Descoberta de Conhecimento Aplicada à Detecção de Anomalias em Bases de Dados

Miguel Artur Feldens, José Mauro Volkmer de Castilho

Universidade Federal do Rio Grande do Sul
Instituto de Informática
Caixa Postal 15064
91501-970 Porto Alegre - RS
Brasil
[feldens, castilho]@inf.ufrgs.br

RESUMO - Este trabalho é composto por uma revisão bibliográfica sobre a área de descoberta de conhecimento em base de dados e de detecção anomalias em bases de dados, a especificação de uma ferramenta que se propõe a detectar entidades anômalas em bases de dados, um tipo de descoberta que pode levar a achados interessantes, auxiliar na detecção de erros em bases de dados e de fraudes. Este trabalho inclui ainda a demonstração do um protótipo que está sendo implementado a partir da especificação realizada, que contem mecanismos de busca sobre bases de dados relacionais.

1. Introdução

A descoberta de conhecimento em bases de dados (DCBD) é uma das áreas que investem no desenvolvimento de tecnologias mais poderosas para a recuperação de informações. Embora os sistemas de gerenciamento de bases de dados (SGBD) [KOR94] sejam capazes de acessar informações explicitamente representadas na base de dados, estes não são capazes de detectar padrões, descobrir o conhecimento que está implicitamente representado nos dados e que pode ser generalizado. Esta é a tarefa a que se propõe a DCBD. A figura 1 ilustra a arquitetura hipotética proposta por [MAT94] para um sistema capaz de descobrir conhecimento em bases de dados:

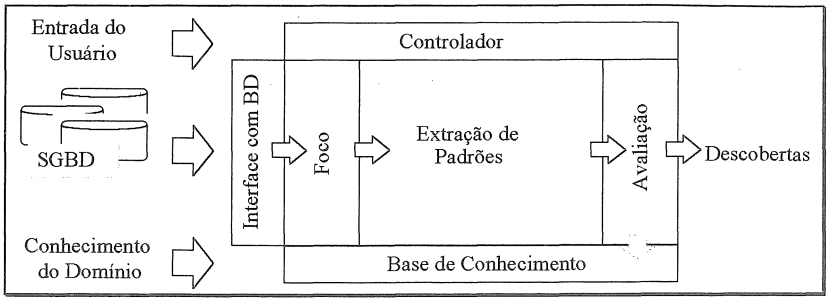


Figura 1 - Modelo de sistema de DCBD

O modelo apresenta seis módulos descritos abaixo:

- **Controlador**: Responsável pelo interfaceamento e a invocação dos demais módulos;
- **Interface com BD**: Responsável pela comunicação com o banco de dados;
- **Base de conhecimento**: Conhecimento do domínio (fornecido e descoberto);
- **Foco**: Aponta quais as seções da base de dados devem ser analisadas;
- **Extração de padrões**: Núcleo do sistema (algoritmos de extração de padrões);
- **Avaliação**: Avalia se os padrões encontrados são "interessantes" e úteis.

O núcleo de um sistema de descoberta de conhecimento tal como apresentado na figura acima é formado pelos algoritmos de extração de padrões. Estes *padrões*, que são as próprias descobertas, denotam associações interessantes entre os elementos contidos na base de dados [MAT94], e podem ser de diferentes tipos. Algumas formas de descoberta são:

- **Análise de dependências**: Há uma dependência entre dois itens quando o valor de um pode influenciar o outro. Em um sistema de informações médicas, por exemplo, poderia ser detectado que determinados procedimentos médicos aparecem sempre associados;
- **Detecção de seqüências**: É a detecção de dependências em relação ao tempo, em uma ordem determinada. Utilizando o exemplo médico, poderia-se descobrir que um determinado procedimento sempre precede outro;

