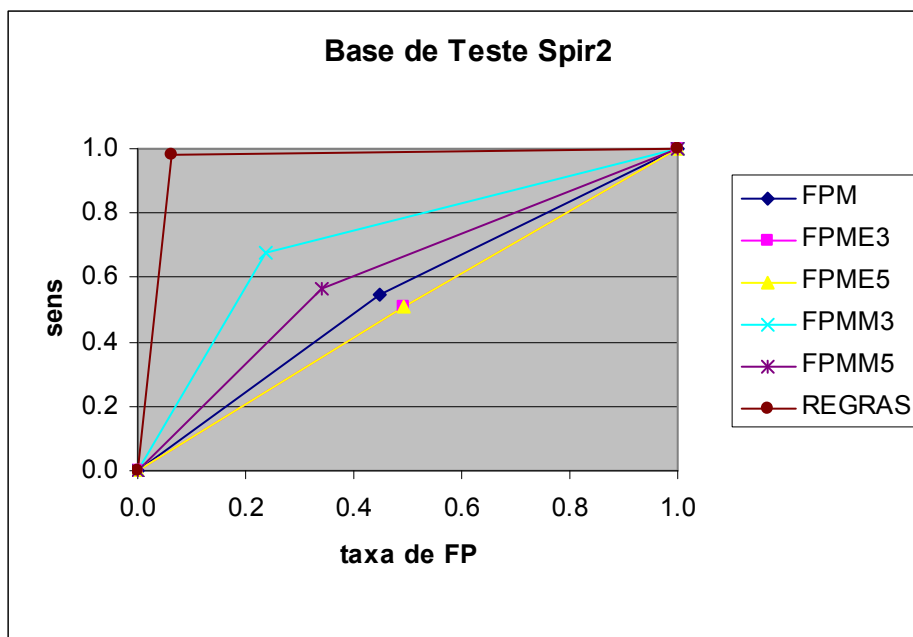


Figura 5.6. Curva ROC – Base *Spir2* – Conjunto de Teste

Para a base de dados *spir3*, novamente o melhor desempenho é verificado no algoritmo de regras de decisão *fuzzy*; embora o classificador FPM tenha apresentado uma acurácia razoável.

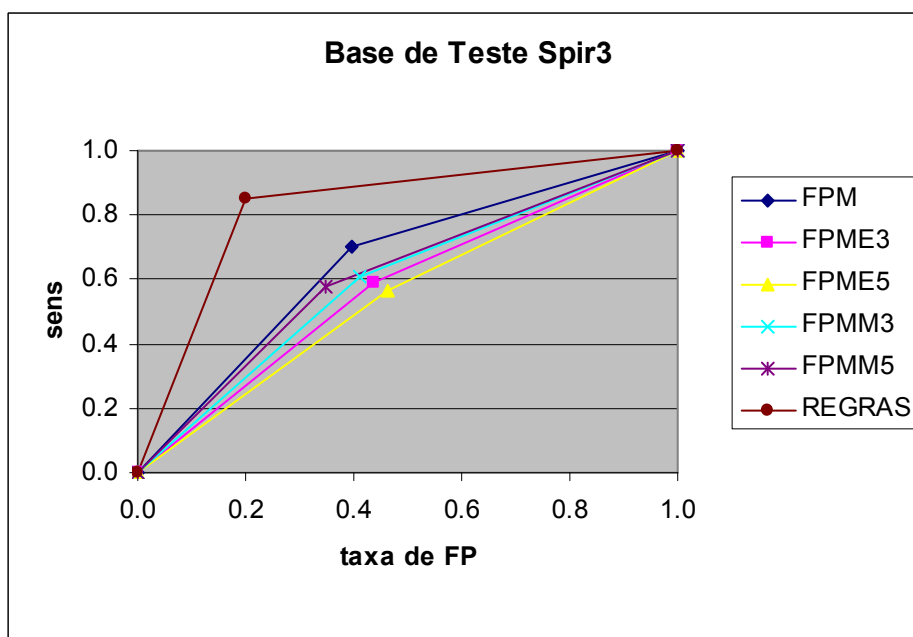


Figura 5.7. Curva ROC – Base *Spir3* – Conjunto de Teste

Na classificação da base *wine*, o melhor desempenho foi obtido no emprego do algoritmo FPMM com 3 subclasses, o que também pode ser verificado nos valores de medida F obtidos. O algoritmo de regras de decisão *fuzzy* também gerou um bom resultado nesta base.

