

# O USO DE POSSIBILIDADE COMO MEDIDA DE SIMILARIDADE NA CLASSIFICAÇÃO BASEADA EM EXEMPLARES EM DOMÍNIOS FUZZY

FLÁVIA OLIVEIRA SANTOS DE SÁ LISBOA

*Instituto de Física de São Carlos, Universidade de São Paulo  
Caixa Postal 369, 13560-970, São Carlos, SP, Brazil  
e-mail: flavia@if.sc.usp.br*

MARIA DO CARMO NICOLETTI

*Departamento de Computação, Universidade Federal de São Carlos  
Caixa Postal 676, 13565-905, São Carlos, SP, Brazil  
e-mail: carmo@dc.ufscar.br*

**Resumo**— Este artigo descreve as principais idéias usadas no desenvolvimento de um sistema de aprendizado de máquina, para domínios *fuzzy*, o qual induz hipóteses representadas por valores *fuzzy*, a partir de um conjunto de exemplos de treinamento, cada um deles descrito por um vetor de números *fuzzy* e uma classe associada representada por um valor *crisp*. O artigo apresenta e discute os resultados obtidos com o uso desse sistema em alguns domínios de conhecimento.

**Abstract**— This paper describes the main ideas used in the development of a machine learning system, for fuzzy domains, which induces fuzzy hypotheses described by fuzzy real numbers from a set of training examples and an associated crisp class. It presents and discusses results obtained using the system in a set of domains.

**Keywords**— fuzzy domain learning, supervised learning, exemplar-based learning.

## 1 Introdução

A Teoria de Exemplos Generalizados Aninhados (NGE – *Nested Generalized Exemplar*) (Salzberg, 1991) pode ser considerada descendente da família de algoritmos *nearest neighbour* (Cover, 1967). O algoritmo básico da teoria NGE, conhecido como EACH, é um algoritmo de aprendizado indutivo de máquina que generaliza um conjunto de exemplos de treinamento em um conjunto de hipóteses, representadas por um conjunto de hiperretângulos em um espaço euclidiano  $n$ -dimensional. Após o aprendizado, um novo exemplo pode ser classificado via cálculo da distância euclidiana entre o exemplo e cada um dos exemplos – a classe do exemplar mais próximo do exemplo é atribuída a ele.

Este artigo apresenta as principais idéias que subsidiaram o desenvolvimento de um sistema de aprendizado para domínios *fuzzy* chamado FuGES (*Fuzzy Generalized Exemplar System*), o qual foi inspirado pela teoria NGE e cujo algoritmo básico pode ser considerado uma evolução daquele descrito em (Nicoletti, 1996).

O artigo está organizado como segue: a Seção 2 apresenta as principais idéias do modelo de aprendizado NGE. A Seção 3 descreve o algoritmo FuGES em detalhes, focalizando: a) duas funções de pertinência que são utilizadas na representação de números reais *fuzzy*; b) uma métrica para avaliar o grau de similaridade entre dois números reais *fuzzy*, baseada na extensão do conceito de possibilidade e c) três novos operadores que implementam o processo de generalização entre dois números reais *fuzzy*. A

Seção 4 apresenta e analisa os resultados experimentais obtidos com o uso do protótipo FuGES em conjuntos de dados extraídos do UCI Repository (Merz, 1998). As conclusões são apresentadas na Seção 5, apontando possíveis extensões para o trabalho de pesquisa descrito neste artigo.

## 2 A Teoria NGE

A teoria NGE é um modelo de aprendizado baseado em exemplos de classes, onde a hipótese induzida é representada como um conjunto de hiperretângulos em um espaço euclidiano  $n$ -dimensional. Os exemplos de classes são hiperretângulos ou então, pontos (referenciados como hiperretângulos triviais).

A entrada para o NGE é um conjunto de exemplos de treinamento (conjunto de treinamento) que são apresentados incrementalmente ao sistema, onde cada exemplo é descrito como um vetor de pares *atributo\_numérico/valor* e uma classe associada. Os  $n$  atributos usados para descrever os exemplos definem o espaço  $n$ -dimensional no qual o conceito será induzido e representado.

