Uma proposta para o uso de agentes de software no processo de geração de árvores de decisão a partir de banco de dados

Adilson Pereira dos Santos
CEETEPS – Centro Estadual de Ensino Tecnológico Paula Souza
adilson.pereira@uol.com.br

Maurício Amaral de Almeida
CEETEPS – Centro Estadual de Ensino Tecnológico Paula Souza
malmeida@inteligenciaartificial.eti.br

Resumo. Este artigo propõe o uso de agentes de software atuando em um banco de dados como um mecanismo para a obtenção de conhecimento através de aprendizado supervisionado ou não. Para um melhor entendimento da proposta é feita uma analogia do sistema de banco de dados com um almoxarifado onde parte dos agentes é comparada com os almoxarifes.

Palavras-chave: Agentes, Banco de dados, Árvores de decisão

1. Introdução

A quase totalidade das informações corporativas são preservadas em sistemas de banco de dados como conseqüência do registro das informações provenientes dos processos internos destas corporações. Com o passar do tempo, no entanto, muito destes dados podem se tornar apenas uma espécie de arquivo morto, quer seja por alguma motivação legal, que exige a manutenção das informações por um período mínimo de tempo ou simplesmente por não se saber exatamente o que fazer com eles.

Manter estes dados gera custos, pois exige a manutenção de sistemas de armazenamento, gerenciamento de *backup*, infra-estrutura de hardware além da alocação de profissionais para cuidar de tudo isto. Sem mencionar que muitos sistemas informatizados geram no usuário dúvidas sobre a necessidade de continuar alimentando sistemas, que não retornam nada além de simples consultas ou relatórios, a partir de um emaranhado de dados.

Usando técnicas adequadas de busca de informações neste emaranhado de dados pode-se obter conhecimento que se transformará em diferencial

competitivo, aumentará a lucratividade da corporação ou simplesmente agregará maior confiabilidade ao produto. Um bom exemplo de como dados armazenados ao longo do tempo podem ajudar a agregar confiabilidade à um produto, é o uso dos dados armazenados durante o funcionamento de disco rígidos de computadores, para prever antecipadamente a ocorrência de falhas deste dispositivos. Tal tecnologia, conhecida como SMART (*Self-Monitoring And Reporting Technology*) é hoje utilizada pelos principais fabricantes de discos rígidos (MURRAY; HUGHES).

O que é proposto aqui é o uso de agentes de software para a obtenção de regras de indução, empregando regras condição-ação, árvores de decisão ou estruturas similares de representação de conhecimento. Como exemplos podemos citar o uso destas técnicas para fazer o diagnóstico de dispositivos mecânicos, prevenir falhas em transformadores elétricos e prever a formação de grandes tempestades (LANGLEY; SIMON). Outra possível aplicação é a determinação de critérios para classificar as falhas de equipamentos eletro-eletrônicos como reincidentes para, a partir daí, propor desde a sua segregação do sistema onde estão instalados até a determinação da causa da falha ou o seu sucateamento e reposição por um novo.

Mas quando uma solução baseada em agentes é apropriada? Segundo Wooldridge (WOOLDRIDGE, 2002), ela é apropriada quando (a) o ambiente é aberto, ou pelo menos altamente dinâmico, incerto ou complexo, (b) agentes são uma metáfora natural, (c) temos dados, controle ou conhecimento distribuídos e (d) serão usados em sistema legados.

Sob este prisma, a base de dados onde os agentes estarão inseridos pode ser considerada complexa devido à estrutura de dados que ela pode apresentar, principalmente se os bancos de dados estiverem distribuídos em uma rede. O ambiente pode ser completamente ou parcialmente observável dependendo das funcionalidades do agente e, como o ideal é que estes agentes não atuem na base transacional, o ambiente pode ser considerado estático. Com base nesta breve análise do ambiente, principalmente no quesito complexidade, pode-se afirmar que uma solução baseada em agentes é apropriada, faltando agora definir como estes agentes atuarão no banco de dados. Além disto os agentes são uma metáfora natural conforme veremos na próxima seção.

2. Os agentes nos bancos de dados

Se fizermos uma analogia entre um sistema de almoxarifados e um sistema de bancos de dados poderemos ver a equivalência entre os banco de dados, que geralmente agrupam e armazenam dados relacionados entre si, e os diferentes almoxarifados de uma empresa, onde cada um deles armazena produtos com algum tipo de semelhança, como, por exemplo, inflamáveis, alimentos, produtos químicos e etc. Nesta mesma linha, as estantes, organizadas em corredores, podem ser vistas como as tabelas do banco de dados, enquanto cada prateleira guarda semelhança com os campos da tabela. Ainda nesta abordagem, os almoxarifes, responsáveis no mundo real pela guarda e recuperação dos diversos itens armazenados, podem ser entendidos como agentes de software com papéis de arquivistas. Os almoxarifes que preparam e acondicionam os itens antes de serem armazenados correspondem aos agentes pré-processadores, que são responsáveis pela “limpeza dos dados”, descarte de informações pouco significativas ou incoerentes e a discretização de dados contínuos.