- **Descrição de conceitos** (aprendizado supervisionado): Dado um atributo que indique a classe a que as entidades pertencem, o algoritmo de extração monta uma descrição para cada classe, identificando características comuns entre os membros da classe. É o tipo de aprendizado realizado por uma criança ao observar objetos e receber informações do tipo "é uma cadeira" (exemplos) ou "não é uma cadeira" (contra exemplos);
- **Identificação de classes** (aprendizado não supervisionado): Neste caso, não é informado a que classe pertencem as entidades, cabe ao algoritmos explorar diferentes alternativas, detectar padrões e então descrever conceitos;
- **Detecção de desvio**: Utilizada para detectar anomalias [PAR93] [FEL96] em bases de dados, pode evidenciar problemas de qualidade de dados, fraude, etc. Dado um conjunto de dependências, seqüências e/ou descrições de conceitos, que podem ser obtidos automaticamente, o algoritmo procura os elementos contidos no banco de dados que estão fora destes padrões.

Este trabalho se concentra na análise de dependências e na detecção de desvios, visando evidenciar anomalias em bases de dados. Na seção 2 são definidas as anomalias e o processo de detecção automática. Nas seções 3 e 4 são é apresentadas, respectivamente, a especificação e o protótipo de um sistema capaz de detectar anomalias em bases de dados. Finalmente, são apresentadas conclusões e sugestões para pesquisa futura.

2. Anomalias em Bases de Dados

2.1. O que são anomalias em bases de dados?

Anomalias são exceções a algum padrão esperado de valores para elementos da base de dados. Elas podem ocorrer em bases de dados por diversas razões, inclusive erros [PAR93]. Por isso, a detecção automática de anomalias pode ser uma ferramenta útil no reforço da qualidade de dados. Para tornar a idéia mais clara, seguem exemplos de possíveis causas de anomalias:

- Erros de aplicativo
- Falhas humanas (na entrada de dados, por exemplo)
- Fraudes
- Casos com ocorrência rara (geralmente interessantes para o especialista do domínio)

Os três primeiros casos estão relacionados com a qualidade de dados, podendo indicar que as restrições de integridade precisam ser revisadas (sendo que os resultados da pesquisa podem não apenas indicar a qualidade, mas ainda servir como base para refinar as restrições de integridade).

Ainda que uma anomalia não aponte um erro (casos válidos, mas cuja ocorrência é rara) esta é interessante para o especialista do domínio [MAT93]. Por exemplo, o diagnóstico de um paciente sofrendo de alguma doença supostamente erradicada é potencialmente mais interessante para o médico do que o daqueles pacientes com doenças mais comuns.

2.2. *Detecção automática de anomalias*

Uma vez que os erros em bancos de dados são difíceis de ser detectados devido ao tamanho [ZYT93] e por estes erros estarem onde não são esperados [PAR93], ferramentas para a manutenção da consistência de bancos de dados lançam mão da descoberta de conhecimento. Tal conhecimento, que é extraído por meio de algoritmos de aprendizado, descreve melhor o conjunto de dados do que as chamadas restrições de integridade o fazem. Isto porque as restrições de integridade são um conjunto (possivelmente incompleto) de regras fornecidas *a priori*, enquanto as regras extraídas automaticamente avaliam o seu comportamento corrente; resultando em uma descrição mais completa dos dados armazenados. Um exemplo de regra que poderia ser extraída:

```
Confiança = 99%
IF
      INTERVENÇÃO = "Cesariana"
THEN
      SEXO = "Feminino"
```

Figura 2 - Exemplo de regra

Note-se que, neste exemplo, esta regra foi validada para 99% dos casos na base de dados, o que torna evidente (baseado no conhecimento de que somente mulheres fazem cesariana) a existência de alguma inconsistência.