O NGE generaliza um conjunto inicial de pontos (chamados sementes) em hiperretângulos. O número de sementes é definido pelo usuário no início da fase de aprendizado. A generalização acontece via expansão de uma ou de mais dimensões de um hiperretângulo, à medida que novos exemplos de treinamento vão sendo apresentados ao sistema. A escolha de qual hiperretângulo generalizar depende de uma métrica de distância. Geralmente quando os atributos têm valores *crisp* a métrica adotada é a

distância euclidiana ponderada, ponto-a-ponto ou ponto-a-hiperretângulo.

O NGE inicializa o processo de aprendizado escolhendo randomicamente um determinado número de sementes que são consideradas hiperretângulos triviais e constituem a representação inicial do conceito. Então, para cada novo exemplo de treinamento  $E_{\text{novo}}$ , o NGE busca entre todos os hiperretângulos construídos até então, o mais 'próximo' de  $E_{\text{novo}}$ , chamado de  $H_{\text{próximo1}}$  e o segundo mais 'próximo',  $H_{\text{próximo2}}$ . Esses são os candidatos a serem generalizados. Se  $E_{\text{novo}}$  e  $H_{\text{próximo1}}$  têm a mesma classe,  $H_{\text{próximo1}}$  é expandido de maneira a incluir  $E_{\text{novo}}$ , num processo chamado generalização; caso contrário, a comparação de classes vai acontecer entre  $E_{\text{novo}}$  e  $H_{\text{próximo2}}$ . Se eles tiverem a mesma classe, o NGE irá especializar  $H_{\text{próximo1}}$  reduzindo seu tamanho, de maneira que se distancie de  $E_{\text{novo}}$ .  $H_{\text{próximo2}}$  se torna então o mais próximo de  $E_{\text{novo}}$  ao longo daquela dimensão e é 'expandido' de maneira a 'absorver'  $E_{\text{novo}}$ . Se a classe de  $E_{\text{novo}}$  diferir tanto da classe de  $H_{\text{próximo1}}$  quanto da de  $H_{\text{próximo2}}$ ,  $E_{\text{novo}}$  se torna um novo exemplar, assumindo a forma de um hiperretângulo trivial.

O ajuste de peso é adotado pelo NGE como uma forma de reforçar a relevância tanto de atributos quanto de exemplares, no processo de classificação. Esse reforço pode ser tanto positivo quanto negativo, dependendo da contribuição de cada atributo (ou exemplar) para a classificação correta dos exemplos. Durante o processo de aprendizado, o aumento na relevância de um atributo se reflete pela diminuição do peso a ele associado e vice-versa. Uma política de pesos semelhante é adotada para exemplares. O algoritmo FuGES discutido na próxima seção adota apenas pesos associados a exemplares e, de forma contrária ao NGE, assume que quanto maior o peso do exemplar maior sua relevância na expressão do conceito.

### 3 O Algoritmo FuGES

O FuGES é uma versão do algoritmo NGE, adequada para aprendizado em domínios *fuzzy*. Uma vez que trabalham com representações completamente distintas, a diferença entre esses dois algoritmos está, principalmente, nas funções usadas para a escolha do exemplar mais próximo do novo exemplo e na generalização de um exemplar, usando o novo exemplo.

No ambiente de aprendizado em domínios *fuzzy*, cada exemplo do conjunto de treinamento é descrito por um vetor de pares *atributo/número\_real\_fuzzy* e uma classe *crisp* associada. Cada número real  $a_0$  do conjunto de treinamento é representado por uma função de pertinência triangular, como mostra a Fig. 1(a).

Durante o processo de aprendizado e após o processo de generalização ter acontecido, a tendência

é que números reais *fuzzy* sejam representados por intervalos *fuzzy* com uma função de pertinência no formato trapezoidal, como mostra a Fig. 1(b). O FuGES consiste de duas fases, a fase de aprendizado e a fase de classificação.

#### 3.1 Fase de Aprendizado

Esta é a fase em que um processo de aprendizado indutivo induz a expressão do conceito. O FuGES inicializa o processo de aprendizado selecionando aleatoriamente a partir do conjunto de treinamento, um determinado número de sementes e as transforma em exemplares *fuzzy* (triviais). A fase de aprendizado do FuGES consiste de dois passos consecutivos: a escolha do melhor exemplar e, então, sua generalização.