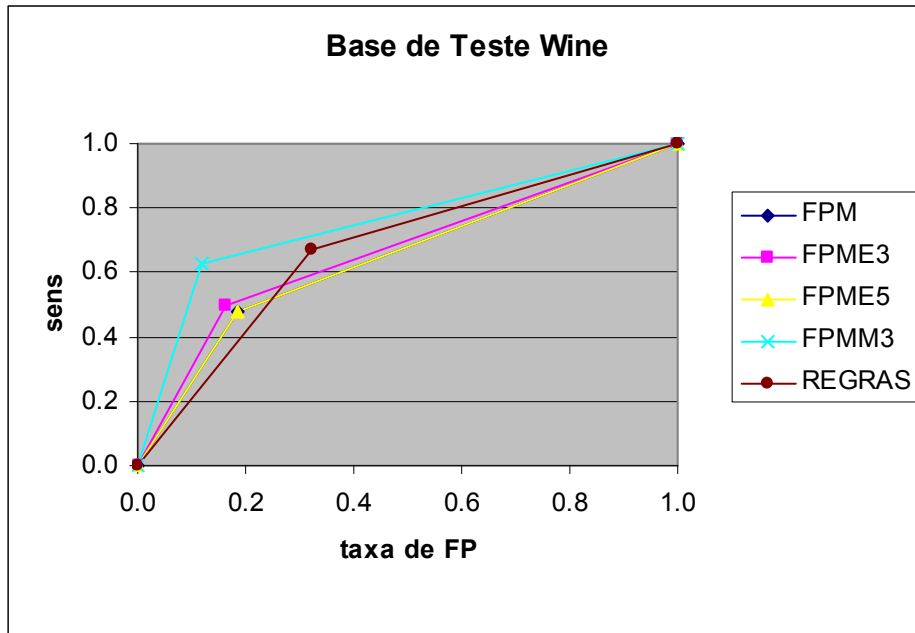


Figura 5.8. Curva ROC – Base *Wine* – Conjunto de Teste

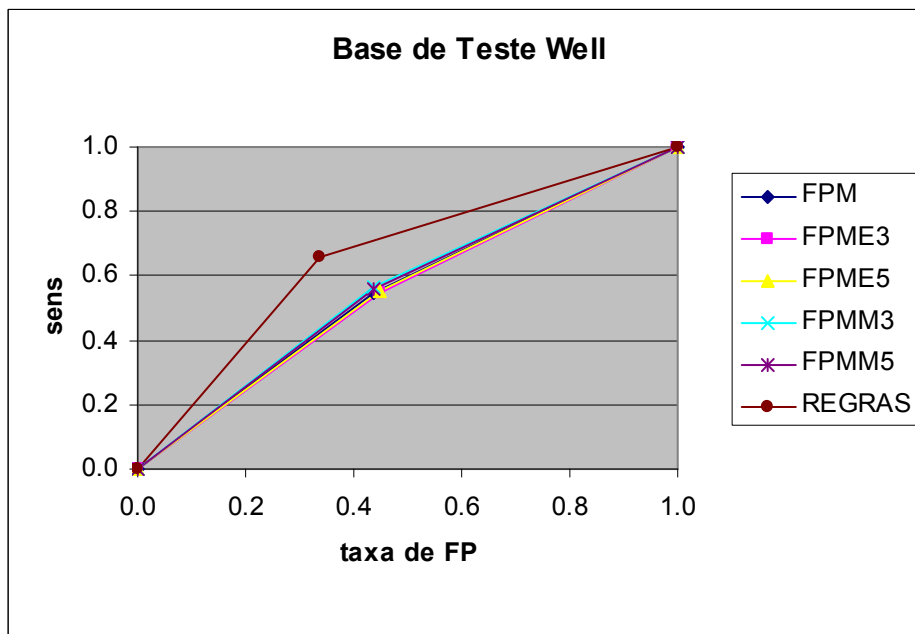


Figura 5.9. Curva ROC – Base *Well* – Conjunto de Teste

Na classificação da base *well*, os classificadores FPM, FPME E FPMM apresentaram um desempenho similar, e, assim como na classificação das bases de dados anteriores, verifica-se que o número de subclasses não causa uma influência significativa no desempenho do classificador. Mas, como no caso das bases *spir2* e *spir3*, o algoritmo de regras de decisão *fuzzy* apresenta o melhor desempenho na classificação dos dados.

Pode-se observar que o algoritmo FPM apresenta, na maioria dos casos, um desempenho inferior ao dos algoritmos FPME e FPMM, o que já era de se esperar, uma vez que o primeiro utiliza somente um protótipo por classe, enquanto que os demais utilizam a divisão em subclasses, respeitando o formato das classes.

Em relação ao algoritmo de regras de decisão *fuzzy*, verifica-se um bom desempenho deste classificador em todas as bases de dados testadas, principalmente nas bases *spir2* e *spir3*.

É notável ainda que o algoritmo de regras *fuzzy* é mais eficiente na classificação dos dados, tanto em relação aos algoritmos de classificação *fuzzy* implementados inicialmente, FPM, FPME e FPMM, quanto em relação ao algoritmo original, não *fuzzificado*.

### **5.3.2. Ensembles**

A análise realizada tem como objetivo responder às seguintes questões:

- i. É possível obter um bom poder de predição combinando poucos classificadores sobre um dado conjunto de registros?
- ii. É possível aumentar o poder de predição comparando o *ensemble* construído em relação aos seus componentes?
- iii. Quais métodos de construção de *ensembles* e quais combinações de classificadores teriam maior poder de predição?

Diante desta série de questões, diversos experimentos foram realizados, variando o número de classificadores que compõem os *ensembles* e diversificando ainda os

algoritmos de classificação usados, combinando os classificadores *fuzzy* implementados com alguns classificadores utilizados na literatura.

A Tabela 5.7 apresenta os algoritmos de classificação retirados da literatura e a Tabela 5.8 exibe as características dos seis diferentes cenários utilizados para a realização destes experimentos.

Tabela 5.7. Classificadores utilizados

<b>Abreviação</b>	<b>Algoritmo</b>
BAYES	Classificador Bayesiano Multivariado com distribuições normais
DISCLINEAR	Análise de Discriminante Linear
NAIF	Classificador Bayesiano Simples
NEURO	Classificador Neuro- <i>Fuzzy</i>

Tabela 5.8. Cenários utilizados para realização dos experimentos

<b>Cenário</b>	<b>Número de Classificadores</b>	<b>Algoritmos de Classificação</b>
<i>ensemble1</i>	3	REGRAS - NEURO - NAIF
<i>ensemble2</i>	4	REGRAS - NEURO - NAIF - FPM
<i>ensemble3</i>	4	REGRAS - NEURO - NAIF - FPMM
<i>ensemble4</i>	4	REGRAS - NEURO - NAIF - DISCLINEAR
<i>ensemble5</i>	5	REGRAS - NEURO - NAIF - DISCLINEAR - FPME
<i>ensemble6</i>	5	REGRAS - NEURO - BAYES - DISCLINEAR - FPMM

Seguem abaixo os resultados experimentais obtidos da implementação dos seis diferentes conjuntos de classificadores, onde a Tabela 5.9 contém os resultados dos

classificadores retirados da literatura para os conjuntos de teste e as Tabelas 5.10 e 5.11 apresentam os resultados dos *ensembles* para os conjuntos de treinamento e teste, respectivamente:

Tabela 5.9. Percentual de Classificações Corretas – Conjunto de Teste

Treinamento	Percentual de Classificações Corretas			
	BAYES	DISCLINEAR	NAIF	NEURO
<i>spir2</i>	52.38	50.00	52.38	99.87
<i>spir3</i>	63.16	66.84	63.16	95.26
<i>wine</i>	98.33	99.91	98.33	98.16
<i>well</i>	75.22	74.78	72.79	75.00

Tabela 5.10. Percentual de Classificações Corretas – Conjunto de Treinamento

Treinamento	Percentual de Classificações Corretas					
	<i>ensemble1</i>	<i>ensemble2</i>	<i>ensemble3</i>	<i>ensemble4</i>	<i>ensemble5</i>	<i>ensemble6</i>
<i>spir2</i>	98.41	83.53	88.49	76.19	77.78	87.90
<i>spir3</i>	91.51	87.53	86.34	78.51	79.18	85.94
<i>wine</i>	77.72	75.53	75.91	74.33	76.64	77.59
<i>well</i>	75.81	75.81	76.85	77.32	76.66	79.03

Tabela 5.11. Percentual de Classificações Corretas – Conjunto de Teste

Teste	Percentual de Classificações Corretas					
	<i>ensemble1</i>	<i>ensemble2</i>	<i>ensemble3</i>	<i>ensemble4</i>	<i>ensemble5</i>	<i>ensemble6</i>
<i>spir2</i>	98.41	82.54	88.89	76.19	77.78	87.30
<i>spir3</i>	90.53	87.37	85.26	77.89	78.95	85.79
<i>wine</i>	77.37	75.22	75.13	73.92	76.01	77.45
<i>well</i>	73.67	75.66	74.78	75.00	74.78	76.55

Ao analisar as curvas ROC da primeira base, *spir2*, pode-se observar um bom desempenho dos diferentes conjuntos de classificadores, destacando-se o conjunto referente ao primeiro cenário estudado.