A segunda etapa do processo de detecção de anomalias é pesquisar os casos que contradizem as regras, dado um limiar de confiança a partir do qual assume-se que uma regra é considerada uma contingência. Para a regra apresentada no exemplo acima seriam detectados os casos em que houvesse INTERVENÇÃO="Cesariana" e SEXO≠"Feminino". A figura 3 ilustra o processo de detecção automática de anomalias em bases de dados:

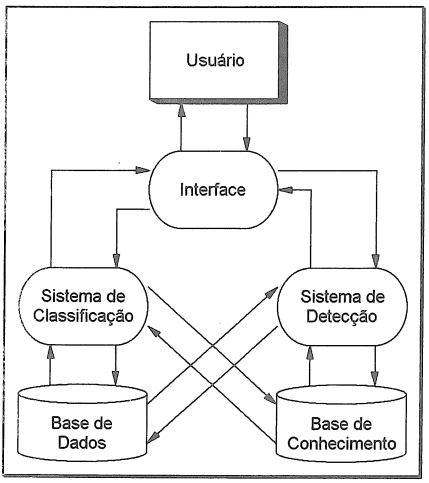


Figura 3 - Modelo para detecção de anomalias

3. DAN - Detector de ANomalias em Bases de Dados

Nesta seção são apresentados conceitos necessários, relacionados com a abordagem utilizada, além da especificação do sistema DAN - Detector de ANomalias.

3.1. Fundamentos

Os valores de atributos associados aos registros ou tuplas das relações indicam as suas categorias [BER]. Sobre o modelo relacional, uma categoria será representada por cada par <atributo,valor> na base de dados. A cada categoria *jc* em uma tabela *D*, sendo um atributo *j* e um valor *c*, corresponde uma *freqüência relativa*, denotada $\|jc\|_D$, correspondente ao número de entidades pertencentes à categoria sobre o número total de registros da tabela.

Uma conjunção de proposições que representam categorias é uma combinação, i.e, uma conjunção de pares <atributo-valor>. O número de categorias que formam uma determinada combinação é o *comprimento* de uma combinação. Assim, a combinação:

$$C = j_1c_1 \wedge \ldots \wedge j_kc_k$$

tem comprimento *k* [PIA93]. A cada combinação corresponde ainda uma *freqüência relativa*. A freqüência de uma combinação *C* sobre uma tabela *D* é denotada $\|C\|_D$.

Dado um par de combinações C_j e C_k, que não contém atributos comuns, podem ser formuladas implicações do tipo $C_j \rightarrow C_k$. Para computar os pesos (*validade* e *cobertura*) associados a uma implicação é criada uma tabela de contingência conforme abaixo:

	C_k	$\neg C_k$
C_j	*a*	*b*
$\neg C_j$	*c*	*d*

onde lê-se:

a é o número de entidades que satisfazem C_j e C_k
b é o número de entidades que satisfazem C_j e não satisfazem C_k
c é o número de entidades que não satisfazem C_j e satisfazem C_k
d é o número de entidades que não satisfazem C_j nem C_k

A *validade* de uma implicação $C_j \rightarrow C_k$ pode ser definida como a freqüência relativa de C_j juntamente com C_k, ou seja, a probabilidade condicional $P(C_j/C_k)$:

$$P(C_j / C_k) = \|C_j \rightarrow C_k\|_D = \frac{\|C_j \wedge C_k\|_D}{\|C_k\|_D} = \frac{a}{a+b}$$

A validade de uma implicação indica quão fortemente a combinação no lado esquerdo (condição) está ligada à combinação do lado direito. Se todas as entidades que satisfazem a condição satisfizerem também a combinação do lado direito, então a validade será igual a um.

A *cobertura* de uma implicação $C_j \rightarrow C_k$ corresponde à freqüência relativa de C_k ocorrendo juntamente com C_j, ou seja, a probabilidade condicional $P(C_k/C_j)$:

