FUNDAÇÃO EDSON QUEIROZ

UNIVERSIDADE DE FORTALEZA - UNIFOR

Ricardo Batista Rebouças

Formação incremental de conceitos probabilísticos a partir de observações com atributos discretos e contínuos

Fortaleza

2003

FUNDAÇÃO EDSON QUEIROZ

UNIVERSIDADE DE FORTALEZA - UNIFOR

Ricardo Batista Rebouças

Formação incremental de conceitos probabilísticos a partir de observações com atributos discretos e contínuos

Dissertação apresentada ao curso de
Mestrado em Informática Aplicada da
Universidade de Fortaleza como requisito para
obtenção de Título de Mestre em Informática.

Orientador: Prof. D.Sc. João José Vasco Furtado

Fortaleza

2003

Formação incremental de conceitos probabilísticos a partir de observações com atributos discretos e contínuos

Banca Examinadora:

_____

Prof. João José Peixoto Furtado, D.Sc.

_____

Prof. Pedro Porfírio Muniz Farias, D.Sc.

_____

Prof. Walfredo Cirne, D.Sc.

Aos meus pais pela inspiração, paciência, carinho e atenção dedicada a mim durante a execução desse trabalho.

Agradecimentos

Muitas foram as pessoas que ajudaram na elaboração e conclusão desse trabalho. Algumas mais diretamente que outras, mas todas aqui lembradas com igual carinho. Devo destacar alguns que presenciaram e participaram de todo o processo da obtenção desse título. Em primeiro lugar Deus, que sempre me iluminava e dava forças nas madrugadas de estudo. Minha família em seguida e meus amigos que também sempre se mostraram presentes, solidários e pacientes na minha jornada.

Meus professores e colegas de mestrado, a quem devo especial gratidão pelas dicas e sugestões a respeito deste trabalho. Ao professor Sérgio Forte ao nos presentear com o Manual de Elaboração de Dissertação de Mestrado, que, sem a menor dúvida, facilitou e orientou por demais meu trabalho de escrita da dissertação. Ao Vasco, meu orientador, professor, chefe e, acima de tudo, amigo, por todo apoio, dedicação e confiança em mim depositada.

Registro aqui, enfim, os meus sinceros agradecimentos a todos.

Abstract

Many real world entities can be represented by a combination of discrete and continuous attributes. In domains with this kind of representation, concept formation systems have a problem due to the use of different evaluation functions for each type of attribute. In this dissertation, this problem is analyzed specifically in probabilistic concept formation systems (PCFS). In such systems, the evaluation function for discrete and continuos attributes has different behavior which results in an unbalanced contribution for each attribute-type evaluation function inside the main evaluation function. Indeed, a bias occurs in hierarchy building, affecting directly the quality of the generated concepts. Basically, this work studies the quality of the generated concepts in terms of its predictability. Since PCFS are generally incremental, they change concept prdicatability for each new entity analyzed. This work describes an approach based on the difference between the individual predictability gain for each attribute type. Indeed, it also favors the creation of concept hierarchies that maximize the individual predictability gain for each attribute. This approch means a way to work around the unbalanced contribution problem in attribute-driven evaluation functions environments. Experiments using the approach presented here have shown higher quality concepts, in terms of predictability, when compared with related works.

Resumo

Muitas entidades do mundo real podem ser representadas através da combinação de atributos discretos e contínuos. O problema básico da formação de conceitos, em domínios com esta representação, deve-se ao uso de diferentes funções de avaliação para cada tipo de atributo. Nesse trabalho, esse problema será analisado especificamente em sistemas de formação de conceitos probabilísticos (SFCP). A análise do problema aponta diferentes comportamentos para as funções de avaliação dos atributos discretos e contínuos. Essa diferença resulta numa participação desbalanceada das funções de avaliação, para cada tipo de atributo, na função de avaliação geral. Isso, conseqüentemente, gera uma tendência na criação da hierarquia, afetando diretamente a qualidade dos conceitos gerados. Basicamente, este trabalho se concentra em estudar a qualidade dos conceitos em termos da capacidade de inferência dos mesmos. Em SFCP, devido o caráter incremental dos mesmos, a cada nova entidade classificada a capacidade de inferência dos conceitos sofre modificações. Essas modificações são de diferente intensidade para cada tipo de atributo. A proposta de solução desse trabalho está baseada no ganho individual de capacidade de inferência para cada tipo de atributo, discretos e contínuos. Assim, busca-se a criação de uma da hierarquia de conceitos que também promova o maior ganho em capacidade de inferência para cada tipo de atributo. A abordagem aqui proposta constitui uma forma de contornar a predominância na função de avaliação geral. Experimentos realizados para medir a qualidade dos conceitos gerados pela proposta apresentada mostraram resultados superiores em comparação com trabalhos similares.

Sumário

Lista de Tabelas

Lista de Figuras

Lista de Equações

Capítulo 1

Introdução

O acúmulo de informações em bases de dados é um fato comum nas mais diversas áreas e o volume dessas informações tornou a extração de conhecimento potencialmente útil uma tarefa não trivial. Uma forma de descobrir conhecimento a partir de bancos de dados é utilizar algoritmos indutivos de aprendizagem automática. Alguns destes algoritmos, ditos não supervisionados, realizam um processo de formação de categorias através da observação gradual de entidades. Esses sistemas são conhecidos como sistemas de formação de conceitos, onde as categorias representam esses conceitos. As entidades que compõem uma categoria são descritas por suas propriedades em termos de atributos e valores associados aos mesmos. Esta descrição denomina-se observação.

Os sistemas de formação de conceitos realizam uma busca heurística, no espaço de todas as estruturas conceituais possíveis, à procura da *melhor* (segundo um critério preestabelecido). Nessa busca, o aspecto fundamental a ser considerado é a função que define o critério para medir a qualidade das estruturas geradas e assim, escolher a melhor dentre elas. Na literatura, essa função é geralmente chamada de função de avaliação heurística.

Dentre os sistemas de formação de conceitos, citamos uma classe particular e muito estudada que usa uma representação do conhecimento chamada conceitos probabilísticos. Conceitos probabilísticos identificam o conhecimento através de uma hierarquia onde os níveis mais altos representam conceitos mais genéricos enquanto os mais baixos representam conceitos mais específicos. Um conceito probabilístico representa uma categoria de entidades através da generalização das propriedades destas entidades. Esta generalização é realizada com o auxílio da representação de uma probabilidade condicional associada a cada uma das propriedades. As propriedades, da mesma forma que uma observação, são pares de atributo/valor e, a probabilidade condicional é a probabilidade de que uma entidade, fazendo parte da categoria representada pelo conceito, possua um determinado valor para um atributo.

Hierarquias de conceitos probabilísticos denotam ainda um meio de classificar entidades. De forma que, entidades representadas pelo mesmo conceito possuem propriedades semelhantes. Ou seja, classificar uma entidade numa hierarquia de conceitos também constitui uma maneira de inferir valores desconhecidos para essa entidade, usando como base o conceito em que esta foi classificada. Estas inferências são efetuadas com base nas probabilidades condicionais das propriedades dos conceitos. Essas probabilidades são fundamentais para determinar a capacidade de inferência do conceito.

Algumas abordagens tratam entidades como sendo representadas exclusivamente por atributos discretos ou contínuos. No entanto, no mundo real, a maioria das entidades é mais bem representada pela combinação desses tipos de atributos. Por exemplo, analisando dados geológicos, percebemos que atributos como *idade, porosidade e permeabilidade* são informações numéricas, enquanto *tipos de rocha* e *estruturas cristalinas* são informações nominais.

O termo “numérico”, usado para denominar um atributo, também pode ser identificado como “contínuo”, por causa da natureza estatística do tipo de dado, indicando sua “continuidade” para com os demais valores do atributo. Da mesma forma, atributos ditos “nominais” podem ainda ser denominados “discretos”. Ou seja, que não estão em continuidade com outros valores do atributo. Nesse trabalho, será utilizado o termo “contínuo” para atributos que representam informações numéricas, enquanto o termo “discreto” será usado para identificar atributos que representam informações nominais.

Para uma maior aplicabilidade da aquisição de conhecimento através da formação de conceitos, faz-se importante uma abordagem que trabalhe com atributos discretos e contínuos em conjunto. As soluções dadas em propostas anteriores, para o tratamento de entidades com atributos, em geral, encaixam-se em uma das seguintes categorias:

- Representação de atributos discretos através de números com aplicação de uma abordagem que trabalhe somente com atributos contínuos;
- Discretização de atributos contínuos, e utilização de um método que use somente atributos discretos;

- Criação de uma função de avaliação heurística para cada tipo de atributo, e utilização do resultado de ambas em uma função de avaliação geral.

Representar atributos discretos através de números, na maioria das vezes, não faz sentido, pois abordagens que tratam atributos contínuos fazem uso de medidas de distância para determinar a proximidade entre entidades. Intuitivamente, não existe distância entre dois valores discretos de um atributo.

No caso da discretização de atributos contínuos, em geral, existe a perda de informações importantes contida nos valores. Podemos citar, principalmente, a distância relativa ou absoluta entre os valores contínuos discretizados.

Este trabalho baseia-se em abordagens que utilizam uma função de avaliação para cada tipo de atributo, pois consistem em uma maneira coerente de tratamento das particularidades dos atributos contínuos e discretos.

1.1 Objetivos

Este trabalho tem como objetivo propor uma abordagem para formação de conceitos probabilísticos em domínios com entidades representadas por atributos discretos e contínuos. Para isso, serão analisadas, comparadas e discutidas soluções anteriores, destacando seus benefícios e problemas persistentes.

A abordagem aqui proposta deverá ser capaz de criar hierarquias de conceitos onde a predição de valores não é prejudicada pelo uso de tipos distintos de atributos. Esta abordagem deve ainda exigir o mínimo de interferência do usuário, pois sendo um algoritmo de aprendizagem automática, espera-se que o conhecimento seja extraído exclusivamente dos dados.

1.2 Hipóteses

Acredita-se que os algoritmos de formação de conceitos probabilísticos existentes apresentam-se deficientes na aquisição de conhecimento em domínios com entidades representadas por atributos discretos e contínuos.

Devido à própria natureza dos diferentes tipos de atributos, a tarefa de comparar informações usando uma mesma medida torna-se difícil. O uso de funções de avaliação distintas, pelo fato de considerar as peculiaridades inerentes a cada tipo de atributo, termina por utilizar diferentes medidas para cada tipo de atributo.

Essa prática causa um desequilíbrio na função de avaliação geral. Portanto, a existência de predominância por um dos tipos de atributos, causada pelo desequilíbrio na função de avaliação geral, é um fato esperado.

Essa predominância, por vezes, tem sido amenizada com o uso de parâmetros externos fornecidos pelo usuário. Parâmetros externos adicionam informações ao processo de aquisição de conhecimento, muitas vezes, alheias às entidades, o que não é um comportamento desejável para abordagens de aprendizagem automática.

Considerar os diferentes comportamentos dos atributos contínuos e discretos, assim como a intensidade da contribuição de cada um na função de avaliação geral constitui uma solução eficaz para o problema da predominância.

1.3 Relevância

A evolução dos meios de comunicação de dados tem aumentado a velocidade de acesso aos mais diversos tipos de informações. Paralelamente a essa evolução, percebeu-se também o aumento na capacidade dos dispositivos de armazenamento de dados. Em outras palavras, nunca foi tão fácil acumular dados.

Em vista do volume de dados acumulados, que hoje facilmente ultrapassa a barreira dos *gigabytes*, métodos de consolidação de dados se fazem extremamente importantes. Uma das maneiras de consolidar dados é convertendo-os em conhecimento, o que pode ser feito através de técnicas de *Data Mining*. A aquisição automática de conhecimento, particularmente a formação automática de conceitos constitui uma forma de *Data Mining* amplamente utilizada.

A aquisição automática de conhecimento, atualmente, exige que o conhecimento extraído dos dados seja identificado por um especialista nos dados em questão. Assim, a representação do conhecimento se torna de essencial importância para que a análise do especialista tenha sucesso com maior rapidez. Nesse ponto, a representação de conhecimento usando conceitos probabilísticos constitui uma forma bastante intuitiva e relativamente fácil de explicação do significado do conhecimento.

Uma abordagem que melhore o tratamento de atributos mistos para sistemas de formação de conceitos probabilísticos contribui na qualidade dos conceitos gerados em termos de capacidade de inferência dos mesmos.

1.4 Estrutura do trabalho

O trabalho aqui apresentado foi organizado da seguinte maneira. O capítulo a seguir irá analisar as abordagens correlatas aos objetivos desse trabalho. Os problemas encontrados através da análise bibliográfica serão detalhados no capítulo 3. O capítulo 4 apresenta uma proposta para os problemas identificados. Essa proposta será avaliada, comparada com abordagens correlatas e apresentados resultados no capítulo 5. Enfim, o trabalho é concluído no capítulo 6 onde ainda são apresentadas suas limitações e sugestões para trabalhos futuros.

2 Estado da Arte

2.1 Aquisição de conhecimento e mineração de dados

Uma forma de descobrir conhecimento a partir de bancos de dados é utilizar algoritmos indutivos de aprendizagem automática. Alguns destes algoritmos, ditos não supervisionados, realizam um processo de formação de categorias através do agrupamento conceitual de entidades(*conceptual clustering* [Michalski 83]). Trata-se aqui de agrupar em categorias entidades similares do mundo real em função de suas propriedades. Particularmente, os sistemas incrementais, são conhecidos como sistemas de formação de conceitos [Fisher 87], [Lebowitz 87], pois as categorias são representadas por conceitos que explicam o agrupamento das entidades.

Esses sistemas de formação de conceitos realizam uma busca heurística, no espaço de todas as hierarquias de conceitos possíveis, da *melhor* (segundo um critério pré-estabelecido) estrutura conceitual a gerar (geralmente uma hierarquia). Nessa busca o aspecto fundamental a ser considerado é a função que define o critério para medir a qualidade das hierarquias geradas e assim, escolher a melhor dentre elas.

Citamos, em particular uma classe de algoritmos que usa uma representação do conhecimento, definida originalmente por Smith e Medin [Smith, 81] no contexto da psicologia cognitiva, chamada conceitos probabilísticos. A representação de um conceito probabilístico consiste de uma lista de pares de atributo/valor onde cada par possui uma probabilidade associada. Esta probabilidade, também chamada de predictabilidade (*predictability*), é a probabilidade condicional de que uma observação possua um atributo *A* com um valor *V*, dado que esta observação pertence ao conceito *C*, $P(A=V|C)$.

O uso de uma bem definida função para mensurar a qualidade de uma hierarquia de conceitos representou um avanço para a área de formação automática de conceitos. Assim como o uso de conceitos probabilísticos na representação dos conceitos facilitou o entendimento do método de *clustering* utilizado. COBWEB

[Fisher, 87] é uma abordagem que aplica essas idéias em sua implementação. Isso o transformou em representante típico dos Sistemas de Formação de Conceitos Probabilísticos (SFCP). A seguir COBWEB será detalhado.

2.1.1 COBWEB

COBWEB [Fisher, 87] é um algoritmo de formação incremental de conceitos que, embora não se apresente explicitamente como um modelo psicológico, foi fortemente influenciado por pesquisas na área da psicologia cognitiva [Rosch, 75].

Em breves palavras, experimentos com humanos sugerem que alguns conceitos são identificados mais rapidamente que outros e seus nomes utilizados com mais freqüência. Existem ainda evidências que, para um certo conceito, algumas entidades são identificadas mais rapidamente que outras e reconhecidas como melhores exemplos para o conceito. Esses conceitos e exemplos são ditos pertencerem ao "*nível básico*" de conhecimento.

COBWEB, através de uma função de avaliação detalhada a seguir, faz uso de uma heurística de criação de conceitos baseada nessas teorias herdadas da psicologia cognitiva. Isso representou um avanço no estudo da formação automática de conceitos em relação a abordagens anteriores [Feigenbaum, 63] [Feigenbaum, 84] [Lebowitz, 83]. A existência de uma função de avaliação que representa a heurística utilizada tem facilitado o avanço de trabalhos posteriores e motivado a escolha de COBWEB como base para técnicas de formação de conceitos.

2.1.1.1 Representação e organização

COBWEB representa as entidades como uma observação de um conjunto de suas propriedades. Uma propriedade de uma entidade é formada por um atributo da entidade e seu respectivo valor.

Na hierarquia construída por COBWEB, cada nó é um conceito que representa uma categoria do mundo real. Esses conceitos são chamados de conceitos probabilísticos, pois generalizam as entidades contidas na categoria em termos da probabilidade condicional de suas propriedades (par atributo/valor).

Essa hierarquia serve ainda para organizar as entidades em conceitos que, por sua vez, são úteis para prever valores de atributos de uma nova entidade. O

conceito onde uma nova entidade foi classificada, por exemplo, serve como base para inferências sobre valores desconhecidos dessa entidade, uma vez que entidades pertencentes a uma mesma categoria compartilham propriedades semelhantes.

Essa utilização da hierarquia para realizar inferências permite que capacidade de inferência dos conceitos gerados possa servir como métrica para a qualidade da hierarquia construída. Entende-se por capacidade de inferência de um conceito como seu poder de prever corretamente valores desconhecidos de atributos.

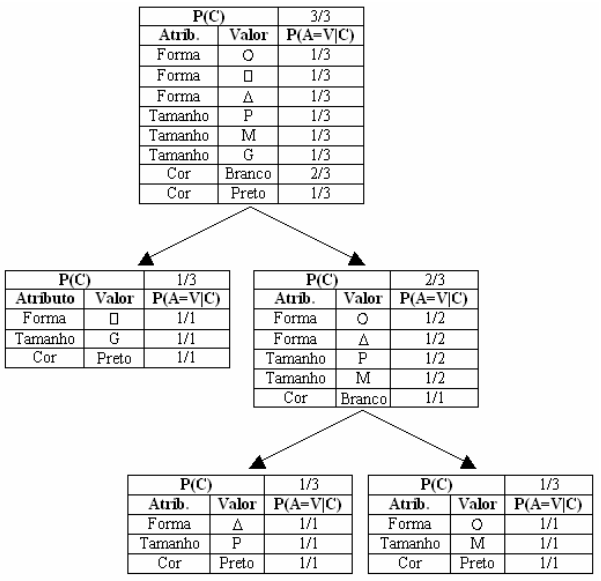

Figura 1: Exemplo de hierarquia de conceitos probabilísticos num domínio de objetos.

A capacidade de inferência para um conceito pode ser descrita em termos da capacidade de inferência dos atributos. Na literatura, os termos *predictiveness* e da *predictability* são usados para identificar dois aspectos dessa capacidade para os atributos. A *predictiveness* do valor *V* de um atributo *A* para uma categoria *C* representa a probabilidade condicional que uma observação *i* seja um membro de *C*, dado que *i* possui o atributo *A* com o valor *V*. Ou seja, *P(C|A=V)*. Analogamente a *predictiveness*, a *predictability* do valor *V* de um atributo *A* para uma categoria *C* é definida como a probabilidade condicional que uma observação *i* possua um valor *V* para o atributo *A*, dado que *i* é um membro de *C*. Em outras palavras, *P(A=V|C)*. A *predictability* representa a quantidade esperada de valores de atributos que podem ser inferidos corretamente para um membro qualquer de uma determinada categoria.

A figura 1 exemplifica uma hierarquia de conceitos, para um domínio de objetos, com as probabilidades associadas aos conceitos e aos valores dos atributos. Onde os níveis mais baixos na hierarquia representam as categorias mais específicas enquanto os mais altos representam as categorias mais genéricas. Em outras palavras, as categorias mais baixas estão ligadas à categoria superior através de uma relação de especialização ou do tipo “é um”.

2.1.1.2 Classificação e aprendizado

A forma de classificação das entidades e o aprendizado em COBWEB se confundem, pois à medida que uma entidade é classificada, percorrendo a hierarquia da categoria mais genérica à categoria mais específica, seus conceitos vão sendo atualizados.

O sistema inicia a hierarquia com a primeira entidade analisada, baseando as informações do primeiro conceito nos valores dos atributos dessa entidade. Quando a segunda entidade é analisada, COBWEB generaliza as propriedades do primeiro conceito em termos das propriedades da primeira e da segunda entidade. Em seguida cria para o primeiro conceito dois nós filhos representando a primeira e a segunda entidade.

Desse ponto em diante, o processo de classificação de novas entidades consiste em COBWEB considerar o encaixe da nova entidade em cada conceito filho do nível hierárquico em questão. Esses conceitos filhos desse nível hierárquico formam uma partição da hierarquia que tem como raiz o conceito que as generaliza.

Além do encaixe, COBWEB ainda considera a criação de um novo conceito, no mesmo nível hierárquico, formado somente pela nova entidade. Cada operação, seja ela de encaixe ou de novo conceito, fará parte de um conjunto de possíveis operações a serem realizadas na hierarquia existente. Uma possível operação consiste de uma partição que terá a qualidade medida através de uma função de avaliação. Essa função de avaliação será descrita na próxima seção. Somente a operação que possuir melhor qualidade será efetivamente aplicada na hierarquia.

A realização da melhor operação significa ainda a atualização das probabilidades condicionais dos valores de atributos das categorias por onde a nova entidade foi classificada, ou seja, a categoria raiz da partição sendo analisado e a

categoria da partição escolhida como melhor encaixe, quando a melhor operação for de encaixe.

A escolha da operação de criação de um novo conceito, intuitivamente, significa que a nova entidade é suficientemente diferente dos conceitos existentes e, portanto, o processo de classificação está encerrado. Caso a melhor operação seja de encaixe em um conceito existente, o processo de classificação continua percorrendo a hierarquia, usando o conceito de melhor encaixe como raiz da próxima partição onde serão realizadas a mesma escolha da melhor operação. Esse processo é realizado até que sejam explorados todos os conceitos.

Embora o processo descrito acima forneça recursos para a construção de hierarquias de conceitos, este fica sujeito a ordem de apresentação das observações ao algoritmo. Ou seja, pode haver diferentes estruturas hierárquicas caso as observações sejam submetidas ao algoritmo em ordens diferentes. Para minimizar esse problema, COBWEB possui ainda duas outras operações: junção e divisão. Cada uma irá criar uma opção de partição com a respectiva qualidade para ser comparada com as opções de encaixe e criação de um novo conceito.

A operação de junção irá aglutinar os dois conceitos escolhidos como melhores opções de encaixe da nova entidade. Ou seja, será criado um novo conceito que irá totalizar as duas melhores opções de encaixe e essas duas opções passarão a ser conceitos filhos do novo conceito criado. Caso essa opção seja escolhida como melhor, o novo conceito criado será o novo conceito raiz para continuidade do processo de classificação. A figura 2 demonstra a operação de junção.

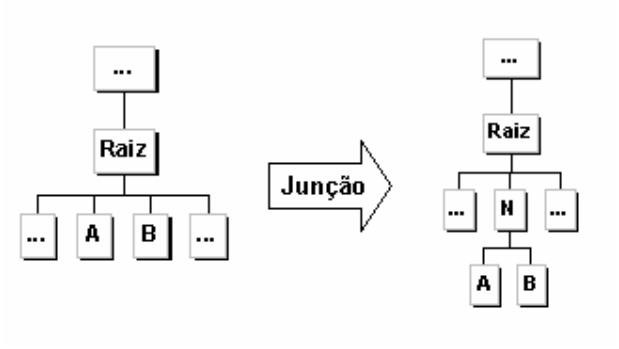

Figura 2: Operação de junção de dois conceitos.

COBWEB também possui a operação inversa da junção, a divisão. A operação de divisão consiste em eliminar o conceito escolhido como melhor opção de encaixe, fazendo com que seus filhos subam um nível hierárquico, de acordo com a figura 3. Com isso, caso um dos *N* conceitos da partição analisada tenha *M* conceitos filhos, a divisão desse conceito fará com que a partição analisada passe a ter *N+M-1* conceitos. Caso essa seja a melhor operação escolhida, a nova observação será submetida novamente ao algoritmo sem alterar o conceito raiz. No entanto, a partição agora terá *N+M-1* conceitos.

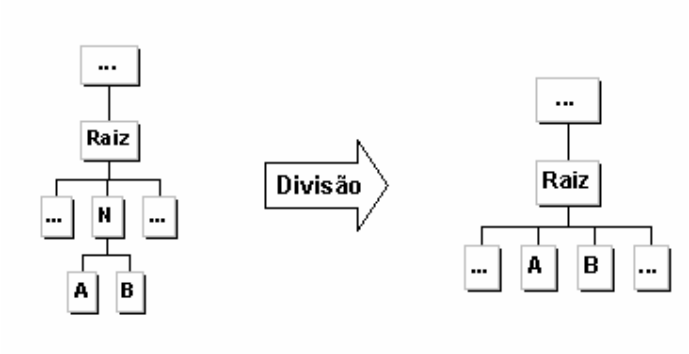

Figura 3: Operação de divisão de um conceito em vários.

A promoção ou rebaixamento em termos de nível hierárquico ajuda a minimizar o problema de ordem de apresentação das entidades ao algoritmo.

2.1.1.3 Função de Avaliação

Na seção anterior foram feitas várias referências à função de avaliação usada por COBWEB, como também anteriormente foi citada a influência da psicologia cognitiva nesse sistema. Essa seção descreverá essa função de avaliação que foi batizada de *Category Utility*.

Category Utility é uma medida que Gluck e Corter [Gluck & Corter, 85] mostraram representar o nível básico de conhecimento encontrado em experimentos da psicologia cognitiva. A função foi derivada por dois caminhos, um através da teoria da informação e outro através da teoria dos jogos.

Category Utility, em linhas gerais, favorece a criação de hierarquias de conceitos que maximizam a capacidade de inferir corretamente informações de um domínio. Com isso, procura maximizar (1) a semelhança de entidades membros de

uma mesmo categoria e, (2) a diferença entre categorias no mesmo nível hierárquico. A medida básica de *Category Utility* assume que os conceitos são, por natureza, descritos em termos de probabilidades.