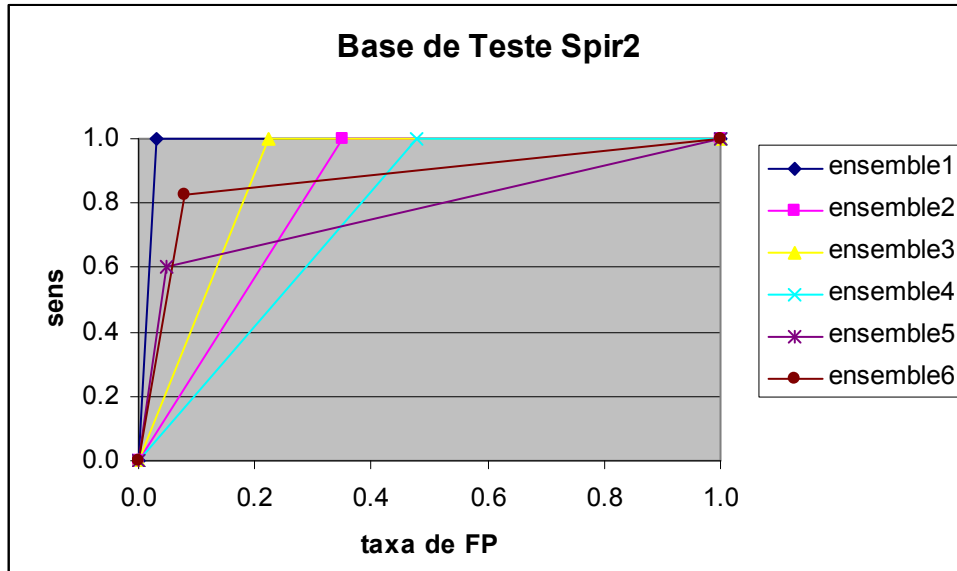


Figura 5.10. Curva ROC – Base *Spir2* – Conjunto de Teste

Já para a base de dados *spir3*, tem-se um bom desempenho dos diferentes conjuntos de classificadores, sendo este ainda melhor do que para o caso da base de dados *spir2*.

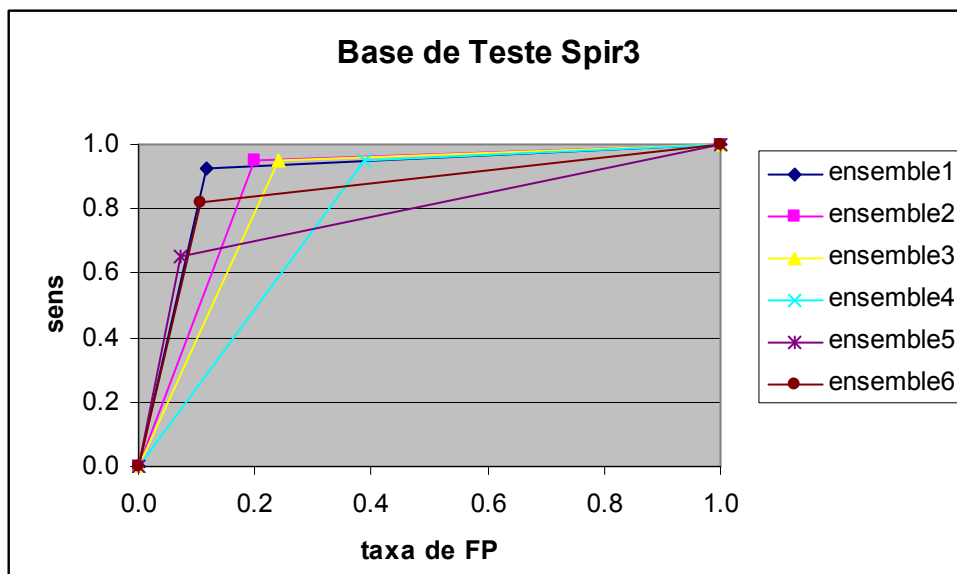


Figura 5.11. Curva ROC – Base *Spir3* – Conjunto de Teste

No caso da base de dados *wine*, observa-se que os seis *ensembles* apresentaram um desempenho similar, mas, de qualquer forma, tiveram uma acurácia razoável.

