



Avaliação da aprendizagem por meio de lógica de *fuzzy* validado por uma Árvore de Decisão ID3

Beatriz Wilges, UFSC/(PGEGC), beaw@inf.ufsc.br

Gustavo Pereira Mateus, UFSC/(PPGCC), gpmateus@inf.ufsc.br

Silvia Modesto Nassar, UFSC/(PPGCC), silvia@inf.ufsc.br

Rogério Cid Bastos, UFSC/(PGEGC), rogerio@inf.ufsc.br

Resumo. Esse trabalho implementa uma comunidade de agentes que utiliza lógica *fuzzy* para determinar o perfil de aprendizagem dos estudantes. Esse perfil é determinado pelo desempenho, que é considerado sob dois enfoques denominados prático e teórico. O objetivo dessa classificação é conduzir o estudante para uma utilização mais apropriada do ambiente de ensino-aprendizagem. As entradas do modelo proposto foram escolhidas a partir de análise estatística e são apresentadas neste artigo. A arquitetura do AVA é implementada em PHP, o modelo *fuzzy* e o motor de inferência foram feitos com bibliotecas do Matlab. Os resultados da integração destas tecnologias bem como a validação do modelo *fuzzy* foram certificados através de um conjunto de dados. Esse conjunto de dados possui vários perfis de aprendizagem de diversos estudantes que foram minerados. Para mineração aplicou-se o método de classificação denominado Árvore de Decisão (AD). Esse trabalho apresenta uma possibilidade de validar e melhorar os resultados do *fuzzy*. O processo de validação pode ser utilizado na remodelagem das características de qualquer proposta de sistema *fuzzy*, ou seja, é uma forma de construir um modelo mais harmônico e condizente com o perfil do estudante.

Palavras-chave: Agente BDI, lógica *fuzzy*, árvore de decisão.

Abstract. *This work implements a community of agents that use fuzzy logic to determine the profile of student learning. This profile is determined by performance, which is considered from two perspectives known as practical and theoretical. The purpose of this classification is to lead the student to a more appropriate use of the teaching learning environment. The inputs of the model were chosen from statistical analysis and presented in this article. The architecture of the AVA is implemented in PHP, the model and the fuzzy inference engine were done with Matlab libraries. The results of integrating these technologies as well as validating the fuzzy model were verified through a set of data. This dataset has several learning profiles of several students that were mined. The classification method applied in data mining is called Decision Tree (AD). This work presents an opportunity to validate and improve the performance of fuzzy. The validation process can be used in the remodeling of the characteristics of any proposed fuzzy system, thus it is a way to build a model which is more harmonious and consistent with the student profile.*

Evaluation of learning through fuzzy logic validated by an ID3 Decision Tree

Keywords: *BDI agent, fuzzy logic, decision tree.*

1. Introdução

A proposta dessa pesquisa vem sendo desenvolvida, adaptada e avaliada por um grupo de pesquisa que busca criar um modelo de adaptação para Ambientes Virtuais de Aprendizagem (AVA), um desafio em constante evolução. A possibilidade de gerar um modelo de adaptação para os AVAs envolve técnicas de IA que já estão bem consolidadas: SMA, lógica *fuzzy* e mineração de dados.

Essa pesquisa trabalha com as características que são implementadas na maioria dos ambientes de aprendizagem. Dessa forma o desenvolvimento dessa proposta não está focado somente no ambiente em que são feitas as validações, mas sim na perspectiva de que qualquer AVA possa implementar este modelo adaptativo. Essas características poderiam envolver, por exemplo, o registro de *logs* e as avaliações do desempenho de estudantes através de tarefas como os questionários.

De acordo com Reategui (2006), Sistemas Multiagentes (SMA) são utilizados como uma alternativa eficiente na modelagem de AVAs interativos. Além disso, segundo Oliveira (1996), os princípios dos SMA mostram-se bastante adequados ao desenvolvimento de ambientes computacionais, tendo em vista que o problema de ensino-aprendizagem é de natureza social e cooperativa.

O termo “agente” é compreendido pela comunidade científica como um programa de software que auxilia o usuário na realização de alguma tarefa ou atividade (Wooldridge, 2002), sendo dotado de inteligência e autonomia. A autonomia, neste caso, significa que cada agente possui sua própria existência, operando sem a intervenção de humanos ou outros agentes e tem controle sobre suas ações e seu estado interno.