As duas probabilidades condicionais associadas aos conceitos são (1) a probabilidade de ocorrência de um determinado par atributo/valor, $A_i=V_{ij}$, dado que esse par pertence à categoria C_k , expressa por $P(A_i=V_{ij}|C_k)$, e (2) a probabilidade de ocorrência de uma determinada entidade pertencer a uma categoria dado que esse categoria possui um certo par atributo/valor, expressa por $P(C_k|A_i=V_{ij})$. Combinando essas duas medidas de pares de atributo/valor, pode-se determinar uma medida de qualidade para a hierarquia. A equação 1 demonstra essa combinação.

Equação 1:Combinação de duas probabilidades condicionais aplicadas em COBWEB

$$\sum_{k=1}^{n} \sum_{i} \sum_{j} P(A_i = V_{ij})\ P(C_k \mid A_i = V_{ij})\, P(A_i = V_{ij} \mid C_k)$$

A probabilidade $P(A_i=V_{ij})$ permite dar um peso a valores individuais de atributos, de forma que pares de atributo/valor mais freqüentes sejam mais significativos que aqueles com menor ocorrência. Através de uma derivação usando o método de Bayes, pode-se transformar a equação 1 na forma exibida na equação 2.

Equação 2: Equação inicial de COBWEB

$$\sum_{k=1}^{n} P(C_k) \sum_{i} \sum_{j} P(A_i = V_{ij} \mid C_k)^2$$

Gluck e Corter [Gluck and Corter 85] fizeram uma associação entre os valores computados pela expressão $\sum_i\sum_j P(A=V|C)^2$ e a quantidade valores, para uma determinada categoria, que se pode inferir para um atributo. Essa associação representa uma estratégia de *probabilidade de encaixe*. De forma que, assume-se que é possível inferir um valor de atributo com uma probabilidade de acerto igual a sua probabilidade de ocorrência. Ou seja, pode-se inferir o valor *V* de atributo *A* de

uma entidade que pertence a categoria *C*, com uma probabilidade de acerto de *P(A=V|C).*

Assim, Gluck e Corter definiram *Category Utility* como o aumento da quantidade de valores de atributos que podem ser corretamente inferidos, dados que esses valores pertencem a um conjunto de *n* categorias, sobre essa mesma quantidade sem saber a que categoria pertence. A equação completa de Category Utility esta na equação 3:

Equação 3: Definição de Category Uility em COBWEB

$$CU\ (C) = \frac{\sum_{k=1}^{n} P(C_k)\left[\sum_i \sum_j P(A_i = V_{ij} \mid C_k)^2 - \sum_i \sum_j P(A_i = V_{ij})^2\right]}{n}$$

Onde a expressão $\sum_i\sum_j P(A_i=V_{ij}|C)^2$ representa a quantidade de valores de atributos que se pode inferir sem conhecimento a que categoria pertencem. Considerando uma partição como sendo o conjunto de categorias que possuem uma mesma categoria como raiz, a divisão da equação por *n* permite a comparação de partições com diferentes quantidades de categorias. Fato necessário devido ao uso de operações como junção, divisão e criação de uma nova categoria.

Uma vez que *Category Utility* está baseado na quantidade de inferências corretas sobre valores de atributos, pode-se dizer que sua habilidade de realizar inferências é uma medida natural de seu comportamento.

2.1.1.4 Conclusão

COBWEB é um sistema de formação automática de conceitos que pode ser visto ainda como um sistema de busca num espaço de hierarquia de categorias, possuindo 4(quatro) operações básicas:

1. Classificar uma entidade em alguma categoria existente;
2. Criar uma nova categoria a partir de uma nova entidade;
3. Combinar duas categorias em uma única categoria (junção), e
4. Dividir uma categoria em várias categorias (divisão).

COBWEB utiliza uma função de avaliação, *Category Utility*, para determinar que operação aplicar na hierarquia durante o processo de classificação.

O uso de medida de qualidade bem definida em uma função significou um avanço na área de formação automática de conceitos. Da mesma forma, a representação da capacidade de inferência de valores de atributos em termos de suas probabilidades condicionais facilitou o entendimento dos agrupamentos conceituais.

Novas operações, como junção e divisão, significaram uma melhoria para sistemas de aprendizado incremental, permitindo recuperar o poder de predição da hierarquia sem perder o caráter incremental.

Mesmo com vários avanços e melhorias para os algoritmos de aquisição automática de conhecimento, COBWEB possui algumas limitações. Sua implementação permite somente o uso de atributos com valores discretos. O que para muitos domínios pode não ser suficiente para uma representação fiel de suas entidades.

Esse trabalho analisa implementações baseadas em COBWEB, principal representante dos SFCP, que atacam essa limitação. As seções a seguir detalharão algumas abordagens nesse sentido.

2.2 Formação de conceitos com atributos discretos e contínuos

Algumas metodologias tradicionais em formação de conceitos assumem que as entidades a serem categorizadas são representadas por atributos contínuos [Duda & Hart, 73] [Jain & Dubes, 1988]. No entanto, quando essas metodologias são aplicadas em áreas como medicina, negócios ou ciências sociais, faz-se necessário trabalhar com atributos como sexo, cor, formato, tipo de doença, entre outros que são discretos por natureza.

No mundo real, a maioria das entidades é melhor representada através de atributos discreto e contínuos em conjunto. Dados geológicos, por exemplo, possuem atributos como idade, porosidade, e permeabilidade que são contínuos, mas também possuem atributos como tipo de rocha e estrutura cristalina que são discretos.

Para atividades de descoberta de conhecimento, é crucial a existência de sistemas que tratem domínios onde suas entidades são representadas pela combinação de atributos discretos e contínuos. Pode-se dizer que sistemas que tratam esses tipos de atributos isoladamente já atingiram sua maturidade. Mas, abordagens que tratam esses atributos em conjunto encontram dificuldades, principalmente, devido às diferentes naturezas dos mesmos.

A maioria das soluções para essa situação encaixa-se em uma das seguintes categorias:

- *Codificação de atributos discretos em valores numéricos inteiros*. Assim é possível utilizar medidas de distância oriunda da categorização a partir de atributos contínuos. Todavia, em muitos casos, essa conversão não faz sentido e a proximidade desses valores é de difícil interpretação;
- *Discretização de atributos contínuos, encaixando cada valor contínuo em um predeterminado intervalo de valores*. Em seguida pode-se aplicar soluções de categorização usando atributos discretos. Nesse tipo de abordagem, o processo de discretização, geralmente, ignora uma parte importante da informação: a diferença relativa ou absoluta entre os valores contínuos dos atributos.
- *Generalização de funções de avaliação direcionadas para cada tipo de atributo.* A principal dificuldade nesse tipo de abordagem está na natureza da função de avaliação utilizada, visto que geralmente são baseadas em probabilidade de distribuição de valores. Essa probabilidade de distribuição de valores para atributos discretos é calculada de uma maneira que não pode ser aproveitada para atributos contínuos.

Nesse trabalho serão analisadas soluções que se encaixam na última categoria de soluções para tratamento de atributos mistos. Inicialmente, será analisada uma abordagem já bem difundida nesse aspecto: CLASSIT [Gennari et al., 89]. Em seguida, outras soluções serão avaliadas, embora a maioria delas seja uma especialização de CLASSIT.

Ao final de cada solução avaliada serão traçados alguns comentários pertinentes à solução, bem como suas limitações no tratamento de diferentes tipos

de atributos em conjunto. A conclusão desse capítulo terá um comparativo entre as vantagens e desvantagens das soluções apresentadas na área de formação de conceitos probabilísticos baseados em atributos discretos e contínuos.

2.2.1 Família CLASSIT

CLASSIT [Gennari et al., 89] é um SFCP fortemente baseado em COBWEB, diferenciado-se basicamente nos seguintes pontos: (1) função de avaliação de qualidade da hierarquia, (2) representação das entidades e (3) dos conceitos.

Particularmente, os sistemas COBWEB/3 [McKusick & Thompson, 90], COBIT [Bond & Hine, 93] e CLASSITALL [Moller, 97], compõem a família CLASSIT, por serem considerados implementações da abordagem de Gennari, com singelas modificações que serão oportunamente apresentadas a seguir.

As modificações propostas por CLASSIT à COBWEB têm como objetivo a formação de conceitos em domínios com entidades representadas por atributos contínuos. A derivação realizada em *Category Utility* original para atingir esse objetivo manteve a essência dos conceitos probabilísticos apresentada em COBWEB, consistindo numa proposta para o tratamento de atributos discretos e contínuos em conjunto. CLASSIT ainda manteve a mesma estratégia de controle e os mesmos operadores que COBWEB. Com isso, CLASSIT serviu como base e inspiração para vários outros SFCP.

As seções a seguir demonstram o modelo proposto por CLASSIT, destacando suas diferenças do modelo de COBWEB. Assume-se que as modificações apresentadas nessas seções aplicam-se a todos os sistemas que compõe a família CLASSIT, exceto melhorias adicionadas por esses outros sistemas, que serão oportunamente explicitadas.

2.2.1.1 Representação e Organização

Embora atributos discretos ocupem um papel importante na descrição de objetos em linguagem natural, eles não fazem tanto sentido na descrição de dimensões de objetos.

É possível dizer que uma pessoa é alta e outra baixa e assim diferenciar as duas, mas o objetivo de CLASSIT é poder diferenciá-las através da menor alteração

no valor real de suas alturas. Essa capacidade permite a representação detalhada de entidades do mundo real em termos do valor quantitativo de seus atributos.

CLASSIT nasceu da necessidade da formação de conceitos num domínio de entidades representadas por suas dimensões físicas, e por isso trabalha somente com atributos contínuos.

O uso de atributos contínuos no lugar de atributos discretos requer ainda a modificação na representação dos conceitos. Existem duas abordagens óbvias a serem adotadas. Primeira, dividir os valores dos atributos em intervalos e usar os intervalos como representação simbólica dos atributos numéricos. Esse método permite a utilização original de COBWEB. Contudo, a prévia determinação desses intervalos introduz um conhecimento externo que pode não representar fielmente a realidade do domínio. Segunda, representar os conceitos diretamente em termos dos valores contínuos dos atributos.

CLASSIT adota essa última abordagem mantendo a mesma idéia de COBWEB de associar uma probabilidade de distribuição para cada atributo em cada conceito. No entanto, em vez de armazenar a probabilidade para cada par atributo/valor (ex. para o conceito C, *P(alto|C)=0.3; P(médio|C)=0.5; P(baixo|C)=0.2)*, CLASSIT assume uma distribuição normal contínua para cada atributo e a armazena, como ilustrado na figura 4. Essa distribuição é representada em termos da média e do desvio padrão dos valores do atributo.

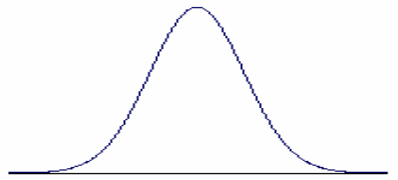

Figura 4: Representação gráfica da distribuição Normal.

CLASSIT organiza os conceitos da mesma maneira hierárquica de COBWEB. Conceitos mais genéricos estão próximos do topo enquanto os conceitos mais específicos estão abaixo destes. Em geral, quanto mais baixo o conceito na hierarquia menor o desvio padrão, uma vez que eles representam conceitos mais específicos com valores mais próximos.

2.2.1.2 Classificação e Aprendizado

O modelo de representação de conceitos de CLASSIT não requer nenhuma alteração na estratégia de controle de COBWEB. O uso dos 04(quatro) operadores de controle permanece o mesmo, assim como a aplicação da função de avaliação nas partições geradas por cada operação e em seguida a escolha da melhor como sendo a de maior resultado. Contudo, foram necessárias poucas, mas importantes, mudanças no algoritmo devido ao tratamento de atributos contínuos.

O sistema, por exemplo, pode decidir que deve parar a classificação em um determinado nível em vez de sempre descer até as folhas da hierarquia. Quando isso acontece significa que o sistema identificou que a entidade é "suficientemente" semelhante a um conceito já existente. O nível de semelhança de um atributo é determinado por um parâmetro do sistema chamado *cutoff,* baseado em sua função de avaliação.

Essa alteração tem duas vantagens básicas. Primeira, pesquisas [Quinlan, 83] mostraram que a criação de hierarquias exaustivamente tende a gerar uma quantidade maior de categorias em domínios com "*ruído*[1]". A segunda vantagem é que sistemas que armazenam todos seus objetos na hierarquia, criam estruturas muito grandes em aplicações reais.

2.2.1.3 Função de Avaliação

O uso de atributos contínuos na representação tanto das entidades quanto na dos conceitos requer uma generalização em *Category Utility* tradicional. Particularmente, os dois somatórios internos, definidos por COBWEB, exibidos na equação 04, precisam ser adaptados para o tratamento de valores contínuos.

Equação 4: Capacidade de Inferência com conhecimento da categoria e capacidade de inferência sem conhecimento da categoria.

$$\sum_i \sum_j P(A_i = V_{ij} \mid C_k)^2 \quad e \quad \sum_i \sum_j P(A_i = V_{ij})^2$$

[1] Entende-se por ruído como valores de atributos informados erroneamente. Esses valores são informações irrelevantes para a representação da observação e interferem na qualidade da hierarquia construída.

Ambos somatórios utilizam a probabilidade de ocorrência de um determinado par de atributo/valor (A_i=V_{ij}). O primeiro com o conhecimento da categoria que o par pertence, enquanto o segundo sem esse conhecimento. Contudo, o uso dessa estratégia, em termos de predição de valores, não se aplica para atributos contínuos, pois um valor contínuo pode nunca se repetir.

Para que a idéia de associação da capacidade de inferência com a probabilidade de ocorrência das propriedades fosse mantida, a abordagem adotada foi substituir o somatório por uma integração, assumindo alguma regra de distribuição para os valores contínuos. No caso, é utilizada uma *função de densidade de probabilidade*[1] para essa tarefa. Sem nenhum conhecimento a respeito da distribuição dos valores dos atributos, é assumido que eles estão distribuídos segundo a *curva normal*[2].

Assim, a probabilidade de ocorrência de um determinado valor de um atributo contínuo foi aproximada pelo valor do ponto, na curva normal, no local identificado pelo valor do atributo. E o somatório do quadrado das probabilidades dos pares de atributo/valor será o quadrado da integral da distribuição normal a partir da média e do desvio padrão dos valores do atributo contínuo.

Para o primeiro somatório da equação 4, a distribuição é calculada para uma categoria C_k em particular, enquanto o segundo somatório da mesma equação será calculado para o nó raiz, representando o desconhecimento da categoria. Logo, para ambos os casos, a integral pode ser representada pela transformação demonstrada na equação 5.

Equação 5: Transformação do somatório em integral

$$\sum_j P(A_i = V_{ij})^2 \Leftrightarrow \int \frac{1}{2\pi\,\sigma^2} \exp\left(\frac{x-\mu}{\sigma}\right)^2 \mathrm{dx} = \frac{1}{\sigma\ 2\sqrt{\pi}}$$

Onde μ é a média dos valores do atributo contínuo e σ é o seu respectivo desvio padrão. Com essa transformação, CLASSIT introduz uma abordagem para

[1] Funções de densidade de probabilidade são usadas na estatística para definir o comportamento da distribuição e concentração de variáveis contínuas.

[2] Curva normal ou distribuição normal são termos comumente utilizados em referência à distribuição de GAUSS.

formação de conceitos com atributos mistos. Fazendo a substituição em *Category Utiltiy* original, temos a equação 6.

Equação 6: Category Utility proposto por CLASSIT. Implementado em COBWEB/3 e CLASSITALL

$$CU\ (C) = \frac{\sum_{k=1}^{n} P(C_k)\left[\sum_i \frac{1}{\sigma_{ik}\ 2\sqrt{\pi}} - \sum_i \frac{1}{\sigma_{ir}\ 2\sqrt{\pi}}\right]}{n}$$

Onde *n* é a quantidade de categorias na partição, *i* é a quantidade de atributos, σ_{ik} é o desvio padrão do atributo *i* na categoria *k*, e σ_{ir} é o desvio padrão do atributo *i* na categoria raiz. Essa equação foi implementada em COBWEB/3 [McKusick & Thompson, 90] e em CLASSITALL [Moller, 97], que trabalham com os dois tipos de atributos, discretos e contínuos.

CLASSIT foi criado a partir da necessidade de representar objetos por suas dimensões. Embora sua proposta possa ser aplicada em domínios com atributos mistos, devido a seu objtivo CLASSIT foi implementado para trabalhar somente com atributos contínuos. Como o resultado da função de avaliação é usado para comparação, a constante $1/2\sqrt{\pi}$ pode ser descartada para a implementação de CLASSIT. Então, *Category Utility* usada em CLASSIT fica como demonstrado na equação 7.

Equação 7: Categoru Utility implementado em CLASSIT

$$CU\ (C) = \frac{\sum_{k=1}^{n} P(C_k)\left[\sum_i \frac{1}{\sigma_{ik}} - \sum_i \frac{1}{\sigma_{ir}}\right]}{n}$$

A implementação de CLASSIT ainda divide o somatório final pela quantidade de atributos sem valor nulo. Em linhas gerais, *Category Utility* final, para o tratamento de atributos discretos e contínuos, fica como demonstrado na equação 8.

Equação 8: Categoru Utility para ambientes com atributos discretos e contínuos

$$CU(C) = CU_{Discreto}(C) + CU_{Contínuo}(C)$$

Ou seja, utiliza uma função quando o atributo for discreto, no caso a proposta de COBWEB, e outra quando este for contínuo, esta que acabou de ser definida na equação 6.

2.2.1.4 Limitações

A função de avaliação de CLASSIT foi idealizada como sendo uma função equivalente à original de COBWEB. Infelizmente essa transformação introduz um problema quando o desvio padrão é zero para um conceito. Para remediar esse fato foi criado um parâmetro chamado *acuity*, que corresponde à noção da menor diferença perceptível nos valores. Esse parâmetro é usado em substituição a $1/\sigma_{ik}$ (ou $1 / \sigma_{ik}\ 2\sqrt{\pi}$) quando o desvio padrão for 0(zero).

A determinação do valor desse novo parâmetro é uma tarefa bastante subjetiva visto que o valor ideal depende de um conhecimento prévio a respeito dos valores a serem analisados, o que é difícil em sistemas incrementais. Por outro lado, a definição aleatória do *acuity* compromete consideravelmente a qualidade da hierarquia [Gennari et. al. 89] [Yoo & Yoo, 95], uma vez que esse valor deve ser compatível com a distribuição analisada. Por exemplo, definir o valor do *acuity* em 1 (um) onde a distribuição desvio padrão na ordem de 0.05 pode não ser interessante, pois o valor do acuity seria alto demais para a realidade da distribuição e vice-versa.

sO cálculo da função de avaliação heurística geral em domínios mistos, como definida na equação 8, é feito de acordo com o tipo de atributo. Por exemplo, quando o atributo for discreto utiliza-se o método original de COBWEB, caso contrário utiliza-se o método proposto por CLASSIT. Com isso, um outro problema surge do fato que os resultados das duas funções têm escalas diferentes.

No caso de COBWEB, o resultado do somatório dos quadrados das probabilidades dos valores de um determinado atributo discreto varia de 0(zero) até 1(um) (ex.: [0,1]). Já em CLASSIT, o resultado da probabilidade de um determinado atributo numérico, que deveria ser equivalente a um atributo discreto, varia de 0(zero) até $\propto$(*infinito*), caso o desvio padrão seja um valor muito próximo a 0(zero). Essa diferença de escalas faz com que os atributos numéricos possam ter predominância sobre os discretos em certos casos.

Em vista do problema de utilizar o parâmetro *acuity* e a diferença de escalas entre as diferentes funções de avaliação, COBIT propõe uma solução que possibilita o uso da função para os atributos contínuos, mesmo quando o desvio padrão de seus valores for 0(zero).

Equação 9: Proposta de modificação de COBIT.

$$\sigma \cong \sqrt{1+\sigma}$$

Ou seja, onde for utilizado o desvio padrão substitui-se pela raiz de 1(um) mais o valor do desvio padrão. Embora essa abordagem resolva, numericamente o problema, não fica bem definido o que representa intuitivamente, uma vez que não se pode dizer que a derivação proposta inicialmente por Gennari foi mantida.

Sob outro ponto de vista, os sistemas da família CLASSIT de modo geral, sofrem um prejuízo quando utilizam somente o desvio padrão: não levam em consideração a distância, relativa ou absoluta, entre os valores dos objetos [Biswas & Li, 98].

A figura 5 demonstra essa situação. Dadas duas categorias intermediárias, C_1 e C_2. μ_1, μ_2 as médias e σ_1, σ_2 os desvios padrão de um atributo contínuo A_1 em cada categoria respectivamente, sendo $\mu_1 < \mu_2$ e $\sigma_1 > \sigma_2$. Uma nova observação de uma entidade a ser submetida à CLASSIT tem valor *Vi* para *A1*. As operações de de encaixe da nova observação nas categorias C_1 e C_2 irão mudar seus desvios padrão para $\underline{\sigma_1}$ e $\underline{\sigma_2}$, respectivamente. Se $\underline{\sigma_1} > \underline{\sigma_2}$, então $CU(C_1) < CU(C_2)$, conseqüentemente o objeto será encaixado em C_2, apesar do fato de $|V_i - \mu_1| < |V_i - \mu_2|$.

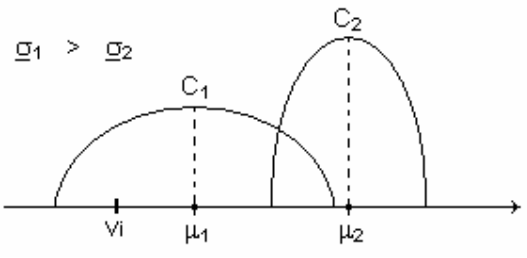

Figura 5: Situação especial para a implementação de CLASSIT

Os sistemas de formação de conceitos que serão descritos a seguir consideram as limitações aqui definidas e de alguma forma apresentam soluções.

2.2.2 ECOBWEB

ECOBWEB (Enhanced COBWEB) nasceu de um sistema de aplicação de aprendizado indutivo na engenharia civil [Reich, 91] [Reich & Fenves, 1991]. ECOBWEB é uma extensão do COBWEB, onde sua implementação sugere algumas soluções para problemas oriundos de CLASSIT como o uso do parâmetro *acuity* e o uso da distribuição normal como função de densidade de probabilidade para os atributos contínuos, uma vez que essa distribuição não era a mais apropriada para alguns domínios.

O tratamento de atributos discretos em ECOBWEB continua sendo o mesmo utilizado por COBWEB. No caso dos atributos contínuos, a função de densidade de probabilidade é definida pela probabilidade de distribuição de um atributo em torno da média aritmética de seus valores.

2.2.2.1 Função de avaliação

Capturando a essência de *Category Utility*, ECOBWEB introduz a seguinte aproximação.

$$\sum_j P(A_i = V_{ij} \mid C_k)^2 \approx P(A_i = \mu_i \mid C_k)^2$$

Onde μ_i representa a média do atributo *i* na categoria *k.* Ou seja, o valor esperado para um atributo contínuo é estimado em função da média dos valores do atributo. Esse valor esperado é calculado por:

$$P(A_i = \mu_i \mid C_k)^2 = \sum_i \left(\int_{-Lik}^{Lik} p_{ik}(v)dv \right)^2$$

Li representa um intervalo definido em torno de μ_i para a mesma categoria *k*. Para um conjunto *N* de vários atributos contínuos, *Category Utility* em ECOBWEB passa a ser:

$$CU_k = P(C_k) \sum_i^N \left(\int_{-Lik}^{Lik} p_{ik}(v)dv \right)^2 - \left(\int_{-Li}^{Li} p_i(v)dv \right)^2$$

Experimentos mostraram que a definição do intervalo L_i, definido em torno da média de um atributo contínuo, tem uma influência significante no resultado da função de avaliação de ECOBWEB, interferindo diretamente na construção da hierarquia de conceitos [Reich & Fenves, 1991]. A forma mais simples de definição desse intervalo é:

$$2L_i = \frac{Intervalo\ de\ valores\ do\ atributo\ Ai}{Quantidade\ de\ Intervalos\ diferentes\ do\ Atributo\ Ai}$$

Depois de calculado, esse intervalo funciona como uma constante, sendo utilizada em todo processo de construção da hierarquia. De forma que, quanto mais especializada for uma categoria, menor será esse intervalo de valores dos atributos contínuos nessa categoria.

Caso o intervalo escolhido, 2Li, seja muito grande, a probabilidade $P(A_i=V_i|C_k)$ para classes com valores de atributos contínuos muito próximos seria sempre 1(um), o que torna essa abordagem falha nesse sentido. Por outro lado, caso o valor do intervalo seja pequeno, a probabilidade de ocorrência em torno de um atributo pode ser igual para dois atributos contínuos diferentes. No entanto, a real distribuição desses dois atributos pode ser totalmente diferente.

Tabela 1: Três métodos implementados por ECOBWEB para definição do intervalo em torno da média de um atributo numérico.

Método	Cálculo do Intervalo
Estático	$\frac{Intervalo\ de\ valores\ do\ atributo\ Ai}{Quantidade\ de\ Intervalos\ diferentes\ do\ Atributo\ Ai}$
Dinâmico	$\frac{2\times\sigma i}{Quantidade\ de\ Intervalos\ diferentes\ do\ Atributo\ Ai}$
Adaptativo	$\frac{\sqrt{2\times\sigma i\times Intervalo\ de\ valores\ do\ atributo\ Ai}}{Quantidade\ de\ Intervalos\ diferentes\ do\ Atributo\ Ai}$

Para evitar a existência desses fatos, ECOBWEB apresenta dois outros métodos para definição do intervalo em torno da média. O primeiro método é

chamado “dinâmico”, pois o tamanho do intervalo de distância da média de um atributo contínuo é calculado em função da variância dos valores desse atributo. Já O segundo método, identificado por “adaptativo”, computa o valor do intervalo em torno da média de um atributo contínuo através da média geométrica dos valores do intervalo encontrados usando os métodos estático e dinâmico.

A tabela 1 exibe a fórmula de cálculo das 3(três) abordagens utilizadas por ECOBWEB para encontrar o valor de *2L*, todas baseadas na “quantidade esperada de intervalos distintos para o atributo contínuo”, a ser definida através da interferência do usuário. Onde σ_i representa o desvio padrão do atributo A_i.

