Um Componente para Mineração de Exceções em Cubos de Dados OLAP

Fábio Pereira[1], Jacques Robin[2]

[1]Universidade Estadual do Sudoeste da Bahia (UESB)

[2]Centro de Informática – Universidade Federal de Pernambuco (UFPE)

{fmp2, jr}@cin.ufpe.br

Abstract. *This paper presents OCCOM, a Java component for the task of mining outliers in a multidimensional and multi-granular data warehouse. OCCOM explores the synergic integration of OLAP and data mining and is based and extends previous theoretical work that mathematically defines the concept of outlier in a data warehouse.*

Resumo. *Este artigo apresenta OCCOM, um componente Java para mineração de exceções em armazéns de dados multidimensionais e multigranulares. OCCOM busca a integração sinérgica entre as tecnologias de OLAP e mineração de dados e fundamenta-se e estende trabalhos prévios que definem matematicamente a noção de exceção em um data warehouse.*

1. Introdução

OLAP (Processamento Analítico On-Line) e mineração de dados têm emergido como duas das técnicas de maior sucesso para suporte a decisão. Elas têm vantagens e limitações complementares. OLAP permite a visualização guiada pelo usuário de dados multidimensionais e pode representar um passo de pré-processamento e seleção eficiente, flexível e iterativo para a mineração de dados. Por sua vez, a mineração de dados permite a descoberta automática de padrões e dados interessantes, podendo representar um guia automático que acelera a procura por percepções (insights) nos cubos OLAP. Investigamos a integração sinérgica entre essas duas tecnologias através da apresentação de OCCOM (OLAP Cuboid Cell Outlier Miner), a primeira ferramenta Java para integração de OLAP e mineração de exceções – uma pequena porção de dados que fogem radicalmente do padrão geral seguido pelos outros dados de um determinado contexto ou amostra.

Analistas de negócios que estão navegando em um cubo OLAP estão freqüentemente procurando por exceções, porque geralmente levam a identificação de áreas problemáticas ou novas oportunidades. Por exemplo, um aumento nas vendas de um produto para uma determinada faixa etária em uma certa época do ano, pode representar uma oportunidade a ser explorada. Durante a navegação em cubos OLAP, o grande volume de dados isolados torna difícil a visualização de exceções através de processo não automático. Para cubos com muitas dimensões e muitos níveis de hierarquia ao longo de cada dimensão, esse processo exploratório manual requer a especificação e avaliação dos resultados de um número combinatorialmente explosivo de consultas OLAP, inviabilizando o uso prático de tal abordagem.

O processo exploratório guiado pela descoberta é uma abordagem alternativa em que medidas pré-computadas que apontam exceções são utilizadas para guiar o usuário

no processo de análise dos dados, em todos os níveis de agregação. OCCOM é a primeira ferramenta que fornece serviços integrados de OLAP e mineração de dados, que seja de fácil utilização, extensão e integração com outras ferramentas, graças às suas seguintes características distintivas: (1) arquitetura de software moderna cliente-servidor multi-usuário, (2) implementação orientada a objetos independente da plataforma, (3) código aberto[1] e (4) interface gráfica amigável para navegação de armazéns guiada pela integração de consultas OLAP com execução de tarefas de mineração. Sendo baseado em uma API Java, os serviços fornecidos por OCCOM podem ser facilmente integrados em qualquer aplicação Java.

O desenvolvimento de OCCOM foi motivado pelo projeto MATRIKS[2] [Robin e Fávero 2001], que propõe a construção de um ambiente abrangente e aberto para KDD (Knowledge Discovery in Databases). O projeto MATRIKS se baseia no principal gargalo dos ambientes atuais de KDD: a falta de integração dos vários serviços computacionais necessários no processo exploratório, iterativo e interativo de KDD.

A próxima seção apresenta os fundamentos teóricos e algorítmicos sobre os quais a mineração de exceção do OCCOM está baseada. A seção 3 apresenta o sistema: arquitetura, funcionalidade e algoritmo. A seção 4 apresenta os resultados alcançados e a seção 5 resume as contribuições de OCCOM e discute direções de trabalho futuro.