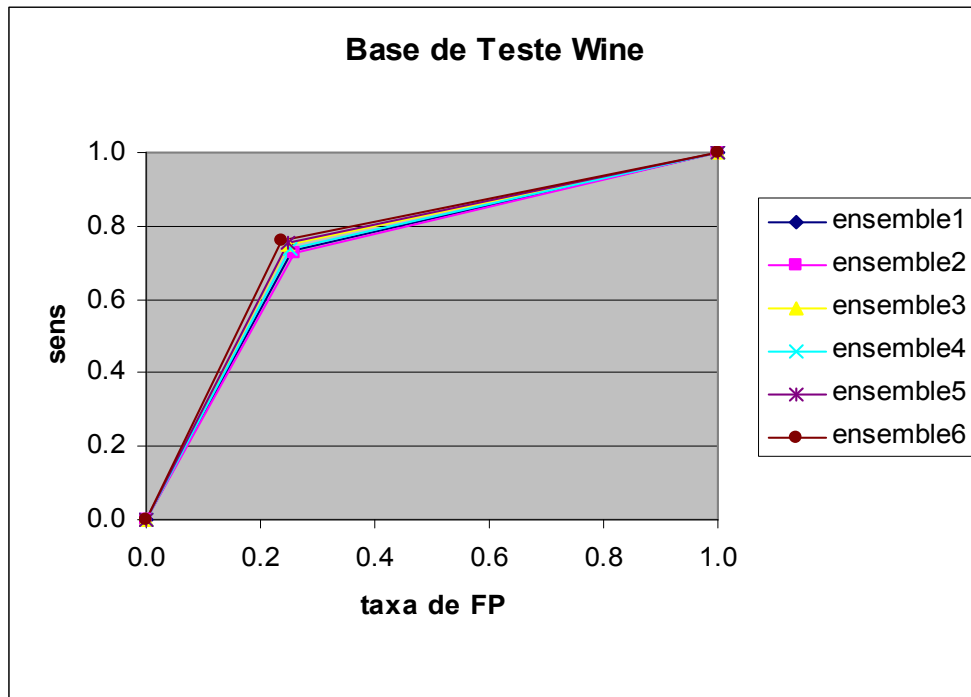


Figura 5.12. Curva ROC – Base *Wine* – Conjunto de Teste

Na classificação da base *well*, mais uma vez os seis *ensembles* apresentaram um desempenho similar, destacando-se o conjunto referente ao sexto cenário estudado.

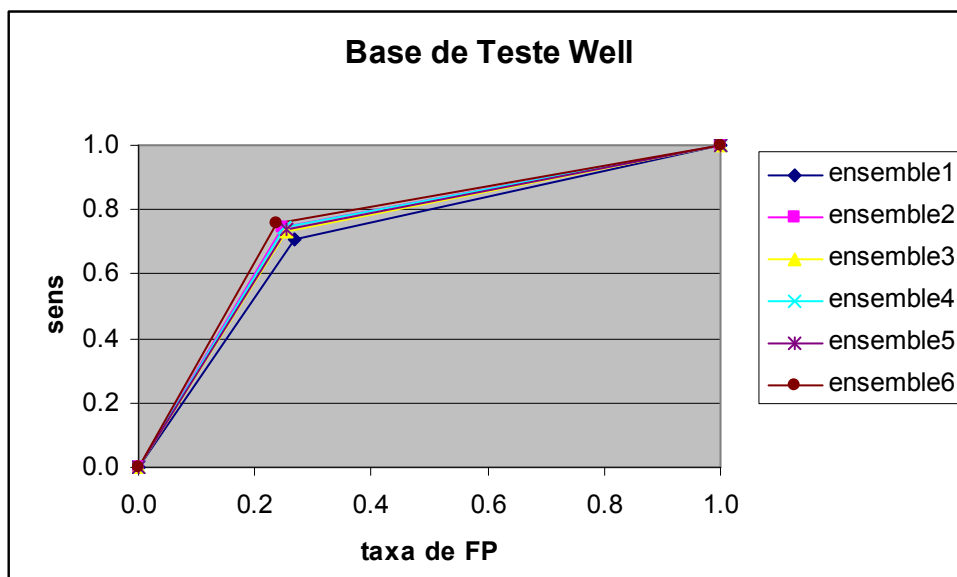


Figura 5.13. Curva ROC – Base *Well* – Conjunto de Teste

Diante dos resultados obtidos com os seis diferentes cenários, conclui-se que é possível obter um bom poder de predição combinando poucos classificadores sobre um dado conjunto de registros, verifica-se ainda uma melhoria significativa no percentual de classificações corretas dos *ensembles* de classificadores em relação a seus classificadores componentes individualmente, confirmando assim a eficácia desta técnica de classificação.

## 6. Conclusão

---

Em muitas aplicações e situações rotineiras presentes na indústria do petróleo, é necessário extrair conhecimento de bases de dados para auxiliar a tomada de decisões futuras. Para extrair conhecimento dessas bases, uma tarefa de mineração de dados prática e geralmente utilizada é a classificação, uma vez que o conhecimento extraído por um classificador é facilmente compreensível e simples de ser aplicado.

O enfoque do trabalho, que é a classificação *fuzzy*, tem grande aplicabilidade e gera bons resultados por ser um processo muito rápido, de grande simplicidade computacional e processual, sem etapas anteriores e sem parametrizações, se tornando muito eficaz quando o objetivo final é classificar dados.

Neste trabalho, além dos algoritmos *fuzzy*, e do algoritmo de regras de decisão, foi estudada a técnica de agrupar classificadores, *ensemble*, e foi verificado o aumento notável do poder de predição dos *ensembles* em relação aos seus classificadores componentes individualmente.

Dentre os métodos de construção de *ensembles* estudados, somente um deles foi testado, o método de votação sem peso. Este método pode ser mais indicado pelo fato de não requerer o cálculo dos pesos das hipóteses, o que implicaria, por exemplo, no cálculo da estimativa da taxa de erro dos classificadores componentes, e diminuiria assim a eficiência computacional do *ensemble*.

Uma proposta para trabalhos futuros é investigar mais profundamente os diversos métodos de construção de *ensembles*, como estes métodos podem influenciar no poder de predição dos *ensembles*, quais métodos seriam mais eficientes e ainda pesquisar a construção de um novo método de construção de *ensembles*, tudo isso de maneira a aumentar o poder de predição dos *ensembles* construídos.

Diante da proposta definida inicialmente neste trabalho, que é a de desenvolver e pesquisar metodologias eficientes para a classificação de dados da indústria do petróleo, este estudo obteve um resultado satisfatório, com a aplicação de uma das áreas mais ativas em aprendizado de máquina, que é a técnica de construção de *ensembles*.