2.2.2.2 Limitações

Percebe-se que a definição do valor da distância definida em torno da média (*2L)* modifica a performance de ECOBWEB. Assim, experimentos [Biswas & Li, 98] realizados analisaram as conseqüências da definição dos 2(dois) parâmetros definidos pelo usuário: (1) o método a ser utilizado para computar o intervalo em torno da média de um atributo contínuo, e (2) a quantidade esperada de intervalos distintos para um atributo contínuo (*n*). Os resultados desses experimentos comprovaram a dependência de ECOBWEB em relação a esses parâmetros. Pode-se dizer que essa dependência não é uma característica desejável em algoritmos de aprendizagem automática. Outro ponto negativo é que a definição do “intervalo esperado de valores de um atributo” não teria nenhuma garantia real da concordância com o domínio, visto que se trata de um algoritmo incremental.

2.2.3 ITERATE

O sistema de formação de conceitos ITERATE [Biswas et. al., 98] foi desenvolvido, principalmente, visando evitar (1) um problema inerente à algoritmos incrementais de aprendizado, a ordem de observação das entidades, e (2) construir uma *melhor* hierarquia de conceitos através da redistribuição das entidades depois da classificação da entidade na hierarquia.

O tratamento de atributos discretos juntamente com atributos contínuos não é considerado meta principal dessa implementação. Assim, serão vistos conceitos gerais do algoritmo, em seguida, será dada maior importância ao aspecto do

tratamento de atributos contínuos e discretos em conjunto, dado este ser o interesse principal deste trabalho.

Os passos gerais da estrutura controle de ITERATE estão apresentados na tabela a seguir:

Tabela 2: Três passos principais da estrutura de controle de ITERATE.

Passos
1) Geração da hierarquia de conceitos. Uma heurística de dissimilaridade é usada para ordenar os objetos analisados a cada classe antes que uma nova partição seja criada;
2) Escolha de um conjunto de categorias que seja representativo para a hierarquia total de acordo com *Category Utility*,
3) Analisar os objetos para redistribuí-los na categoria mais similar, de acordo com uma medida adotada de encaixe dos objetos nas categorias.

A intenção por trás da estrutura de controle de ITERATE é usar o resultado de uma classificação da hierarquia como ponto de partida para todo o processo de categorização. O primeiro passo do algoritmo classifica hierarquicamente as entidades previamente ordenadas. O segundo passo escolhe um conjunto inicial de categorias como as categorias que genericamente melhor representam o domínio, e com essas categorias, ITERATE cria uma larga partição inicial para o processo de categorização do terceiro passo. Por fim, o terceiro passo além de ser a categorização propriamente dita, representa a fase de otimização. Esse passo utiliza uma medida de similaridade entre uma observação e uma classe para determinar qual categoria melhor representa a nova entidade. A otimização é repetida até que seja encontrada uma partição estável, ou seja, quando nenhuma observação muda de categoria. Nesta dissertação se estudará o processo de otimização com mais detalhe, pois é onde se encontra o tratamento de atributos discretos e numéricos em conjunto.

O processo de reorganização da hierarquia não interfere no tratamento de atributos mistos e, portanto, não será abordado nesse trabalho.

2.2.3.1 Função de avaliação

A versão inicial de ITERATE utiliza a abordagem de COBWEB para atributos discretos. Para atributos contínuos, a abordagem de COBWEB foi reformulada de

maneira similar à CLASSIT, contudo através da abordagem de PARZEN WINDOW [Duda & Hart, 73], que será detalhada a seguir.

A reformulação de ITERATE consistiu basicamente em adaptar a função de avaliação inicial para considerar atributos contínuos. Dois pontos foram considerados, (1) a transformação dos somatórios em integrais, e a (2) escolha da distribuição de probabilidade a ser utilizada.

A forma de conversão da parte da equação que utiliza somatório em cálculos de integrais utilizados foi a mesma adotada por CLASSIT, ou seja, a analogia feita entre a probabilidade de distribuição de atributos discretos e uma função de densidade de probabilidade foi a mesma. Contudo, a função de densidade de probabilidade utilizada não foi a distribuição normal. De fato, muitos domínios onde ITERATE foi aplicado não passaram pelo *teste de normalidade* [1].

Uma variação do método de PARZEN WINDOW, a abordagem do vizinho mais próximo [Duda & Hart, 73], foi usada como função de densidade de probabilidade em ITERATE. A tabela 3 exemplifica os passos do método de cálculo dessa abordagem.

Os primeiros dois passos ordenam o atributo contínuo *A* e determinam a *área da vizinhança*[2], a qual será denominada *J,* calculada a partir da quantidade de observações do domínio. Essa área se desloca da esquerda para a direita segundo os valores ordenados do atributo. Cada valor representa um passo de deslocamento de *J*. A densidade de probabilidade de J a cada ponto é estimada por y_i *,* segundo a equação do 3º passo da tabela 3, onde *i* representa o passo e *Xi* representa o valor contínuo referente ao passo.

Para garantir que a soma dos valores de densidade de probabilidade sejam, no máximo, 01(um), a densidade de probabilidade em cada ponto y_i é normalizada segundo o passo 4. O fator de normalização é aproximado através do somatório das áreas do trapézio definido pelos pontos que forma a área da vizinhança *J* (x_i, x_{i+J}, y_{i+1}, y_i) , de acordo com a figura 6 . As densidades de probabilidade, Y_i, formam a curva de densidade de probabilidade.

---

[1] Teste que verifica se determinado conjunto de valores contínuos está distribuído segundo o método de GAUSS.
[2] A área da vizinhança é referenciada nos artigos como *WINDOW SIZE*.

Segundo a analogia usada em CLASSIT para tratamento de atributos contínuos, deve-se calcular o valor do quadrado da função de densidade de probabilidade. Novamente, a integral do quadrado da função de densidade de probabilidade; e aproximada pela área sob a curva formada por Y_i^2 *segundo* a equação do passo 5 da tabela 3.

O valor final da parte dos atributos contínuos em *Category Utility* de ITERATE é calculado através do somatório do cálculo acima descrito para cada atributo. Esse método de cálculo será utilizado para o nó raiz de uma partição e para os nós que formam a partição. Sendo que nesse último caso existe uma particularidade: quando a quantidade de entidades que uma categoria representa for inferior a 5(cinco), é utilizada uma distribuição uniforme de valores em vez do método de PARZEN WINDOW.

Tabela 3: Função de avaliação para atributos contínuos em ITERATE.

Passos do método PARZEN WINDOW
1) Ordenar os valores contínuos do atributo *A*
2) Dado uma base de dados com *n* observações, o tamanho da área da vizinhança será $J=\sqrt{n}$.
3) $\forall i,\ i=1,2,...,n$ calcule: $$y_i = p\left(\frac{1}{2}[x_i + x_{i+J}]\right) = \frac{\frac{J}{n}}{x_{i+J} - x_i}$$
4) A integral é aproximada pelo cálculo da área sob a curva através do método do trapézio. Antes disso, calcula-se um fator de normalização para garantir que o valor da integral da *fdp*[1] seja 1. $$A = \int fdp = \sum \frac{1}{2}(y_i + y_{i+1}) \times d_i$$ onde $d_i = x_{i+J} - x_i$. então calcula-se $Y_i = \frac{y_i}{A}$
5) Calcula-se $\int fdp^2$ através de $\sum Y_i^2 \times d_i$

[1] Fdp é a abreviação de Função de Densidade de Probabilidade.

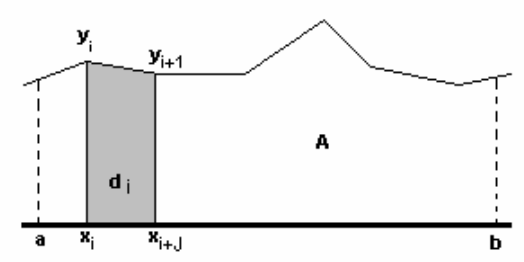

Figura 6: Cálculo do fator de normalização do método PARZEN WINDOW.

Category Utility final em ITERATE possui uma função para os atributos discretos, a mesma de COBWEB, e outra para atributos contínuos, a que foi citada a pouco. A forma de utilizar as duas em conjunto é a mesma proposta por CLASSIT, ou seja, quando o atributo for discreto utiliza-se a função específica dos discretos, e assim também para os atributos contínuos. Ao final, o somatório do resultado de cada função específica para cada tipo diferente de atributo dará o valor de Category Utility para uma partição.

2.2.3.2 Limitações

Alguns problemas foram identificados na proposta de Biswas. Inicialmente, quando uma categoria é formada por uma única entidade, a curva que deveria existir para o cálculo da densidade de probabilidade é representada por um ponto e a área sob um ponto é indefinida. Foi feita, mais uma vez, uma analogia à proposta de Gennari, quando uma categoria representa somente uma observação. No caso de CLASSIT, a ocorrência desse fato faz com que o desvio padrão da categoria seja 0(zero), tornado-se também uma situação indefinida.

Em CLASSIT, a solução dada foi o uso do parâmetro *acuity*[1]. Em ITERATE, essa idéia também foi aproveitada, assim como seus problemas. Demais problemas encontrados em ITERATE são decorrentes da tentativa de contornar os problemas inerentes do uso de *acuity*.

Tentou-se, para evitar o uso do *acuity*, utilizar o desvio padrão da categoria pai. Entretanto, a própria avaliação de ITERATE mostrou que esse valor era alto demais para categorias com somente uma observação. Com isso, tentou-se utilizar metade

[1] Em CLASSIT, acuity representa a máxima precisão que se pode obter do desvio padrão.

do valor do desvio padrão da categoria pai da categoria em questão, o que da mesma forma não apresentou bons resultados. Outras tentativas foram realizadas, contudo a performance desse método fica dependendo da característica de distribuição dos dados.

ITERATE permaneceu com os problemas oriundos do uso do *acuity*, como também os resultados de seus experimentos não foram suficientemente melhores que COBWEB/3. Isso motivou Biswas a propor uma segunda abordagem para atributos contínuos através da discretização de seus valores, o que foi visto no começo deste capítulo que não constitui uma boa abordagem.

Além do problema do uso do *acuity*, outro fato que pode ser visto como limitação para ITERATE é a interpretação da função de avaliação para os atributos contínuos. A complexidade do cálculo dessa função torna o entendimento do significado intuitivo da função uma tarefa não trivial. O que não é desejável para sistemas de formação de conceitos, pois os conceitos construídos serão interpretados por um especialista.

2.2.4 COBWEB95

COBWEB95 [Yoo & Yoo, 95] é outro algoritmo de categorização que busca atacar os problemas identificados em algoritmos da família CLASSIT. Sua proposta segue a mesma linha de raciocínio de ECOBWEB e ITERATE. Atributos discretos são tratados como no COBWEB original. Assim como seus antecessores, COBWEB95 também possui uma *CategoryUtility* para cada tipo de atributo e a medida heurística final é a soma cada CU individual.

Em COBWEB95, assume-se também que a distribuição dos atributos contínuos é regida pela curva normal. Contudo, utiliza-se uma probabilidade de aproximação da média dos valores contidos em uma categoria. Essa aproximação é dada por um intervalo ao redor da média que é definido por uma *margem de tolerância.* Essa margem de tolerância é um percentual sobre o valor da média. Então, para atributos contínuos, dada uma observação $X \in C_k$, a probabilidade de inferir corretamente um valor com uma margem de tolerância δ é $P[\ |X_i - \mu_{ik}| < \delta\ ]$ que é calculado por

$$\int_{-\delta}^{\delta} \frac{1}{\sigma_{ik}\sqrt{2\pi}} e^{-\frac{x^2}{2\mu_{ik}^2}} dx$$

onde μ_{ik} e σ_{ik} são os valores da média e do desvio padrão do atributo A_i para a categoria *k*. O valor dessa integral está ilustrado na figura 7. Sem o conhecimento a que categoria uma observação pertence, a probabilidade com a mesma tolerância é dada por $P[\ |X_i - \underline{\mu_{ik}}| < \delta\ ]$ que é a mesma equação definida anteriormente, exceto que $\underline{\mu_{ik}}$ e $\underline{\sigma_{ik}}$ são respectivamente o valor da média e do desvio padrão do atributo na categoria raiz.

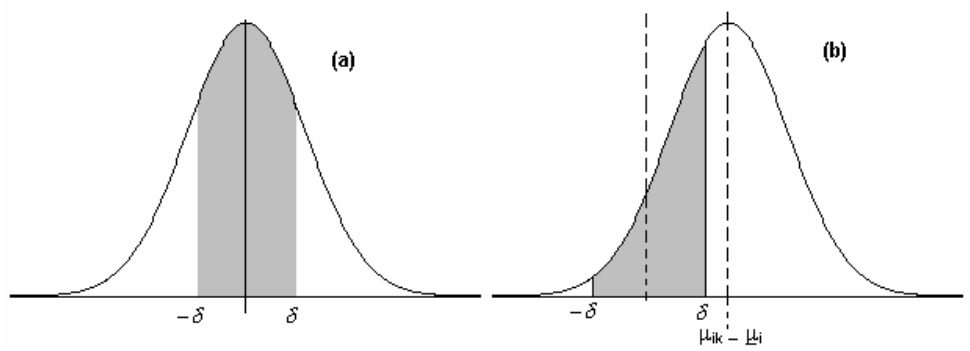

Figura 7: Curvas de distribuição normal.

Logo, a função de avaliação para atributos contínuos é definida na equação 10.

Equação 10: Category Utility proposto por COBWEB95.

$$CU_{Contínuos}(C) = \frac{1}{n}\sum_{k=1}^{n} P(C_k)\left[\sum_j P[|\ Xi - \mu_{ik}\ | < \delta] - \sum_j P[|\ Xi - \mu_{ik}\ | < \delta]\right]$$

Assim, Yoo definiu *Category Utility* para atributos contínuos como o aumento esperado da probabilidade de inferir corretamente um valor de um atributo para uma categoria com uma determinada margem de erro em relação a mesma probabilidade sem o conhecimento da categoria. Ou seja, na figura 7(a), a maior probabilidade de inferir corretamente um determinado valor de atributo contínuo é alcançada quando o valor da variância dos valores dos atributos é pequeno. Por outro lado, sem o conhecimento da classe, quanto menor a diferença entre $\underline{\mu_i}$ e $\underline{\mu_{ik}}$, maior a área, de acordo com a figura 7(b). Isso significa que a maior probabilidade de inferir

corretamente o valor contínuo de um atributo é alcançada quando a média de um atributo *A* numa determinada categoria pai é próxima da média da categoria C_k.

2.2.4.1 Limitações

COBWEB95 é um sistema de formação de conceitos baseado nas idéias de CLASSIT e que propõe soluções para suas limitações.

O uso do cálculo da probabilidade dispensa a necessidade de normalização dos dados. No cálculo da probabilidade de ocorrência é levado em consideração tanto o desvio padrão quanto a média, ou seja, a real distância entre os valores de um atributo é considerada. No caso do desvio padrão ser zero, a probabilidade passa a ser 1, dispensando ainda a necessidade do *acuity.* Contudo, o maior benefício apresentado por COBWEB95 foi a derivação de *Category Utility* original para o tratamento de atributos contínuos de forma equivalente a CLASSIT, no entanto mais robusta no sentido de evitar a necessidade de interferência do usuário.

Estudos comparativos foram realizados com COBWEB95 e COBWEB/3 [Yoo & Yoo, 95]. Para COBWEB/3, foram normalizados os dados e foi utilizado o melhor valor para o *acuity*, encontrado através de tentativa e erro. COBWEB95 apresentou os mesmos resultados que COBWEB/3, sem a necessidade da definição do *acuity*.

Apesar dessa abordagem apresentar uma solução que aparentemente soluciona antigos problemas, análises e experimentos realizados nesta dissertação mostraram uma limitação no que diz respeito à equivalência das funções de avaliação. Essa limitação será abordada com maior detalhe no capítulo 4*,* onde mostrará que se trata de um problema inerente ao uso de diferentes funções de avaliação para os diferentes tipos de atributos. Este trabalho usou COBWEB95 como base para sua proposta.

2.3 Conclusão

Esta seção apresenta um comparativo entre os sistemas de formação de conceitos apresentados nesse capítulo. A comparação foi realizada em termos da abordagem utilizada para atributos contínuos, das melhorias acrescentadas ao contexto de formação de conceitos em domínios com atributos mistos, e das respectivas limitações. Os sistemas não foram comparados em termos do

tratamento dado aos atributos discretos, pois todos usaram a mesma técnica apresentada em COBWEB. A tabela 4 resume as abordagens estudadas.

Pode-se dizer que o uso da curva normal para medir a probabilidade de ocorrência de atributos contínuos possa ser uma limitação em diferentes domínios. No entanto, para efeito de comparação, assume-se que cada sistema utilizou a melhor abordagem para os domínios em que foram aplicados.

O aspecto da diferença entre escalas das diferentes funções de avaliação, embora não tenha sido citada em todas as abordagens como limitação, representa uma limitação comum entre os sistemas. Para o quadro foi dada prioridade para os aspectos mais explícitos de cada sistema. Ao final, pode-se dizer que COBWEB95 foi a abordagem que mais se sobressaiu em termos de proposta de aplicação e independência de parâmetros externos.

O capítulo a seguir demonstrará o problema da diferença entre escalas das funções de avaliação, uma sugestão para evitar esse problema e um estudo comparativo de performance em termos de poder de aprendizado entre alguns sistemas.

Tabela 4: Resumo das abordagens de tratamento de atributos discretos e contínuos.

Sistema	Abordagem Numérica	Melhoria	Limitação
CLASSIT	Curva Normal; Probabilidade igual ao valor da curva que representa o valor do atributo.	Primeira abordagem numérica descendente de COBWEB; Analogia com o tratamento dos atributos discretos.	Somente atributos contínuos; Acuity; Não considera a real distância entre valores contínuos.
COBWEB/3	Mesmo que CLASSIT	Primeira implementação de CLASSIT para dois tipos de atributos; Primeiras análises em ambiente misto	Acuity; Não considera a real distância entre valores contínuos.
COBIT	Mesmo que CLASSIT	Resolve o problema do acuity substituindo o desvio padrão pela raiz deste mais um.	Não se sabe o que intuitivamente representa a solução para o problema de acuity; Não considera a real distância entre valores contínuos.
CLASSITALL	Mesmo que CLASSIT	Apresenta estudos iniciais sobre diferença de escalas entre funções de avaliação	Acuity; Não considera a real distância entre valores contínuos.
ECOBWEB	Probabilidade de distribuição de valores em torno da média dos atributos.	Abordagem que se adequa a distribuição existente.	Dois parâmetros devem ser escolhidos pelo usuário que interferem consideravelmente na formação dos conceitos.
ITERATE	PARZEN WINDOW	Apresenta abordagem para domínio com distribuição diferente da normal.	Acuity; Não considera escalas de resultados entre funções de avaliação.
COBWEB95	Curva Normal; Probabilidade igual a área de um intervalo sob a normal.	Solução para acuity; Solução para real distância entre valores.	Não considera escalas de resultados entre funções de avaliação.

3 Problemática

A principal dificuldade na utilização de atributos contínuos e discretos, em sistemas de formação de conceitos, deve-se à natureza da função de avaliação utilizada. No caso de conceitos probabilísticos, tais funções de avaliação dependem do cálculo de probabilidades baseadas na freqüência de ocorrência dos valores dos atributos.

Para atributos discretos, a probabilidade de ocorrência de valores pode ser estimada simplesmente pela contagem de ocorrências do valor de um atributo. Considerar essa mesma probabilidade para os atributos contínuos não é factível, pois um valor numérico não se repete freqüentemente, ou ainda, pode nunca se repetir. Portanto, o uso da abordagem utilizada para atributos discretos em atributos contínuos, em termos de predição de valores, não seria interessante.

Existem várias abordagens visando determinar a probabilidade de ocorrência para atributos com valores contínuos como visto no capítulo anterior. Contudo, o problema básico, em sistemas de formação de conceitos em domínios com atributos discretos e contínuos, é fazer as diferentes funções de avaliação trabalharem em conjunto. Como essas funções serão utilizadas na heurística de criação da hierarquia através de uma função de avaliação geral, é importante que seus resultados sejam, no mínimo, coerentes entre si, pois a predominância de qualquer uma das funções, em se tratando de predição de valores, pode levar a conclusões indesejadas.

3.1 Amplitude de resultados

No caso de *Category Utility* usada em COBWEB, a função que mede a probabilidade de ocorrência de um atributo discreto em uma determinada categoria é dada por $\sum P(A=V|C)^2$. Onde $P(A=V|C)$ é a probabilidade do atributo *A* ter o valor igual a *V* dado que esse par de atributo/valor pertence à categoria C. Essa função terá resultados que variarão de acordo com a quantidade de valores do atributo. Suponha um conceito *C* com um atributo *A* discreto com 02(dois) valores distintos, *X*

e *Y*. Dado que o cálculo é feito através da equação $\sum P(A=V|C)^2$, o resultado da probabilidade de ocorrência do atributo *A* pode variar entre $[(1/2)^2+(1/2)^2] = 0,5$ e $[0^2 + 1^2] = 1$. O resultado da função será 0,5 quando o conceito *C* tiver a mesma quantidade de ocorrências para o valor *X* e para o valor *Y*. O valor máximo dessa função, 1(um), será atingido quando todas as ocorrências do atributo *A* no conceito *C* tiver somente um valor, *X* ou *Y*. Os outros possíveis resultados da função, quando o atributo possuir 02(dois) valores, variarão de acordo com as diferentes quantidades de ocorrência dos valores do atributo *A* no conceito *C,* sempre respeitando os limites.

O mesmo comportamento também se verifica quando um atributo possui 3(três) valores. Nesse caso, os limites de resultado da função de probabilidade de ocorrência do atributo ficam entre [0,33 ... 1]. Para quatro valores seriam [0,25 ... 1], e assim por diante, como a tabela 5 exemplifica.

Tabela 5: Exemplo dos limites inferiores do resultado da função de probabilidade de ocorrência utilizada para atributos discretos em COBWEB. Os limites superiores sempre serão 1(um).

Quantidade de Valores	Limite Inferior
02	0,5000
03	0,3333
04	0,2500
05	0,2000
06	0,1667
07	0,1429
08	0,1250
09	0,1111
10	0,1000
11	0,0909
12	0,0833
13	0,0769
14	0,0714
15	0,0667
16	0,0625
17	0,0588
18	0,0556
19	0,0526
20	0,0500

Conclui-se que, quanto maior a quantidade de valores de um atributo discreto, maior será a amplitude de possíveis resultados de sua função de probabilidade de ocorrência.

Na função de avaliação para atributos contínuos, entretanto, os valores do resultado têm um comportamento diferente. O cálculo da probabilidade de ocorrência para esse tipo de atributo depende, em geral, do conhecimento prévio de como estão distribuídos os valores do atributo. Para as abordagens que assumem que os valores numéricos estão distribuídos segundo a curva normal, seus resultados são calculados a partir do valor da média e do desvio padrão do atributo.

CLASSIT, por exemplo, mede a probabilidade de ocorrência de um atributo contínuo em um determinado conceito através da função $(1 /(2\sqrt{\pi}\ \sigma))$, onde σ representa o desvio padrão dos valores do atributo contínuo em uma categoria.

Os limites de resultado dessa função variarão de acordo com o comportamento do desvio padrão do atributo. Em outras palavras, quanto menor o desvio padrão dos valores de um atributo, maior será o resultado de sua função. Assumindo um desvio padrão de valor 02 (dois) para um atributo contínuo, por exemplo, o resultado da função de probabilidade de ocorrência para esse atributo seria 0,1410 ($1/2\sqrt{\pi}\ \sigma = 1 / 2 * 1,772 * 2$). Para um desvio padrão igual a 10 (dez), o resultado passaria para 0,0282. A tabela 6 exemplifica a variação de resultados da função de probabilidade de ocorrência de atributos contínuos em algumas diferentes abordagens.

Assim, percebe-se que o resultado da função de probabilidade de ocorrência para os atributos contínuos possui um domínio de valores mais amplo que o resultado da função para atributos discretos.

A diferença de amplitude do domínio das funções de probabilidade de ocorrência significa que a função para os atributos contínuos pode assumir valores que a função para os atributos discretos dificilmente assumiria. Com isso, a função de avaliação dos atributos contínuos pode ser tanto predominante como irrelevante em relação à dos discretos para o cálculo da função de avaliação geral. A função será predominante quando tiver resultado maior que 01(um), pois a função de probabilidade de ocorrência dos atributos discretos tem valor máximo de 01(um). A mesma função será irrelevante quando seu resultado for menor que o limite inferior da função dos atributos discretos.

Qualquer situação, de predominância ou irrelevância, não é desejável para formação de conceitos, pois a hierarquia gerada estaria negativamente influenciada pelo tipo de atributo predominante.

Tabela 6: Exemplos de resultados de algumas funções de probabilidade de ocorrência para atributos contínuos. Para COBWEB95 foi utilizada média igual a 10 e tolerância de 10%. O termo acuity, usado em CLASSIT, representa a variação mínima perceptível para o atributo quando seu desvio padrão é zero.

Desvio Padrão	COBIT	CLASSIT	COBWEB95
0,00	1,0000	***Acuity***	1,00000
0,01	0,9950	28,2095	1,00000
0,10	0,9535	2,8209	1,00000
0,20	0,9129	1,4105	1,00000
0,50	0,8165	0,5642	0,97725
0,60	0,7906	0,4702	0,95221
0,70	0,7670	0,4030	0,92344
0,80	0,7454	0,3526	0,89435
0,90	0,7255	0,3134	0,86674
1,00	0,7071	0,2821	0,84134
2,00	0,5774	0,1410	0,69146
3,00	0,5000	0,0940	0,63056
6,00	0,3780	0,0470	0,56618
7,00	0,3536	0,0403	0,55680
8,00	0,3333	0,0353	0,54974
9,00	0,3162	0,0313	0,54424
10,00	0,3015	0,0282	0,53983
20,00	0,2182	0,0141	0,51994

3.2 Velocidade de convergência para limites de resultados

Além da diferença de amplitude entre as diferentes funções de probabilidade de ocorrência, existe outro comportamento essas funções que deve ser destacado: a velocidade de convergência para os limites de resultado. Entende-se por velocidade de convergência para os limites de resultado como a rapidez com que o resultado de uma função de probabilidade de ocorrência tende em direção aos seus limites, inferior ou superior.