Assim, os agentes que vão atuar diretamente no banco de dados, armazenando ou recuperando dados, recebem solicitações de outros agentes e devolvem o resultado da solicitação para o agente solicitante. Também podem ser instruídos a negociar com outros agentes de banco de dados, o acesso e a retirada de dados, quando estes estivem distribuídos em outras bases de dados.

Outros agentes de software, no entanto, podem receber papeis que permitam solicitar e analisar os dados em busca de padrões de comportamento, num processo de aprendizagem não supervisionado durante uma mineração de dados ou, como parte de um sistema de suporte à decisão, estruturando estes dados para a construção de uma árvore de decisão.

No contexto de geração de uma árvore de decisão, os agentes responsáveis por analisar os dados, os agentes analistas, teriam a incumbência

de executar a seqüência de testes nos nós da árvore para cada um dos registros fornecidos pelos agentes arquivistas.

Agora, nesta sociedade de agentes que se forma, é preciso definir qual a melhor arquitetura para cada um dos tipos de agentes, a organização desta sociedade e a linguagem de comunicação entre eles.

3. Definição dos tipos de agentes

Existem basicamente quatro tipos de agentes: (a) agentes reativos simples, (b) agentes reativos baseados em modelo, (c) agentes baseados em objetivos e (d) agentes baseados na utilidade. Estes quatro tipos básico podem ainda ser convertidos em agentes com aprendizado (RUSSEL; NORVIG, 2004).

O agente arquivista tem como função percorrer as tabelas dos bancos de dados procurando por dados relacionados à uma chave de busca do tipo: - *qual registro se refere ao equipamento cujo número de série é X?* - e disponibilizar estes dados para o agente analista que solicitou. Não faz diferença para este agente, conhecer ou não como o ambiente onde ele está inserido evolui e nem como suas ações interferem neste ambiente. Da mesma forma, durante a realização de sua tarefa, o agente arquivista não precisa tomar decisões para atingir seus objetivos – recuperar a informação solicitada – como faria um agente baseado em objetivos. Pela mesma razão ele não precisa melhorar o seu desempenho ou ser mais eficiente como um agente baseado na utilidade.

Baseado nas premissas mencionadas de que o agente arquivista deve simplesmente recuperar e disponibilizar os dados para os agentes analistas, eles podem ser implementados como o tipo mais simples de agente, o agente reativo simples, já que não existe a necessidade de aprendizado para a execução de suas tarefas.

O agente analista por sua vez, tem como função analisar os dados fornecidos pelo agente arquivista e baseado em inferências indutivas, aprender como este se relacionam, descobrindo regularidades ocultas em um processo de aprendizado não supervisionado (GUNNAR, 2004). Outra possibilidade para este agente é, num processo de aprendizado supervisionado, inferir a partir de

exemplos, uma função que retorne uma saída baseada nos dados de entrada (RUSSEL; NORVIG, 2004). Para a proposta em questão, onde se que obter um sistema de apoio à decisão com base nos dados presentes em um banco de dados, propõe-se a utilização de agentes baseados na utilidade, já que eles podem lidar com a incerteza inerente aos ambientes parcialmente observáveis (RUSSEL; NORVIG, 2004).

4. Organização da sociedade de agentes

Uma vez definidos os tipos de agentes e os papeis desempenhados por cada um deles é preciso estabelecer a organização da sociedade de agentes que vão atuar no sistema. Caberá aos agentes arquivistas, dentro desta sociedade, o papel de acessar o banco de dados e recuperar os dados solicitados pelos agentes analistas. Estes por sua vez farão a análise destes dados de acordo com o processo de aprendizagem mais adequado aos objetivos do trabalho. Os agentes analistas poderão, sempre que julgarem necessário, solicitar dados adicionais aos agentes arquivistas.

A quantidade de agentes envolvidos dependerá da complexidade da organização dos bancos de dados e da distribuição destes na arquitetura da rede, se for o caso.

5. Comunicação entre os agentes

Na proposta de organização de agentes, os dois tipos de agentes comunicam-se para fazer solicitações de dados e avisar quando estes estão disponíveis. Um exemplo básico da comunicação entre eles pode ser expresso pelo diálogo a seguir:

O **Agente Analista** solicita dados sobre a falha do equipamento *X* ao agente arquivista.

Agente Analista
(evaluate)Agente arquivista, me informe os próximos dados mais recentes e ainda não recuperados sobre a falha do equipamento de número de série *X*.

Agente Arquivista
(reply) Ok.

O **Agente Arquivista**, após encontrar os dados solicitados entrega o resultado da busca ao agente solicitante.

Agente Arquivista
(stream-about) Agente analista, aqui estão os dados.

Após a primeira análise dos dados o **Agente Analista** decide que precisa de informações adicionais.

Agente Analista
(evaluate) Agente arquivista, me informe os próximos dados mais recentes sobre o reparo realizado no equipamento *X* na data *D*.