## Referências Bibliográficas

---

- AMINZADEH, F., 1994, “Applications of fuzzy expert systems in integrated oil exploration”, *Computers and Electrical Engineering*, v. 2, pp. 89–97.
- BERNARDINI, F. C., 2006, *Combinação de classificadores simbólicos utilizando medidas de regras de conhecimento e algoritmos genéticos*. Tese de D.Sc., ICMC/USP, São Carlos, SP, Brasil.
- BEZDEK, J. C., TRIVEDI, M., EHRLICH, R., FULL, W. C., 1981, “Fuzzy clustering: A new approach for geostatistical analysis”, *International Journal of System, Measurement and Decision*, v. 1, pp. 13–23.
- BLAKE, C. L., MERZ, C. J., 1998, UCI Repository of ML Databases, University of California, Department of Information and Computer Science. Disponível na Internet via: <http://www.ics.uci.edu/~mlearn/MLRepository.html>.
- BOIS, P., 1983, “Some applications of pattern recognition to oil and gas exploration”, *IEEE Transactions on Geoscience and Remote Sensing*, v. 21, n. 4, pp. 687–701.
- BOIS, P., 1984, “Fuzzy seismic interpretation”, *IEEE Transactions on Geoscience and Remote Sensing*, v. 22, n. 6, pp. 692–697.
- BRAUNSCHWEIG, B., DAY, R., 1995, *Artificial intelligence in the petroleum industry. Symbolic and computational applications*. Paris, Editions Technip.
- BREIMAN, L., 1996, “Bagging predictors”, *Machine Learning*, v. 24, n. 2, pp. 123–140.
- CHAPPAZ, R. J., 1977, “Application of the fuzzy sets theory to the interpretation of seismic”, *Abstract of the 47th SEG meeting*, Calgary, Paper R-9.
- CHEN, H. C., FANG, J. H., KORTRIGHT, M. E., CHEN, D. C., 1995, “Novel Approaches to the Determination of Archie Parameters II: Fuzzy Regression Analysis”, *SPE Advanced Technology Series*, v. 3, n. 1.
- CHUNG, T. H., CARROLL, H. B., LINDSEY, R., 1995, “Application of Fuzzy Expert Systems for EOR Project Risk Analysis”, *SPE paper 30741, SPE Annual Technical Conference and Exhibition*, Dallas, October 22-25.
- CUDDY, S. J., 2000, “Litho-facies and permeability prediction from electrical logs using fuzzy logic”, *SPE Reservoir Evaluation and Engineering*, v. 3, n. 4, pp. 319–324.

- DEVILLEZ, A., 2004, "Four fuzzy supervised classification methods for discriminating classes of non-convex shape", *Fuzzy Sets and Systems*, v. 141, pp. 219-240.
- DIETTERICH, T. G., 2000, "Ensemble methods in machine learning", *First International Workshop on Multiple Classifier Systems, Lecture Notes in Computer Science*, v. 1857, pp. 1-15.
- DUBOIS, D., PRADE, H., 1987, *Théorie des possibilités: Application à la représentation des connaissances en informatique*, deuxième édition, Masson, Paris.
- DUBOIS, D., PRADE, H., TESTEMALE, C., 1988, "Weighted fuzzy pattern matching", *Fuzzy Sets and Systems*, v. 28, pp. 313-331.
- EFRON, B., TIBSHIRANI, R., 1993, *An Introduction to Bootstrap*. Springer Verlag.
- ESPÍNDOLA, R. P., 2004, *Sistema Inteligente para Classificação de Dados*. Tese de D.Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil.
- ESPÍNDOLA, R. P., EBECKEN, N. F. F., 2002, "A Hybrid Fuzzy Genetic System to Data Classification in the Petroleum Industry", *23rd Iberian Latin American Congress on Computational Methods in Engineering*, Giulianova, Itália, 24-26 June.
- EVSUKOFF, A. G., GONÇALVES, F. T. T., BEDREGAL, R. P., *et al.*, 2004, "Fuzzy Classification of Surface Geochemistry Data Applied to the Determination of HC Anomalies", *2<sup>nd</sup> International Conference on Soft Computing and Intelligent Systems and 5<sup>th</sup> International Symposium on Advanced Intelligent Systems*.
- FAYYAD, U. M., PIATETSKY-SHAPIRO G., SMYTH, P., 1996, "From data mining to knowledge discovery: an overview", *Advances in Knowledge Discovery and Data Mining*, AAAI Press, Cambridge, MA, pp. 1 - 36.
- FINOL, J., GUO, Y. K. and JING, X. D., 2001, "A rule based fuzzy model for the prediction of petrophysical rock parameters", *Journal of Petroleum Science and Engineering*, v. 29, pp. 97-113.
- FLETCHER, A., DAVIS, J. P., 2002, "Decision-Making with Incomplete Evidence", *SPE 77914, Proceedings, 13th European Petroleum Conference*, Aberdeen, Scotland, U.K., 29–31 October.
- FREUND, Y., SCHAPIRE, R., 1997, "A decision-theoretic generalization of on-line learning and an application to boosting", *Journal of Computer and System Sciences*, v. 55, pp. 119-139.

- FUNG, C. C., WONG, K. W., WONG, P. M., 1997, "A self-generating fuzzy rules inference for petrophysical properties prediction", *Proceedings of the IEEE International Conference on Intelligent Processing System*, Beijing, China, pp. 205-208.
- GARCIA, A., MOHAGHEGH, S. D., 2004, "Forecasting U.S. Natural Gas Production Into Year 2020: A Comparative Study", *SPE 91413, Proceedings, SPE Eastern Regional Conference and Exhibition*, Charleston, West Virginia, September 15 – 17.
- GARROUCH, A. A., LABABIDI, H. M., 2001, "Development of an Expert System for Underbalance Drilling Using Fuzzy Logic", *Journal of Petroleum Science and Engineering*, v. 31, pp. 23-39.
- GOLDBERG, D., 1989, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley.
- HAN, J., KAMBER, M., 2001, *Data Mining: Concepts and Techniques*, 1 ed., California, Morgan Kaufmann.
- HAND, D. J., TILL, R. J., 2001, "A Simple Generalization of the Area Under the ROC Curve for Multiple Class Classification Problems", *Machine Learning*, v. 45, pp. 171-186.
- HANSEN, L., SALAMON, P., 1990, "Neural networks ensembles", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 12, pp. 993-1001.
- HAYKIN, S., 1999, *Neural Networks: A Comprehensive Foundation*, 2 ed., New Jersey, Prentice Hall.
- HOLMSTRÖM, L., KOISTINEN, P., LAAKSONEN, J., OJA, E., 1996, *Comparison of neural and statistical classifiers - theory and practice*, Rolf Nevanlinna Institute Research Reports A13, Finland.
- HONGJIE, X., 1995, "A comprehensive approach to formation damage diagnosis and corresponding stimulation type and fluid selection", *SPE paper 29531, Production Operations Symposium*, Oklahoma, 2-4 April.
- HONGJIE, X., HOLDTICH, S., 1994, "Investigation into the application of fuzzy logic to well stimulation treatment design", *Proceedings of the Permian Basin Oil & Gas Recovery Conference*, Midland, pp. 457-468.
- HSIEH, B. Z., LIN, Z. S., LEWIS, C., 2005, "Lithology identification of aquifers from geophysical well logs and fuzzy logic analysis: Shui-Lin Area, Taiwan", *Computers & Geosciences*, v. 31, pp. 263-275.