Para demonstração desse outro comportamento, considere um conceito qualquer com 02(dois) valores distintos para um atributo discreto. Suponha o encaixe de uma nova observação nesse conceito. A nova observação possui um terceiro valor para o atributo discreto, de forma que o atributo discreto citado passa a ter 03(três) valores distintos para o conceito. O resultado da função de probabilidade de ocorrência passaria de 0,5 , calculado com 02 valores, para 0,33 , calculado com 03 valores. Caso o valor do atributo na nova observação fosse um dos valores já existentes na categoria, o resultado da mesma função passaria de 0,5 para 0,55.

Suponha a mesma situação dos atributos discretos para os atributos contínuos. O atributo contínuo possui desvio padrão igual a 02(dois) para o conceito. Suponha que a nova observação encaixada no conceito possui um valor para o atributo contínuo que modifica o desvio padrão para de 02(dois) para 10(dez). No caso de CLASSIT, o resultado da função de probabilidade de ocorrência para os atributos contínuos passa de 0,1410(com desvio padrão igual a 02) para 0,0282(com desvio padrão igual a 10).

O exemplo colocado ilustra que, com o encaixe de uma única nova observação, o resultado da função de probabilidade de ocorrência dos atributos discretos sofreu um decréscimo de 34%, enquanto teve-se aproximadamente 80% de decréscimo para o resultado da função dos atributos contínuos.

A mudança abrupta do resultado da função de probabilidade de ocorrência dos atributos contínuos em relação à dos discretos faz com que o problema da diferença de amplitude possa acontecer a cada nova observação classificada na hierarquia.

A seguir será demonstrado um exemplo que destaca o prejuízo causado pelos problemas citados.

3.3 Ilustração do problema

Considere a hierarquia de conceitos da figura 8, gerada a partir dos dados da tabela 7. Nela, a coluna ANIMAL é o identificador da observação e não é considerada no processo de categorização, o atributo “NINHADA” representa a quantidade de filhotes do animal em uma gestação (para o exemplo esse atributo será considerado discreto), o atributo ALIMENTO representa o principal alimento do animal e o atributo ALTURA indica a altura média do animal e será tratado como contínuo para o exemplo.

Tabela 7: Tabela com informações de animais

Animal	Ninhada	Alimento	Altura
Vaca	1	Vegetal	175,5 cm
Leão	4-5	Carne	75,9 cm
Búfalo	1	Vegetal	176,0 cm
Onça	4-5	Carne	75,4 cm
Antílope	1	Vegetal	175,8 cm
Tigre	4-5	Carne	76,0 cm

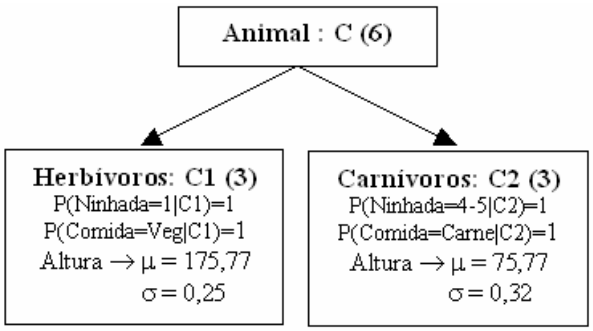

Figura 8 : Hierarquia gerada a partir dos dados da Tabela 7.

Na hierarquia descrita na Figura 8, no nível subseqüente ao nó raiz, dois conceitos foram criados, *C1* e *C2*. Cada conceito é representado através de suas propriedades e respectiva probabilidade de ocorrência no conceito. O termo *P(Ninhada=1|C1)=1*, por exemplo, indica que a propriedade *"Ninhada=1",* dado que o animal é herbívoro, possui a probabilidade de ocorrência igual a 1. No caso do atributo contínuo ALTURA, a probabilidade de ocorrência, para esta demonstração, será calculada através de uma distribuição normal com média e desvio padrão respectivamente representados por μ e σ.

O conceito identificado por *C1* representa a categoria dos animais herbívoros enquanto os carnívoros são representados pelo conceito *C2*. O termo "herbívoro" e "carnívoro" são meramente ilustrativos para facilitar o entendimento da explanação, não sendo utilizada nenhuma informação adicional para tal classificação. Foram usadas essas duas terminologias visto que as propriedades do conceito *C1* referem-se tradicionalmente a animais herbívoros enquanto as propriedades do conceito *C2* referem-se aos carnívoros da mesma forma. O número ao lado da identificação de cada conceito é a quantidade de entidades que este representa.

Imagine uma observação de uma nova entidade, um bode por exemplo, com as seguintes propriedades: "NINHADA=1", "ALIMENTO = Vegetal", e "ALTURA=75,6 cm". Essa nova observação será classificada dentro da hierarquia existente. O processo de classificação aqui utilizado consiste na operação de encaixar a nova observação em uma das categorias existentes ou, criar uma nova categoria com a nova observação. Vamos desconsiderar as operações de reestruturação da hierarquia, junção e divisão, pois os mesmos não são aplicáveis ao problema aqui exposto. A função de avaliação geral será calculada a cada operação, medindo a

qualidade da mudança sugerida pela operação. O maior resultado dessa função indica a melhor opção de construção da hierarquia.

A tabela 8 mostra o resultado da função de avaliação geral para alguns sistemas de formação de conceitos nas operações realizadas: encaixe da nova observação na categoria *C1*, na categoria *C2* e criação de uma nova categoria.

Tabela 8: Resultado de diferentes funções de avaliação para três operações diferentes.

Sistema	Encaixe em C1	Encaixe em C2	Nova Categoria
COBWEB/3	0,6335	0,7744	0,6028
COBIT	0,6703	0,6860	0,6034
COBWEB95	0,6384	0,6544	0,5791

Na situação apresentada, de acordo com a tabela 8, todas as abordagens têm maior resultado para suas funções de avaliação na operação de encaixe na categoria C2. Ou seja, a melhor opção para construção da hierarquia seria inserir a nova observação na categoria C2, como exemplificado na figura 9.

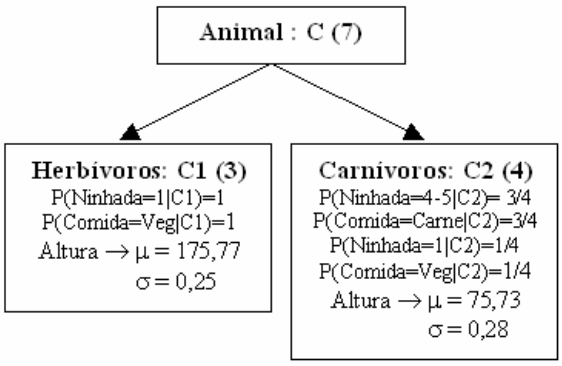

Figura 9 : Hierarquia gerada a partir dos dados a Tabela 7 com a nova observação encaixada na categoria C2.

Analisando a figura 8, intuitivamente percebemos que a nova observação estaria mais bem representada se inserida no conceito *C1*(herbívoros), de acordo com a figura 10, por possuir os mesmos valores para os atributos discretos, ou seja, "NINHADA" e "COMIDA". Contudo, os sistemas avaliados consideram como melhor opção o encaixe da nova observação sob a categoria C2 (carnívoros), como ilustrado na figura 9, onde somente um valor de atributo se assemelha[1] ao da categoria preferida, no caso o atributo "ALTURA".

[1] O termo "semelhante" usado para os sistemas indica que, assumindo uma distribuição normal, o valor em questão está a uma distância da média do atributo menor que uma tolerância admitida.

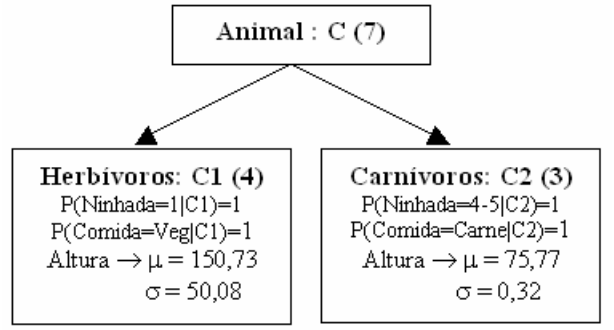

Figura 10 : Hierarquia gerada a partir dos dados a Tabela 7 com a nova observação encaixada na categoria C1.

Assumindo que todos os atributos têm igual importância, podemos afirmar ainda que a hierarquia ilustrada na figura 9 não é a melhor opção para fins de inferências futuras. A hierarquia apresentada na figura 9 teve dois atributos com a probabilidade de ocorrência diminuída, os dois atributos discretos, onde dois de seus valores passaram de 1 para ¾ . Enquanto a hierarquia da figura 10, que exemplifica a situação onde a nova observação seria inserida no conceito *C1*, teria somente um atributo com capacidade de inferência diminuída, ou seja, o atributo contínuo que o desvio padrão passou de 0,25 para 50,08. Uma vez que a capacidade de inferência de CLASSIT é inversamente proporcional ao desvio padrão (*$1/(2\sqrt{\pi}\,\sigma)$*).

Esse fato acontece devido à diferença de amplitude de resultados entre as diferentes funções de probabilidade de ocorrência. Apresenta-se aqui uma situação onde a construção da hierarquia de conceitos estaria sendo "guiada" pelos atributos contínuos. Para melhor esclarecer esse comportamento, será analisado detalhadamente o cálculo de cada função de avaliação quando o encaixe da nova observação é realizado em C1 e em C2.

O exemplo acima descrito será detalhado de forma que serão analisadas as funções de avaliação para duas situações: encaixe em *C1* e encaixe em *C2*. Portanto, a hierarquia inicial será a mesma da figura 8. Ou seja, o conceito identificado por *C* será a raiz da hierarquia e que os conceitos *C1* e *C2* formam a partição onde o algoritmo irá considerar o encaixe da nova observação em cada conceito (*C1 e C2*).

Inicialmente serão explicadas algumas reduções feitas nas funções avaliação, visando a objetividade da explicação, em seguida a análise será discutida em detalhes.

Considere, por exemplo, a função de avaliação geral implementada por COBWEB/3. Para essa explicação das funções de avaliação, será definida uma *Category Utility* reduzida ($CU_{Reduzida}$) onde serão removidas duas partes da função de avaliação original para deixar a explicação mais clara e objetiva. No caso em questão, as partes removidas são constantes e, por isso, sua retirada não interfere na identificação do problema. Vale ressaltar ainda que a retirada dessas partes não acentuam nem atenuam o problema, como será demonstrado a seguir.

A primeira parte a ser removida é a que na função original é utilizada para ganho de informação ($\sum\sum P(A_i=V_{ij})^2$). O valor de $P(A=V)$ é calculado a partir do nó raiz e, especificamente para nosso caso, terá o mesmo valor para ambas situações.

A segunda parte removida será a divisão pela quantidade de categorias(n) que a partição possui, visto que ambas partições possuem 02(duas) categorias. A equação 11 ilustra $CU_{Reduzida}$ que será utilizada para a explanação do comportamento das funções de avaliação.

Equação 11: Category Utility para explicação de comportamento de abordagens que trabalham com tipos mistos de atributos

$$CU_{reduzida}(C) = \sum_{k=1}^{n} P(C_k) \begin{cases} \sum_i \sum_j P(A_i = V_{ij} \mid C_k)^2 & \rightarrow \quad Discreto \\ \sum_i \frac{1}{\sigma_{ik}\ 2\sqrt{\pi}} & \rightarrow \quad Contínuo \end{cases}$$

Onde k representa a categoria, i representa o atributo e j o valor do atributo, no caso dos atributos discretos. A porção, na equação 11, destacada pela chave "{" representa o cálculo da *predictability* de *Category Utility* tradicional, onde a parte superior será usada quando o atributo i for discreto enquanto a inferior quando o mesmo for contínuo.

Embora tenha sido definida somente uma função de avaliação para o cálculo da partição, o valor resultado para cada tipo de atributo será feito separadamente como parte da explanação. Esses resultados parciais serão identificados por Ck_{Tipo}

do Atributo. Onde *k* representa a categoria, e *"Tipo do Atributo"* é o mnemônico representando um atributo *discreto* ou *contínuo.*

Considere a partição da figura 9, formada pelas categorias *C1* e *C2*, onde a nova observação está encaixada em *C1*. Essa partição será chamada de Pt_{C1}. Aplicando-se a equação de $CU_{Reduzida}$ em Pt_{C1} teríamos o seguinte desenvolvimento.

$$C1_{\,Discreto} \rightarrow P(C1) * \{\mathrm{P}(\mathrm{Ninhada} = 1 \mid \mathrm{C1})^2 + \mathrm{P}(\mathrm{Comida} = \mathrm{Veg} \mid \mathrm{C1})^2\}$$

$$+$$

$$C1_{\,Contínuo} \rightarrow P(C1) * \{\frac{1}{\sigma_{Altura\ \ em\ C1} * 2\sqrt{\pi}}\}$$

$$+$$

$$C2_{\,Discreto} \rightarrow P(C2) * \{\mathrm{P}(\mathrm{Ninhada} = 4-5 \mid \mathrm{C2})^2 + \mathrm{P}(\mathrm{Comida} = Carne \mid \mathrm{C2})^2\}$$

$$+$$

$$C2_{\,Contínuo} \rightarrow P(C2) * \{\frac{1}{\sigma_{Altura\ \ em\ C2} * 2\sqrt{\pi}}\}$$

$$\downarrow$$

$$\boldsymbol{CU_{Reduzida}(Pt_{C1}) = C1_{Discreto} + C1_{Contínuo} + C2_{Discreto} + C2_{Contínuo}}$$

Substituindo as variáveis da fórmula pelos valores da categoria tem-se a seqüência a seguir.

$$C1_{\,Discreto} \rightarrow {}^{4}\!/_{7} * \{(1)^2 + (1)^2\} = 1{,}1428$$

$$+$$

$$C1_{\,Contínuo} \rightarrow {}^{4}\!/_{7} * \{(0{,}0056)\} = 0{,}0032$$

$$+$$

$$C2_{\,Discreto} \rightarrow {}^{3}\!/_{7} * \{(1)^2 + (1)^2\} = 0{,}8571$$

$$+$$

$$C2_{\,Contínuo} \rightarrow {}^{3}\!/_{7} * \{(0{,}8815)\} = 0{,}3777$$

$$\downarrow$$

$$\boldsymbol{CU_{reduzida}(Pt_{C1}) = 1{,}1428+0{,}0032+0{,}8571+0{,}3777 = 2{,}3808}$$

Considere agora uma partição Pt_{C2}, representada pelas categorias C1 e C2 da figura 8, onde a nova observação está encaixada na categoria C2. Analogamente à situação de encaixe em C1, aplicando-se a fórmula de $CU_{Reduzida}$ em Pt_{C2} teremos.

$$C1_{\,Discreto} \rightarrow P(C1) * \{\mathrm{P(Ninhada} = 1 \mid \mathrm{C1})^2 + \mathrm{P(Comida} = \mathrm{Veg} \mid \mathrm{C1})^2\}$$
$$+$$
$$C1_{\,Contínuo} \rightarrow P(C1) * \{\frac{1}{\sigma_{Altura\ em\ C1} * 2\sqrt{\pi}}\}$$
$$+$$
$$C2_{\,Discreto} \rightarrow P(C2) * \{\mathrm{P(Ninhada} = 4-5 \mid \mathrm{C2})^2 + \mathrm{P(Comida} = Carne \mid \mathrm{C2})^2 + \mathrm{P(Ninhada} = 1 \mid \mathrm{C2})^2 + \mathrm{P(Comida} = \mathrm{Veg} \mid \mathrm{C2})^2\}$$
$$+$$
$$C2_{\,Contínuo} \rightarrow P(C2) * \{\frac{1}{\sigma_{Altura\ em\ C2} * 2\sqrt{\pi}}\}$$
$$\downarrow$$

$$\boldsymbol{CU_{Reduzida}(C^2) = C1_{Discreto} + C1_{Contínuo} + C2_{Discreto} + C2_{Contínuo}}$$

Também substituindo as variáveis da fórmula pelos valores das respectivas categoria para essa outra situação, tem-se a seguinte seqüência.

$$C1_{\,Discreto} \rightarrow 3/7 * \{(1)^2 + (1)^2\} = 0,8571$$
$$+$$
$$C1_{\,Contínuo} \rightarrow 3/7 * \{(1,1284)\} = 0,4836$$
$$+$$
$$C2_{\,Discreto} \rightarrow 4/7 * \{(3/4)^2 + (3/4)^2 + (1/4)^2 + (1/4)^2\} = 0,7143$$
$$+$$
$$C2_{\,Contínuo} \rightarrow 4/7 * \{(1,0075)\} = 0,5757$$
$$\downarrow$$

$$\boldsymbol{CU_{reduzida}(C^2) = 0,8571 + 0,4836 + 0,7143 + 0,5757 = 2,6307}$$

Comparando o resultado das *Category Utility Reduzidas* para as duas partições, Pt_{C1} e Pt_{C2}, tem-se que o cálculo da função de avaliação geral para a partição Pt_{C2} tem resultado final maior que para a partição Pt_{C1}.

Separando o resultado de $CU_{Reduzida}$ para cada tipo de atributo, de acordo com a tabela 9, percebe-se que esse fato acontece devido ao resultado individual da função dos atributos contínuos. Embora o valor da função em Pt_{C1} para os atributos discretos tenha resultado superior que em Pt_{C2}, indicando um melhor encaixe para esse tipo de atributo, a amplitude de resultados da função dos atributos contínuos faz com que essa melhora se torne irrelevante em relação ao resultado da função dos atributos discretos, reduzindo o valor final da função de avaliação geral.

Tabela 9: Resultado de Category Utility Reduzida separada por tipo de atributo.

Partição	$CU_{Reduzida}$ Discreto	$CU_{Reduzida}$ Contínuo	Total
Pt_{C1}	1,9999	0,3809	2,3808
Pt_{C2}	1,5714	1,0593	2,6307

Esse comportamento garante aos atributos contínuos uma predominância em relação aos discretos, pois somente um atributo contínuo garantiu o resultado final maior para uma categoria. Ou seja, um atributo contínuo teve preferência sobre dois atributos discretos. A proporção de predominância de 01 (um) atributo contínuo sobre 2 (dois) atributos discretos pode aumentar em situações com mais atributos contínuos.

CLASSIT, COBWEB95 e outras abordagens compartilham o mesmo problema, utilizam uma função de avaliação para atributos contínuos com uma amplitude de resultados e velocidade de convergência diferente da função para os atributos discretos, e não consideram esse comportamento na heurística geral de criação da hierarquia. A próxima seção apresenta uma abordagem que ataca este problema.

4 FORMVIEW2

Esta dissertação propõe um sistema de formação de conceitos probabilísticos em domínios com atributos discretos e contínuos. A implementação desse sistema foi feita em FORMVIEW2.

FORMVIEW2 é um sistema de aprendizado automático que estende FORMVIEW originariamente definido por Furtado [Furtado, 97]. FORMVIEW, genericamente, gera hierarquias de conceitos e descobre relacionamentos entres hierarquias de diferentes perspectivas, o que fornece um mecanismo para prover comunicação entre diferentes *expertises*. A característica de ligação entre duas *expertises* distintas foi o fator chave para a escolha desse sistema.

FORMVIEW foi definido originariamente para trabalhar somente com atributos discretos. Um dos resultados desse trabalho consiste em dar a FORMVIEW a capacidade do tratamento de dois tipos diferentes de atributos.

Inicialmente, será discutida a forma de classificação e aprendizado dos sistemas aqui envolvidos. A partir da classificação desses sistemas, a abordagem aqui proposta definirá alguns novos conceitos de aprendizado que permitirão o tratamento de atributos discretos e contínuos em conjunto. Um exemplo prático, em seguida, irá demonstrar a implementação e a melhoria da proposta.

4.1 Classificação e aprendizado

Sistemas de formação de conceitos probabilísticos (SFCP) representam o conhecimento adquirido através de uma hierarquia de conceitos. Esses sistemas apresentam praticamente a mesma metodologia de classificação de observações. Essa metodologia será rapidamente relembrada para facilitar a leitura.

O sistema inicia a hierarquia com a primeira observação analisada, criando o primeiro conceito a partir dos valores dos atributos dessa observação. Quando a segunda observação é analisada, o sistema generaliza as informações do primeiro conceito em termos das informações da primeira e da segunda observação. Em seguida cria para o primeiro conceito dois nós filhos representando a primeira e a

segunda observação. Desse ponto em diante, o sistema classifica novas observações, escolhendo entre 4 operações que permitem a construção e reestruturação da hierarquia, até encontrar um conceito que não possua mais filhos.

A melhor operação, a ser aplicada efetivamente na hierarquia, será escolhida de acordo com sua respectiva qualidade medida através de uma função de avaliação. A aplicação da melhor opção de estrutura significa, além da mudança na estrutura da hierarquia, a atualização das informações para cálculo das probabilidades condicionais dos atributos[1] nas categorias por onde a nova observação foi classificada. Em outras palavras, essa atualização de informações promove uma variação que afeta diretamente as probabilidades condicionais dos atributos nas categorias, como veremos na próxima seção.

4.2 Variação de Predictabilidade

A predictabilidade é, genericamente, uma métrica que representa a probabilidade de uma observação possuir um atributo com um determinado valor[1] dado que essa observação pertence a um certa categoria. Fisher aplicou fortemente esse conceito em sua função de avaliação *Category Utility.* O trabalho de Fisher teve uma intensa influência em pesquisas posteriores que mesmo outras implementações da função de avaliação, inclusive para atributos contínuos, destacam internamente a porção que representa essa predictabilidade. Ou seja, a noção de predictabilidade constitui uma das bases da construção sistemas de formação de conceitos probabilísticos.

Percebe-se, a partir da atualização das informações dos atributos nas categorias no processo de classificação de novas observações, que a predictabilidade de um atributo pode sofrer uma variação positiva ou negativa. Será positiva quando a atualização aumentar a probabilidade. Por outro lado, será negativa quando diminuí-la.

Cognitivamente, essa variação denota um fator para medir o aumento da capacidade de inferência de um atributo. Ou seja, um fator de ganho ou perda de poder de previsão de valores de atributos. Apesar do apelo de cognitivo que essa

---

[1] Para os atributos discretos são armazenadas individualmente informações para o cálculo da probabilidade condicional dos valores do atributo.

variação possui, tem sido até agora ignorada pelos sistemas correlatos de formação de conceitos probabilísticos. Nosso trabalho define uma métrica capaz de representar essa variação.

A forma proposta nesse trabalho para medir a variação da capacidade de inferência de um atributo foi através do aumento percentual da predictabilidade gerado a partir da atualização das categorias durante o processo de classificação de uma nova observação. Em outras palavras, a heurística proposta nesse trabalho define que a variação da capacidade de inferência de um atributo é medida através do quociente entre a predictabilidade de um atributo para uma categoria depois da atualização das informações dos atributos nessa categoria, e a mesma predictabilidade na situação antes da atualização das informações, assim como demonstrado na equação 12.

Equação 12: Variação da capacidade de inferência de um atributo de uma categoria.

$$\Omega\,(C,A) = \frac{predictability(C, A)}{predictability(C_A, A)}$$

Onde *C* representa uma categoria depois da atualização das informações dos atributos, enquanto C_A representa a mesma categoria antes da atualização das informações. Portanto, $\Omega(C,A)$ é a variação da predictabilidade do atributo *A* na categoria *C*.

Essa métrica aqui definida foi usada como subsídio para nossa proposta de solução para o problema da predominância da função de avaliação de atributos contínuos sobre atributos discreto em sistemas de formação de conceitos probabilísticos, como será discutido na próxima seção.

4.3 Ganho de Capacidade de Inferência da Categoria

O capítulo anterior caracterizou o problema da existência de predominância entre funções de avaliação de diferentes tipos de atributos dentro da função de avaliação geral de um sistema de formação de conceitos probabilísticos. Podemos recordar que o prejuízo maior da predominância é causado pelo fato que: a

[1] Esse valor do atributo pode ser um intervalo de valores para o caso dos atributos contínuos.

preferência de classificação é dada a categoria onde um único atributo contínuo da observação melhor se adequar, mesmo que a nova observação possua dois valores de atributos discretos coincidindo com os valores dos mesmos atributos em uma outra categoria. Esse comportamento prejudica a performance do sistema em termos de inferências futuras.

Aplicando o conceito de variação de predictabilidade, pode-se ver a questão da predominância por outro prisma. A coincidência entre dois valores discretos da nova observação e os valores da categoria pode ser vista como uma variação positiva da predictabilidade desses dois atributos discretos nessa categoria. Analogamente, houve variação positiva da predictabilidade de somente um atributo contínuo na outra categoria.

Percebe-se uma relação entre a predominância dos atributos contínuos sobre os discretos e a variação de capacidade de inferência desses tipos de atributos. De forma que a predominância entre atributos pode ser atribuída ao fato que as heurísticas atuais ignoram a variação de capacidade de inferência entre os diferentes tipos de atributos.

A abordagem aqui proposta defende que, além da qualidade medida através da função de avaliação geral do sistema em questão, deve-se considerar a variação de capacidade de inferência dos atributos discretos e contínuos. Ou seja, a categoria em que um determinado tipo de atributo tiver maior variação positiva de capacidade de inferência, deve ter preferência sobre outras categorias.

Para isso, definimos aqui uma função que mede o ganho de capacidade de inferência de uma categoria em termos de cada tipo de atributo. Essa função é calculada através do somatório da variação de predictabilidade para todos atributos do domínio, considerando a proporção de cada tipo de atributo dentro do domínio. A proporção de cada atributo dentro do domínio significa a quantidade de atributos de um determinado tipo sobre a quantidade total de atributos do domínio. A equação 13 exemplifica essa função.