Este trabalho é direcionado para o modelo de agentes Belief-Desire-Intention (BDI) (Georgeff, 1999). Os conceitos deste modelo foram inicialmente propostos por Bratman (1987). O modelo consiste de crenças, desejos e intenções. Isso envolve as atitudes mentais que geram a ação humana. Rao e Georgeff (1995) adaptaram o modelo proposto por Bratman (1987), transformando-o em uma teoria formal e um modelo de execução para agentes de software baseados na noção de crenças, objetivos e planos.

Para reconhecer a qual tipo de categoria um determinado estudante pertence foi aplicada a lógica *fuzzy*. De acordo com Jameson (1995), a lógica *fuzzy* exige poucos cálculos e é mais fácil de inicializar e atualizar o modelo do aluno. Ainda assim, os resultados que o sistema atribui para um determinado estudante são analisados.

Nesse trabalho propomos a utilização da mineração de dados. Aplicamos a técnica de classificação baseada em Árvores de Decisão (AD). De acordo com Quinlan (1993) alguns dos algoritmos de classificação mais tradicionais e comuns são os que usam os valores dos atributos de forma combinada para delimitar regiões no espaço de atributos que definem as classes. A vantagem das AD é que elas explicam claramente que decisões são tomadas sobre quais atributos para classificação e sumarização. Existem diversos algoritmos na literatura utilizados para a construção de árvores de decisão, tais como ID3, C4.5 e CHAID. O algoritmo baseado em árvore de decisão mais utilizado é o C4.5 (Quinlan, 1993), que tem sua origem no ID3 (Quinlan, 1979).

O interesse deste trabalho é apoiado por um ambiente de aprendizagem desenvolvido no Laboratório de Estatística Aplicada. O objetivo do AVA é o desenvolvimento de projetos e ideias que possam agregar conhecimentos de estatística para estudantes de graduação e pós-graduação. O ambiente utilizado como estudo de caso e validação desta pesquisa é denominado SestatNet. Ele é considerado um ambiente flexível de ensino-aprendizagem de Estatística por meio da web.

Esse artigo está organizado da seguinte forma: a próxima seção detalha aspectos do AVA utilizado nesta pesquisa como estudo de caso. A seção 3 detalha os agentes BDI que são implementados na arquitetura proposta, as definições e estratégias adotadas por cada um deles. A seção 4 explica quais as variáveis do AVA que foram consideradas pertinentes para diagnosticar/determinar o perfil de aprendizagem do estudante. A seção 5 detalha como foram modeladas as curvas de pertinência em relação as variáveis definidas na seção 4. Também é apresentada nesta seção a forma como a lógica *fuzzy* foi utilizada neste trabalho para apoiar o SMA na tomada de decisão sobre o perfil de aprendizagem do estudante. A seção 6 mostra a validação realizada utilizando Árvore de Decisão e os resultados dessa validação. E por fim, na seção 7 são apresentadas as conclusões desta proposta.

2. O Ambiente Virtual de Aprendizagem (AVA)

O AVA é usado para ensino semipresencial na disciplina de estatística pelos estudantes de engenharia da Universidade Federal de Santa Catarina. No ambiente, o estudante é conduzido a projetar uma pesquisa quantitativa. Após a coleta, os dados são estruturados em uma planilha que é importada para o ambiente. Nesse AVA o estudante pode trabalhar os dados de sua pesquisa em dois módulos: tutoria e avançado. Inicialmente o estudante trabalha no módulo tutoria e lá aprende o conteúdo estatístico guiado pelo Mapa Conceitual (MC) do AVA. Cada passo do estudante está vinculado a uma planilha de dados, que pode ser de sua pesquisa ou um exemplo disponibilizado no AVA.

O MC é um mapa que contém os principais tópicos e respectivos subtópicos abordados no curso. Nesse módulo de tutoria o estudante tem total liberdade para escolher seus caminhos de aprendizagem. O MC mostra ao estudante, em formato de Árvore de Decisão, o caminho de navegação que uma dada interação percorre até chegar ao resultado estatístico final, e também os caminhos que o sistema pode seguir no caso de respostas diferentes. Essa característica tem como objetivo localizar o estudante dentro do raciocínio estatístico.

No módulo avançado do AVA o estudante aplica diretamente, sobre seus dados, a análise desejada, sem o apoio do módulo de tutoria. Ou seja, sem o uso do MC. Normalmente este módulo é usado pelos estudantes depois que eles se sentem seguros sobre as análises que desejam realizar, assim que eles tenham aprendido como se faz uma análise de dados no módulo tutoria.