- HUANG, Y., GEDEON, T. D., WONG, P. M., 1999, "A practice fuzzy interpolator for prediction of reservoir permeability", *Proceedings of the IEEE International Conference on Fuzzy Systems*, South Korea, pp. III-1528–III-1533.
- ISAKSEN, G. H., KIM, C. S., 1997, "Interpretation of molecular geochemistry data by the application of artificial intelligence technology", *Organic Geochemistry*, v. 26, n. 1/2, pp. 1-10.
- KRAMAR, U., 1995, "Application of limited fuzzy clusters to anomaly recognition in complex geological environments", *Journal of Geochemical Exploration*, v. 55, pp. 81-92.
- KURGAN, L. A., CIOS, K. J., DICK, S., 2006, "Highly Scalable and Robust Rule Learner: Performance Evaluation and Comparison", *IEEE Systems, Man, and Cybernetics – Part B: Cybernetics*, v. 36, n. 1, pp. 32-53.
- LING, C. X., HUANG, J., ZHANG, H., 2003, "AUC - a Statistically Consistent and more Discriminating Measure than Accuracy", *Proceedings of the International Joint Conference on Artificial Intelligence IJCAI 2003*, pp. 519-526, Acapulco, Mexico, 9-15 August.
- LIU, B., HSU, W., MA, Y., 1998, "Integrating classification and association rule mining", *Proc. 1998 Int. Conf. Knowledge Discovery and Data Mining*, pp. 80-86, New York, August.
- MARTINEZ-TORRES, L. P., 2002, *Characterization of naturally fractured reservoirs from conventional well logs*. M.Sc. Dissertation, University of Oklahoma, 155 pp.
- MICHIE, D., SPIEGELHALTER, D. J., TAYLOR, C. C., 1994, *Machine Learning, Neural and Statistical Classification*, 1 ed., Hertfordshire, Ellis Horwood.
- MOHAGHEGH, S. D., 2000, "Virtual Intelligence Applications in Petroleum Engineering: Part 3 – Fuzzy Logic", *Journal of Petroleum Technology, Distinguished Author Series*, pp. 82-87.
- NIKRAVESH, M., ADAMS, R. D., LEVEY, R. A., 2001, "Soft Computing: Tools for Intelligent Reservoir Characterization (IRESC) and Optimum Well Placement (OWP)", *Journal of Petroleum Science and Engineering*, v. 29, pp. 239-262.
- NIKRAVESH, M., AMINZADEH, F., 2001a, "Mining and fusion of petroleum data with fuzzy logic and neural network agents", *Journal of Petroleum Science and Engineering*, v. 29, pp. 221-238.

- NIKRAVESH, M., AMINZADEH, F., 2001b, "Past, present and future intelligent reservoir characterization trends", *Journal of Petroleum Science and Engineering*, v. 31, pp. 67-79.
- QUINLAN, J. R., 1993, *C4.5: Programs for Machine Learning*, 1 ed., California, Morgan Kaufmann.
- REZENDE, S. O., PUGLIESI, J. B., MELANDA E. A. & DE PAULA, M. F., 2003, "Mineração de Dados", *Sistemas Inteligentes: Fundamentos e Aplicações*, Barueri, SP, Brasil, Rezende, S. O.(coord.), Editora Manole Ltda., cap. 12, pp.307-336.
- RIVERA, V. P., 1994, "Fuzzy Logic controls pressure in fracturing fluid characterization facility", *SPE paper 28239, SPE Petroleum Computer Conference*, Dallas, Aug. 3.
- VELEZ-LANGS, O., 2005, "Genetic algorithms in oil industry: An overview", *Journal of Petroleum Science and Engineering*, v. 47, pp. 15-22.
- WEISS, W. W., BALCH, R. S., STUBBS, B. A., 2002, "How Artificial Intelligence Methods Can Forecast Oil Production", SPE 75143, *Proceedings, SPE/DOE Improved Oil Recovery Symposium*, Tulsa, Oklahoma, 13–17 April 2002.
- ZADEH, L. A., 1965, "Fuzzy sets", *Information and Control*, v. 8, pp. 338-353.
- ZADEH, L. A., 1978, "Fuzzy sets as a basis for a theory of possibility", *Fuzzy Sets and Systems*, v. 1, pp. 3-28.

## Anexo

Tabela A.1. Sensitividade – Conjunto de Treinamento

Treinamento	Sensitividade				
	FPM	FPME (3)	FPME (5)	FPMM (3)	FPMM (5)
Spir2	0.5568	0.5220	0.5491	0.6858	0.5751
Spir3	0.6756	0.5878	0.5654	0.5938	0.5813
Wine	0.5084	0.4970	0.5084	0.6553	.....
Well	0.5482	0.5517	0.5523	0.5641	0.5652

Tabela A.2. Sensitividade – Conjunto de Teste

Teste	Sensitividade				
	FPM	FPME (3)	FPME (5)	FPMM (3)	FPMM (5)
Spir2	0.5454	0.5098	0.5098	0.6800	0.5682
Spir3	0.7031	0.5897	0.5634	0.6118	0.5806
Wine	0.4779	0.4995	0.4779	0.6298	.....
Well	0.5475	0.5496	0.5503	0.5632	0.5621

Tabela A.3. Especificidade – Conjunto de Treinamento

Treinamento	Especificidade				
	FPM	FPME (3)	FPME (5)	FPMM (3)	FPMM (5)
Spir2	0.5625	0.5150	0.5393	0.7644	0.6755
Spir3	0.5935	0.5624	0.5393	0.5821	0.6527
Wine	0.8047	0.8019	0.8047	0.8570	.....
Well	0.5647	0.5541	0.5558	0.5629	0.5624

Tabela A.4. Especificidade – Conjunto de Teste

Teste	Especificidade				
	FPM	FPME (3)	FPME (5)	FPMM (3)	FPMM (5)
Spir2	0.5500	0.5067	0.5067	0.7647	0.6579
Spir3	0.6032	0.5625	0.5378	0.5905	0.6515
Wine	0.8163	0.8363	0.8163	0.8830	.....
Well	0.5631	0.5527	0.5535	0.5622	0.5614