Equação 13: Ganho de capacidade de inferência de uma categoria.

$$\Psi(C) = P(C) \times \sum_i \phi(A_i) \times \Omega(C, A_i)$$

Onde *P(C)* é a probabilidade de uma observação pertencer a categoria *C*. $\phi(A_i)$ representa a proporção do tipo do atributo A_i dentro do domínio. Essa proporção é medida através divisão da quantidade de atributos do tipo de A_i pela quantidade total de atributos. E $\Omega(C,A_i)$ indica a variação da predictabilidade do atributo A_i na categoria *C*. É importante lembrar que assume-se a igual importância de todos atributos.

O uso desse novo conceito aplica-se nas situações onde é possível medir a variação da predictabilidade dos atributos nas categorias. Em outras palavras, nas categorias onde o processo de classificação atualiza as informações dos atributos.

O primeiro momento em que a função para medir o ganho de capacidade de inferência pode ser utilizada é após o encaixe de uma nova observação nas categorias existentes. A seção a seguir apresenta uma proposta de modificação no processo tradicional de classificação de novas observações utilizando $\Psi(C)$.

4.4 Nova classificação

Nossa proposta de correção do problema de predominância entre funções de avaliação adiciona um processo após a escolha das duas melhores categorias de encaixe no processo tradicional de classificação.

Esse novo processo consiste em utilizar o valor de Ψ como peso para uma segunda comparação entre melhores as opções de encaixe. Garantindo que o ganho de capacidade de inferência dos atributos participe do processo de classificação e aprendizado dos sistemas de formação de conceitos probabilísticos.

A utilização de Ψ como peso consiste em multiplicar o resultado da função de avaliação geral do sistema para cada uma das duas categorias escolhidas, pelo valor de Ψ para as categorias correspondentes. De forma que o processo todo aconteça da seguinte maneira.

Depois da escolha das duas melhores categorias de encaixe, tem-se que a melhor categoria de encaixe é C_1 e a segunda melhor C_2. A partição onde a nova

observação foi encaixada em C_1 é representada por Pt_{C1}. Enquanto Pt_{C2} é a partição em que a nova observação está encaixada em C_2.

A medida de qualidade, calculada através de *Category Utility(CU),* que garantiu a situação de encaixe em C1 como melhor, passa a ser $CU(Pt_{C1}) \times \Psi(C_1)$. Por outro lado a medida de qualidade para a situação de encaixe da nova observação em C2 será $CU(Pt_{C2}) \times \Psi(C_2)$.

Esses novos valores calculados serão novamente comparados e então re-classificadas as categorias como primeira e segunda melhor categoria de encaixe.

Caso o resultado final da função de avaliação com o uso de Ψ para a categoria de segundo melhor encaixe (C_2) for superior ao resultado para a melhor categoria (C_1) também usando Ψ, percebe-se uma inversão de posições. Essa inversão é, então, corrigida através da promoção da segunda melhor categoria de encaixe para a primeira melhor categoria de encaixe e vice-versa. Em outras palavras, a melhor categoria de encaixe passa a ser a segunda melhor e a segunda melhor categoria passa a ser a melhor, de acordo com o esquema demonstrado na equação a seguir:

Equação 14: Verificação de inversão de posições, caracterizando a predominância entre atributos contínuos e discretos.

$$SE\ (CU(Pt_{C1}) \times \Psi(C_1) \leq CU(Pt_{C2}) \times \Psi(C_2))\ ENTÃO$$

$$\mathbf{CU_{temp} = CU_{primeiro}}$$

$$\mathbf{CU_{primeiro} = CU_{segundo}}$$

$$\mathbf{CU_{segundo} = CU_{temp}}$$

Onde $CU_{primeiro}$ representa o valor do melhor resultado de *Category Utility* na melhor categoria de encaixe. Enquanto $CU_{segundo}$ representa o valor do segundo melhor resultado de *Category Utility* na segunda melhor categoria de encaixe.

Assim, a melhor escolha no processo encaixe de FORMVIEW para categorias existentes passa a ser também aquela que possua melhor ganho de capacidade de inferência para ambos tipos de atributos. A próxima seção detalha a função de avaliação geral que será utilizada na implementação de nosso sistema.

4.5 Função de Avaliação

A abordagem de FORMVIEW implementada nesse trabalho foi baseada em COBWEB95, ou seja, utiliza uma função de avaliação para os atributos discretos e outra para os atributos contínuos. Para melhor entendimento, deve-se recordar a função de avaliação tradicional utilizada por COBWEB95.

Equação 15: Função de avaliação utilizada por COBWEB95

$$CU(C) = \frac{1}{n}\sum_{k=1}^{n} P(C_k)\begin{cases} \sum_i \sum_j P(A_i = V_{ij} \mid C_k)^2 - \sum_i \sum_j P(A_i = V_{ij})^2 \rightarrow discretos \\ \sum_i P[\mid Xi - \mu_{ik} \mid < \delta] - \sum_i P[\mid Xi - \mu_{ik} \mid < \delta] \rightarrow contínuos \end{cases}$$

Onde *C* representa o nó raiz de uma partição formada pelas categorias C_k. $P(C_k)$ representa a probabilidade da classe C_k. A porção superior da equação é utilizada para atributos discretos. Onde $P(A_i=V_{ij}|C_k)$ é a probabilidade do atributo Ai ser igual a V_{ij} na categoria C_k. $P(A_i=V_{ij})$ é a probabilidade do atributo A_i ter valor igual a V_{ij}.

A porção inferior será usada para atributos contínuos. Onde $P[|X_{ik} - \mu_{ik}| < \delta]$ representa a probabilidade de inferir corretamente um valor numa distribuição normal com uma margem de tolerância δ dado que esse valor pertence à categoria Ck. Enquanto a equação identificada por $P[|X_i - \mu_i| < \delta]$ representa a mesma probabilidade sem o conhecimento da categoria, ou seja, na raiz da hierarquia. Ambas sendo definidas pela equação 16.

Equação 16: Integral de cálculo de probabilidade de ocorrência de valores contínuos em COBWEB95.

$$\int_{-\delta}^{\delta} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\mu^2}}\, dx$$

Onde δ indica a tolerância utilizada e, σ e μ são, respectivamente, o desvio padrão e a média da categoria em questão.

Embora FORMVIEW utilize praticamente a mesma abordagem de COBWEB95, ele não negligencia o fato das diferentes funções de avaliação terem

comportamentos distintos, como demonstrado no capítulo 3 de caracterização do problema.

Para aplicar a abordagem do ganho de capacidade de inferência da categoria definida nesse trabalho, foram destacadas das respectivas funções de avaliação as partes que representam a o cálculo da predictabilidade do atributo A para a categoria C, pois representam a capacidade de inferência do atributo. Essas partes estão ilustradas na equação 17.

Equação 17: Equação que mede a capacidade de inferência de um atributo.

$$predictability(C,A) = \begin{cases} \sum_{j} P(A = V_j \mid C)^2 \quad \rightarrow \quad Discreto \\ \int_{-\delta}^{\delta} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{x^2}{2\mu^2}} dx \rightarrow \quad Contínuo \end{cases}$$

A porção superior da equação é referente aos atributos discretos enquanto a porção inferior para atributos contínuos. A seção a seguir mostrará detalhadamente essa implementação.

4.6 Exemplo

Essa seção do trabalho irá demonstrar um exemplo da nova heurística de busca definida em FORMVIEW. O exemplo aqui utilizado será o mesmo apresentado na seção de caracterização do problema. Para efeito didático, algumas etapas serão somente referenciadas, assim como alguns valores, tornando a explanação mais objetiva. Para facilitar a leitura, será feita uma breve recordação do exemplo utilizado naquela seção.

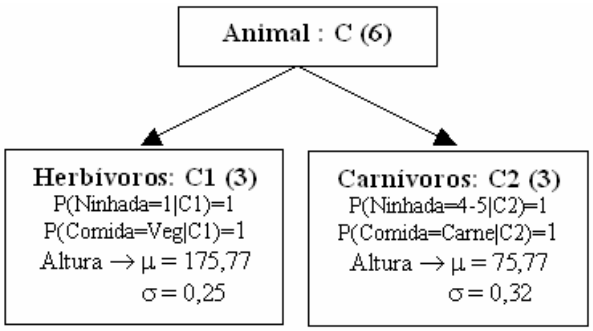

Figura 11 : Hierarquia geradas a partir dos dados a Tabela 7 do capítulo anterior.

O exemplo do capítulo anterior trata de um domínio de animais onde cada animal é descrito em termos de 03 (três) atributos: ALIMENTO, NINHADA e ALTURA. Os atributos ALIMENTO e NINHADA são discretos, enquanto o atributo ALTURA é contínuo. Considere uma hierarquia de conceitos formada por estas seis observações desse domínio, de forma que as categorias do nó raiz estão dispostas de acordo com a figura 11.

Considere uma nova observação com as características descritas na tabela 10.

Tabela 10: Nova observação a ser inserida na hierarquia da figura 11 .

ALIMENTO	NINHADA	ALTURA
VEGETAL	1	75,6 cm

A nova observação será classificada na hierarquia da figura 11, segundo a função de avaliação de FORMVIEW. Seguindo o procedimento normal de escolha da melhor categoria, o resultado da função de avaliação de FORMVIEW para encaixe da nova observação em *C1* seria 0,6384, de acordo com a tabela 8 de resultados da seção anterior. O encaixe em *C2* teria como resultado 0,6544, também de acordo com a referida tabela. A figura 12 demonstra a situação de encaixe da nova observação em *C1* enquanto a figura 13, o encaixe em *C2*.

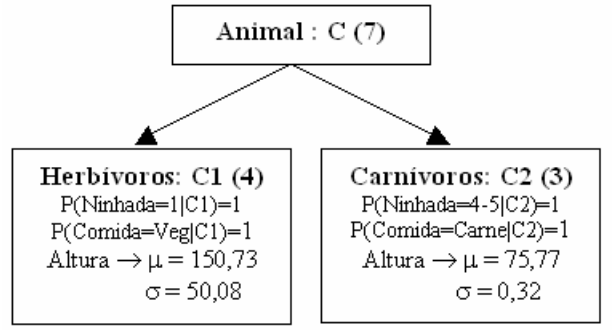

Figura 12 : Hierarquia gerada a partir dos dados a Tabela 7, do capítulo 3, com a nova observação encaixada na categoria C1.

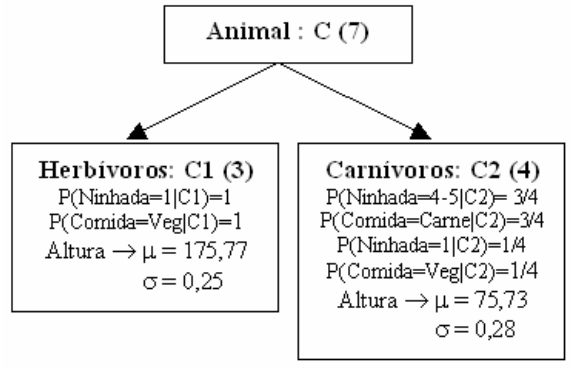

Figura 13 : Hierarquia gerada a partir dos dados a Tabela 7, do capítulo anterior, com a nova observação encaixada na categoria C2.

Aplicando a abordagem proposta por FORMVIEW, inicialmente é necessário calcular a predictabilidade dos atributos nos conceitos. Para o cálculo da *predictability* dos atributos contínuos foi utilizada uma tolerância de 10%, assim como no exemplo da seção anterior. Considere agora a situação antes da avaliação da nova observação, segundo a figura 11. Os conceitos nesse estado serão identificados por *C1a* e *C2a*. O valor da predictabilidade dos atributos em *C1a* está demonstrado na seqüência abaixo.

$$predictability(C1a, ALIMENTO) = \sum_i P(ALIMENTO = VEGETAL \mid C1a)^2 = 1$$

$$predictability(C1a, NINHADA) = \sum_i P(NINHADA = 1 \mid C1a)^2 = 1$$

$$predictability(C1a, ALTURA) = \int_{158,19}^{193,34} \frac{1}{0,25\sqrt{2\pi}} e^{-\frac{x^2}{2x175,77^2}} dx = 1$$

Da mesma forma, para *C2a*, o cálculo da predictabilidade seria como a seguir.

$$predictability(C2a, ALIMENTO) = \sum_i P(ALIMENTO = CARNE \mid C2a)^2 = 1$$

$$predictability(C2a, NINHADA) = \sum_i P(NINHADA = 4-5 \mid C2a)^2 = 1$$

$$predictability(C2a, ALTURA) = \int_{68,19}^{83,34} \frac{1}{0,32\sqrt{2\pi}} e^{-\frac{x^2}{2x75,77^2}} dx = 1$$

Continuando o processo, o cálculo também deve ser realizado para a situação após a avaliação da nova observação, onde os conceitos serão identificados por $C1_D$ e $C2_D$. A seqüência apresentada abaixo demonstra esse cálculo para, respectivamente, $C1_d$ e $C2_d$.

$$predictability(C1d, ALIMENTO) = \sum_i P(ALIMENTO = VEGETAL \mid C1d)^2 = 1$$

$$predictability(C1d, NINHADA) = \sum_i P(NINHADA = 1 \mid C1d)^2 = 1$$

$$predictability(C1d, ALTURA) = \int_{135,65}^{165,8} \frac{1}{50,08\sqrt{2\pi}} e^{-\frac{x^2}{2x150,73^2}} dx = 0,2358$$

$$predictability(C2d, ALIMENTO) = \sum_i \begin{matrix} P(ALIMENTO = VEGETAL \mid C2d)^2 + \\ P(ALIMENTO = CARNE \mid C2d)^2 \end{matrix} = 0,625$$

$$predictability(C2d, NINHADA) = \sum_i \begin{matrix} P(NINHADA = 1 \mid C2d)^2 + \\ P(NINHADA = 4-5 \mid C2d)^2 = 0,625 \end{matrix}$$

$$predictability(C2d, ALTURA) = \int_{68,15}^{83,30} \frac{1}{0,28\sqrt{2\pi}} e^{-\frac{x^2}{2x75,73^2}} dx = 1$$

Por substituição, agora se pode calcular o valor do ganho de capacidade de inferência do conceito (Ψ). O valor da proporção de cada tipo de atributo é, respectivamente, 2/3 e 1/3, para atributos discretos e atributo contínuos.

A equação a seguir demonstra detalhadamente o cálculo de Ψ para cada conceito *C1d e C2d*.

$$\Psi(C1d) = 4/7 \times \{ [2/3 \times (predictability(C1d,ALIMENTO) / predictability(C1a,ALIMENTO)] + [2/3 \times (predictability(C1d,NINHADA) / predictability(C1a,NINHADA)] + [1/3 \times (predictability(C1d,ALTURA) / predictability(C1a,ALTURA)]\} = \mathbf{0,806}$$

$$\Psi(C2d) = 4/7 \times \{ [2/3 \times (predictability(C1d,ALIMENTO) / predictability(C1a,ALIMENTO)] + [2/3 \times (predictability(C1d,NINHADA) / predictability(C1a,NINHADA)] + [1/3 \times (predictability(C1d,ALTURA) / predictability(C1a,ALTURA)]\} = \mathbf{0,666}$$

Utilizando o valor de COBWEB95 para o encaixe da nova observação em *C1*, 0,6384, multiplicado por *Ψ(C1d)*, 0,806, tem-se um novo valor, 0,5145, que será utilizado posteriormente para comparação. Da mesma forma, com o resultado de COBWEB95 para o encaixe da observação em *C2*, 0,6544, multiplicado por *Ψ(C2d)*, 0,666, encontra-se um valor igual a 0,4319.

Percebe-se que o novo valor de avaliação para o conceito *C1*, 0,5145, é superior ao novo valor para conceito *C2*, 0,4319. Portanto, de acordo com a metodologia apresentada por FORMVIEW, o conceito que a nova observação melhor se encaixa seria *C1* ao invés de *C2*. Intuitivamente podemos ver que esta é a melhor alternativa, pois o conceito *C1* teve um ganho de predictabilidade para dois atributos em relação a somente um do conceito *C2*.

4.7 Conclusão

FORMVIEW é um sistema de formação incremental de conceitos para uso em ambientes com atributos discretos e numéricos. Sua abordagem ataca um problema clássico na área de formação de conceitos com atributos mistos onde cada tipo de

atributo tem uma função de avaliação específica: o comportamento de diferentes funções de avaliação.

A proposta de FORMVIEW constitui-se uma solução inovadora, pois não modifica as funções de avaliação da hierarquia. Ao invés, inicia-se por aprofundar-se no entendimento das distintas funções de avaliação, o que constitui um avanço para formação de conceitos probabilísticos em ambiente com atributos discretos e contínuos.

Sua solução contribui ainda para o avanço das heurísticas dos mecanismos de busca, demonstrando o importante papel que esses mecanismos desenvolvem e que o molde desses mecanismos a uma necessidade específica pode significar aumento da performance do sistema, em termos de inferências futuras, como é o caso de FORMVIEW.

O capítulo a seguir vai demonstrar, através de experimentos, o ganho de performance em termos de capacidade de aprendizado de FORMVIEW sobre os sistemas avaliados nesse trabalho.

5 Avaliação

Este capítulo irá mostrar alguns experimentos que visam avaliar a capacidade de aprendizado de FORMVIEW em relação a outros sistemas de formação de conceitos probabilísticos. Tanto FORMVIEW quanto os demais sistemas usados nessa avaliação foram implementados e executados a partir do software *SmartBASE* [Rebouças e Furtado, 2000] que será detalhado no Apêndice I.

Os experimentos aqui realizados foram motivados pelo modelo de aprendizado proposto por Dietterich [Dietterich & Michalski, 83]. Esse modelo, em linhas gerais, aborda o aprendizado sob três aspectos: base de conhecimento, capacidade de aprendizado e forma de aprendizado.

O aspecto da base de conhecimento diz respeito à representação dos conceitos gerados. No caso de FORMVIEW e dos demais sistemas aqui avaliados, a representação do conhecimento é feita através de conceitos probabilísticos. Esse modelo de representação de conhecimento já está bastante consolidado, sendo por vezes identificado como um modelo cognitivo de representação de conhecimento [Rosch, 75] [Mervis & Rosch, 81] [Gluck & Corter, 85]. Reavaliar esse modelo de representação não é o objetivo desse trabalho.

Sob o aspecto da capacidade de aprendizado, a metodologia de Dietterich avalia os sistemas em termos da eficácia do conhecimento adquirido em fazer inferências. A abordagem de FORMVIEW propõe uma solução que afeta diretamente essa capacidade. Assim, esse será o aspecto que daremos maior ênfase nos experimentos.

No aspecto da forma de aprendizado são discutidos as operações realizadas por um sistema para construção dos conceitos, o custo computacional dessas operações e o impacto dessas operações, no nosso caso, na avaliação incremental das observações.

Inicialmente será explicada a metodologia utilizada para dirigir os experimentos visando, principalmente, o aspecto da capacidade de aprendizado. As bases de dados utilizadas nos experimentos serão descritas em seguida. A seção 5.3 apresentará experimentos e discussões sobre o aspecto da capacidade de

aprendizados dos sistemas. O aspecto da forma de aprendizado será abordado na seção 5.4. Com base nos resultados, será feita na seção 5.5 a avaliação dos sistemas em domínios do mundo real com atributos discretos e contínuos. Por fim, esse capítulo será concluído com um resumo dos experimentos.

5.1 Método de Avaliação

O processo utilizado para avaliação de performance dos algoritmos é realizado através da comparação do poder de inferência dos mesmos. Ou seja, comparar a capacidade que os algoritmos tem de inferir corretamente valores desconhecidos de atributos do domínio em questão.

A metodologia utilizada para medir essa capacidade consiste em ignorar um atributo de uma observação de teste e classificar essa observação em uma hierarquia de conceitos previamente construída. O algoritmo será responsável por encontrar o conceito que *melhor*[1] encaixa a observação de teste sem o conhecimento do valor de um atributo. A partir do conceito encontrado, o algoritmo deverá sugerir um valor para um atributo.

Para atributos discretos, o valor sugerido será o de maior probabilidade de ocorrência para o atributo. Considera-se que o sistema realizou uma inferência correta quando esse valor inferido coincide com o valor do atributo. No caso dos atributos contínuos, a idéia é a mesma, entretanto considera-se uma margem de erro para acerto de um valor. Nos experimentos foi utilizada uma margem de erro de 20%.

Para cada domínio avaliado, algumas observações foram utilizadas para criar uma hierarquia de conceitos enquanto outras observações foram usadas para testar essa hierarquia. As primeiras observações serão chamadas de observações de treinamento e as outras de observações de teste. Nos experimentos aqui apresentados as observações de treinamento representam, aproximadamente, 80% das observações do domínio. Enquanto as observações de teste constituem os 20% restantes, sendo estas diferentes das observações de treinamento.

O processo de classificação é realizado para todas observações de teste, sendo um atributo escolhido por vez. De forma que, um domínio em que suas

observações são representadas por *M* atributos, e que serão usadas *N* observações de teste, pode-se dizer que serão realizadas *M x N* classificações. Em outras palavras, esse valor também representa a quantidade de inferências sobre valores de atributos. A cada atividade de busca será computada a quantidade de inferências corretas. Essas inferências corretas serão chamadas de "acertos". Ao final desse processo, o resultado do poder de inferência de um sistema é dado em termos do percentual de acertos.

Assim como proposto por Quinlan [Quinlan, 83], a capacidade de aprendizado de um sistema pode ser verificada através da avaliação da quantidade de inferências corretas com diferentes quantidades de observações de treinamento. Para isso, as observações de treinamento foram divididas em subconjuntos com aumento progressivo de quantidade. Esses subconjuntos possuem, respectivamente, 25%, 50%, 75% e 100% das observações de treinamento. As observações de teste serão, então, avaliadas nas hierarquias geradas para cada subconjunto de observações de treinamento na ordem acima especificada.

Os resultados da avaliação em FORMVIEW foram comparados com COBWEB95, COBWEB/3 e COBIT. Para FORMVIEW e COBWEB95, a margem de tolerância usada nessas abordagens foi de 20%. O valor de *acuity* usado em COBWEB/3 foi de 1(um), como sugerido por GENNARI [Gennari et. al., 89].

A avaliação dos sistemas foi realizada inicialmente em domínio artificial, em seguida as abordagens foram aplicadas em domínios reais.

5.2 Domínios artificiais

A proposta dessa dissertação, apresentada no capítulo 4, indica que FORMVIEW gera melhores hierarquias de conceitos em domínios com atributos contínuos e discretos, de forma que essa hierarquia é capaz de representar mais fielmente o domínio e, conseqüentemente, de fazer mais inferências corretas.

O propósito da construção das bases de dados artificiais para avaliar os sistemas foi representar cenários do mundo real através de domínios com atributos discretos e contínuos. Esses cenários dizem respeito a diferentes quantidades de atributos contínuos e discretos, visto que, no capítulo 3, foi identificada uma

---

[1] A melhor categoria é encontrada baseando-se na função de avaliação heurística do sistema

predominância dos atributos contínuos sobre os discretos. É importante então destacar os casos que evidenciam esse comportamento.

Para confecção das várias situações, foram criadas 35 bases de dados experimentais, todas possuindo 100 observações. Onde a quantidade de atributos dessas bases varia de 6 a 10 atributos. De forma que, cada uma das bases terá quantidades diferentes de atributos discretos e contínuos, possuindo pelo menos um atributo de cada tipo. Em outras palavras, uma base com 6 atributos, por exemplo, poderá ter no mínimo 1(um) e no máximo 5(cinco) atributos contínuos. Quando possuir 1(um) atributo contínuo terá 5(cinco) atributos discretos, e quando possuir 5(cinco) atributos discretos terá 1(um) atributo contínuo. Com isso, serão criadas 5 bases diferentes possuindo 6 atributos.

Na geração das observações, a quantidade de valores para cada atributo discreto foi uniformemente escolhida, contanto que possuam no mínimo 2 e no máximo 4 valores. No caso dos atributos contínuos, foram definidos 4 intervalos numéricos. Os números contidos em cada intervalo foram gerados a partir de uma distribuição normal definida por uma média e um desvio padrão, ambos aleatoriamente escolhidos para cada intervalo. Cada atributo contínuo terá números pertencentes a pelo menos 2 desses intervalos, sendo os intervalos aleatoriamente escolhidos.

A seguir serão apresentados os experimentos realizados nas bases experimentais de acordo com a proposta de Dietterich.

5.3 Classificando para inferir

A eficácia dos métodos usados em SFCP depende da regularidade dos valores de atributos importantes para o domínio [Cheng & Fu, 85] [Pearl, 85]. Os experimentos realizados nessa seção têm como objetivo mostrar que a abordagem proposta nesta dissertação, implementada em FORMVIEW, é capaz de representar essa regularidade através de hierarquias, em domínios com atributos discretos e contínuos, de maneira mais eficiente que as demais abordagens aqui avaliadas.

5.3.1 Capacidade de inferência

A análise inicial da capacidade de inferência buscou avaliar os sistemas em termos da capacidade de aprendizado incremental dos mesmos. Para tanto, foi verificada a performance dos sistemas para diferentes quantidades de observações de treinamento, assim como descrito na seção 5.1. Essa análise inicial considerou 3 aspectos para os sistemas COBIT, COBWEB/3, COBWEB95 e FORMVIEW:

(a) O primeiro aspecto diz respeito ao percentual de inferências corretas ou acertos realizados pelos sistemas. Em outras palavras, consiste da capacidade de inferência propriamente dita. Foram destacados, os valores médios, mínimos, máximos e desvios padrões nas bases utilizados para o experimento.

(b) O segundo aspecto apresenta um comparativo em termos da diferença percentual dos acertos de FORMVIEW em relação aos outros sistemas. Os acertos de COBIT, COBWEB/3 e COBWEB95 foram comparados com FORMVIEW em cada base experimental. De forma que o resultado dessa comparação foi computado em termos da diferença percentual de acertos que FORMVIEW teve em relação a cada um dos demais sistemas. Para essa diferença, destacam-se também os valores médio, mínimo e máximo atingido por essa diferença. Valores positivos da diferença indicam que FORMVIEW obteve ganho percentual em relação a outro sistema, valores negativos indicam perda e 0(zero) valor igual.