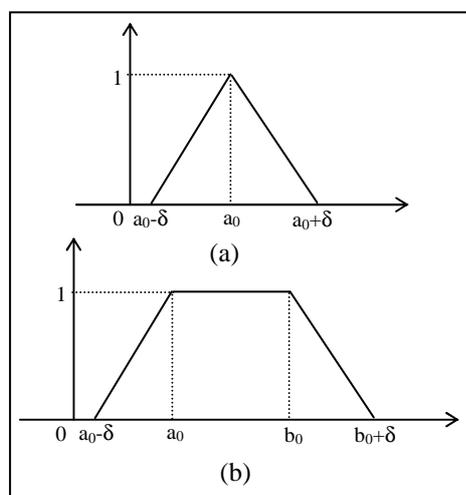


Figura 1. Possíveis representações de números *fuzzy* em FuGES

#### a) Escolha do Melhor Exemplar

No contexto deste algoritmo, a escolha do 'melhor' exemplar significa a escolha do exemplar 'mais próximo' do exemplo de treinamento, onde 'mais próximo' é entendido como o 'mais similar'. O FuGES usa como métrica de similaridade uma extensão do conceito de possibilidade entre conjuntos *fuzzy*. A definição de possibilidade, como apresentada em (Klir, 1995) é:

$$\text{possibilidade}(A,B) = \max_x (A \wedge B) \quad (1)$$

onde  $\wedge$  é implementado como o operador min. A Fig. 2 ilustra a medida de possibilidade para os conjuntos *fuzzy* A e B, definidos no conjunto  $\{0,1,2,3,4,5,6\}$  e representados por suas funções de pertinência. No exemplo da figura,  $\text{possibilidade}(A,B) = 0.3$ .

Neste trabalho optou-se por uma métrica de similaridade que estendesse o conceito de possibilidade, de maneira a poder também avaliar proximidade quando as funções de pertinência, relativas aos dois valores reais *fuzzy*, fossem disjuntas.

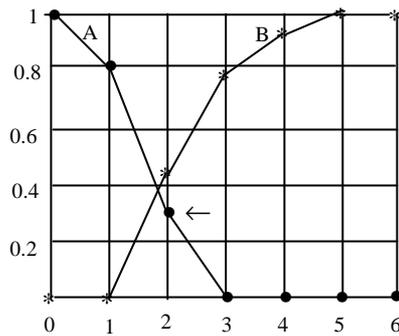


Figura 2. Conjuntos *fuzzy* A e B e o valor da similaridade entre eles.

Inicialmente, devido à própria definição de possibilidade, quando funções de pertinência eram disjuntas, o FuGES considerava o valor de sua similaridade como zero. Entretanto verificou-se empiricamente que esta era uma medida bastante drástica, que acabava descartando um grande número de exemplares-candidatos e que também não diferenciava entre funções de pertinência que, embora disjuntas, tinham diferentes 'proximidades'. Optou-se então por estender o conceito de possibilidade, para evitar este problema. Esta extensão é tratada logo a seguir.

Assim, dado um novo exemplo de treinamento  $E_{\text{novo}}$ , o FuGES calcula o grau de similaridade entre  $E_{\text{novo}}$  e todos os exemplares disponíveis H, construídos até então, com o objetivo de escolher os dois mais próximos de  $E_{\text{novo}}$ .

A primeira vez que esse passo do algoritmo é executado, os únicos exemplares disponíveis são os exemplares triviais, o que significa que as funções de pertinência representando os valores de atributo em ambos, exemplo de treinamento e exemplar, têm o formato triangular. Três possíveis situações podem ocorrer quando da avaliação da similaridade entre um novo exemplo e um exemplar, com relação ao valor de um determinado atributo, representado em ambos (exemplo e exemplar), por funções de pertinência.