2. Algoritmos para Mineração de Exceções em Cubos OLAP

Intuitivamente, uma exceção é o valor de uma célula de um cubo de dados que é significantemente diferente de um valor antecipado, baseado em um modelo estatístico. O modelo considera variações e padrões no valor medido através de todas as dimensões a que a célula pertence. Por exemplo, se a análise das vendas de um item revela um aumento no mês de dezembro em comparação a todos os outros meses, isto poderia ser visto como uma exceção na dimensão tempo. Entretanto, isto não é uma exceção se a dimensão item é considerada, já que ocorre um aumento similar nas vendas para outros itens durante dezembro. O modelo considera exceções escondidas em todas as agregações *group-by* do cubo de dados. Indicações visuais, como a cor de fundo, são utilizadas para refletir o grau de exceção em cada célula, baseado em um valor pré-computado. Desta forma, o indicador de exceção pode ser utilizado para guiar a descoberta de anomalias interessantes nos dados [Han e Kamber 2001].

A seguir veremos modelos estatísticos para computação de exceções em cubos OLAP que serviram como base para desenvolvimento de OCCOM.

2.1 Algoritmo de Sarawagi, Agrawal e Meggido

Sarawagi et.al. (1998) observaram os seguintes fatores para a elaboração de seu modelo: a necessidade de considerar variações e padrões no valor da medida ao longo de *todas as dimensões* a que a célula pertence e a necessidade de encontrar exceções em todas as possíveis agregações do cubo, e não somente no nível mais detalhado, facilitando o entendimento do usuário final por meio de uma representação em camadas concisa. Além disso, o usuário deve ser capaz de interpretar a razão de certos valores estarem marcados como exceção: um usuário típico de OLAP é um executivo, não necessariamente um estatístico sofisticado.

Segundo o modelo, para uma célula na posição i_r da r-ésima dimensão d_r ($1 \le r \le n$), definimos um valor antecipado $\hat{y}_{i1,\ i2,\ \ldots,\ in}$ como uma função f de contribuição das várias agregações de níveis mais altos como:

$$\hat{y}_{i_1,i_2,\ldots,i_n} = f\left(\gamma^{G}_{i_r,d_r \in G} \mid G \subset \{d_1, d_2 \ldots d_n\}\right) \quad (1)$$

A diferença absoluta entre o valor atual, $y_{i1,\ i2,\ \ldots,\ in}$ e o valor antecipado $\hat{y}_{i1,\ i2,\ \ldots,\ in}$ é chamado valor residual do modelo. Qualquer valor com um resíduo de tamanho relativamente alto é uma exceção. Uma definição estatisticamente válida para "relativamente alto" requer que utilizemos uma escala de valores baseada também no desvio padrão antecipado $\sigma_{i1,\ i2,\ \ldots,\ in}$ associado ao resíduo, onde é predefinido algum limiar τ. Então, consideramos $y_{i1,\ i2,\ \ldots,\ in}$ uma exceção se

$$s_{i_1,i_2,\ldots,i_n} = \frac{\left| y_{i_1,i_2,\ldots,i_n} - \hat{y}_{i_1,i_2,\ldots,i_n} \right|}{\sigma_{i_1,i_2,\ldots,i_n}} > \tau \quad (2)$$

A função f na equação 1 pode tomar uma das formas: aditiva ou multiplicativa. Em suas experiências com dados OLAP, a forma multiplicativa forneceu um melhor enquadramento que a forma aditiva.

O procedimento para computar exceções deve ser eficiente e escalável para grandes volumes de dados, comum em bases de dados OLAP. Neste sentido, Sarawagi et al. apresentam dois métodos para computação das medidas de exceção: o método *up-down*, que leva em conta todas as agregações a que a célula pertence; e o método de *reescrita*, mais eficiente e que utiliza técnicas para efetuar o cálculo das exceções praticamente ao mesmo tempo em que agregações são computadas.

2.2 Algoritmo de Chen

Segundo Chen (1999), a forma da função f na equação do valor antecipado (eq. 1) pode ser: (1) Aditiva – modelos lineares; (2) Multiplicativa – modelos log-lineares; (3) Complexa: f é uma mistura das formas aditiva e multiplicativa – modelos mistos. Os modelos mistos não são práticos para cubos com mais de 2-3 dimensões, por causa da sobrecarga na computação do cubo.

A principal diferença do modelo de Chen para o de Sarawagi et al. é o fato de utilizar modelos diferentes, a depender do tipo de função de agregação: quando a função de agregação (medida) é uma quantidade (*count*) ou soma (*sum*), o modelo log-linear introduzido é a melhor escolha; quando a função de agregação é baseada na média (*average*) o modelo linear, introduzido por Chen traz melhores resultados.