(c) O terceiro aspecto apresenta um outro comparativo. Agora em termos da quantidade de bases experimentais em que FORMVIEW obteve ganho percentual em relação a outro sistema. Ou seja, paralelamente ao cálculo da diferença percentual de acerto de FORMVIEW, definida no segundo aspecto, foi armazenada a quantidade de bases em que FORMVIEW teve diferenças percentuais positivas, negativas e nulas. Denotando um “placar” entre FORMVIEW e os demais sistemas, onde é informado o percentual de bases em que FORMVIEW “ganhou”, “perdeu” ou “empatou”.

O resultado da aplicação dos sistemas nas bases experimentais foi agrupado pela quantidade de observações de treinamento usadas para realizar as inferências, assim como definido na seção 5.1. No caso, os resultados foram divididos em grupos de resultados possuindo, respectivamente, 20, 40, 60 e 80 observações de treinamento.

A tabela 11 mostra as informações para cada aspecto recém definido (a,b,c) em função do grupo de resultados e dos sistemas avaliados. O último grupo de resultados, identificado por "Geral", representa a média[1] dos 4(quatro) grupos de resultados.

Tabela 11: Resultados consolidados sistemas em bases com atributos contínuos escolhidos aleatoriamente.

Obs	Sistema	(a) % Acerto				(b) % Diferença			(c) Placar		
		Med	Min	Max	D. P.	Med	Min	Max	Ganhou	Perdeu	Empatou
20	COBIT	81,81	71,43	88,89	4,5	5	-5	23	71,4	8,6	20
	COBWEB/3	63,35	40	77,5	10,67	40	6	117	100	0	0
	COBWEB95	80,02	70	90	4,68	7	-4	29	74,3	8,6	17,1
	FORMVIEW	85,64	80	91,43	2,54	-	-	-	-	-	-
40	COBIT	95,67	83,33	100	4,96	3	-5	16	45,7	20	34,3
	COBWEB/3	69,05	56,67	86	7,17	44	16	76	100	0	0
	COBWEB95	96,89	90	100	3,28	1	-5	11	40	11,4	48,6
	FORMVIEW	98,05	90	100	2,76	-	-	-	-	-	-
60	COBIT	94,91	82,50	100	5,24	5	-6	21	65,7	2,6	31,4
	COBWEB/3	63,74	54	73,75	5,7	56	23	85	100	0	0
	COBWEB95	96,11	90	100	2,9	3	-7	9	68,6	2,9	28,6
	FORMVIEW	98,94	90	100	2,46	-	-	-	-	-	-
80	COBIT	94,67	85	100	4,83	5	-3	18	68,6	2,9	28,6
	COBWEB/3	64,53	51,11	80	7,5	56	21	80	100	0	0
	COBWEB95	98,23	90	100	2,9	1	-7	9	37,1	2,9	60
	FORMVIEW	99,25	93,33	100	1,64	-	-	-	-	-	-
Geral	COBIT	91,8	81,25	97,2	4,5	4	-2	17	77,1	13,0	9,9
	COBWEB/3	65,2	56,6	79,3	4,52	47	28	71	100,0	0,0	0,0
	COBWEB95	92,8	86,6	96,11	2,69	3	-2	8	94,3	5,7	0,0
	FORMVIEW	95,5	90	97,9	1,46	-	-	-	-	-	-

5.3.1.1 Interpretação dos resultados

O primeiro grupo de resultados, com 20 observações de treinamento, é um grupo particularmente peculiar, pois representa a capacidade que as abordagens tem em criar *"boas[2]"* hierarquias de conceitos a partir de poucas observações. Esse grupo será especialmente analisado com gráficos ilustrativos, explicando como foram interpretadas as informações da tabela 11. Os gráficos apresentados na figura 14 ilustram os as informações da tabela 11 para esse grupo de análise.

A figura 14(a) representa o aspecto (a) considerado nessa análise. Portanto apresenta os valores médio, mínimo e máximo do percentual de acerto para cada sistema. O desvio padrão desse percentual está demonstrado na figura 14(d).

Na figura 14(b) está exposto o aspecto (b). Conseqüentemente, demonstra a diferença entre os resultados de FORMVIEW e as demais abordagens. O placar de

[1] Os dados dos aspectos comparativos da análise representam a diferença percentual entre essas médias de acertos e não a média da diferença percentual dos 4 grupos de resultados.

[2] Em termos de capacidade de inferência.

FORMVIEW em relação aos demais sistemas está ilustrado na figura 14(c), assim como previsto no aspecto (c).

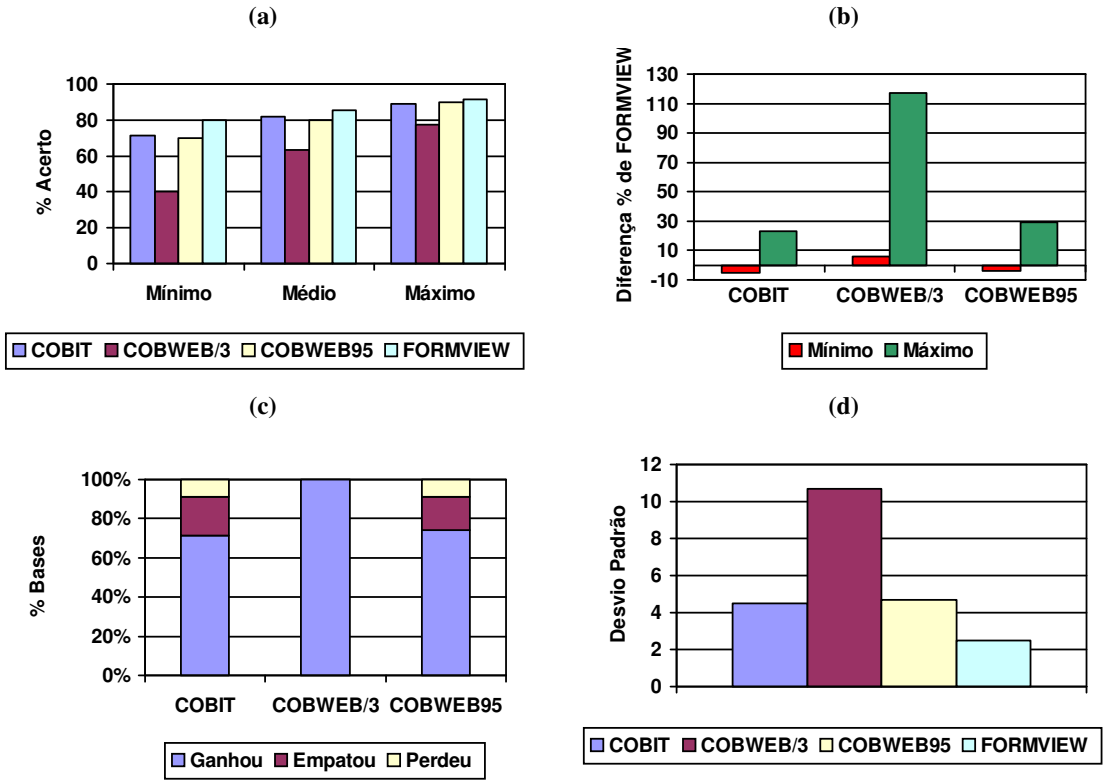

Figura 14: Gráficos de performance dos sistemas usando 20 observações de treinamento. (A) Percentual de Acerto, (B) Diferença percentual de acerto em relação a FORMVIEW, (C) Placar e (D) Desvio Padrão do Percentual de acerto.

A partir dos resultados, podemos dizer que FORMVIEW, com 20 observações ou 25% das observações de treinamento, superou os resultados dos outros sistemas, figura 14(a). O desvio padrão do percentual de acerto, figura 14(d), de FORMVIEW é inferior às outras abordagens. Esse valor baixo indica que FORMVIEW apresenta uma menor variação no percentual de acertos, de forma que pode ser visto como uma abordagem que, além de possuir resultados superiores, se apresenta mais constante.

Os casos em que FORMVIEW teve performance inferior representam menos de 9%, figura *14(c):Perdeu*, das bases avaliadas. A melhoria de performance desses casos não foi superior a 2%, figura 14(b). Por outro lado, nos casos em que FORMVIEW se sobressaiu, mais de 71% como mostra a figura *14(c):Ganhou*, a melhoria chegou a 23% em relação a COBIT, 117% em relação a COBWEB/3 e 29%

em relação a COBWEB95. FORMVIEW se apresentou superior a COBWE/3 em todos os casos.

Podemos enfim afirmar que FORMVIEW tem uma capacidade de inferência superior ao outros sistemas usando com uma menor quantidade de observações.

A interpretação dos dados do grupo de resultados com 20 observações de treinamento foi a mesma utilizada para os grupos com 40, 60 e 80 observações. Os resultados desses outros grupos apresentam algumas variações que podem ser destacadas. O placar do *"grupo de resultados 40"*, por exemplo, não se apresenta tão superior como o do grupo anterior. Essas variações, contudo, não foram consideradas nem significativas em termos quantitativos, nem representativas em termos de quantidade de bases. O que, em linhas gerais, mantém a superioridade de FORMVIEW.

O grupo de resultados "Geral" terá uma conotação conclusiva da performance geral dos sistemas, como também realizadas algumas análises adicionais. Os gráficos apresentados na figura 15 ilustram a conclusão.

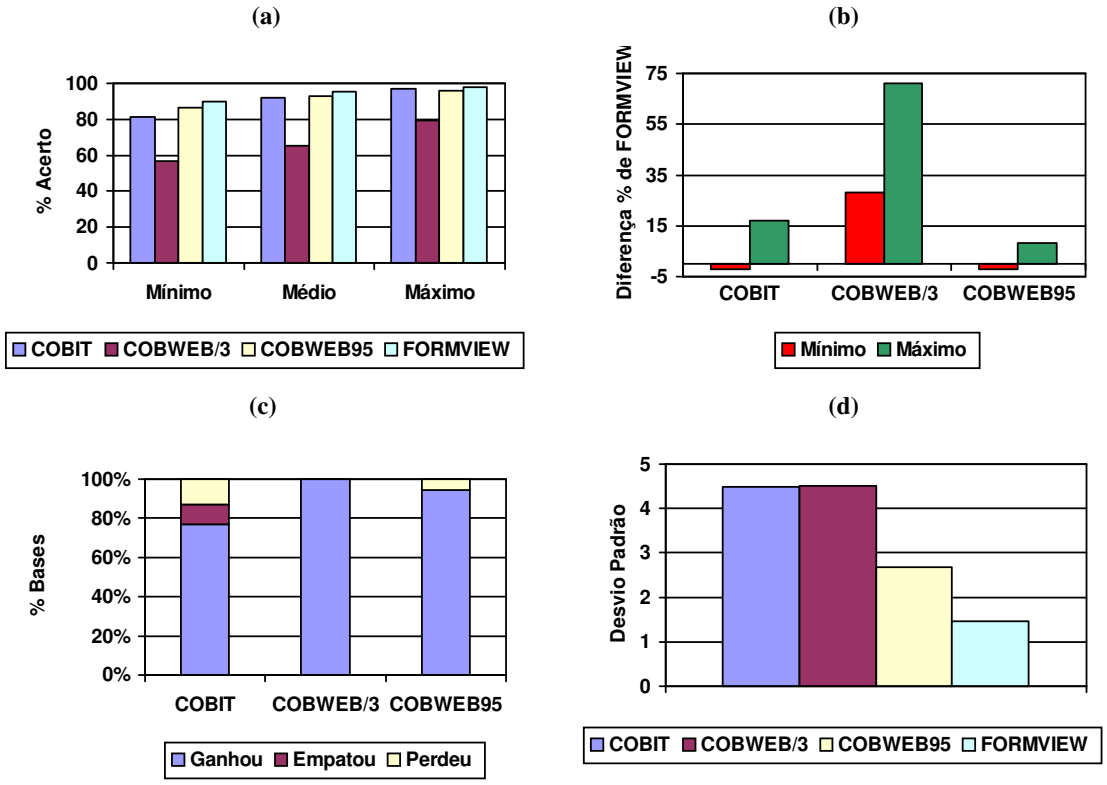

Figura 15: Gráficos de performance geral dos sistemas. (a) Percentual de Acerto, (b) Diferença percentual de acerto em relação a FORMVIEW, (c) Placar e (d) Desvio Padrão do Percentual de acerto.

O gráfico da figura 15(a) demonstra a constante superioridade de FORMVIEW em relação aos outros sistemas. Essa superioridade constante é comprovada pelo desvio padrão reduzido de FORMVIEW, figura 15(d).

O gráfico das figuras 15(b) e 15(c) mostram que a melhoria de FORMVIEW em relação a COBIT ocorreu em mais de 77% das bases utilizadas e em relação à COBWEB95 chegou a mais de 94%. Nesses casos de ganho de FORMVIEW, a diferença percentual em relação a COBWEB95 chegou a 8%. No caso de COBIT, essa diferença atingiu 17%. A superioridade em relação a COBWEB/3 é incontestável.

5.3.1.2 Convergência

Em SFCP, é desejável que o percentual de acertos de um sistema seja o mais alto possível com a menor quantidade de observações de treinamento, e que, com o aumento da quantidade de observações de treinamento, esse percentual não diminua. Esse comportamento denota um meio de medir a convergência dos sistemas para uma situação ótima[1] de acertos. O objetivo desse experimento é comparar essa convergência dos sistemas.

Para medir a convergência dos algoritmos, foram aproveitados os resultados do experimento anterior. Nesse caso, foi computada a quantidade de bases experimentais onde cada sistema obteve 100% de acerto usando 40 e 60 observações de treinamento, e permaneceu assim para as quantidades subseqüentes de observações de treinamento. Não foram considerados os casos que o sistema atingiu essa meta com 80 observações, pois é a quantidade total de observações de treinamento.

A partir da quantidade de bases que os sistemas possuem essa convergência, pode-se comparar a performance dos mesmos nesse sentido. A quantidade de bases para cada sistema está apresentada em termos percentuais em relação as 35 bases analisadas.

O gráfico da figura 16 mostra o percentual de bases que cada sistema obteve para os casos com 40 e 60 observações de treinamento.

---

[1] Considera-se a situação ótima os casos com 100% no percentual de acerto.

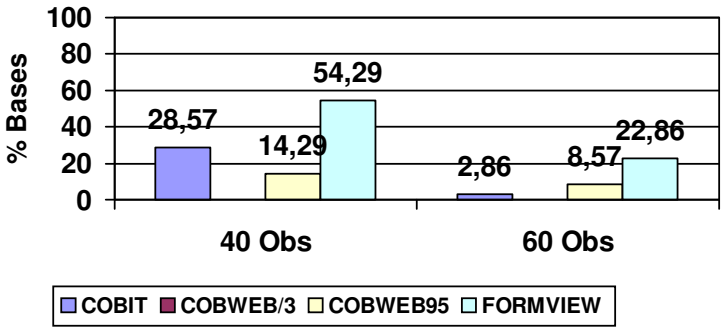

Figura 16: Divisão de performance ótima por sistema

Percebe-se que, FORMVIEW atinge 100% de inferência corretas, com 40 observações de treinamento, em mais da metade das bases. Somando esses casos com os casos onde obteve êxito com 60 observações, FORMVIEW totaliza mais de 76% das bases experimentais.

5.3.2 Diferentes quantidades de tipos de atributos

Os experimentos descritos anteriormente indicam que FORMVIEW tem uma capacidade de aprendizado superior em relação aos demais sistemas. Outro aspecto que será discutido refere-se a avaliação dos algoritmos em termos das diferentes quantidades de atributos discretos e contínuos. O objetivo dessa segunda análise é procurar identificar alguma relação entre a performance dos sistemas e a quantidade de atributos contínuos de um domínio.

Nesse outro experimento foram utilizadas as mesmas 35 bases experimentais. O percentual de acerto será resultado da média de percentual de acerto da avaliação dos sistemas nas diferentes quantidades de observações de treinamento. Esse valor representa o percentual médio de inferências corretas de um sistema para uma base experimental. Cada sistema possuirá um percentual médio de acerto para cada base analisada. Para cada base, foi calculado o percentual de atributos contínuos em relação a todos atributos que cada base possui.

Através do cálculo do coeficiente de correlação entre o percentual médio de acertos e o percentual de atributos contínuos, pôde-se perceber uma tendência de variação do percentual médio de acertos e a quantidade de atributos contínuos das

bases. A tabela 13 a seguir apresenta o valor do coeficiente de PEARSON entre o percentual de acerto e o percentual de atributos contínuo para cada sistema.

Tabela 12: Coeficiente de correlação de PEARSON entre percentual de acerto e proporção de atributos contínuos.

Sistema	PEARSON
COBIT	-0,90
COBWEB3	0,84
COBWEB95	-0,69
FORMVIEW	-0,63

Em outras palavras, constatou-se que para COBIT, COBWEB95 e FORMVIEW, há uma forte correlação inversa entre a quantidade de atributos contínuos e o percentual de acerto. Ou seja, há um forte indício que quanto maior a quantidade de atributos contínuos para um domínio, menor será a quantidade de inferências corretas sobre esse domínio.

Embora tenha sido identificada uma tendência de queda de performance em relação à quantidade de atributos contínuos, os resultados do primeiro experimento demonstram que FORMVIEW mantém-se superior na maioria das bases experimentais. Esses resultados indicam que FORMVIEW está menos predisposto a tendência apresentada.

No caso de COBWEB/3, identificou-se que existe uma forte correlação direta entre a performance do sistema e a quantidade de atributos contínuos. Em outras palavras, quanto maior a quantidade de atributos contínuos, melhor a performance de COBWEB/3. Pode-se imaginar que em alguma situação[1] COBWEB/3 superaria FORMVIEW. Contudo, nas bases experimentais utilizadas, o percentual de atributos contínuos variou de 10% à 90% e a diferença mínima geral entre FORMVIEW e COBWEB/3 foi de 28%. Portanto, esses dados indicam que mesmo com uma tendência de aumento de performance com o aumento da quantidade de atributos contínuos, FORMVIEW ainda teria resultados superiores.

5.3.3 Outras características dos atributos

O experimento anterior mostrou uma forte correlação da performance dos sistemas e a quantidade de atributos contínuos de um domínio. Vale ressaltar que

[1] Uma base com um alto percentual de atributos contínuos, por exemplo.

essa correlação indica uma tendência em relação quantidade de atributos contínuos. Haja vista que houve casos, para COBWEB95, por exemplo, que o sistema obteve melhores resultados em uma base com maior quantidade de atributos contínuos que em outra com menor quantidade.

Com isso, pode-se imaginar que a quantidade de atributos contínuos não é o único fator que diferencia a performance dos sistemas. Ou seja, embora apresente uma tendência, a quantidade de atributos contínuos não determina que a performance será maior ou menor.

Um aspecto importante que deve ser considerado na formação de conceitos está ligado à relevância de um atributo para o domínio. Trabalhos na área da psicologia cognitiva [Seifert, 89] e de aprendizado de máquina [Stepp, 86] [Merckt & DeCaestecker, 94] defendem firmemente a importância de determinar a relevância de um atributo para um domínio.

Uma das maneiras de medir a relevância de um atributo é através de sua dependência em relação aos outros atributos [Fisher, 87]. A equação 17 a seguir define uma métrica para medir o grau de dependência de um atributo *Am* em relação as outros atributos *Ai*. Pesquisas realizadas na área de formação de conceitos probabilísticos [Fisher, 87] [Furtado, 98] mostraram que atributos relevantes são os mais beneficiados, em termos de inferências sobre seus valores, pela classificação utilizando hierarquias de conceitos.

$$\frac{1}{n}\sum_{i}\sum_{ji} P(A_i = V_{ij})\sum_{jM}\left[P(A_M = V_{M_{jm}} \mid A_i = V_{ij})^2 - P(A_M = V_{M_{jm}})^2\right]$$

Figura 17: Métrica para determinar o grau de dependência de um atributo discreto em relação aos demais.

O interesse desse trabalho na relação de atributos relevantes com poder de inferência é verificar qual o impacto de atributos contínuos relevantes na performance dos sistemas aqui avaliados. Buscando esclarecer esse ponto, outro experimento foi realizado. Para o novo experimento, foi necessária a criação de outras bases experimentais.

Inicialmente foi criada 1(uma) base com 100 observações. As observações dessa base são representadas por 10 atributos discretos onde cada atributo é

identificado por uma letra de *A* a *J*. A quantidade de valores de cada atributo discreto foi aleatoriamente escolhida, sendo de 2 a 4 valores.

Essa base serviu como modelo para a criação de outras bases, onde efetivamente serão realizados os experimentos. Cada atributo discreto dessa base modelo foi transformado em contínuo dando origem a uma base experimental com atributos mistos. Assim, têm-se dez bases experimentais cada uma com um atributo contínuo diferente.

Para transformar um atributo discreto em contínuo foi feita a atribuição de uma média e um desvio padrão a cada valor de discreto do mesmo. Foram escolhidos números aleatórios para a média atribuída aos valores de um atributo discreto, de forma que a distância entre duas médias seja superior aos seus respectivos desvios padrões. O desvio padrão adotado foi de 10% em relação à média escolhida.

Os valores discretos nas observações serão substituídos por valores contínuos. Cada valor contínuo é, aleatoriamente, escolhido a partir de uma distribuição normal definida pela média e desvio padrão que foi atribuído ao valor discreto sendo substituído.

Essa forma de transformação de um atributo discreto em contínuo permite que seja mantida a correlação entre os atributos. Como o objetivo desse trabalho não é estabelecer a dependência entre atributos discretos e contínuos, é possível utilizar a equação 17 para calcular a dependência de um atributo contínuo. Assim, pode-se dizer que a dependência de um atributo contínuo em relação aos outros é a mesma de seu atributo discreto de origem.

Ao final, cada base será identificada pelo atributo discreto que foi convertido e a dependência do atributo discreto convertido está apresentada na tabela 13. Onde o atributo *J* possui maior dependência em relação aos outros, enquanto os atributos *A* e *F* são os que possuem menores dependências.

Tabela 13: Coeficiente de correlação de PEARSON entre percentual de acerto e proporção de atributos contínuos.

Atributo:	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**
Dependência:	0,27	0,35	0,30	0,34	0,33	0,27	0,33	0,32	0,35	0,37

As 10 bases foram submetidas aos algoritmos segundo a metodologia definida na seção 5.1. Para esse experimento foi utilizado o valor do percentual médio de

acerto, calculado a partir do resultado dos sistemas em diferentes quantidades de observações de treinamento. Ou seja, cada sistema possuirá um percentual médio de acerto para cada base analisada.

Calculando o coeficiente de correlação de PEARSON entre a dependência dos atributos contínuos e o percentual de acerto dos sistemas nas respectivas bases, temos os seguintes resultados apresentados na tabela 14.

Tabela 14: Coeficiente de correlação de PEARSON entre percentual de acerto e dependência dos atributos.

Sistema	PEARSON
COBIT	0,54
COBWEB3	0,58
COBWEB95	0,52
FORMVIEW	0,08

Os resultados mostram uma considerável relação entre a performance dos outros sistemas avaliados e o grau de dependência de um atributo contínuo. Isso quer dizer que os outros sistemas avaliados, para domínios com atributos contínuos e discretos, embora continuem favorecendo a inferência em atributos relevantes, estão mais suscetíveis a variações de performance quando os atributos relevantes são contínuos.

A proposta aqui definida, implementada em FORMVIEW, não apresenta essa mesma relação. De fato, a relação presente para FORMVIEW é considerada fraquíssima (0,08).

5.4 Efeitos da definição da margem de erro

Essa seção visa avaliar o impacto de diferentes margens de erro para os sistemas. A margem de erro é um parâmetro definido pelo usuário que, intuitivamente, representa o quanto, em termos percentuais, é permitido distanciar-se do verdadeiro valor de um atributo contínuo para que uma inferência seja considerada correta.

É importante que esse valor seja determinado pela necessidade de precisão dos atributos contínuos de um domínio. Por exemplo, um domínio para diagnóstico de doenças deve exigir uma precisão de valores maior que um domínio para classificação de automóveis. Em SFCP, é desejável que a abordagem faça uso da margem de erro de forma que atenda a necessidade de precisão de valores. Outra

interpretação dessa margem pode resultar na criação de conceitos que não representam o domínio e, conseqüentemente, terão baixa capacidade de inferência.

Na análise do comportamento dos sistemas em diferentes situações de margem de erro, utilizamos as 35 bases definidas no primeiro experimento do capítulo. Contudo, as bases foram submetidas aos sistemas utilizando diferentes margens de erro. Inicialmente, alterou-se a margem de erro para 10% e em seguida para 30%. Uma vez que para os resultados do primeiro experimento foi utilizada uma margem de erro de 20%.

A comparação dos resultados dos sistemas foi realizada em termos do percentual médio de acerto de inferências, calculado com base nas avaliações com diferentes quantidades de observações de treinamento. Aqui também foram consideradas as 03 categorias de análise definidas no primeiro experimento. Os resultados dos sistemas para cada margem de erro distinta estão apresentados na tabela 15.

Tabela 15: Resultados consolidados da aplicação dos sistema com diferentes margens de erro.

Margem de Erro	Sistema	Categoria 1				Categoria 2			Categoria 3		
		% Acerto				% Diferença			Placar		
		Med	Min	Max	D. P.	Med	Min	Max	Ganhou	Perdeu	Empatou
10%	COBIT	76,64	56,25	93,57	10,8	6	-3	22	82	9	9
	COBWEB/3	56,69	51,75	61,94	2,9	43	15	66	100	0	0
	COBWEB95	77,23	60,50	90,25	8,25	5	-1	11	94	6	0
	FORMVIEW	80,90	62,50	93,00	8,76	-	-	-	-	-	-
20%	COBIT	91,8	81,25	97,2	4,5	4	-2	17	77,1	13,0	9,9
	COBWEB/3	65,2	56,6	79,3	4,52	47	28	71	100,0	0,0	0,0
	COBWEB95	92,8	86,6	96,11	2,69	3	-2	8	94,3	5,7	0,0
	FORMVIEW	95,5	90	97,9	1,46	-	-	-	-	-	-
30%	COBIT	92,14	81,67	98,33	4,62	4	-2	14	66	17	17
	COBWEB/3	65,61	56,8	74,4	4,72	46	26	71	100	0	0
	COBWEB95	94,76	88,33	96,75	1,8	1	-2	4	60	26	14
	FORMVIEW	95,46	88,75	97,86	1,63	-	-	-	-	-	-

Os resultados mostraram que FORMVIEW mantém a superioridade com diferentes margens de erro. Contudo, o comportamento do placar merece alguns comentários.