...

O exemplo mostra o dialogo entre os dois agentes usando *performativas* da linguagem de comunicação KQML, conforme a representação proposta por ITO (2006, Apêndice 2). No entanto, a determinação de qual das linguagens de comunicação entre agentes disponíveis será utilizada depende de um estudo mais aprofundado do sistema.

6. Construção da árvore de decisão

Conforme mencionado no item 3, onde são definidos os tipos básicos de agentes, o agente analista tem a função de analisar os dados fornecidos pelo agente arquivista. Dentre as formas de analisar estes dados, a indução de uma árvore de decisão a partir de um conjunto de dados de treinamento, é, segundo Rezende (2005), bastante simples.

Árvores de decisão são estruturas de dados que podem ser representadas como um conjunto de regras disjuntas que tem seu início na raiz da árvore e caminha até os nós folhas ou classes, passando por nós de decisão, que testam atributos específicos. No entanto, a grande dificuldade ao se trabalhar com árvores de decisão está em determinar o melhor atributo para particionar a árvore, já que o particionamento é a chave para o sucesso do algoritmo de aprendizado por árvores de decisão (REZENDE, 2005).

Outro fator importante para o sucesso do processo de indução da árvore de decisão, é evitar que ela seja muito específica para o conjunto de dados utilizados no treinamento, produzindo um *overfitting*. Este superajustamento provoca o aumento da taxa de erro quando a árvore é aplicada à um conjunto de exemplos de teste com um grande número de nós (Mitchell, 1998 *apud* Rezende, 2005).

A solução para tentar minimizar este efeito é reduzir o número de nós internos, diminuindo a complexidade da árvore e, por conseguinte, melhorando o seu desempenho. Este processo, conhecido como poda, pode ser aplicado antes ou depois da indução da árvore de decisão, existindo vários métodos para a realização deste processo de poda.

Ao agente analista, caberá a tarefa de induzir a árvore de decisão e realizar a sua poda com base em exemplos de treinamento, aplicá-la em exemplos de teste, para validar o resultado, e finalmente, utilizá-la nos dados que se quer analisar.

Cabe lembrar que, assim como no caso da definição da linguagem de comunicação entre os agentes, a determinação do algoritmo de indução e de poda mais apropriados, vai depender da complexidade dos dados e da precisão desejada.

7. Conclusão

O que foi proposto aqui ainda precisa de um melhor detalhamento, com o uso de técnicas de modelagem adequadas. Esta modelagem será útil para, (a) a determinação precisa dos tipos de agentes e dos papeis que eles devem desempenhar, (b) ter um melhor entendimento destes papéis e das ações a serem executadas, (c) determinar qual a linguagem mais adequada, (d) determinar os algoritmos de indução e poda da árvore de decisão e (e) determinar a melhor forma de implementá-los.

Independente desta caracterização mais precisa dos agentes, o uso deles para o aprendizado supervisionado e não supervisionado, em sistemas de banco de dados, é particularmente interessante quando aplicados em sistemas distribuídos, pois tiram proveito da mobilidade que os agentes podem apresentar para percorrer diferentes bases de dados em diferentes pontos de uma rede, levando consigo algoritmos consagrados de extração de conhecimento.

7. Referências

GUNNAR, R., **A Brief Introduction into Machine Learning**. Disponível em <http://www.ccc.de/congress/2004/fahrplan/files/105-machine-learning-paper.pdf>. Acesso em: 05/05/2006.

ITO, M. **Um modelo de gestão de paciente crônico baseado nos conceitos de relacionamento com o cliente**. 2006. 140 f. Tese (Doutorado) – Escola Politécnica, Universidade de São Paulo, São Paulo, 2006.

LANGLEY, P., SIMON, H. A., **Applications of Machine Learning and Rule Induction**. Disponível em <http://www.cs.uml.edu/~gary/classes/91_421/L9/langley 95 applications.pdf> . Acesso em: 10/07/2006.

MITCHELL, T. M., **Machine Learning**. Boston, USA: McGraw-Hill, 1998 *apud* REZENDE, S. O. (Org.), **Sistemas Inteligentes**: fundamentos e aplicações. Barueri, SP: Manole, 2005.

MURRAY, F. M., HUGHES, G. F., KREUTZ-DELGADO, K, **Machine Learning Methods for Predicting Failures in Hard Drives: A Multiple-Instance Application**, Journal of Machine Learning Research 6 (2005) 783-816. Disponível em <http://jmlr.csail.mit.edu/> Acesso em: 24/07/2006.

REZENDE, S. O. (Org.), **Sistemas Inteligentes**: fundamentos e aplicações. Barueri, SP: Manole, 2005.

RUSSEL S., NORVIG P., **Inteligência artificial.** Rio de Janeiro: Elsevier, 2004.

WOOLDRIDGE, M., **An Introduction to Multiagent Systems**. Indianopolis, USA: Wiley Publishing Inc, 2002.