3. Os agentes BDI

Implementar agentes *Belief-Desire-Intention* (BDI) em um Ambiente Virtual de Aprendizagem (AVA) envolve a definição de quais serão as crenças, desejos e intenções dos agentes dentro do ambiente. Essa pesquisa propõe que as crenças representem o nível de conhecimento que o agente presume que o estudante tenha sobre um conteúdo

específico. Os desejos do agente são estimular e ajudar o estudante no processo de aprendizagem, através de mensagens de apoio e dicas de utilização do ambiente. Enquanto que as intenções estão relacionadas com o modo como os agentes vão conduzir os estudantes aos módulos mais adequados do ambiente.

De acordo com Weiss (1999), em Sistemas Multiagentes os agentes se comunicam e interagem para alcançarem objetivos comuns e individuais. Considerando que o foco dos agentes BDI está centrado na análise que os mesmos farão sobre os dados dos estudantes obtidos do banco de dados do AVA. Foram identificados os seguintes agentes: estudante, avaliação, tutoria e avançado.

O AVA é implementado com a linguagem de programação PHP e o banco de dados MySQL. Ambas as tecnologias adotadas nesta pesquisa são baseadas em software livre. Além disso, toda proposta do ambiente, apresentado na seção 2, já foi desenvolvida sobre essas duas tecnologias, incluindo a linguagem de programação PERL.

Alguns destes agentes são extremamente simples, necessitando apenas de consultas a determinadas tabelas de um banco de dados do sistema. Nesse contexto inserem-se os agentes “tutoria” e “avaliação”, que apenas coletam informações de utilização do ambiente.

A aplicação do modelo BDI envolve a definição de como podem ser criados e manipulados os agentes dentro de um AVA. Agentes BDI têm crenças, que são armazenadas em uma base de crenças. Objetivos representam motivações concretas, como, por exemplo, estados a serem atingidos, e influenciam no comportamento do agente. Para atingir seus objetivos, os agentes executam planos.

A arquitetura projetada utiliza quatro tipos de agentes, denominados de “estudante”, “avaliação”, “tutoria” e “avançado”. Para cada estudante que utiliza o ambiente existe uma instância do “agente estudante”. Os agentes “avaliação”, “tutoria” e “avançado” representam os recursos disponíveis para o estudante no ambiente.

Inicialmente os estudantes interagem no ambiente com os módulos de tutoria e avaliação que registram suas interações através de arquivos de *log*, armazenando as ações do estudante no AVA. Além das questões respondidas e salvas na base de dados os agentes, que representam os estudantes, atualizam suas bases de crenças com informações a respeito do desempenho dos estudantes, obtidas dos arquivos de *log*.

Com o objetivo de indicar o módulo mais adequado ao estudante, o agente estudante processa as informações obtidas, através de planos de ação, e inicia a troca de mensagens com o agente do módulo tutoria ou com o agente do módulo avançado. O plano de ações é definido por meio da aplicação da lógica *fuzzy*. É através da lógica *fuzzy* que é determinado se estudante tem um escore suficiente para mudar de módulo ou se precisa continuar estudando no módulo tutoria. Assim o estudante é conduzido a utilizar o módulo que mais se adapta ao seu atual nível de aprendizagem. Os detalhes sobre o funcionamento do módulo *fuzzy* são apresentados na seção 5.

4. Análise das variáveis do modelo proposto

Muitas características dos usuários são armazenadas no servidor do AVA. Essas características correspondem ao tempo de sessão nas páginas visitadas pelos estudantes, o desempenho deles nos questionários avaliativos, o tempo utilizado para responder o questionário entre outras informações. Aplicar essas informações para melhorar a

interface adaptativa do AVA é o objetivo deste trabalho. Para isso adotou-se a técnica de lógica *fuzzy*.

Como já apresentado o desempenho no AVA é avaliado com dois enfoques: um mais prático e outro mais teórico. Esses dois enfoques foram analisados considerando os resultados do AVA e da avaliação presencial. Portanto, nessa pesquisa foi considerado um mapeamento dessas diferentes avaliações (ambiente e presencial) de desempenho (prático e teórico) do estudante no intuito de encontrar uma analogia, que poderia ser contemplada no modelo *fuzzy* proposto.

Segundo Bardin (2000) é um equívoco privilegiar o conhecimento imediato (empírico) em detrimento daquele fundamentado em conceitos, na falsa ideia de que a teoria (uma abstração) é desprovida de objetividade. Inexiste prática sem teoria, bem como teoria desprovida de prática. Sendo assim, o foco desta pesquisa concentrou-se tanto na avaliação teórica como a avaliação prática justificando e preservando o movimento contínuo de teoria-prática.