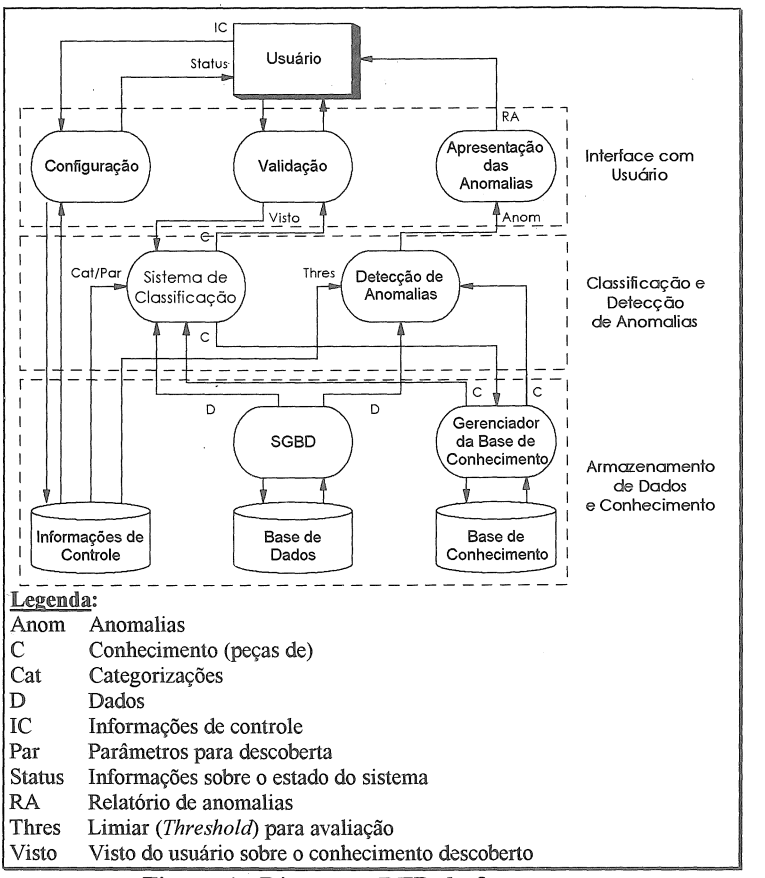
$$P(C_k / C_j) = \|C_k \rightarrow C_j\|_D = \frac{\|C_k \wedge C_j\|_D}{\|C_j\|_D} = \frac{a}{a+c}$$

A cobertura de uma implicação indica quão fortemente o lado direito da implicação está ligado à condição do lado esquerdo. Se todos os objetos que satisfazem o lado direito satisfizerem aquela condição, então a cobertura será igual a um.

3.2. Especificação do sistema

O diagrama abaixo é uma representação funcional do sistema DAN. As principais partes do sistema são descritas a seguir, nas seções 3.2.1 e 3.2.2.

Observa-se que o subsistema de armazenamento de dados e de conhecimento, no protótipo considerado, está baseado em bancos de dados relacionais. Tal opção foi feita buscando simplicidade para o armazenamento de conhecimento descoberto e para implementação de mecanismos de inferência. Em [FEL96] são apresentadas as representações relacionais para regras, implicações e categorias adotadas.



Legenda:

Anom	Anomalias
C	Conhecimento (peças de)
Cat	Categorizações
D	Dados
IC	Informações de controle
Par	Parâmetros para descoberta
Status	Informações sobre o estado do sistema
RA	Relatório de anomalias
Thres	Limiar (*Threshold*) para avaliação
Visto	Visto do usuário sobre o conhecimento descoberto

Figura 4 - Diagrama DFD da ferramenta

3.2.1. Interface com o usuário

Os módulos de *Configuração*, *Validação* e *Apresentação das Anomalias* na figura 4 representam o nível de interface com o usuário do Detetor de ANomalias.

O módulo de *Configuração* recebe do usuário uma série de informações de controle. Algumas destas informações devem, necessariamente, ser fornecidas pelo usuário - por exemplo, nome e localização das tabelas sobre as quais o sistema deve operar, o relacionamento entre as tabelas e sua cardinalidade. Algumas informações podem ser ajustadas pelo usuário ou, em caso

de omissão, assumidos os valores *default*. Um exemplo de ajuste opcional é o valore de limiar de confiança a partir do qual o conhecimento descoberto deve ser utilizado na detecção de anomalias.

A *Validação* é a apresentação ao usuário das regras e/ou implicações entre categorias, já avaliadas pelo sistema, antes de serem pesquisadas as anomalias. Neste momento, o usuário tem oportunidade de descartar regras ou editá-las (caso em que o peso deverá ser recomputado).