A Fig. 3 mostra a primeira situação, quando as funções de pertinência relativas ao valor de um determinado atributo em ambos, exemplo e exemplar, se interceptam. Esta situação tem duas possibilidades, dependendo da representação da função de pertinência. A Fig. 3(a) exibe a situação em que ambas as funções de pertinência são triangulares e a Fig. 3(b) exibe a situação em que a função que descreve o valor do atributo no exemplo é triangular e a que descreve o valor do mesmo atributo, no exemplar, é intervalar. Quando exemplares foram generalizados, alguns de seus valores de atributos passam a ser representados por intervalos *fuzzy*. Nessas duas situações o valor da similaridade é definido aplicando-se a equação 1. A segunda situação acontece quando as funções de pertinência coincidem, o que torna o valor da similaridade entre elas igual a 1.

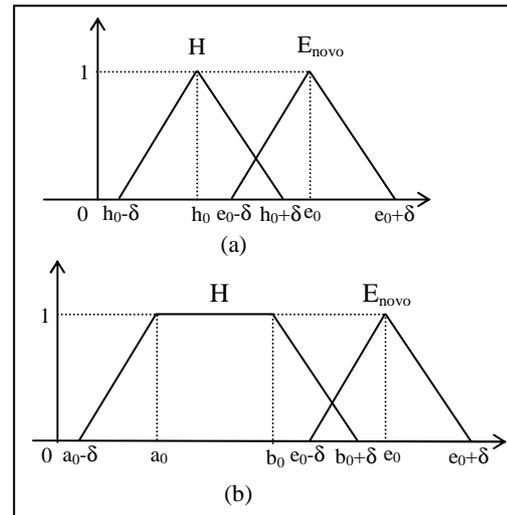


Figura 3. Determinação da medida de similaridade entre dois valores de atributos representados por suas funções de pertinência.

A terceira situação ocorre quando as funções de pertinência que representam os valores associados ao atributo, são disjuntas. Como comentado anteriormente, para esta situação optou-se por estender o conceito de possibilidade. Para tanto, decidiu-se por permitir uma distância máxima ( $\tau$ ) entre as funções de pertinência disjuntas, onde apenas para distâncias maiores que  $\tau$ , a similaridade entre as funções de pertinência é definida como zero. Para funções de pertinência distantes até no máximo  $\tau$ , o FuGES considera um valor positivo pequeno de proximidade, que é inversamente proporcional à distância entre elas (Fig. 4).

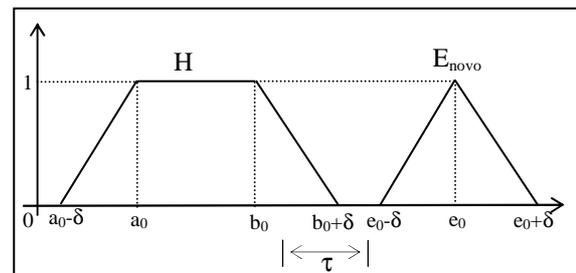


Figura 4. Medida de similaridade estendida entre dois valores de atributos representados por suas funções de pertinência disjuntas.

O próximo passo para o cálculo da similaridade entre  $E_{\text{novo}}$  e H é combinar todas as similaridades individuais atributo-a-atributo em um único número. Tal número é chamado similaridade atributo-a-atributo (saa) entre  $E_{\text{novo}}$  e H e definido como:

$$\text{saa}(E_{\text{novo}}, H) = \frac{\sum_{i=1}^n \text{possibilidade}(\text{val\_at}_i(E_{\text{novo}}), \text{val\_at}_i(H))}{n} \quad (2)$$

Uma vez obtido, o valor saa é ponderado pelo peso do exemplar, gerando a similaridade total ponderada (stp), dada por:

$$\text{stp}(E_{\text{novo}}, H) = \text{saa}(E_{\text{novo}}, H) * \text{peso}_H \quad (3)$$

Este processo é repetido para cada exemplar existente. Usando os valores obtidos de  $stp$ , o FuGES define o primeiro e o segundo exemplares mais similares a  $E_{\text{nov}}$ , identificados como  $H_{\text{próximo1}}$  e  $H_{\text{próximo2}}$ , e inicializa o passo de generalização.