Para obter essas informações e definições dos parâmetros para a modelagem *fuzzy* foi realizada uma análise estatística nos *logs* da turma do semestre 2010/1. Além disso, era necessário extrair os diferentes estados que podem levar a caracterização do perfil do estudante. No módulo de tutoria do AVA, os estudantes eram conduzidos a explicitar qual a decisão estatística que estavam fazendo naquele momento. Nesse instante o AVA reconhece se o estudante acertou ou errou, mas não interrompe o processo de aprendizagem. Essa informação fica armazenada no registro de *logs* e está em mãos do agente estudante. Essa informação é caracterizada com uma avaliação prática do AVA.

Esse desempenho no AVA foi comparado com a respectiva avaliação presencial considerando também a parte prática da prova. Nesse sentido buscou-se perceber a existência de algum tipo de relação entre os desempenhos práticos do AVA e o desempenho na avaliação presencial, de forma a identificar diferentes grupos para definição dos desafios pedagógicos.

Em seguida relacionou-se, também, o desempenho dos estudantes na avaliação teórica tanto no AVA como na avaliação presencial, ambas contemplam diferentes níveis de conhecimento. A partir dos resultados e da forte correlação entre essas variáveis considerou-se pertinente adotar o desempenho prático e teórico do AVA com entradas para o modelo *fuzzy* proposto.

Em princípio o tempo nas sessões de aprendizagem seria avaliado com uma variável linguística para o sistema *fuzzy*. No entanto, depois de realizada algumas análises notou-se que não havia correlação entre tempo no AVA e desempenho do estudante. Por meio das análises percebemos que estudantes com pouco tempo nas sessões do AVA apresentaram um desempenho que variava entre 50% a 100%. Ainda nessa análise percebeu-se que existem estudantes com um tempo extremamente alto o que pode evidenciar não só o tempo de estudo, mas sim o tempo em outras atividades externas ao AVA. Desta forma para acompanhar e reconhecer o nível de aprendizagem foram consideradas as variáveis de desempenho no AVA.

5. O plano de ações determinado pelo *fuzzy*

Diversas pesquisas propõem métodos, ferramentas e ambientes de acompanhamento de desempenho baseados em conjuntos *fuzzy* (por exemplo, satisfatório, insatisfatório e

regular), onde o professor relaciona o grau de pertinência desses valores dentro de intervalos de notas, assim como a base de regras. (Nolan, 1998; Weon e Kim, 2001; Fabri e Fabri, 2002; Rasmani, 2002; Hadjileontiadou et al, 2004; Rieder e Brancher, 2004; Rissoli e Giraffa, 2006; Stathacopoulou et al, 2005; Huang et al, 2007; Ribeiro, 2007; Silva et al, 2008).

Nesta pesquisa foram definidos os seguintes estados possíveis de nível de aprendizagem: regular, bom e excelente. Os pontos na região clara representam os estudantes com desempenho/tempo aceitáveis e os pontos na região escura representam estudantes com mais dificuldades. Assim, o ambiente agirá de formas diferentes, pois será guiado pelo desempenho do estudante no AVA.

Nessa Figura 1 assume-se como ponto de corte o valor 0,6. Esse valor é determinado em função de representar a nota mínima para aprovação nessa universidade. Ou seja, é um parâmetro institucional e pode variar de acordo com a instituição que implementar o modelo.

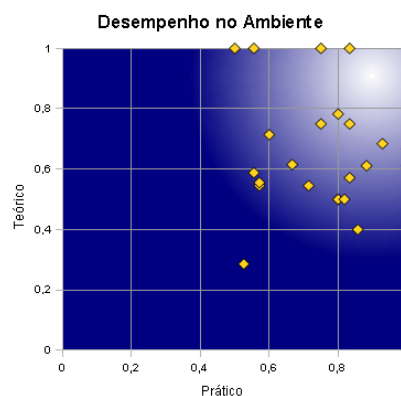


Figura 1. Estados baseados nas entradas: desempenho teórico e prático

Nessa Figura 1 observa-se que o grupo que se encontra na faixa acima do 0,8 tanto para o desempenho teórico com para o desempenho prático pode receber um estímulo pedagógico. Esse estímulo hoje corresponde ao agente estudante indicar o módulo mais adequado ao estudante. Após o processamento das informações obtidas, o agente estudante inicia a troca de mensagens com o agente do módulo avançado. Nesse momento o estudante é conduzido a trabalhar no módulo avançado.