Percebe-se uma diminuição da liderança de FORMVIEW com o aumento da margem de erro. Os resultados com 30% de margem de erro, por exemplo, apresentam que FORMVIEW perdeu em 17% dos casos em relação a COBIT e em 26% dos casos em relação a COBWEB95.

Embora possam ser considerados percentuais elevados de casos, ambos sobressaíram-se à FORMVIEW em, no máximo, 2% de diferença de acertos. Enquanto nos casos em que FORMVIEW foi superior, mais 60% dos casos, essa diferença em relação à COBWEB95 chegou a 4% e em relação a COBIT chegou a 14%.

Contudo, o que deve ser ressaltado é que a diminuição dos casos em que FORMVIEW tem melhor performance acontece com o aumento da margem de erro. Percebe-se que FORMVIEW, nesses casos, mantém um percentual médio de acerto elevado, mais de 95%. Isso nos leva concluir que a diminuição dos casos de superioridade de FORMVIEW acontece devido à melhora das outras abordagens e não devido a queda de performance de FORMVIEW.

Portanto, esses resultados indicam que COBIT e COBWEB95 passaram a representar melhor os domínios quando utilizam uma margem de erro maior. O que indica uma dependência da margem de erro no que diz respeito não somente à precisão nos dados, mas também à estrutura hierárquica.

A partir desses fatos, pode-se imaginar que, em casos que necessitem uma margem de erro reduzida, COBIT e COBWEB95 deixariam a desejar enquanto FORMVIEW apresentara-se menos predisposto ao fator margem de erro.

5.5 Custo de FORMVIEW

A proposta de uma nova abordagem baseada em outra sugere um novo custo computacional da nova proposta. A análise desse custo faz-se importante, pois mesmo que a proposta apresente melhores resultados, seu custo elevado pode inviabilizar sua utilização.

Para calcular o custo de FORMVIEW para incorporar uma única observação em uma hierarquia de conceitos preexistente, considere que *B* é a quantidade média de categorias por nível da hierarquia e que *N* é o número de observações já incorporadas.

Considere ainda que o domínio em questão é representado por uma quantidade de atributos discretos *D* e uma quantidade de atributos contínuos *C*. A quantidade média de valores dos atributos discretos é *V*. No caso dos atributos contínuos, para efeito de contagem de comparações, assumiremos que um atributo

contínuo possui somente um valor, que seria a distribuição normal dos valores do atributo.

No processo de verificação da melhor categoria de encaixe, FORMVIEW realiza um ciclo onde a nova observação irá atualizar as informações para cálculo das probabilidades condicionais do conceito sendo avaliado enquanto *Category Utility* avalia todos os conceitos do nível sendo analisado. Portanto, o custo desse ciclo é aproximado por $O(B(DV+C))$ e deve ser realizado para cada categoria do nível sendo avaliado para que a melhor categoria possa ser escolhida. Ou seja, o custo total de verificação da melhor categoria de encaixe é aproximado por $O(B^2(DV+C))$.

Depois da verificação da melhor categoria, existem ainda a avaliação da criação de uma nova categoria (mais um ciclo), a avaliação da junção das duas melhores categorias de encaixe (mais um ciclo), a divisão da melhor categoria de encaixe (mais B ciclos) e a aplicação da proposta desse trabalho que feita sobre as duas melhores categorias de encaixe (mais um ciclo). Como esses ciclos são adicionais e estamos trabalhando com informações médias para cálculo do custo, o valor $O(B^2(DV+C))$ continua sendo uma aproximação válida para o custo de avaliação de uma nova observação a cada nível da hierarquia.

Em geral, o processo de incorporação avalia a nova observação até o último nível da hierarquia, pode-se aproximar a profundidade média da hierarquia por $log_B n$. De forma que o custo total de FORMVIEW para incorporação de uma nova observação em uma hierarquia de conceitos existente é $O(B^2 \times log_B N \times (DV+C))$.

Enfim, percebe-se que a proposta implementada em FORMVIEW mantém o baixo custo computacional inerente aos sistemas baseados em COBWEB.

5.6 Domínio Real

O experimento final analisa o comportamento de FORMVIEW numa situação real. Para esse experimento foram usadas 4(quatro) bases de dados de domínio público em UCI ML Repository (http://www.ics.uci.edu/~mlearn/MLRepository.html).

A idéia de analisar as abordagens em ambientes reais com dados mistos consiste em verificar seus desempenhos frente a situações com dados incorretos, erros de precisão, ausência valores de atributos, atributos irrelevantes etc. Enfim,

situações que farão parte do dia-a-dia do uso dos sistemas.

O método de avaliação de performance dos sistemas continua o mesmo usado na seção anterior. As observações de cada base foram separadas em conjunto de treinamento e conjunto de teste. O conjunto de treinamento foi formado por aproximadamente 80% das observações do domínio, enquanto o conjunto de teste pelos 20% restantes. Da mesma forma, o conjunto de treinamento foi subdividido em 4, cada um representando um crescimento de 25% no número de observações. A margem de erro utilizada para verificação de acerto dos atributos contínuos foi de 20%.

As bases de dados do UCI ML Repository foram aleatoriamente escolhidas, somente tendo como requisito possuírem atributos discretos e contínuos. As bases escolhidas foram: AUTO-MPG, BRIDGES, ECHOCARDIOGRAM, e HEART DISEASE. A tabela 16 a seguir demonstra as características de cada base de dados em termos das quantidades de atributos discretos e contínuos, tamanho do conjunto de treinamento e de teste, e o conteúdo das bases.

Tabela 16: Informações de bases de dados do UCI ML Repository.

Base	Discretos	Numéricos	Treinamento	Teste	Conteúdo
AUTO-MPG	3	5	336	70	Consumo de automóveis
BRIDGES	8	3	88	20	Pontes
ECHOCARDIOGRAM	3	7	108	24	Resultados de Ecocardiogramas
HEART DISEASE	9	5	252	51	Pacientes com problemas cardíacos

Os resultados da aplicação dos sistemas nessas bases serão apresentados em gráficos. Os gráficos demonstram através de barras verticais as diferenças de performance entre os sistemas avaliados em termos de poder de predição de valores.

O eixo vertical do gráfico indica o percentual de inferências corretas, enquanto o eixo horizontal representa o aumento na quantidade de observações usadas no subconjunto de treinamento. De modo que, quanto maior a barra, maior o número de inferências corretas sobre o domínio.

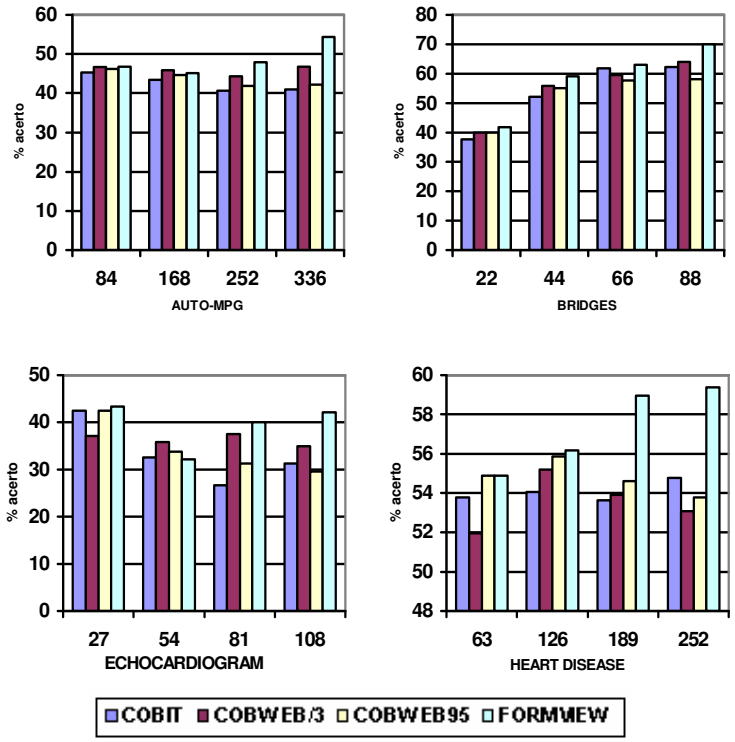

Figura 18: Performance dos sistemas em bases de dados reais.

De acordo com os gráficos apresentados, FORMVIEW demonstrou resultados satisfatórios nas 04(quatro) bases reais avaliadas. Tendo se destacado na base HEART-DISEASE.

5.7 Conclusão

Esta seção apresentou experimentos medindo a capacidade de inferência de FORMVIEW sempre comparando com abordagens correlatas. FORMVIEW apresentou resultados satisfatórios sendo aplicado em domínios com atributos discretos e contínuos.

Percebeu-se uma relação entre a performance das abordagens, em termos de poder de previsão, e a quantidade de atributos contínuos de um domínio. Apesar do indício da relação, FORMVIEW apresentou-se menos predisposto e com resultados superiores.

Outra relação percebida ocorreu entre a performance das outras abordagens, em termos de poder de previsão, e a grau de dependência de atributos contínuos em relação aos demais atributos. De forma que COBIT, COBWEB/3 e COBWEB95,

apresentaram melhores resultados quando o atributo contínuo é relevante, no sentido de apresentar um alto grau de dependência do demais atributos. Nesse caso, FORMVIEW mostrou-se menos sensível a essa relação possibilitando atingir melhores resultados.

Embora tenham acontecido situações onde a melhora não tenha sido tão significativa, não foram consideradas relevantes a ponto de comprometer a proposta de FORMVIEW. Contudo, seria interessante um estudo mais aprofundado dos motivos que levaram a ocorrência desses casos. Assim como também merece atenção um estudo da relação entre a performance dos sistemas e a quantidade de atributos contínuos dependentes. Acredita-se que nesses casos FORMVIEW apresente ainda melhores resultados.

Nesse capítulo, ficou ainda mais claro a necessidade de se definir corretamente o parâmetro *acuity* para COBWEB/3, uma vez que sua performance deixou muito a desejar em relação às demais.

Na próxima seção a conclusão é efetuada, sendo discutidas algumas limitações de nossa abordagem, assim como propostas de trabalhos futuros.

6 Conclusões, Limitações e Sugestões.

FORMVIEW é um algoritmo incremental de aprendizagem indutiva que realiza a formação de conceitos em domínios com atributos discretos e contínuos. O problema básico das abordagens nesses cenários é o tratamento em conjunto de diferentes tipos de atributos, pois se deve levar em consideração a equivalência de valores das diferentes funções de avaliação de cada tipo de atributo.

O capítulo 3 mostrou que a contribuição das diferentes funções de avaliação em relação a função de avaliação geral da hierarquia acontece devido à suas diferenças de amplitude de resultados. A análise da velocidade de convergência do resultado das funções de avaliação mostrou que a diferença de contribuição, para a função de avaliação geral, pode acontecer a cada nova entidade submetida aos algoritmos.

Esse trabalho apresenta uma abordagem que considera essa disparidade baseando-se no ganho de capacidade de inferência que é diferenciado para cada tipo de atributo, assim como demonstrado no capítulo 4. Experimentos de avaliação de nossa abordagem, apresentados no capítulo 5, mostraram resultados satisfatórios em relação aos objetivos inicialmente almejados.

Em domínio com entidades representadas usando esses dois tipos de atributos, FORMVIEW sobressaiu-se às demais abordagens mostrando-se estar menos predisposto a fatores inerentes a esses ambientes, como diferentes quantidades de atributos discretos e contínuos, e atributos contínuos com alto grau de dependência de outros atributos. No que diz respeito a parâmetros informados pelo usuário, como a margem de erro, FORMVIEW também se destacou apresentando comportamento mais consistente em relação à COBWEB95 e COBIT.

6.1 Contribuições

Dentre as contribuições à área de formação de conceitos, pode-se destacar as seguintes.

1. O estudo das diferentes funções de avaliação para a heurística geral apresentou um comportamento diferenciado e desbalanceado dessas funções, o que até então era desconsiderado.
2. Nossa proposta utilizando o ganho individual dos atributos em termos da capacidade de inferência, revelou um conhecimento adicional a ser considerado incrementando a qualidade dos conceitos construídos.
3. Sua aplicação juntamente com a modificação de um operador de construção da hierarquia mostrou uma abordagem para minimizar o problema da predominância de um tipo de atributo em relação a outro. Essa abordagem, sem a necessidade da redefinição completa da heurística de busca, apresenta uma forma de otimização de métodos existentes.

Ainda como resultado desse trabalho, pode-se citar a implementação de vários algoritmos de formação incremental de conceitos probabilísticos em um aplicativo com interface gráfica, SmartBASE [Rebouças e Furtado, 2000].

O uso de uma ferramenta gráfica, como SmartBASE, para auxiliar o processo de aprendizado dos SFCP constitui um facilitador para o entendimento dos problemas relacionados com formação automática de conceitos [Rebouças e Furtado, 98]. No nosso caso, SFCP a partir de entidades representadas por atributos discretos e contínuos.

A existência de vários algoritmos implementados e o ambiente gráfico de SmartBASE permitiram sua aplicação em dois trabalhos na área de aquisição de conhecimento.

No primeiro trabalho, SmartBASE serviu como ferramenta de auxílio à gestão do conhecimento [Gomes, 2002]. No desenvolvimento de uma metodologia para facilitar a gestão do conhecimento na secretaria da fazenda do estado do Ceará, SmartBASE serviu para classificar contribuintes sob a perspectiva da tributação e da arrecadação. Essa aplicação permitiu a análise da relação entre essas duas perspectivas [Gomes et. al., 2001].

Para o segundo trabalho, SmartBASE foi utilizado como base para o desenvolvimento de uma abordagem de formação de conceitos probabilísticos com o auxílio de cores, permitindo a interferência do usuário a partir do acompanhamento visual do processo de classificação [Cavalcante, 2003].

6.2 Limitações

Experimentos mostraram a eficiência e a importância de se considerar as diferenças entre as funções de avaliação. Apesar dos satisfatórios resultados iniciais, esta proposta apresenta algumas limitações que devem ser consideradas quando aplicada em um domínio desconhecido.

FORMVIEW assume que os valores contínuos estão distribuídos segundo a curva normal, o que pode não ser verdade em alguns casos. Essa distribuição foi adotada devido ao caráter incremental da abordagem que torna impossível fazer algum teste de normalidade nos valores numéricos.

A idéia do ganho individual de capacidade de inferência foi possível, pois as funções são baseadas em conceitos probabilísticos. A aplicação da mesma idéia em ambientes com funções distintas requer um estudo mais aprofundado.

O ganho individual para atributos contínuos apresentou bons resultados por que a heurística escolhida possuía amplitude de resultados bem definida. A aplicação dessa idéia em COBWEB/3, por exemplo, necessitaria de ajustes adicionais, pois como não existe limite superior para os resultados de sua função de avaliação, a predominância dos atributos contínuos poderia continuar existindo.

6.3 Trabalhos futuros

Durante o desenvolvimento desse trabalho foram identificados alguns pontos que constituem avanços para a área de formação de conceitos probabilísticos, mas que não puderam ser atacados nesse trabalho.

O estudo da problemática da predominância das diferentes funções de avaliação permitiu o desenvolvimento da solução apresentada. Contudo, o esclarecimento do problema também permitiu a consideração de outras abordagens.

Outra possível solução foi vislumbrada através da ponderação de cada função de avaliação distinta. A idéia de um peso para as funções visa balancear a contribuição das mesmas na função de avaliação geral. Para tanto, a definição do valor desse peso tem primordial importância para a eficácia da abordagem.

Foi verificado que a função de avaliação dos atributos contínuos possui amplitude de resultados maior que a dos atributos discretos. Baseando-se nesse

fato, definiu-se um peso para a função de avaliação dos atributos discreto a partir da probabilidade complementar de ocorrência dos atributos discretos. Esse valor complementar de probabilidade foi uma tentativa de representar a diferença de amplitude de resultados e, assim, igualar a contribuição das funções.

De fato, experimentos mostraram que, em determinados casos, a capacidade de inferência dessa abordagem apresentou-se superior em relação a outras propostas. No entanto, a existência de casos em que a performance não foi melhor demonstra que o peso utilizado não representa uma medida genérica nem adaptativa para a função de avaliação dos atributos discretos. Embora os casos de melhor resultado tenham motivado o avanço dessa abordagem, seu significado intuitivo não ficou muito claro. Portanto, optou-se por avaliar outras alternativas. Maiores detalhes dessa implementação e dos experimentos podem ser encontrados em [Rebouças e Furtado, 2003].

A proposta definida nessa dissertação, considerada em seguida, se apresentou mais robusta tanto no sentido da representação intuitiva como em capacidade de inferência. Experimentos mostraram que a proposta atual apresentou resultados melhores e mais constantes quando comparados com a abordagem de uso do peso.

Entretanto, alguns testes realizados com uma terceira abordagem, criada a partir da combinação da proposta de uso do peso com a proposta de ganho individual de capacidade de inferência, apresentaram performance superior à proposta atual. Todavia não tão constantes, pois permanece com o mesmo problema da definição mais precisa do valor do peso. Deste modo, acredita-se que a combinação das duas propostas constitua uma otimização para FORMVIEW. Porém, faz-se necessário um estudo mais aprofundado visando sintonizar o peso com o comportamento das diferentes funções de avaliação.

Outro ponto identificado que pode representar um avanço para a formação de conceitos probabilísticos é o uso do ganho individual de capacidade de inferência dos atributos. Esse fator, que até então era desconsiderado, representou um conhecimento no processo de classificação que deveria ser considerado. Sua eficácia vislumbrou a modificação da heurística de Category Utility, por exemplo, visando aprimorar algoritmos existentes em domínios com entidades representadas somente por um tipo de atributo.

Essa idéia vem do fato que em *Category Utility* a medida de qualidade para escolha da melhor operação a ser realizada na hierarquia é baseada na partição toda. Ou seja, a variação individual na capacidade de inferência de cada atributo ou de cada conceito passa desapercebida.

Em vista disso, acredita-se haver situações em que, embora o resultado de *Category Utility* seja superior para uma partição, intuitivamente a melhor opção seja outra. Considere, por exemplo, duas partições geradas por operações de encaixe de uma nova observação. Imagine que a aplicação do algoritmo na primeira partição tem resultado de *Category Utility* maior que a aplicação na segunda partição. Porém, percebe-se que o conceito, na primeira partição onde a nova observação foi encaixada, promoveu um ganho de capacidade de inferência, para os atributos, inferior ao ocorrido no conceito de encaixe da segunda partição. Esse exemplo mostra um caso entre operações de encaixe, mas que pode acontecer entre uma operação de encaixe e junção, por exemplo. Enfim, abordagens que utilizam somente um tipo de atributo poderiam também estar se beneficiando da idéia.

Os pontos apresentados podem ser vistos como otimização da proposta atual. Contudo, trabalhos de avanço na formação de conceitos probabilísticos podem ser desenvolvidos visando representar conceitos a partir de outros tipos de atributos, como datas, por exemplo. Domínios com vários tipos de atributos representam o mundo real com maior fidelidade.

Anexo I – SmartBASE

Introdução

O software SmartBASE é uma ferramenta visual de auxílio a aquisição de conhecimento em bases de dados. SmartBASE foi fruto de um trabalho de pesquisa desenvolvido pelo grupo de pesquisa de Inteligência Artificial da Universidade de Fortaleza (UNIFOR). Seu desenvolvimento foi motivado pela exploração de métodos de aquisição automática de conhecimento, umas das linhas de pesquisa desse grupo.

O desenvolvimento de uma ferramenta que suporte todo processo de aquisição de conhecimento em bases de dados é uma forma de aumentar a compreensão e identificar os problemas do tema. As fases do processo de aquisição de conhecimento a partir de bases de dados que SmartBASE auxilia são: Preparação dos dados, extração do conhecimento e sua respectiva análise.

Na etapa de preparação dos dados, foi disponibilizada uma forma mais significativa de identificar as dados, mascarando para o usuário as convenções tradicionalmente usadas no desenvolvimento de aplicações, como o uso de siglas para representar os campos nos bancos de dados, por exemplo.

Na fase de extração de conhecimento, foram implementadas três formas de aquisição automática de conhecimento: extração de árvores de decisão, formação de conceitos e formação de conceitos em múltiplas perspectivas.

No caso do processo de análise do conhecimento extraído, para cada forma de aquisição de conhecimento procurou-se fazer uma representação visual do conhecimento encontrado.

O objetivo desse anexo é apresentar, em linhas gerais, o potencial de SmartBASE, como uma ferramenta de aquisição de conhecimento esclarecendo o auxílio que foi dado para o desenvolvimento desse trabalho. Ou seja, essa descrição do software não pretende apresentar detalhadamente todas as funcionalidades da ferramenta.

A seção a seguir apresentará o funcionamento de SmartBASE. Em seguida será feita uma conclusão onde serão abordados aspectos finais da da ferramenta.

Funcionamento

O processo de aquisição de conhecimento a partir de bases de dados utilizando SmartBASE, segue, em linhas gerias, as seguintes etapas. Inicialmente, é preciso definir o domínio onde será realizada a extração do conhecimento. Em seguida, as propriedades do domínio devem ser informadas, como os atributos e seus respectivos tipos. No caso de atributos discretos, é necessário ainda informar os valores que o atributo pode assumir. Depois de definido o domínio, escolhe-se o algoritmo que será utilizado para aquisição do conhecimento. Por fim, o conhecimento extraído fica disponível para análise. Pode-se dizer que esse é um procedimento padrão para o processo de aquisição de conhecimento. Contudo, a contribuição de SmartBASE são algumas ferramentas que auxiliam e facilitam a execução dessas tarefas.

Preparação dos dados

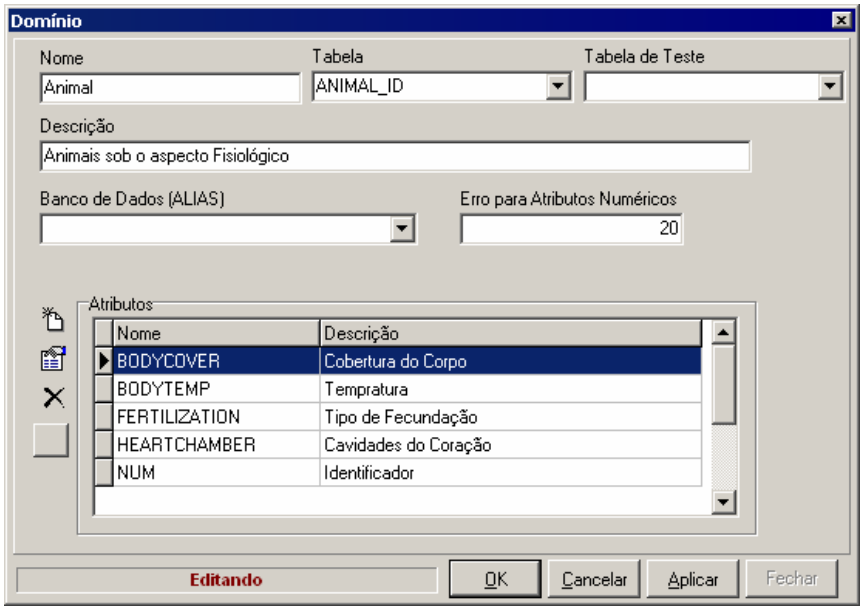

Figura 19: Tela de preenchimento de informações sobre domínio.

Na definição do domínio, assume-se que este é representado por uma tabela de um SGBD. Uma vez que o domínio é representado por um tabela, conseqüentemente, os registros serão as observações das entidades. Algumas informações adicionais sobre o domínio são utilizadas para facilitar a posterior análise do conhecimento. As informações adicionais são:

- Nome: Apelido significativo para o domínio;
- Tabela: Nome da tabela que representa o domínio no banco de dados;
- Tabela de Teste: Quando as informações para teste do conhecimento estão em outra tabela;
- Descrição: Espaço para breve descrição do domínio;
- Banco de Dados: Conexão ODBC do banco de dados onde está a tabela, e
- Erro para atributos numéricos: Margem de erro *default* para atributos contínuos.

A tela de SmartBASE usada para preenchimento dessas informações está apresentada na figura 19.

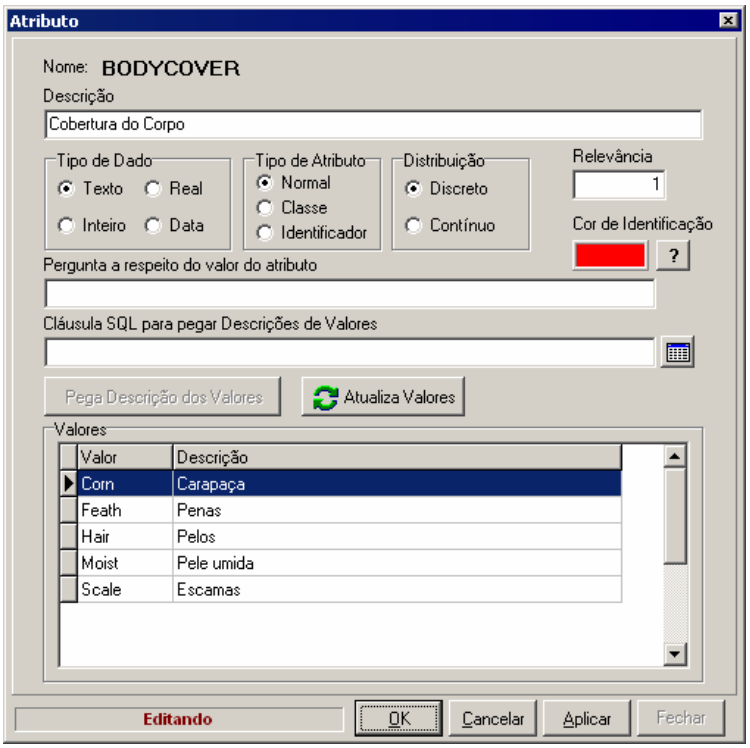

Figura 20: Tela do SmartBASE de características de um atributo.