A versão FuGES assume que quanto maior o peso associado a um exemplar, mais relevante é o papel desse exemplar na expressão do conceito. Pesos são dinamicamente modificados durante a fase de aprendizado. Quando  $H_{\text{próximo1}}$  classifica o exemplo corretamente e, conseqüentemente,  $H_{\text{próximo2}}$  não é usado, apenas  $H_{\text{próximo1}}$  tem seu peso atualizado. A estratégia de peso implementada é a sugerida em (Salzberg, 1991), a qual atualiza os pesos de acordo com as regras  $\omega_i \leftarrow \omega_i(1 + \Delta_f)$  (ou  $\omega_i \leftarrow \omega_i(1 - \Delta_f)$ ), onde  $\Delta_f$  é uma constante global de ajuste (escolhida como 0.05 para os experimentos) e  $\omega_i$  o peso associado ao exemplar  $H_i$ . Os pesos de todos os exemplares são inicializados com 1.

### b) Generalização do Exemplar

O algoritmo FuGES se comporta exatamente como o algoritmo original NGE, com relação à escolha de qual exemplar generalizar,  $H_{\text{próximo1}}$  ou  $H_{\text{próximo2}}$ . Dependendo do resultado da comparação entre as classes (do novo exemplo e do exemplar),  $H_{\text{próximo1}}$  ou  $H_{\text{próximo2}}$  pode ser generalizado ou, então,  $E_{\text{nov}}$  pode se tornar um novo exemplar (trivial).

O processo de generalização de um exemplar  $H$  (representando  $H_{\text{próximo1}}$  ou  $H_{\text{próximo2}}$ ) usando um exemplo  $E_{\text{nov}}$  pode ser descrito como uma absorção de  $E_{\text{nov}}$  por  $H$ , a qual é realizada via 'extensão' dos limites do exemplar, atributo-a-atributo, de maneira a que este possa incluir o exemplo.

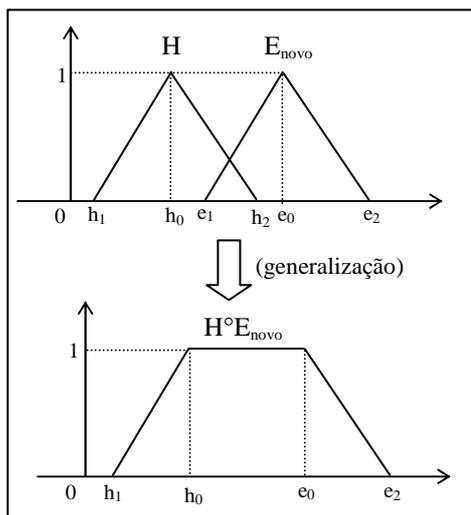


Figura 5. Operador de generalização ( $^{\circ}$ ) entre duas funções de pertinência triangulares.

Dependendo do formato da função de pertinência que representa o valor do atributo no exemplar (triangular ou trapezoidal), dois tipos diferentes de operação de generalização (identificadas pelos operadores  $^{\circ}$  e  $\bullet$  respectivamente) podem acontecer e são mostradas nas figuras 5 e 7. Os algoritmos referentes

às operações  $^{\circ}$  e  $\bullet$  são apresentados nas figuras 6 e 9 respectivamente.

Quando os atributos são disjuntos, o FuGES implementa (via operador  $\Omega$ ), um processo de generalização confinado, de maneira a generalizar o atributo o mínimo possível, no exemplar. O valor  $\delta$  é usado como limite para a 'extensão' dos limites do exemplar (ver figuras 8 e 10).

```
% definição do operador ° entre dois
% conjuntos fuzzy triangulares H e E_novo
% situação onde e0 > h0 e e1 < h2
h1 ← h0 - δ;
h2 ← h0 + δ;
e1 ← e0 - δ;
e2 ← e0 + δ;
if x < h0 then μ_{H°E_novo}(x) = x + y(h1 - h0) - h1
if x > e0 then μ_{H°E_novo}(x) = x - y(e0 - e2) - e2
if h0 ≤ x ≤ e0 then μ_{H°E_novo}(x) = 1
```

Figura 6. Algoritmo para o cálculo do operador  $^{\circ}$ .