Os estudantes que ficam abaixo da faixa de 0,6 devem continuar sendo orientados a trabalhar no módulo tutoria, estudando situações mais pontuais (teoria + prática). Os estudantes que ficam acima da faixa de 0,6 somente em um dos tipos de avaliação, devem ser conduzidos a desafios pedagógicos ou mais teóricos ou mais práticos dependendo dos resultados observados.

Tomando como base as variáveis analisadas na seção anterior foi construído o modelo *fuzzy*. As entradas deste modelo correspondem ao desempenho na parte prática e o desempenho na parte teórica no AVA. Foi utilizada uma função de pertinência gaussiana neste trabalho, considerando que este tipo de função trata a mudança de “estado” entre as variáveis linguísticas de uma maneira mais suave, e não crisp como nas funções triangulares. O modelo dessa proposta se encontra na Figura 2.

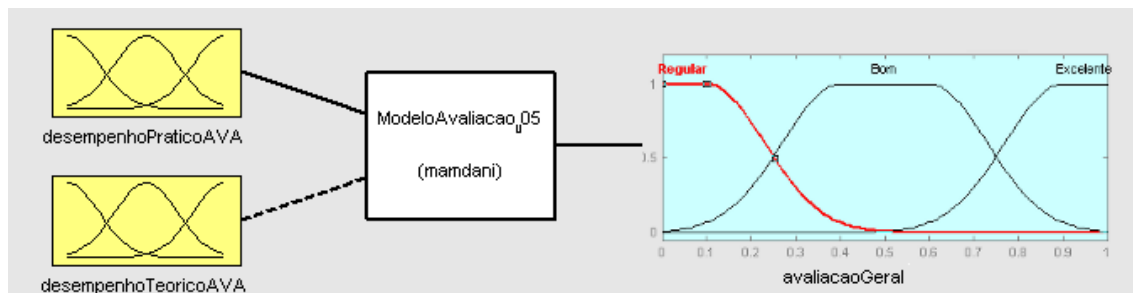


Figura 2. Modelo de avaliação de desempenho

O conjunto que define as regras do sistema *fuzzy* é apresentado na Figura 3. Os valores dos estados são: regular, bom e excelente e foram definidos conforme a subdivisão apresentada na Figura 1.

```
1. If (desempenhoPraticoAVA is Regular) and (desempenhoTeoricoAVA is Regular) then (avaliacaoGeral is Regular) (1)
2. If (desempenhoPraticoAVA is Regular) and (desempenhoTeoricoAVA is Bom) then (avaliacaoGeral is Regular) (1)
3. If (desempenhoPraticoAVA is Regular) and (desempenhoTeoricoAVA is Excelente) then (avaliacaoGeral is Bom) (1)
4. If (desempenhoPraticoAVA is Bom) and (desempenhoTeoricoAVA is Regular) then (avaliacaoGeral is Regular) (1)
5. If (desempenhoPraticoAVA is Bom) and (desempenhoTeoricoAVA is Bom) then (avaliacaoGeral is Excelente) (1)
6. If (desempenhoPraticoAVA is Bom) and (desempenhoTeoricoAVA is Excelente) then (avaliacaoGeral is Excelente) (1)
7. If (desempenhoPraticoAVA is Excelente) and (desempenhoTeoricoAVA is Regular) then (avaliacaoGeral is Bom) (1)
8. If (desempenhoPraticoAVA is Excelente) and (desempenhoTeoricoAVA is Bom) then (avaliacaoGeral is Excelente) (1)
9. If (desempenhoPraticoAVA is Excelente) and (desempenhoTeoricoAVA is Excelente) then (avaliacaoGeral is Excelente) (1)
```

Figura 3. Conjunto de regras do sistema *fuzzy*

O modelo funcionando no AVA trabalha com os parâmetros das funções de pertinência obtidos por meio da análise estatística. Assim as regras de inferência *fuzzy* são aplicadas nas variáveis linguísticas para determinar os objetivos dos agentes em conduzir o estudante para o módulo “tutoria” ou para o módulo “avançado”. Este raciocínio é feito no agente “estudante”, pois ele coleta todas as informações de utilização do ambiente obtidas a partir de outros agentes.

O módulo *fuzzy* proposto funciona no AVA, que é implementado em PHP, através de um toolbox disponibilizado pelo Matlab na linguagem C. No *toolbox* existem dois arquivos em C denominados *fismain.c* e *fis.c*, que são fornecidos como códigos-fonte para um motor de inferência *fuzzy* no modo *stand-alone*. O código *stand-alone* em C para o motor de inferência *fuzzy* pode ler um arquivo FIS, que é a extensão dos arquivos *fuzzy* gerados no Matlab, e um arquivo de dados de entrada para realizar inferência diretamente.