2.3 Discussão

Existem ainda outros modelos para detecção de exceções em células de cubos OLAP. Knorr e Ng (1997) desenvolveram uma abordagem utilizando métodos não estatísticos, onde são apresentados algoritmos para detecção de exceções em grandes volumes de dados utilizando testes de discordância específicos. No entanto, a complexidade da aplicação de métodos estatísticos prejudica o desempenho dos algoritmos de mineração.

Escolhemos implementar e estender os modelos desenvolvidos por Sarawagi et al. e Chen pelo fato destes terem sido desenvolvidos especificamente para detectar

[1] Disponível em <http://www.cin.ufpe.br/~fmp2/OCCOM>

[2] Multidimensional Analysis and Textual Reporting for Insight Knowledge Search – projeto do CIn/UFPE (Centro de Informática da Universidade Federal de Pernambuco)

exceções em células de cubos OLAP, além de serem computacionalmente viáveis. Acreditamos que a integração dos dois modelos forneceu uma base teórica sólida e confiável para o desenvolvimento de nossa ferramenta. É importante salientar que não encontramos implementações disponíveis para esses modelos. Se existem, estão embutidas em sistemas de KDD proprietários, monolíticos e fechados e que não podem ser expandidos, não podendo, portanto, ser utilizados fora destes ambientes.

3. OCCOM: OLAP Cuboid Cell Outlier Miner

Para alcançar seus objetivos e metas, as organizações dependem de uma tomada de decisão efetiva: a identificação de problemas e de novas oportunidades representa uma atividade crítica para qualquer organização. Neste contexto, o projeto MATRIKS (Figura 1) visa o desenvolvimento de um ambiente abrangente e aberto para KDDW (Knowledge Discovery in Data Warehouses) e que pretende melhorar o estado da arte em sistemas de suporte à decisão através da integração em um único ambiente das tecnologias de data warehouse, OLAP, mineração de dados e geração automática de hipertextos em linguagem natural.

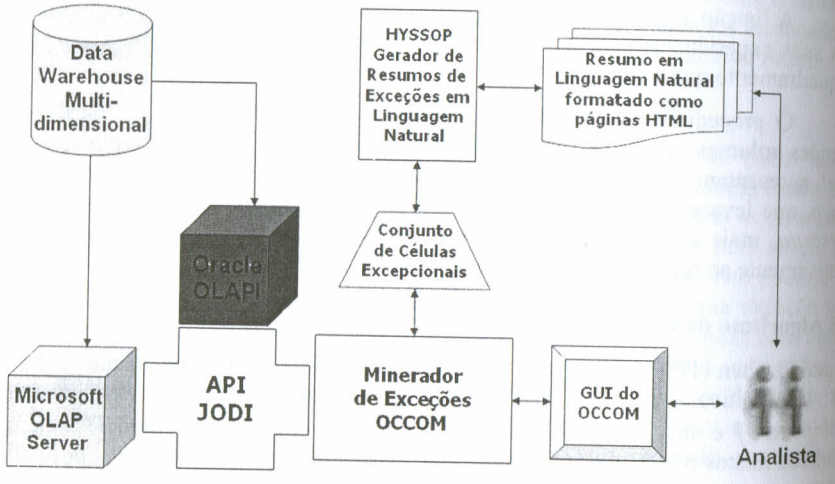

Figura 1. Arquitetura MATRIKS.

O papel principal de OCCOM dentro da arquitetura proposta é localizar anomalias em todos os níveis de agregação em cubos de dados OLAP, fornecendo objetos enriquecidos por mineração que auxiliarão usuários a explorarem cubos OLAP eficientemente. Para mostrar essas informações de uma forma amigável para o usuário, OCCOM pode fazer uso de uma interface gráfica ou se integrar a HYSSOP (Hypertext Summary System for OLAP), um sistema que gera resumos de percepções decisórias obtidas por KDD no formato de hipertexto em linguagem natural [Robin e Favero 2001].

Para a implementação de OCCOM, foi necessário o desenvolvimento do componente chamado JODI – Java OLAP Data Interface, que possibilita o acesso aos dados do servidor OLAP. Detalhes sobre as suas características são mostrados a seguir.