A *Apresentação das Anomalias* é responsável pela confecção dos relatórios de descobertas para o usuário. Devem ser apresentadas as anomalias, o conhecimento em que estas anomalias se baseia e as estatísticas gerais sobre o processo de descoberta.

3.2.2. Sistemas de classificação (algoritmos de busca)

Aqui são descritos e apresentados os algoritmos de busca do Detector de ANomalias. O Algoritmo de Descrição de Classes (ADC), responsável pela extração de regras ainda não foi implementado no protótipo e está detalhado em [FEL96]. O Cálculo de Implicações entre Classe e Valor de Atributo (CICVA) verifica valores de atributo que aparecem relacionados às classes (análise de dependências). Esta é uma alternativa para a simplificação do processo de busca, podendo ser utilizado em conjunto ou alternativamente à descrição de classes.

Propõe-se aqui que se utilize implicações entre classes e valores atributos, tipo de implicação que gera regras com antecessor cujo comprimento é igual a um. Se, por um lado, esta representação é menos expressiva que as regras, seu formato é mais simples e compreensível para o usuário. Além disso, acredita-se que as exceções a este tipo de regra são anomalias mais importantes, por serem facilmente investigadas pelo usuário e por serem infrações a regras simples e fortes. O algoritmo abaixo computa as implicações entre classes e valores de atributos:

- **IMP** é uma lista, inicialmente vazia, de implicações $jc \rightarrow C$ com as respectivas validade e cobertura
- **OPEN** é uma lista de implicações $jc \rightarrow C$, ordenada pela freqüência relativa de *jc*, onde a freqüência relativa de *jc* está no intervalo de <*fmin, fmax*>
- Para cada implicação $jc \rightarrow C$ em **OPEN**
  - compute a validade *P(C/jc)* e a cobertura *P(jc/C)*;
  - se (*P(C/jc)* ∈ <*Pmin, Pmax*> & *P(jc/C)* ∈ <*Qmin, Qmax*>)
    - Adicione $jc \rightarrow C$ e as respectivas validade e cobertura na lista de implicações **IMP**

Este algoritmo, tal como apresentado acima, não foi otimizado para o modelo relacional. Seu caráter é experimental e seu desempenho pode ser melhorado através da depuração e testes. O objetivo de sua inclusão neste trabalho é demonstrar o princípio de detecção destas implicações.

3.2.3. Detecção de anomalias

A partir da base de regras e/ou relações, é possível detectar anomalias. Para tanto o algoritmo simplesmente seleciona da base de dados todas as entidades que respeitam o lado esquerdo de uma regra ou relação e não respeitam o lado direito; casos que conferem com a descrição de uma classe e não pertencem a esta classe. Uma vez que esta especificação está sendo construída sobre uma banco de dados relacional e utilizando SQL, o procedimento de busca de anomalias deriva consultas SQL para cada regra a ser considerada. O mesmo algoritmo é válido tanto para uma base de regras quanto de implicações entre classe e valores de atributo.

- **KB** é a lista de regras induzidas, ordenada pelo peso *w* de cada regra *r*
- **R** é a primeira regra de **KB**
- **W** é o peso de **R**
- enquanto **W** > *limiar* e **KB** ainda contém regras
  Início
  - selecionar casos que respeitam antecedente *A* de **R** e não respeitam conseqüente *C* da regra **R**
  - **R** é a próxima regra de **KB**
  - **W** é o peso de **R**

  Fim

4. O Protótipo de DAN

Nesta seção apresenta-se o protótipo para detecção de anomalias implementado.

4.1. Dados utilizados para os testes

Os testes do protótipo foram realizados utilizando dados reais, obtidos através do CPD da UFRGS. Esta etapa do trabalho proporcionou que se experimentasse a seleção de dados para a descoberta, etapa crítica e não automatizada da descoberta de conhecimento, que se baseia no sentimento do engenheiro da descoberta do conhecimento, que deve coletar subconjuntos da base de dados potencialmente interessantes.