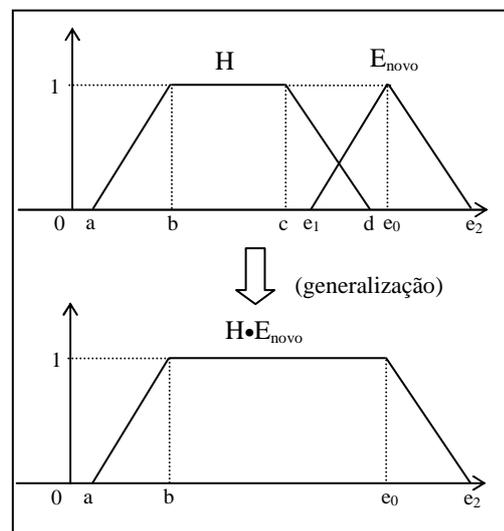


Figura 7. Operador de generalização ( $\bullet$ ) entre uma função de pertinência intervalar e uma triangular.

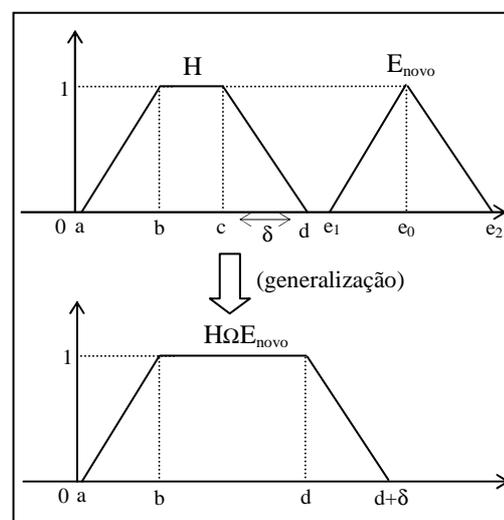


Figura 8. Operador de generalização ( $\Omega$ ) entre atributos disjuntos.

```

% definição do operador • entre dois conjuntos
% fuzzy H (trapezoidal) e Enov (triangular)
% situação onde e0>c (c > b) e e1<d
e1 ← e0 - δ;
e2 ← e0 + δ;
if x < b then μH•Enov(x) = x + y(a - b) - a
if x > e0 then μH•Enov(x) = x - y(e0 - e2) - e2
if b ≤ x ≤ e0 then μH•Enov(x) = 1

```

Figura 9. Algoritmo para o cálculo do operador •.

```

% definição do operador Ω entre dois conjuntos
% fuzzy H (trapezoidal) e Enov (triangular) disjuntos,
% situação onde e0>c (c > b), e1>d e (e1-d)≤τ
e1 ← e0 - δ;
e2 ← e0 + δ;
if x < b then μHΩEnov(x) = x + y(a - b) - a
if x > d then μHΩEnov(x) = x - y(d - (d+δ)) - (d+δ)
if b ≤ x ≤ d then μHΩEnov(x) = 1

```

Figura 10. Algoritmo para o cálculo do operador Ω.

### 3.2 Fase de Classificação

Uma vez terminada a fase de aprendizado, o conjunto de exemplares *fuzzy* obtidos constitui a expressão do conceito aprendido e pode ser usado, a seguir, para a classificação de novos exemplos. A classificação de um novo exemplo é feita medindo a proximidade deste exemplo a cada um dos exemplares existentes. A classe do exemplar mais próximo do exemplo é atribuída ao exemplo.

## 4 O Sistema de Classificação FuGES

O protótipo do sistema FuGES foi desenvolvido usando uma abordagem orientada a objetos e codificado na linguagem C++ (compilador g++), executado sob a plataforma Linux.

O sistema inicialmente espera intervenção do usuário, que deve informar: o nome do arquivo que contém os dados do conjunto de treinamento; o número de atributos que descrevem os exemplos de treinamento/teste; o número de sementes; o valor da variável  $\delta$ , que define o formato triangular da função de pertinência e o valor de  $\tau$  (valor máximo possível para que a métrica de similaridade entre duas funções de pertinência disjuntas não seja zero).