3.1 JODI: Java OLAP Data Interface

JODI é uma interface de programação (API) que implementa o modelo ODCI (Object Data Cube Interface) [Fidalgo 2000] – um modelo abstrato e orientado a objetos para disponibilização e integração de serviços OLAP. A finalidade de JODI é disponibilizar e integrar serviços OLAP, providos pelo servidor *OLE DB for OLAP*[3], com outras tecnologias de suporte à decisão e KDD, no nosso caso, com o minerador de exceções em células de cubos OLAP – OCCOM.

JODI é implementado em uma arquitetura cliente-servidor. A comunicação entre o servidor e o cliente JODI é feita através da API Java RMI (*Remote Method Invocation*), que permite o acesso remoto aos objetos reescritos e exportados pelas interfaces de JODI. Como neste primeiro protótipo escolhemos implementá-lo para o servidor MSOLAP [Peterson e Pinkelman 1999] e este está limitado à plataforma Windows por manipular tipos proprietários COM, o servidor JODI teve que ser desenvolvido para esta plataforma. O cliente JODI é independente de plataforma e, portanto, um sistema portável. JODI mantém a compatibilidade com a proposta de *OLE DB for OLAP*, além de corrigir algumas de suas limitações. Através de JODI é possível acessar os metadados do esquema multidimensional e realizar consultas MDX (MultiDimensional eXpressions).

3.2 O Componente OCCOM

OCCOM foi desenvolvido com com a finalidade de ser facilmente reusado, integrado e distribuído em diferentes aplicações. Para a sua utilização devemos, inicialmente, abrir uma conexão com o servidor JODI, responsável pelo acesso ao servidor OLAP desejado. O cliente deve então escolher e buscar um objeto de cubo de dados (*iCube*), sobre o qual irá aplicar o algoritmo de mineração. Esse objeto deve então ser repassado a OCCOM que, com base nos objetos de metadados multidimensionais de JODI (*iDimension, iHierarchy, iLevel* e *iLevelMember*) irá realizar o cálculo de exceções.

Opcionalmente o cliente OCCOM poderá indicar qual o algoritmo empregado para o cálculo de exceções (modelo linear ou log-linear) e qual o nível de sensibilidade a desvios a ser utilizado (mínimo de 0% e máximo de 30%). Para efetuar o cálculo de exceções, cubóides de dados serão gerados através da realização de consultas MDX a JODI. OCCOM conta ainda com operações como *drill-down* e *roll-up* para que o cliente possa navegar entre cada um dos cubos gerados.

Algoritmo de OCCOM

OCCOM implementa e estende os algoritmos propostos por Sarawagi et. al. (1998) e por Chen (1999). Porém, uma de suas principais características é o fato de ser totalmente parametrizado, assim, o usuário pode escolher qual algoritmo de mineração deseja aplicar: o algoritmo log-linear ou linear.

Diferentemente dos algoritmos de mineração vistos, onde é estabelecido um limiar que indica se uma célula é ou não exceção, OCCOM retorna o grau de exceção com base nos seguintes valores, definidos após a realização de testes em diferentes cubos de dados: se o valor de exceção encontrado for menor ou igual a 1.64 (limiar

[3] *OLE DB for OLAP* é um dos componentes da arquitetura da Microsoft para acesso universal a dados chamada de UDA (*Universal Data Access*).

correspondente a 90% em uma distribuição normal), a célula não possui exceção (grau 0); se o valor encontrado for maior que 1.64 e menor ou igual a 1.96 (correspondente a 95% em uma distribuição normal), a célula possui um grau baixo de exceção (grau 1); se o valor for maior que 1.96 e menor ou igual a 2.58 (correspondente a 99% em uma distribuição normal), a célula possui um grau de exceção médio (grau 2); e se o valor encontrado for superior a 2.58, a célula possui um alto grau de exceção (grau 3).

OCCOM implementa seis medidas de exceção, divididas em duas categorias:

(1) Medidas de Exceção da Célula: (a) *SelfExp* – grau de exceção da célula do cubo; (b) *UnderExp* – grau de exceção que pode ser encontrado abaixo de cada célula caso realizemos uma operação de *drill-down* apenas um nível abaixo da célula; (c) *PathExp* – grau de exceção que pode ser encontrado abaixo de cada célula caso realizemos uma operação de *drill-down* ao longo de determinada dimensão.

(2) Medidas de Exceção do Cubo: (a) *MaxSelfExp* – maior grau de exceção encontrado em alguma célula do cubo; (b) *UnderExp* – grau de exceção encontrado em alguma célula de algum cubo um nível abaixo, resultante de uma operação de *drill-down*; (c) *PathExp* – grau de exceção encontrado em alguma célula de algum cubo resultante de uma operação de *drill-down* ao longo de determinada dimensão.