Tabela A.5. Precisão – Conjunto de Treinamento

Treinamento	Precisão				
	FPM	FPME (3)	FPME (5)	FPMM (3)	FPMM (5)
Spir2	0.5833	0.4246	0.4881	0.8056	0.8056
Spir3	0.4695	0.4880	0.4244	0.5544	0.7586
Wine	0.6774	0.6689	0.6774	0.7704	.....
Well	0.5433	0.6041	0.6190	0.6221	0.6473

Tabela A.6. Precisão – Conjunto de Teste

Teste	Precisão				
	FPM	FPME (3)	FPME (5)	FPMM (3)	FPMM (5)
Spir2	0.5714	0.4127	0.4127	0.8095	0.7936
Spir3	0.4737	0.4842	0.4210	0.5474	0.7579
Wine	0.6494	0.6660	0.6494	0.7659	.....
Well	0.5362	0.6034	0.6058	0.6194	0.6415

Tabela A.7. Medida F – Conjunto de Treinamento

Treinamento	Medida F				
	FPM	FPME (3)	FPME (5)	FPMM (3)	FPMM (5)
Spir2	0.5697	0.4683	0.5168	0.7409	0.6711
Spir3	0.5540	0.5333	0.4849	0.5734	0.6582
Wine	0.5809	0.5703	0.5809	0.7082	.....
Well	0.5457	0.5767	0.5838	0.5917	0.6035

Tabela A.8. Medida F – Conjunto de Teste

Teste	Medida F				
	FPM	FPME (3)	FPME (5)	FPMM (3)	FPMM (5)
Spir2	0.5581	0.4561	0.4561	0.7391	0.6622
Spir3	0.5660	0.5318	0.4819	0.5778	0.6575
Wine	0.5506	0.5709	0.5506	0.6912	.....
Well	0.5418	0.5752	0.5767	0.5900	0.5992

Tabela A.9. Média GSP – Conjunto de Treinamento

Treinamento	Média GSP				
	FPM	FPME (3)	FPME (5)	FPMM (3)	FPMM (5)
Spir2	0.5699	0.4708	0.5177	0.7433	0.6807
Spir3	0.5632	0.5356	0.4899	0.5738	0.6641
Wine	0.5868	0.5766	0.5868	0.7105	.....
Well	0.5457	0.5773	0.5847	0.5924	0.6048

Tabela A.10. Média GSP – Conjunto de Teste

Teste	Média GSP				
	FPM	FPME (3)	FPME (5)	FPMM (3)	FPMM (5)
Spir2	0.5582	0.4587	0.4587	0.7419	0.6715
Spir3	0.5771	0.5344	0.4870	0.5787	0.6634
Wine	0.5571	0.5768	0.5571	0.6945	.....
Well	0.5418	0.5759	0.5774	0.5906	0.6005

Tabela A.11. Média GSE – Conjunto de Treinamento

Treinamento	Média GSE				
	FPM	FPME (3)	FPME (5)	FPMM (3)	FPMM (5)
Spir2	0.5596	0.5185	0.5442	0.7240	0.6233
Spir3	0.6332	0.5750	0.5522	0.5879	0.6160
Wine	0.6396	0.6313	0.6396	0.7494	.....
Well	0.5564	0.5529	0.5540	0.5635	0.5638

Tabela A.12. Média GSE – Conjunto de Teste

Teste	Média GSE				
	FPM	FPME (3)	FPME (5)	FPMM (3)	FPMM (5)
Spir2	0.5477	0.5082	0.5082	0.7211	0.6114
Spir3	0.6512	0.5759	0.5505	0.6011	0.6150
Wine	0.6246	0.6463	0.6246	0.7457	.....
Well	0.5552	0.5511	0.5519	0.5627	0.5617

Tabela A.13. Sensitividade - Algoritmo de Regras

<b>Sensitividade</b>		
	<b>Treinamento</b>	<b>Teste</b>
spir2	0.9643	0.9841
spir3	0.8568	0.8526
wine	0.6713	0.6692
well	0.6645	0.6580

Tabela A.14. Especificidade - Algoritmo de Regras

<b>Especificidade</b>		
	<b>Treinamento</b>	<b>Teste</b>
spir2	0.9365	0.9365
spir3	0.7984	0.8000
wine	0.6851	0.6783
well	0.6705	0.6645

Tabela A.15. Precisão - Algoritmo de Regras

<b>Precisão</b>		
	<b>Treinamento</b>	<b>Teste</b>
spir2	0.9382	0.9394
spir3	0.8095	0.8100
wine	0.7002	0.6958
well	0.7429	0.7460

Tabela A.16. Medida F - Algoritmo de Regras

<b>Medida F</b>		
	<b>Treinamento</b>	<b>Teste</b>
spir2	0.9511	0.9612
spir3	0.8325	0.8308
wine	0.6854	0.6822
well	0.7015	0.6992

Tabela A.17. Média GSP - Algoritmo de Regras

<b>Média GSP</b>		
	<b>Treinamento</b>	<b>Teste</b>
spir2	0.9512	0.9615
spir3	0.8328	0.8310
wine	0.6856	0.6824
well	0.7026	0.7006

Tabela A.18. Média GSE – Algoritmo de Regras

<b>Média GSE</b>		
	<b>Treinamento</b>	<b>Teste</b>
spir2	0.9503	0.9600
spir3	0.8271	0.8259
wine	0.6782	0.6737
well	0.6675	0.6612

Tabela A.19. Sensitividade – Conjunto de Treinamento

<b>Treinamento</b>	<b>Sensitividade</b>					
	<i>ensemble1</i>	<i>ensemble2</i>	<i>ensemble3</i>	<i>ensemble4</i>	<i>ensemble5</i>	<i>ensemble6</i>
spir2	1.0000	1.0000	1.0000	1.0000	0.5992	0.8373
spir3	0.9416	0.9602	0.9576	0.9522	0.6658	0.8276
wine	0.7351	0.7345	0.7561	0.7449	0.7593	0.7683
well	0.7273	0.7290	0.7440	0.7536	0.7445	0.7717

Tabela A.20. Sensitividade – Conjunto de Teste

<b>Teste</b>	<b>Sensitividade</b>					
	<i>ensemble1</i>	<i>ensemble2</i>	<i>ensemble3</i>	<i>ensemble4</i>	<i>ensemble5</i>	<i>ensemble6</i>
spir2	1.0000	1.0000	1.0000	1.0000	0.6032	0.8254
spir3	0.9263	0.9474	0.9474	0.9474	0.6526	0.8210
wine	0.7304	0.7289	0.7513	0.7392	0.7522	0.7614
well	0.7109	0.7429	0.7345	0.7423	0.7384	0.7594