A partir do diagrama E-R do sistema acadêmico da UFRGS e do esquema da base de dados correspondente, foram escolhidos subconjuntos relativamente pequenos para os testes, uma vez que ainda não se buscava efetivamente descobrir conhecimento significativo. O subconjunto que aparece neste trabalho consiste nos históricos de 406 alunos de graduação do Instituto de Informática, com todas as respectivas disciplinas aprovadas e matriculadas. O atributo objetivo foi o "status" do aluno (formando ou não). O sistema detecta automaticamente as disciplinas obrigatórias, e as possíveis anomalias seriam os alunos formandos que não cursaram alguma disciplina obrigatória.

Por questões de sigilo, não foram coletados da base de dados os nomes de qualquer pessoa. Pelo mesmo motivo, as chaves dos alunos nos conjuntos de dados foram encriptadas.

4.2. O protótipo

Nesta seção é apresentado o protótipo e os testes realizados. Algumas ferramentas implementadas para viabilizar os testes foram omitidas, mas estão descritas em [FEL96]. O protótipo foi implementado em linguagem Delphi [BOR95], operando diretamente sobre tabelas

em formatos populares e realizando as consultas utilizando SQL [KOR93], tornando viável sua implementação para a arquitetura cliente-servidor.

Até o momento foram implementados os algoritmos de cálculo de implicação entre classes e valores de atributos atributos e os algoritmos que dão suporte. Além disso, assume-se que a classe a que cada entidade pertence é indicada por um único atributo.

Uma vez abertas as tabelas, estas aparecem nas listas de tabelas abertas. Neste momento o usuário deve informar os relacionamentos entre as tabelas. Antes de disparar o processo de busca, o usuário seleciona o atributo que é considerado como objetivo da busca, neste caso, o atributo "Formando" da tabela "Alunoc.db", conforme pode ser visto na figura abaixo:

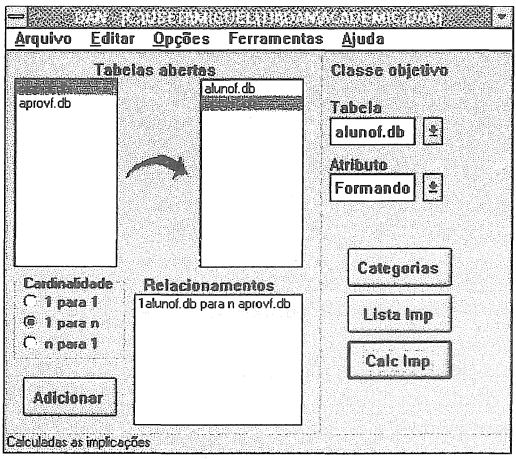


Figura 5 - Interface do protótipo DAN

Os botões que disparam o processo de busca correspondem à geração da lista de categorias da base de dados, à geração da lista de implicações e ao calculo das implicações em si.

A figura 7 contem implicações fortes encontradas nos testes. Note-se que aparecem validades de implicação ("Val") altas para disciplinas obrigatórias do currículo de Informática.

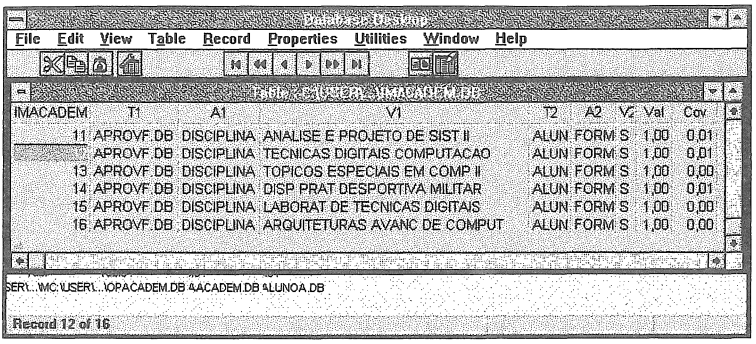


Figura 6 - Resultado de uma busca

Os atributos T1, A1 e V1 correspondem, respectivamente, ao nome da tabela, atributo e valor de atributo no antecessor de cada implicação. Os atributos T2, A2 e V2 correspondem ao nome, atributo e valor do lado direito de cada implicação.