Nesse AVA é gerado um arquivo para o conjunto de entrada (teórica e prática). Esse conjunto de entrada considera os estudantes mais assíduos, ou seja, são realizadas inferências somente com os estudantes que mais utilizam o ambiente considerando os últimos 30 dias. Após a geração desse arquivo, o mesmo é submetido juntamente com o modelo *fuzzy* proposto (arquivo FIS) ao motor de inferência. O código *stand-alone* retorna um arquivo com a saída desfuzzificada para cada entrada (cada estudante).

A avaliação do modelo comparando a nota no ambiente e na avaliação presencial mostrou uma alta correlação conforme apresentado no trabalho de Mateus (2010). As regras do modelo *fuzzy* foram avaliadas por um processo de classificação através de um processo de classificação baseado em mineração de dados.

6. Avaliação das regras do modelo *fuzzy*

Foi utilizada uma amostra de 57 estudantes matriculados na disciplina do primeiro semestre de 2010. Foram observadas as notas obtidas da parte teórica, prática e geral no AVA e as notas obtidas nas avaliações presenciais.

A avaliação teórica no AVA é composta por diversas questões em diferentes níveis de aprendizagem. Cada questão tem diferentes números de opções de respostas. A avaliação teórica é um atributo que apresenta uma baixa entropia, o que caracteriza uma informação significativa para construção da árvore de decisão (AD). Isso acontece porque a parte teórica do AVA é muito mais complexa e rica quando comparada com a avaliação prática, também do AVA. A parte prática é caracterizada por uma tomada de decisão do tipo “sim” e “não” que expressa a interpretação de cálculos realizados no AVA. E que determina a próxima interação do AVA com o estudante.

Os dois módulos de avaliação: teórico e prático são módulos independentes desprovidos de uma hierarquia entre as avaliações. Ou seja, o estudante tem total liberdade de realizar as avaliações teóricas e práticas de modo independente e no tempo em que achar conveniente.

A mineração dos dados considerou as seguintes variáveis extraídas do BD do AVA: conceito teórico e prático obtidos do AVA com a respectiva saída desfuzzificada do modelo apresentado na Figura 2. Para realizar essa classificação adotamos o algoritmo da Árvore de Decisão (AD).

O algoritmo adotado neste trabalho foi o ID3 por apresentar uma resposta mais adequada ao sistema proposto, sem considerar as “podas” que acontecem no C4.5. No algoritmo ID3 a escolha dos atributos a serem utilizados pela árvore se dá a partir de informações de entropia e ganho de informação. Para medir o nível de informação de um atributo se utiliza o conceito de entropia da Teoria da Informação. O valor da entropia corresponde à impureza do atributo e o ganho de informação é a variação da impureza. Quanto menor o valor da entropia, menor a incerteza e mais utilidade tem o atributo para a classificação.

O processamento foi realizado na ferramenta *RapidMiner*. Essa ferramenta é open-source, anteriormente chamada Yale. Conforme RapidMiner (2008), a ferramenta possui interface gráfica do usuário (GUI) e scripts baseados em XML, tornando esta uma *Integrated Development Environment* (IDE) e um interpretador para *Knowledge Discovery and Data Mining* (KDD). É desenvolvida sob a plataforma Java, o que facilita integração com outras aplicações sob esta arquitetura.

Os dados utilizados na classificação foram tratados para as etapas que envolveram a descoberta de conhecimento. As etapas adotadas foram: seleção dos dados, pré-processamento para retirar dados duplicados ou inconsistentes e discretização por frequência. O resultado dessa validação e aproximação de valores é apresentado na Figura 4.

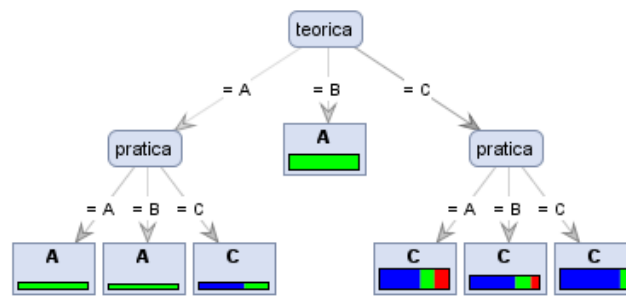


Figura 4. Avaliação do modelo

Após a transformação dos dados, as notas extraídas do BD foram discretizadas em conceitos: A, B e C. Correspondendo as variáveis linguísticas regular, bom e excelente. Os alunos que ficam com um desempenho abaixo de 0,6 são classificados em C. Os estudantes entre 0,61 e 0,79 são B, e os estudantes entre 0,8 e 1,0 são A. A árvore apresentada na Figura 4 mostra a relação entre as regras do modelo *fuzzy* proposto.