O FuGES foi usado em alguns domínios de conhecimento com o objetivo de experimentar o sistema, determinar seu alcance e calibrar alguns de seus parâmetros. A escolha dos domínios foi determinada principalmente pelo tipo dos atributos (atributos numéricos). Uma breve descrição quantitativa desses domínios está mostrada na Tabela 1 e qualquer informação adicional pode ser encontrada no UCI Repository (Merz, 1998). Como o domínio *Excipients* não está descrito na referência acima, detalhes sobre esse domínio são dados a seguir.

O domínio *Excipients* é um domínio farmacêutico contendo 170 instâncias relacionadas à produção industrial de comprimidos. O tipo do excipiente empregado na confecção de comprimidos tem um papel fundamental na liberação de seu componente ativo. As instâncias deste domínio são descritas por 14 características: densidade *bulk*, densidade *tapped*, compressibilidade, ângulo de repouso, preenchimento relativo, fluidez, tamanho da partícula (com porcentagem de retenção de 840, 420, 250, 177, 149 e < 149  $\mu$ m), umidade e índice de Hausner.

As 170 instâncias estão distribuídas em 17 classes diferentes (10 relativas a cada excipiente). Todas as medições foram obtidas em um laboratório. As 17 classes estão distribuídas em 9 tipos diferentes de Celulose microcristalina (Avicel PH-102®, Avicel PH-200®, Microcel MC-101®, Microcel MC-102®, Microcel MC-250®, Microcel MC-301®, Microcel MC-302®, Vivapur type 101® and Vivapur type 102®), Celactose, Lactose, Encompress®, Ludipress®, Manitol em pó e granular, Cloreto de Sódio e Amido de Milho. Este domínio foi previamente utilizado em aprendizado neural-*fuzzy* e detalhes dos experimentos conduzidos podem ser encontrados em (Ramer, 2000).

Tabela 1. Número de exemplos (Orig=conj. original; Tre=conj. treinamento; Tes=conj. teste); At=número de atributos; Clas=número de classes.

Domínio	Orig	Tre	Tes	At	Clas
Iris	150	105	45	4	3
<i>Excipients</i>	170	119	51	14	17
<i>Wine</i>	178	125	53	13	3

Os principais objetivos dos experimentos foram detectar quão bem o paradigma NGE lida com números *fuzzy* e quão bem a medida de possibilidade (estendida) pode ser usada como uma métrica que reflete proximidade. Alguns experimentos iniciais foram conduzidos com os três domínios escolhidos com o objetivo de encontrar os valores das variáveis  $\delta$  e  $\tau$  que melhor definissem cada domínio, bem como evidenciar o melhor número de sementes. Os valores obtidos são mostrados na Tabela 2. O número de sementes usadas em todos os experimentos foi 3. Esta escolha não foi arbitrária mas sim resultado de vários experimentos realizados em diversos domínios. Percebeu-se que quanto mais próximo o número de sementes for do número de classes existentes no domínio, melhor o desempenho do sistema com relação ao número de exemplares induzidos. Um número alto de sementes eventualmente produz um número excessivo de exemplares na expressão final do conceito.

Para os testes de classificação dos domínios, foram criados 10 diferentes grupos de arquivos (10 arquivos de treinamento e os correspondentes 10 arquivos de teste). Cada arquivo de treinamento foi criado com 70% das instâncias do arquivo original, e os 30% restantes compõem o arquivo de teste.

Tabela 2. Valores aplicados às variáveis  $\delta$  e  $\tau$  nos experimentos.

Domínio	$\delta$	$\tau$
Iris	0.5	1
Excipients	0.8	3
Wine	0.2	3

O FuGES foi então usado com cada um dos conjuntos de treinamento, para a indução do conceito e, a seguir, a expressão induzida foi testada com o conjunto de teste correspondente. O desempenho do sistema foi, então, avaliado considerando-se a média dos valores obtidos para cada um dos 10 arquivos de teste. Na Tabela 3 e Tabela 4 abaixo,  $\#H$  corresponde à média de exemplares gerados,  $\#fuzzy$  corresponde à média do mínimo de generalizações que ocorreram durante o processo de aprendizado,  $\#Corr$  é o desempenho médio calculado sobre as 10 execuções para cada domínio e  $\#nc$  é a média de exemplos de teste que não foram classificados. Os exemplos de teste não classificados são aquelas instâncias do conjunto de teste que têm, pelo menos, um atributo cuja função de pertinência está a uma distância maior que  $\tau$  da função de pertinência relativa ao mesmo atributo, no exemplar.