Os campos das tabelas representam os atributos das observações. De forma que, os valores dos atributos discretos são todos os valores, para o atributo, encontrados nas observações. Para os atributos, também são utilizadas algumas informações adicionais, como ilustrado na figura 20. Essas informações para atributos são:

- Nome: O próprio nome do campo que o atributo representa.

- Descrição: Breve descrição significativa para o atributo;
- Tipo de Dado: Se o dado é texto, número ou data;
- Tipo do atributo: Se o atributo é um identificador de observação, identificador de classe para a observação ou normal.
- Distribuição do atributo: Se o atributo é discreto ou contínuo;
- Relevância: Permite dar um peso maior ao atributo (não implementado ainda);
- Cor: Atribui uma cor ao atributo para identifica-lo visualmente no processo de análise do conhecimento extraído.
- Pergunta: Como deve ser feita a pergunta para o usuário para saber um valor para o atributo. Usado no *Shell* da árvore de decisão.
- Cláusula SQL: Utilizada para pegar automaticamente a descrição dos valores do atributo, caso os valores sejam chave estrangeira de outra tabela.

A informação adicional requerida para os valores dos atributos discretos é somente uma descrição do significado do valor. Essa é uma informação importante no momento de análise do conhecimento, visto que, em geral, são utilizados códigos para representar os valores. A figura 21 ilustra a tela de cadastro da descrição de um valor discreto.

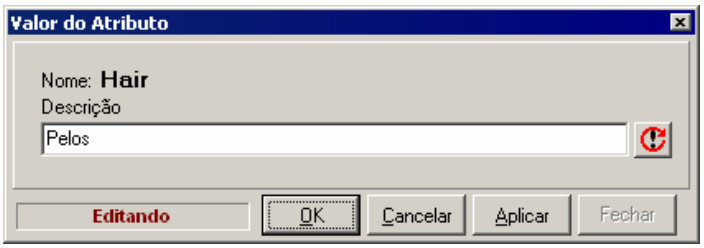

Figura 21:Tela de cadastro da descrição de um valor discreto.

Vale ressaltar que todas essas informações precisam ser preenchidas somente uma vez. Ou seja, análises posteriores aproveitam esses dados. Ainda assim, o preenchimento dessas informações pode se tornar cansativa, caso o domínio possua muito atributos com muitos valores. Em vista disso foi desenvolvido um assistente de preenchimento que facilita esse cadastro, pois muitas dessas informações podem ser retiradas automaticamente do banco de dados. Por exemplo, o nome da tabela, os nomes dos atributos, os tipos dos atributos, sua distribuição e os valores dos atributos discretos.

O preenchimento de todas as informações não é necessário para o processo de aquisição do conhecimento. O verdadeiro ganho com o preenchimento dessas

informações acontece no processo de análise do conhecimento, pois serão apresentadas as informações personalizadas. Portanto, as informações adicionais podem ser definidas depois do conhecimento extraído.

Aquisição do conhecimento

Depois de definido o domínio onde será realizada a busca por conhecimento, o passo seguinte é escolher o método de análise dos dados. Foram implementados em SmartBASE 3(três) tipos de análises: geração de árvores de decisão, formação de conceitos e formação de conceitos em múltiplas perspectivas.

Para geração das árvores de decisão foi implementada uma versão do algoritmo C4.5. Esse algoritmo, em linhas gerais, realiza um preocesso de generalização das observações baseando-se num atributo que identifica a categoria da observação. De maneira que cada ramo da árvore, da raiz até a folha, denote uma regra que define um padrão nas observações.

Todas as regras são armazenadas para serem utilizadas posteriormente. Para essa utilização futura, foi desenvolvido em SmartBASE uma máquina de inferência interativa para aplicação das regras. De forma que o software vai interativamente perguntando sobre propriedades de uma nova observação e a partir das regras é inferido a que categoria pertence a nova observação.

A formação de conceito foi a forma de aquisição de conhecimento mais desenvolvida em SmartBASE, visto que foi tema para essa dissertação. Foram implementados vários algoritmos de formação de conceito, mais precisamente, conceitos probabilísticos. Os algoritmos implementados foram COBWEB, CLASSIT, COBWEB/3, COBWEB95, COBIT e FORMVIEW.

FORMVIEW, onde foi implementada a proposta desse trabalho, também realiza a formação de conceitos probabilísticos em múltiplas perspectivas, caso seja informado dois domínios cada um representando a mesma entidade vista por duas perspectivas. Nesse caso, FORMVIEW descobre relações entre os conceitos das diferentes perspectivas. Essa relação é apresentada por pontes entre as duas hierarquias, onde cada ponte possui o percentual de observações que estão simultaneamente nas duas extremidades da ponte. Quanto maior o percentual de

observações que estão simultaneamente em dois conceitos em hierarquias distintas, maior a relação entre esses conceitos.

Vale lembrar que todo conhecimento extraído em SmartBASE é armazenado em bancos de dados. Possibilitando tanto uma aquisição incremental como uma interpretação gradual do significado do conhecimento.

A seção a seguir irá apresentar as formas de análise dos resultados dos métodos de aquisição de conhecimento.

Análise do conhecimento

Pode-se dizer que foi dada uma atenção especial à análise do conhecimento extraído por SmartBASE. Para cada método de aquisição de conhecimento procurou-se uma representação visual que facilitasse a interpretação dos resultados por um especialista. Nas aplicações de SmartBASE, esses especialistas eram, na maioria das vezes, pessoas com pouco contato com computadores. Esse fato motivou o desenvolvimento de SmartBASE visando ser uma ferramenta naturalmente intuitiva.

Árvore de decisão

Para análise da árvore de decisão, o conhecimento foi apresentado em forma de uma hierarquia onde os nós representam os atributos, os ramos representam os valores dos atributos e as folhas representam as categorias do domínio. De forma, percorrendo a hierarquia a partir da raiz até uma folha tem-se uma regra que define um padrão nas observações. Como citado anteriormente, para analisar o conhecimento dessas árvores também pode ser usada a máquina de inferência de SmartBASE.

Formação de conceitos

Na análise da formação de conceitos, foram desenvolvidos outros meios para facilitar a interpretação do conhecimento. Tradicionalmente, algoritmos de formação de conceitos representam o conhecimento através de uma hierarquia onde cada nó representa um conceito. Portanto, foram usadas estruturas hierárquicas de aparência visual popularmente conhecida, como uma árvore de diretórios (figura 22).

Outra funcionalidade desenvolvida foi a possibilidade de rastrear, através de relatórios do sistema, os valores internos das funções de avaliação. Como também é possível escolher um determinado conceito da hierarquia e calcular sua capacidade de inferência baseado no algoritmo que foi usado para construir a hierarquia. A inferência, um a um, de todos atributos de uma determinada observação e acompanhamento dos erros e dos acertos, também compõe essa aplicação.

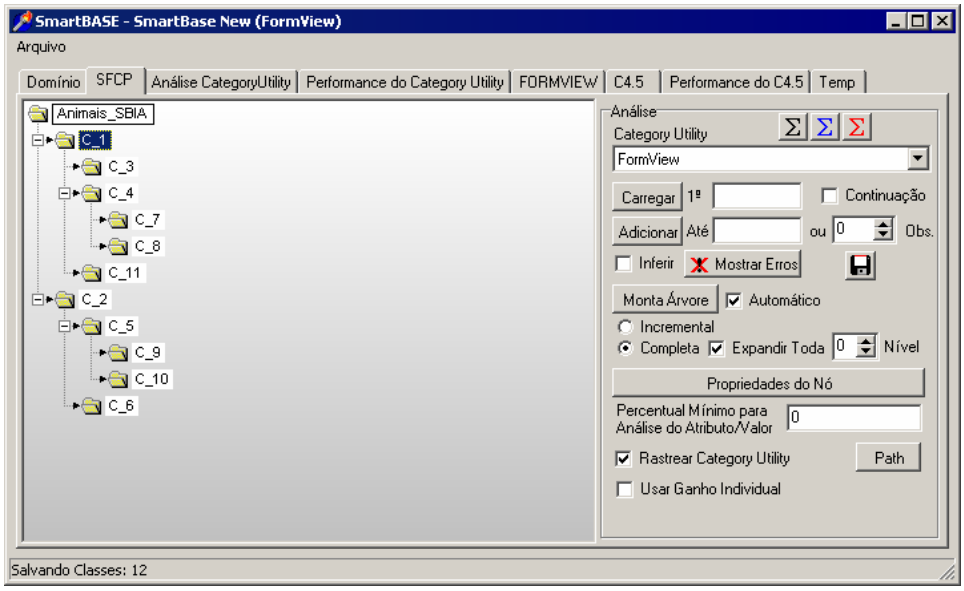

Figura 22: Hierarquia de conceitos em SmartBASE.

No entanto, para representar os conceitos probabilísticos foi feito uso de gráficos utilizados tradicionalmente em estudos estatísticos. A representação gráfica de um conceito probabilístico está ilustrada na figura 23.

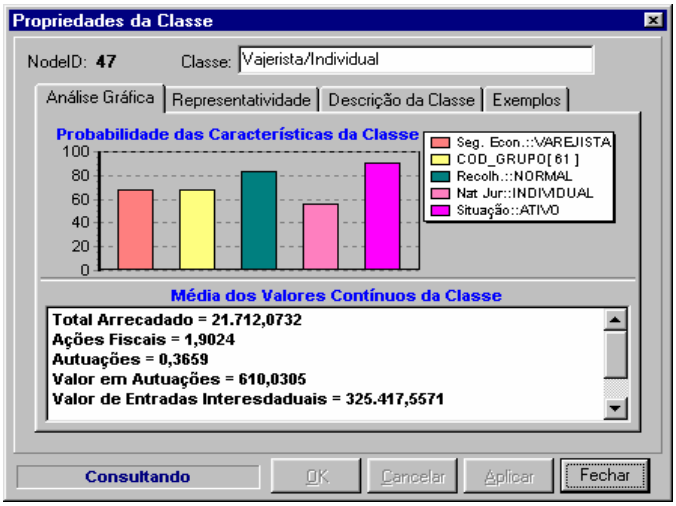

Figura 23: Representação gráfica de um conceito probabilístico.

Onde o gráfico em barra representa as probabilidades de ocorrência das propriedades discretas em um determinado conceito. As propriedades contínuas do conceito estão em termos da média do valor atributo para o conceito no quadro abaixo do gráfico.

Figura 24: Representação gráfica da quantidade de observações que o conceito representa.

Outra forma de análise é feita em função da quantidade de observações que um conceito representa. A figura 24 ilustra um gráfico de pizza que representa o percentual de observações que são membros do conceito. A partir dessas duas análises é possível chegar a conclusões do que se refere o conceito. Essas conclusões podem ser registradas como um rótulo para o conceito ou em termos descritivos em um local especialmente reservado para essa função.

O desenvolvimento dessa dissertação motivou o desenvolvimento de um módulo de análise de algoritmos de formação de conceitos. De forma que, nessa aplicação de visualização de uma hierarquia foram adicionadas funcionalidades para análise dessa classe de algoritmos. Como por exemplo, é possível facilmente alternar entre algoritmos para criação da hierarquia.

Contudo, para fazer um comparativo entre diferentes algoritmos foi necessário o desenvolvimento de uma outra aplicação. A figura 25 apresenta a tela dessa outra aplicação.

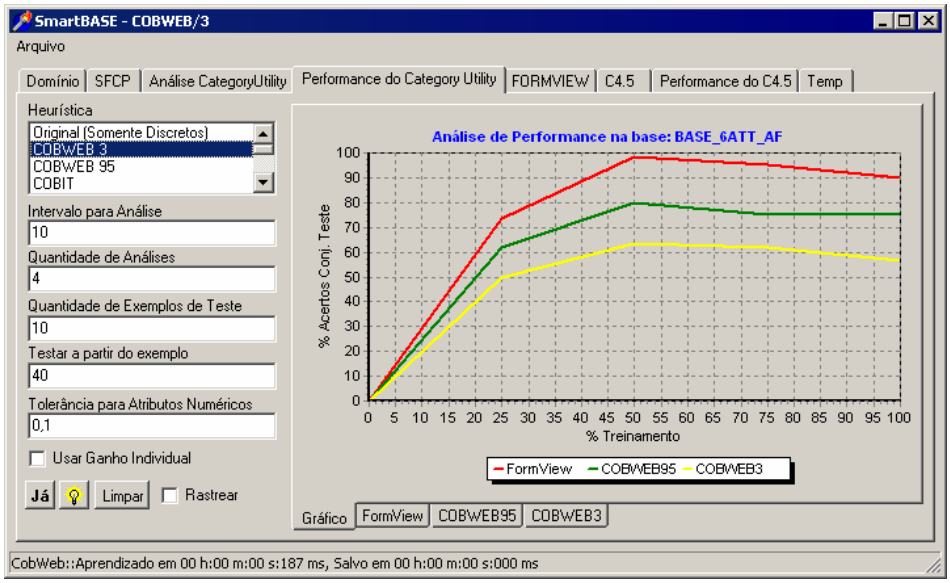

Figura 25: Aplicação para comparativo entre algoritmos.

Nessa aplicação, é possível escolher os algoritmos que se deseja comparar, aplicando-os em um mesmo domínio. Para essa comparação, pode-se variar a quantidade de observações que serão utilizadas no conjunto de treinamento assim como no conjunto de teste. Pode-se ainda variar a margem de erro para atributos contínuos.

Os resultados de cada algoritmo são apresentados através de um gráfico da quantidade de inferências corretas em função da quantidade do conjunto de treinamento. Para essa análise também são apresentadas informações detalhadas da quantidade de erros e que atributos foram mais errados. Essas informações são disponibilizadas em forma de relatório e são automaticamente gravadas ao final de uma análise.

Formação de conceitos a partir de múltiplas perspectivas

No caso da formação de conceitos a partir de múltiplas perspectivas, foi necessária uma maneira particular de representar o conhecimento, pois além das hierarquias de conceitos é preciso representar a relação entre os conceitos das duas hierarquias. A figura 26 ilustra a solução que foi dada para essa representação.

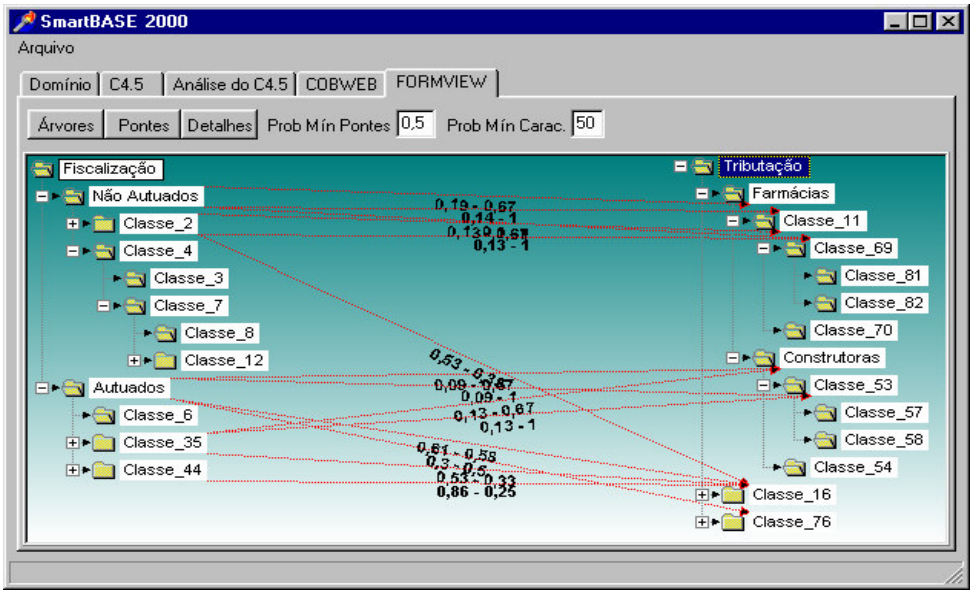

Figura 26: Representação da formação de conceitos a partir de duas perspectivas.

As hierarquias são apresentadas simultaneamente e, entre elas, as pontes são representadas por setas que tem espessura diretamente proporcional ao percentual de observações em ambas hierarquias.

Conclusão

O software SmartBASE é um produto que nasceu no meio acadêmico e vem sendo desenvolvido a mais de 4 anos. Pesquisas e aplicações na área de aquisição de conhecimento tem contribuído bastante para seu amadurecimento. De forma, que sua utilização nos últimos anos tem sido mais consistentes tanto pelo auxílio ao entendimento da área de aquisição de conhecimento como pela simplicidade que trata o tema.

Seu desenvolvimento em linguagem orientada a objetos, possibilitou sua implementação em módulos tanto na parte de infraestrutura do software como na implementação dos algoritmos. Deste modo, o aproveitamento de partes do seu código para aplicações mais específicas foi facilitado.

SmartBASE, hoje, constitui uma ferramenta que serve como base para novas implementações de algoritmos de aprendizado, principalmente no que diz respeito a formação de conceitos.

Bibliografia

- [Biswas et. al., 94] Biswas, G., Weinberg, J. and Li, C.. Iterate: A conceptual clustering method for knowledge discovery in databases. In B. Braunschweig and R. Day, editors, Innovative Applications of Artificial Intelligence in the Oil and Gas Industry. Editions Technip, 1994. to appear.
- [Biswas et. al., 98] Biswas G., Weinberg J.B., and Fisher D., "ITERATE: A Conceptual Clustering Algorithm for Data Mining," IEEE Transactions on Systems, Man and Cybernetics, Vol. 28, pp. 100-111, 1998.
- [Biswas & Li, 98] Biswas, G. and Li, C.: Conceptual Clustering with Numeric and Nominal Mixed Data - A New Similarity Based System. IEEE Transcript on Knowledge and Data Engineering, 1998.
- [Bond & Hine, 93] Bond, A. and Hine, J. H.: Predicting task resource use through classification. In Proceedings of the Annual Conference of the New Zealand Computer Society, August 1993.
- [Cheng & Fu, 85] Cheng, Y. and Fu, K.: Conceptual clustering in knowledge organization. IEEE Transactions on Pattern Analysis and Machine Intelligence, 7. 592-598, 1985.
- [Cavalcante, 2003] Cavalcante, A.S.: Uso de cores para auxiliar a interatividade no processo de formação incremental de conceitos. Tese Mestrado: UNIFOR, Fortaleza, CE, 2003.
- [Decaestecker , 93] Decaestecker, C.: Apprentissage et outils statistiques en classification incrémentale. Revue d'intelligence artificielle 7 - n°1: pp. 33-71, 1993.
- [Dietterich & Michalski, 83] Dietterich, T. and Michalski, R. A comparative review of selected methods from learning from examples. In Michalski, R., Carbonell, J., and Mitchell, T., editors, Machine learning: An artificial intelligence approach. Morgan Kaufmann Publishers, 1983.
- [Duda & Hart, 73] Duda, R. and Hart, P.: Pattern classification and scenes analysis. Wiley, New York. 1973.
- [Forte, 97] FORTE, Sérgio. Manual de elaboração de dissertação de mestrado e monografias de especialização da Unifor. Fortaleza: Gráfica UNIFOR,1997.
- [Fisher, 87] Fisher, D. Knowledge Acquisition via Incremental Conceptual Clustering. Machine Learning, v.2,n.2,1987.
- [Furtado et. al., 96] Furtado, J.J., Faucher, C., Chouraqui, E.: Knowledge Acquisition via Multi-perspective Concept Formation. Journal of Brazilian Computer Society, v.3, 1996.

- [Furtado, 97] Furtado, J. J.: Formation de concepts dans le contexte des languages de schémas. Thèse de doctorat. Université d'Aix Marseille III, IUSPIM/DAIM, 1997.
- [Furtado, 98] Furtado, J. J.: Determining property relevance in concept formation by computing correlation between properties. In Proceedings of the Tenth European Conference on Machine Learning, ECML98, volume 1398 of Lecture Notes in Articial Intelligence, pages 310-315, Chemnitz, Germany, Springer Verlag. 13; 1998.
- [Gennari et al., 89] Gennari, J., Langley, P., and Fisher, D. Models of Incremental concept formation. Artificial Intelligence, 40, (pp. 11-62), 1989.
- [Gomes et. al., 2001] Gomes, J.A.S., Rebouças, R.B. e Vasco, J.J.F.: Descoberta de Conhecimento em Múltiplas Perpectivas em Banco de Dados do ICMS. ENIA - Encontro Nacional de Inteligência Artificial, Fortaleza, CE, 2001.
- [Gomes, 2002] Gomes Jr., J.A.S.: Descoberta de Conhecimento em Múltiplas Perpectivas: Aplicação em bases de dados do ICMS. Tese de Mestrado, Fortaleza: UNIFOR, CE, 2001.
- [Gluck & Corter, 85] Gluck, M., and Corter, J. Information, uncertainty, and the utility of categories. Proceedings of the 7th Annual Conference of Cognitive Science Society. (pp 283-287). Irvine, CA: Lawrence Erlbaum. 1985.
- [Jain & Dubes, 1988] Jain, A.K. and Dubes, R.C.: Algorithms for Clustering Data. Prentice Hall, Englewood Cliffs NJ, U.S.A., 1988.
- [Lebowitz, 87] Lebowitz, M.: Experiments with Incremental Concept Formation:UNIMEM. Machine Learning, v. 2, n. 2, 1987.
- [McKusick & Thompson, 90] McKusick, K. and Thompson, K. COBWEB/3: A Portable Implementation, Technical Report FIA-90-6-18-2, NASA Ames Research Center, 1990.
- [McKusick & Langley, 91] McKusick, K. and Langley, P. Constraints on Tree Structure in Concept Formation, Proceendings of the 12th International Joint Conference on Artificial Intelligence, (pp. 810-816), Sydney, Australia, 1991.
- [Merckt & DeCaestecker, 94] Merckt, T. V. and DeCaestecker, C.: A general framework for concept learning using hybrid systems: The two-functional model. In TR 93-19, IRIDIA, Universit'e Libre de Bruxelles, 1994.
- [Mervis & Rosch, 81] Mervis, C. and Rosch, E.: Categorization of natural objects. Annual Review of Psychology, 32:89--115, 1981.
- [Michalski, 83] Michalski, R., Carbonnel, J., Mitchell, T. : Machine Learning, An Intelligence Approach. v.I, Tioga Publishing, CA. 1983.
- [Moller, 97] Möller, J.-U.: CLASSITALL: Incremental and Unsupervised Learning in the DIA-MOLE Framework. In European Conference on Machine Learning, Workshop Notes on Empirical Learning of Natural Language Processing Tasks, 95-104. Prague, Czech Republic, 1997.
- [Pearl, 85] Pearl, J.: Learning hidden causes from empirical data. Proceedings of the Ninth International Conference on Artificial Intelligence. San Mateo, CA., Morgan Kaufmann. Pp.567-572, 1985.

- [Quinlan, 83] Quinlan, R.: Learning efficient classification procedures. Machine Learning: an artificial intelligence approach, Michalski, Carbonell & Mitchell (eds.), Morgan Kaufmann, p. 463-482, 1983.
- [Rebouças e Furtado, 98] Rebouças, R.B. e Furtado, J.J.: Visualização de Conceitos Organizados Hierarquicamente. IV Encontro de Iniciação à Pesquisa, UNIFOR, Fortaleza, CE, 1998.
- [Rebouças e Furtado, 2000] Rebouças, R.B. e Furtado, J.J.: SmartBASE: Uma ferramenta de Data Mining. I Encontro de Pesquisa e Pós-graduação, UNIFOR, Fortaleza, CE, 2000.
- [Rebouças e Furtado, 2003] Rebouças, R.B. e Furtado, J.J.: Formação de conceitos probabilísticos através de observações contendo atributos discretos e contínuos. ENIA - Encontro Nacional de Inteligência Artificial, Campinas, SP, 2003.
- [Reich, 91] Reich, Y., Building and Improving Design Systems: A Machine Learning Approach. in PhD Tesis, Department of Civil Engineering, Carnegue Melloon University. 1991.
- [Reich & Fenves, 91] Reich, Y. and Fenves, S.J.: The Formation and Use of Abstract Concepts in Design. in Concept Formation: Knowledge and Experience in Unsupervised Learning, Fisher, D.H. and Pazzani, M.J. and Langley, P. eds. (pp. 323-353). Morgan Kaugmann, LosAltos, CA, 1991.
- [Rosch, 75] Rosch, E.: Cognitive representations of semantic categories. Journal of Experimental Psychology: General, 104, 192—233, 1975.
- [RUSSEL, 95] RUSSEL, Stuart & NORVING, Peter. Artificial intelligence: a modern approach. New Jersey: Prentice Hall, 1995.
- [Seifert, 89] Seifert, C. M.: A retrieval model using feature selection. Proceedings of the Sixth International Workshop on Machine Learning (pp. 52-54). Ithaca, NY: Morgan Kaufmann, 1989.
- [Smith & Medin, 81] Smith, E.E. & Medin, D.L.: Categories and Concepts. Harvard University Press, 1981.
- [Stepp & Michalski, 86] R.E. Stepp and R.S. Michalski. Conceptual clustering: Inventing goal-oriented classification of structured objects. In R.S. Michalski et al., editor, Machine Learning: An Artificial Intelligence Approach, Vol. II, pages 471--498. Morgan-Kaufman Publishers, 1986.
- [Yoo & Yoo, 95] Yoo, J., Yoo, S. (1995). Concept Formation in Numeric Domains. ACM 0-89791-737-5, (pp. 36-41).

Livros Grátis

(http://www.livrosgratis.com.br)

Milhares de Livros para Download:

Baixar livros de Literatura
Baixar livros de Literatura de Cordel
Baixar livros de Literatura Infantil
Baixar livros de Matemática
Baixar livros de Medicina
Baixar livros de Medicina Veterinária
Baixar livros de Meio Ambiente
Baixar livros de Meteorologia
Baixar Monografias e TCC
Baixar livros Multidisciplinar
Baixar livros de Música
Baixar livros de Psicologia
Baixar livros de Química
Baixar livros de Saúde Coletiva
Baixar livros de Serviço Social
Baixar livros de Sociologia
Baixar livros de Teologia
Baixar livros de Trabalho
Baixar livros de Turismo