Os vértices da AD representam os atributos preditivos, as folhas representam a saída do modelo *fuzzy* (*label*). As arestas correspondem ao conceito obtido no atributo do vértice correspondente. Cada conceito é representado por uma cor. O verde representa o conceito A, o B é o vermelho e o C é representado pelo azul. As folhas, onde aparece mais de uma cor, são a representação de outros conceitos que também foram saídas além do conceito descrito e determinante para a folha.

De acordo com a AD estudantes que tiram A na teórica e A e B na prática são classificados com A na saída do sistema *fuzzy*. Assim como, percebe-se que estudantes que tem um desempenho B na teórica apresentam na saída do sistema um desempenho avaliado com A. E por fim, os alunos que tiram C na teórica em linhas gerais mantêm o C na saída do sistema.

De acordo com esse resultado percebe-se que é interessante realizar um ajuste nos parâmetros das funções de pertinência das entradas do módulo apresentado na Figura 2. Como o atributo de avaliação teórica é classificado, segundo o algoritmo ID3, como raiz da árvore ele é uma variável que possui uma entropia menor, o que significa que ele é o atributo que possui mais informação na entrada do modelo *fuzzy*. A Figura 5 representa as três funções de pertinência do modelo *fuzzy* proposto já ajustadas conforme análise da AD.

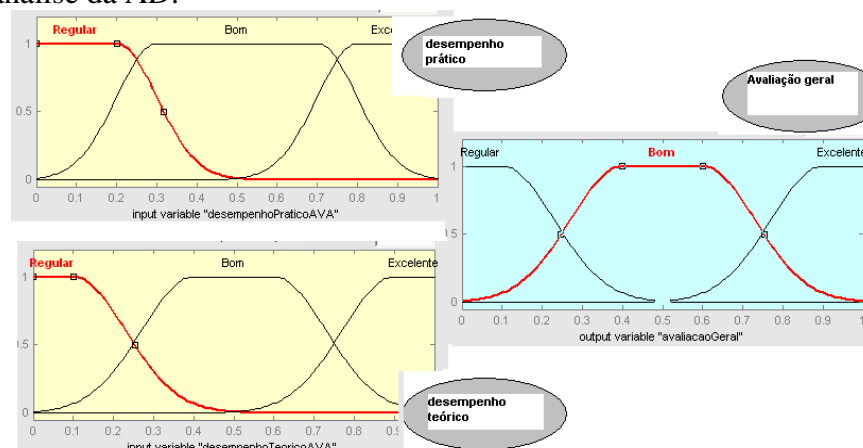


Figura 5. Funções de pertinência do modelo *fuzzy* proposto

O objetivo desse ajuste foi caracterizar a maior incerteza presente no desempenho prático na classificação de um estudante. A função de pertinência da avaliação prática apresenta-se ampliada no intuito de representar um modelo mais adequado conforme análise dos dados descritos pela AD.

7. Conclusões

O uso de lógica *fuzzy* na análise do perfil do estudante pode flexibilizar a categorização do processo de ensino-aprendizagem. Não sendo, simplesmente, um tratamento “crisp” do tipo certo e errado e sim um valor lógico difuso que pode ser qualquer valor no intervalo entre 0 e 1. Depois desse processo de fuzzificação, o agente *fuzzy* envolvido nesse processo comunica o agente estudante que é o responsável por determinar o recurso mais adequado a ser ativado. Este processo ocorre de forma dinâmica e transparente a cada ação do usuário no ambiente.

Além disso, o modelo proposto teve a preocupação de validar os resultados. Essa comparação entre a árvore de decisão e as regras do sistema *fuzzy* apresentam uma correlação importante. Além disso, minerar e classificar esses dados abre a possibilidade de ajustes no modelo proposto. Um refinamento visando uma possível melhoria.