Como pode ser observado na tabelas 3 e 4, o sistema teve um bom desempenho nos domínios considerados. Apenas 1 exemplo foi não classificado, em trinta execuções do sistema. É importante notar, entretanto, que apenas no domínio *Iris* o sistema induziu um hiperretângulo por classe. Nos domínios *Excipients* e *Wine*, a média foi de 1.41 e 6.47 respectivamente. O domínio *Excipients* quando usado no contexto de aprendizado indutivo de regras, via sistema CN2, induziu uma média de 1 regra para cada classe de excipiente (Nicoletti, 1999).

Tabela 3. Média/Desvio Padrão

Domínio	$\#H$	$\#Corr$	$\%Corr$
Iris	3,6/0,97	38,5/3,37	85,5
Excipients	24,0/1,49	40,2/3,42	78,82
Wine	19,4/2,22	47,8/2,65	90,19

Tabela 4. Novos conjuntos *fuzzy* criados

Domínio	$\#fuzzy$	$\#nc$
Iris	64	0
Excipients	204,1	0
Wine	633,1	1

É importante lembrar aqui que os valores obtidos com o sistema ( $\%Corr$  na Tabela 3) nos domínios utilizados (*iris*, principalmente) está abaixo daqueles obtidos por sistemas de Aprendizado de Máquina (por exemplo, CN2 (Clark, 1989)) que lidam com o dado 'crisp'. Isto em absoluto inviabiliza a proposta/sistema apresentados e discutidos neste artigo uma vez que os domínios utilizados não são naturalmente *fuzzy* e foram tratados como tais em virtude da dificuldade de se encontrar dados originalmente coletados como *fuzzy*.

## 5 Conclusões

Existem ainda muitas variáveis a serem investigadas, bem como avaliações a serem conduzidas. O trabalho descrito neste artigo terá continuidade com a experimentação do FuGES em outros domínios de conhecimento, buscando refinar tanto a representação *fuzzy* utilizada, quanto os operadores propostos.

Uma outra linha de pesquisa interessante a investigar é o impacto da escolha do valor de  $\tau$  quando do cálculo da similaridade estendida entre atributos bem como no processo de classificação após o aprendizado.

Pretende-se também introduzir um processo de "defuzzificação" ao final do aprendizado para viabilizar, entre outros, uma avaliação dos resultados do FuGES versus NGE.

## Referências Bibliográficas

- Clark, P. and Niblett, T. (1989). The CN2 induction algorithm. *Machine Learning*, Vol. 3, pp. 261–283.
- Cover, T. and Hart, P. (1967). Nearest neighbour pattern classification. *IEEE Transactions on Information Theory*, Vol. 13, pp. 21–27.
- Klir, G.J. and Yuan, B. (1995). *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice-Hall.
- Merz, C. J. and Murphy, P.M. (1998). *UCI Repository of Machine Learning Databases* [<http://www.ics.uci.edu/~mllearn/MLRepository.html>]. Irvine, CA: University of California, Department of Information and Computer Science.
- Nicoletti, M.A. and Nicoletti, M. C. (1999). Aprendizado indutivo simbólico de categorias de excipientes em domínio farmacêutico. *II Simposio de Informatica y Salud*, Argentina, pp 1–10.
- Nicoletti, M.C. and Santos, F.O. (1996). Learning fuzzy exemplars through a fuzzified nested generalized exemplar theory. *Proceedings of the European Workshop on Fuzzy Decision Analysis for Management, Planning and Optimization*, Dortmund, Germany, pp 140–145.
- Ramer, A.; Nicoletti, M.C. and Sung, S. Y. (2000). *Evaluative studies of fuzzy knowledge discovery through NF systems*. Chapter 19. pp 387–411, N. Kasabov (ed.), *Future Directions for Intelligent Systems and Information Sciences*, Physica-Verlag.
- Salzberg, S. (1991). A nearest hyperrectangle learning method. *Machine Learning*, Vol. 6, pp 252–276.