3.3. GUI OCCOM

Buscamos desenvolver uma interface gráfica simples e amigável, já que OCCOM tem como alvo usuários responsáveis pelo processo de tomada de decisão nas organizações e não usuários com conhecimentos avançados na operação de computadores. A interface possui praticamente duas telas: uma onde o usuário poderá determinar o servidor JODI e o servidor OLAP a que irá se conectar, e setar parâmetros de configuração, além de escolher o cubo de dados e as dimensões sobre as quais aplicará o algoritmo de mineração; e a tela de navegação dos cubos OLAP, onde o usuário poderá verificar as exceções encontradas nas células e realizar operações de *drill-down, roll-up* e *pivot*.

Uma tela no formato da apresentada na Figura 2 será então mostrada ao usuário, que poderá navegar pelas dimensões, efetuando operações de *drill-down* e *roll-up*. Nesta tela, retirada da interface Web de OCCOM, a cor da célula mostra o seu grau de exceção (*SelfExp*), e a cor do texto o grau de exceção encontrado imediatamente abaixo da célula (*UnderExp*). A cor do nome da dimensão mostrada na parte superior da tela irá mostrar o valor da medida *PathExp* para aquela dimensão. Na parte inferior da tela é mostrado o identificador do cubo selecionado. O grau de exceção do cubo (*MaxSelfExp*) é mostrado pela cor do cubo e o grau de exceção que pode ser encontrado imediatamente abaixo dele (*UnderExp*), pela cor de sua letra. No exemplo da Figura 2 podemos visualizar algumas células excepcionais (linha *[Beverages]+[Q4]*). Encontramos ainda exceções abaixo de algumas células das linhas *[Alcoholic Beverages]+[Q1]* e *[Alcoholic Beverages]+[Q3]*, sugerindo ao usuário realizar uma operação de *drill-down* na dimensão *[Time]*.

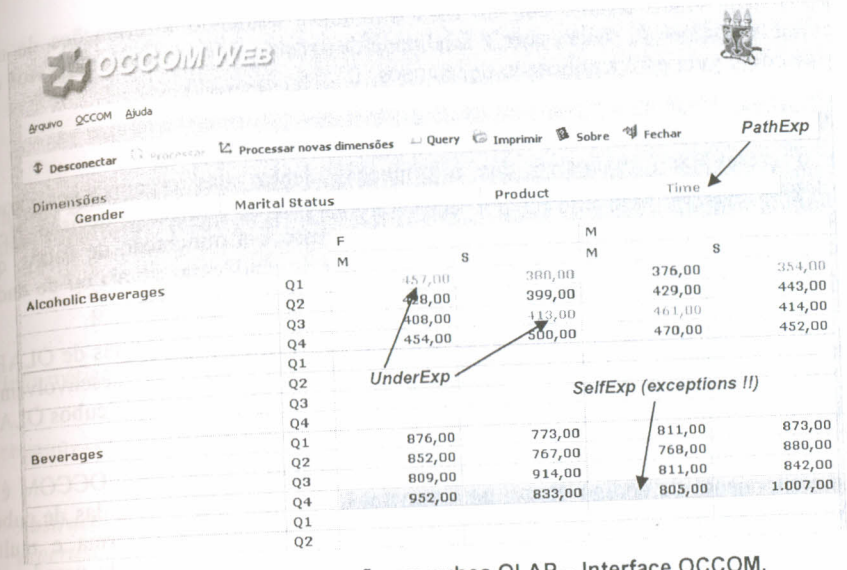

Figura 2. Tela de navegação em cubos OLAP – Interface OCCOM.

4. Testes e Resultados

Com o objetivo de detecção e correção de erros nas aplicações JODI e OCCOM, e assegurar que o software respeite os requisitos especificados, foram realizados testes funcionais, estruturais e de desempenho. Utilizamos para testes a base de dados de exemplo da Microsoft, FoodMart 2000.

Para validação do cálculo de exceções, os resultados de duas metodologias diferentes foram comparados: comparamos os resultados do cálculo utilizando o método *up-down* com os resultados do método de *reescrita*. A estrutura interna das seguintes rotinas críticas do sistema foram testadas em busca de erros: criação de cubóides gerados a partir do cubo base; comunicação com JODI para realização das consultas MDX; cálculo de *All Value* (valor do maior nível de agregação – cubo *All*); cálculo de exceções, que envolve as subrotinas de cálculo do resíduo, cálculo do valor antecipado, cálculo do desvio padrão e cálculo de *SelfExp* (exceção da célula); e cálculo de *UnderExp* e *PathExp* (medidas de exceção derivadas de *SelfExp*).