É importante destacar que uma AD pode ser utilizada com duas finalidades: previsão e descrição. Nesse trabalho buscamos elaborar uma descrição que fornecesse informações interessantes a respeito das relações entre os atributos preditivos e o atributo classe em cima da BD do AVA. O objetivo foi observar se as regras do modelo *fuzzy* se ajustavam a estrutura da árvore apresentada. O ajuste do modelo *fuzzy* gera como resultado final uma possibilidade de um acompanhamento mais adequado das atividades teóricas e práticas.

Referências

- BARDIN L. Análise de conteúdo. Lisboa: Edições 70; 2000.
- BRATMAN, M. E. Intention, Plans, and Practical Reason. Cambridge, MA. 1987
- FABRI, J.A.; FABRI, M.G.S. (2002) “Ferramenta Fuzzy para Acompanhamento do Desempenho dos Alunos nos Cursos à Distância”, In: Anais do XXII Congresso da SBC. WIE, Campinas.
- GEORGEFF, M., Pell, B., POLLACK, M., Tambe, M., and Wooldridge, M. The belief desire intention model of agency. In Proceedings of the 5th International Workshop on Intelligent Agents V: Agent Theories, Architectures, and Languages, Springer-Verlag: Heidelberg, Germany. 1999
- HADJILEONTIADOU, S.J.; NIKOLAIDOU, G.N.; HADJILEONTIADIS, L.J.; BALAFOUTAS, G.N. (2004) “On Enhancing On-Line Collaboration Using *Fuzzy* Logic Modeling”, In: Educational Technology & Society, 7, p. 68-81.
- JAMESON, A. (1995) “Numerical Uncertainty Management In User And Student Modeling: An Overview Of Systems And Issues”, In: User Modeling And User-Adapted Interaction, 5, p. 193-251.
- MATEUS, G.P.; WILGES, B.; NASSAR, S. M.; SILVEIRA, R.A.; BASTOS, R.C. (2010) “Reconhecendo as características dos estudantes por meio de um SMA e lógica *Fuzzy*”, In: Revista de Novas Tecnologias na Educação RNOTE. Ed. 2, 2010.

- NOLAN, J.R. (1998) “An Expert *Fuzzy* Classification System For Supporting The Grading Of Student Writing Samples”, In: Expert Systems with Applications 15, p. 59-68.
- OLIVEIRA, F. M.; Vicari, R. M. (1996) “Are learning systems distributed or social systems.” In: European Conference on AI in Education. Lisbon: [s.n].
- QUINLAN, J.R. Discovering rules by induction from large collection of examples. Expert Systems in the Micro Electronic Age. Edinburgh, UK: Edinburgh University Press, 1979.
- QUINLAN, J.R. C4.5: Programs for Machine Learning. São Francisco: Morgan Kaufmann, 1993.
- RASMANI, K.A. (2002) “A Data-Driven *Fuzzy* Rule-Based Approach for Student Academic Performance Evaluation”, Ph.D. Dissertation Proposal, Central for Intelligent Systems and their Applications. School of Informatics. Universidade de Edinburgo, Edingurgo, U.K.
- RAPIDMINER, Dortmund – Alemanha. RapidMiner 4.1 User Guide. Dortmund, 2008.
- RAO, A. S. and GEORGEFF, M. P. Bdi-agents: from theory to practice. In Proceedings of the First Intl. Conference on Multiagent Systems, San Francisco. 1995
- REATEGUI, Eliseo; BOFF, Elisa; CERON, Rafael Fernando; VICARI, Rosa Maria. Um Agente Animado Sócio-Afetivo para Ambientes de Aprendizagem. Revista Brasileira de Informática na Educação – Edição Especial Aprendizagem Colaborativa. v. 14 n.3. Setembro-Dezembro de 2006. p27-38.
- RIEDER, R.; BRANCHER, J.D. (2004) “Aplicação da Lógica *Fuzzy* a Jogos Didáticos de Computador – A Experiência Do Mercadão GL. VII Congresso Iberoamericano de Informática Educativa.
- SILVA, Ismênia Galvão Lourença da. Projeto e Implementação de Sistemas Multi-Agentes: O Caso Tropos. Dissertação de Mestrado. Universidade Federal de Pernambuco – UFPE. 2005.
- WEISS, G. (1999). Multiagent systems - a modern approach to distributed artificial intelligent. MIT Press, Cambridge, MA.
- WEON, S.; KIM. J. (2001) “Learning Achievement Evaluation Strategy Using *Fuzzy* Membership Function”, Paper presented at the 31st ASEE/IEEE Frontiers in Education Conference, October 10-13, Reno, NV,
- WOOLDRIDGE, M. (2002) Introduction to Multi Agent Systems. Chichester: Wiley.