5. Conclusões e trabalho futuro

O trabalho realizado, consistiu no estudo de DCBD, especificação de um sistema para detecção de anomalias em bases de dados e implementação da primeira versão de seu protótipo. Neste trabalho verificou-se a viabilidade do desenvolvimento de sistemas de DCBD sobre uma linguagem de uso geral, com o acesso a bancos de dados via SQL. Acredita-se que com a otimização dos algoritmos levando em conta as características do modelo relacional e da implementação dos demais algoritmos, o sistema DAN pode ser utilizado sobre diferentes bases de dados com objetivos práticos.

Com a implementação dos algoritmos que realizam a descoberta de regras pretende-se comparar as abordagens de detecção de anomalias através de implicações entre classes e valores de atributos e através de regras. Supõe-se que eventualmente a perda de expressividade possa ser considerada aceitável em face do custo da extração de regras.

Outra sugestão de pesquisa futura é a implementação dos algoritmos utilizados sob a forma de uma biblioteca de objetos para DCBD. Este seria o primeiro passo para a generalização do próprio processo engenharia da descoberta de conhecimento. Posteriormente, através da modelagem das associações entre os problemas de descoberta e as abordagens/algoritmos utilizados, imagina-se ser possível a implementação de uma ferramenta que auxilie e automatize parcialmente a implementação de protótipos de descoberta de conhecimento.

6. Bibliografia

[FEL94] FELDENS, M. A. & PALAZZO, L. A. M. Descoberta de conhecimento em bases de dados relacionais. Trabalho de conclusão. Pelotas, Universidade Católica de Pelotas. Novembro de 1994.

[FEL96] FELDENS, M. A. Descoberta de conhecimento aplicada à detecção de anomalias em bases de dados. Trabalho Individual, No. 508, CPGCC, UFRGS, Janeiro de 1996.

[HOL89] HOLLAND, J.; HOLYOAK, K.; NISBETT, R. E. THAGARD, P. R. Induction, processes of inference, learning, and discovery. Cambridge, The MIT Press, 1989.

[HOL94a] HOLSHEIMER, M. & SIEBES, A. Data mining: the search for knowledge in databases. Amsterdan, Netherlands. Available via ftp from ftp.cwi.nl:pub/CWIreports/AA/CS-R9406.ps.Z

[HOL94b] HOLSHEIMER, M. & KERSTEN, M. L. Architectural support for data mining. Amsterdan, Netherlands. Available via ftp from ftp.cwi.nl:/pub/CWIreports/AA/CS-R9429.ps.Z

[KOR93] KORTH, H. F. & SILBERSCHATZ, A. Sistema de bancos de dados. São Paulo, Makron Books do Brasil Editora Ltda. 2a. edição, 1993.

[KDD] KDD-NUGGETS LIST. (Lista de correspondência eletrônica Internet. Moderador: PIATETSKY-SHAPIRO, G. Assinaturas: kdd-request@gte.com. Números antigos via ftp anônimo em ftp.gte.com diretório /pub/kdd/nuggets ou via www: http://info.gte.com/~kdd/)

[MAT93] MATHEUS, C. J.; CHAN, P. K. & PIATETSKY-SHAPIRO, G. Systems for knowledge discovery in databases. IEEE Transactions on Knowledge and Data Engineering. Vol. 5 No. 6. Dezembro de 1993.

[MIC94] MICHIE, D; SPIEGELHALTER, D. J. & TAYLOR, C. C. Machine learning, neural and statistical classification. Ellis Horwood, England, 1994.

[PAR93] PARSAYE, K. & CHIGNELL, M. Data quality control with smart databases. AI Expert. May, 1993.

[PIA93] PIATETSKY-SHAPIRO, G.; MATHEUS, C.; SMYTH, P. & UTHURUSAMY, R. KDD-93: Progress and challenges in knowledge discovery in databases. November, 1993.

[SCH] SCHUMABET et all. Navigation modeling in hypermedia applications.