Tabela A.21. Especificidade – Conjunto de Treinamento

<b>Treinamento</b>	<b>Especificidade</b>					
	<i>ensemble1</i>	<i>ensemble2</i>	<i>ensemble3</i>	<i>ensemble4</i>	<i>ensemble5</i>	<i>ensemble6</i>
spir2	0.9682	0.6706	0.7698	0.5238	0.9563	0.9206
spir3	0.8886	0.7904	0.7692	0.6180	0.9178	0.8912
wine	0.7436	0.7432	0.7525	0.7518	0.7537	0.7635
well	0.7505	0.7505	0.7620	0.7675	0.7602	0.7852

Tabela A.22. Especificidade – Conjunto de Teste

<b>Teste</b>	<b>Especificidade</b>					
	<i>ensemble1</i>	<i>ensemble2</i>	<i>ensemble3</i>	<i>ensemble4</i>	<i>ensemble5</i>	<i>ensemble6</i>
spir2	0.9682	0.6508	0.7778	0.5238	0.9524	0.9206
spir3	0.8842	0.8000	0.7579	0.6105	0.9263	0.8947
wine	0.7432	0.7415	0.7519	0.7491	0.7503	0.7623
well	0.7314	0.7537	0.7449	0.7480	0.7454	0.7640

Tabela A.23. Precisão – Conjunto de Treinamento

<b>Treinamento</b>	<b>Precisão</b>					
	<i>ensemble1</i>	<i>ensemble2</i>	<i>ensemble3</i>	<i>ensemble4</i>	<i>ensemble5</i>	<i>ensemble6</i>
spir2	0.9692	0.7522	0.8129	0.6774	0.9321	0.9134
spir3	0.8942	0.8209	0.8058	0.7137	0.8901	0.8838
wine	0.7738	0.7517	0.7586	0.7414	0.7643	0.7721
well	0.7951	0.7792	0.7934	0.7935	0.7872	0.8058

Tabela A.24. Precisão – Conjunto de Teste

<b>Teste</b>	<b>Precisão</b>					
	<i>ensemble1</i>	<i>ensemble2</i>	<i>ensemble3</i>	<i>ensemble4</i>	<i>ensemble5</i>	<i>ensemble6</i>
spir2	0.9692	0.7412	0.8182	0.6774	0.9268	0.9123
spir3	0.8889	0.8257	0.7965	0.7087	0.8986	0.8864
wine	0.7713	0.7506	0.7527	0.7395	0.7592	0.7718
well	0.7830	0.7758	0.7648	0.7694	0.7670	0.7733

Tabela A.25. Medida F – Conjunto de Treinamento

<b>Treinamento</b>	<b>Medida F</b>					
	<i>ensemble1</i>	<i>ensemble2</i>	<i>ensemble3</i>	<i>ensemble4</i>	<i>ensemble5</i>	<i>ensemble6</i>
spir2	0.9844	0.8586	0.8968	0.8077	0.7295	0.8737
spir3	0.9173	0.8851	0.8752	0.8159	0.7618	0.8548
wine	0.7540	0.7430	0.7573	0.7431	0.7618	0.7702
well	0.7597	0.7533	0.7679	0.7730	0.7652	0.7884

Tabela A.26. Medida F – Conjunto de Teste

<b>Teste</b>	<b>Medida F</b>					
	<i>ensemble1</i>	<i>ensemble2</i>	<i>ensemble3</i>	<i>ensemble4</i>	<i>ensemble5</i>	<i>ensemble6</i>
spir2	0.9844	0.8514	0.9000	0.8077	0.7308	0.8667
spir3	0.9072	0.8824	0.8654	0.8108	0.7561	0.8524
wine	0.7503	0.7396	0.7520	0.7393	0.7557	0.7666
well	0.7452	0.7590	0.7493	0.7556	0.7524	0.7663

Tabela A.27. Média GSP – Conjunto de Treinamento

<b>Treinamento</b>	<b>Média GSP</b>					
	<i>ensemble1</i>	<i>ensemble2</i>	<i>ensemble3</i>	<i>ensemble4</i>	<i>ensemble5</i>	<i>ensemble6</i>
spir2	0.9845	0.8673	0.9016	0.8230	0.7473	0.8745
spir3	0.9176	0.8878	0.8784	0.8244	0.7698	0.8552
wine	0.7542	0.7431	0.7573	0.7431	0.7618	0.7702
well	0.7604	0.7537	0.7683	0.7733	0.7656	0.7886

Tabela A.28. Média GSP – Conjunto de Teste

<b>Teste</b>	<b>Média GSP</b>					
	<i>ensemble1</i>	<i>ensemble2</i>	<i>ensemble3</i>	<i>ensemble4</i>	<i>ensemble5</i>	<i>ensemble6</i>
spir2	0.9845	0.8609	0.9045	0.8230	0.7477	0.8678
spir3	0.9074	0.8844	0.8687	0.8194	0.7658	0.8531
wine	0.7506	0.7397	0.7520	0.7393	0.7557	0.7666
well	0.7461	0.7592	0.7495	0.7557	0.7526	0.7663

Tabela A.29. Média GSE – Conjunto de Treinamento

<b>Treinamento</b>	<b>Média GSE</b>					
	<i>ensemble1</i>	<i>ensemble2</i>	<i>ensemble3</i>	<i>ensemble4</i>	<i>ensemble5</i>	<i>ensemble6</i>
spir2	0.9840	0.8189	0.8774	0.7237	0.7570	0.8780
spir3	0.9147	0.8712	0.8582	0.7671	0.7817	0.8588
wine	0.7393	0.7388	0.7543	0.7483	0.7565	0.7659
well	0.7388	0.7397	0.7529	0.7605	0.7523	0.7784

Tabela A.30. Média GSE – Conjunto de Teste

<b>Teste</b>	<b>Média GSE</b>					
	<i>ensemble1</i>	<i>ensemble2</i>	<i>ensemble3</i>	<i>ensemble4</i>	<i>ensemble5</i>	<i>ensemble6</i>
spir2	0.9840	0.8067	0.8819	0.7237	0.7579	0.8717
spir3	0.9050	0.8706	0.8474	0.7605	0.7775	0.8570
wine	0.7368	0.7352	0.7516	0.7441	0.7512	0.7618
well	0.7211	0.7483	0.7397	0.7451	0.7419	0.7617