A partir dos resultados dos testes de desempenho em OCCOM, verificamos que a comunicação com JODI representa cerca de 80% do tempo necessário para o cálculo de exceções. O desafio de melhorar o desempenho de OCCOM passa necessariamente pelo aperfeiçoamento da interface JODI. Notamos que o tempo necessário para o cálculo de exceções está mais ligado ao número de hierarquias e, conseqüentemente, ao número de agregações possíveis, do que ao número de células do cubo base.

Aplicado a um data warehouse contendo informações acadêmicas de uma universidade, OCCOM revelou-se extremamente útil na busca por anomalias de diferentes níveis durante a navegação dos cubos OLAP. Podemos facilmente identificar padrões interessantes como: disciplinas que apresentaram alto grau de reprovação em determinado período, professores responsáveis por altos índices de reprovação, cursos com altos índices de evasão em determinados períodos, entre outros. Atualmente,

OCCOM vem sendo testado em um data warehouse contendo informações de uma secretaria municipal de saúde, com a finalidade de apontar áreas críticas e desvios que servirão como guia para a aplicação de recursos.

5. Conclusão

Com o crescimento explosivo das informações dentro das organizações, novas tecnologias surgiram com o intuito de auxiliar o processo de tomada de decisão. Neste contexto, uma técnica que emergiu com grande sucesso é a mineração de dados, que disponibiliza um serviço automático para descoberta de tendências gerais ou de dados atípicos, fornecendo percepções sobre o domínio abordado.

No intuito de realizar uma integração sinérgica entre as tecnologias de OLAP e mineração de dados, aproveitando a complementaridade entre elas, desenvolvemos OCCOM, um componente Java para mineração de exceções em células de cubos OLAP, que busca guiar usuários a explorarem cubos de maneira eficiente.

Destacamos as seguintes contribuições do trabalho realizado: (1) OCCOM é o único componente de código aberto para mineração de exceções em células de cubos OLAP com características de ambiente cliente-servidor, multi-plataforma e multi-usuário; (2) a combinação de OCCOM com o previamente desenvolvido gerador de hipertexto em linguagem natural HYSSOP, permite a geração automática de resumos, no formato de páginas web, de células de dados que são estatisticamente consideradas exceção em um *datawarehouse* OLAP; (3) OCCOM cobre a principal lacuna na arquitetura do projeto MATRIKS, possibilitando a sua implementação.

Acreditamos que o aperfeiçoamento de OCCOM passa pelas seguintes possibilidades: (1) as células enriquecidas por mineração podem ser persistentes, evitando que o cálculo seja executado sempre que o usuário acesse o sistema; (2) a integração com CWM pode ser uma das alternativas no futuro; (3) desenvolvimento da interface JODI para integração com o Oracle OLAPI; (4) aplicação a um número maior de casos reais, a fim de torná-lo uma ferramenta robusta.

Bibliografia

Chen, Q. (1999) "Mining Exceptions and Quantitative Association Rules in OLAP Data Cube", Master of Science Thesis, Simon Fraser University, 1999, p. 33-77.

Fidalgo, R. N. (2000) "JDCI: Uma API Java para disponibilização e integração de serviços OLAP", Dissertação de Mestrado, Universidade Federal de Pernambuco, Brasil, p. 41-59.

Han, J., Kamber, M. (2001) "Data Mining: Concepts and Techniques", Morgan Kaufmann Publishers, p. 39-104.

Knorr, E. M., Ng, Raymond T. (1997) "A Unified Notion of Outliers: Properties and Computation", In: Proc. of the International Conference on Knowledge Discovery and Data Mining, p. 219-222.

Peterson, T., Pinkelman, J. (1999) "Microsoft OLAP Unleashed", Sams.

Robin, J.; Favero, E. (2001) "HYSSOP: Natural Language Generation Meets Knowledge Discovery in Databases", In: Third International Conference on Information Integration and Web-Based Applications and Services, p. 243-256.

Sarawagi, S., Agrawal, R., Megiddo, N. (1998) "Discovery-driven Exploration of OLAP Data Cubes", In: Proc. of the 6th Int'l Conference on Extending Database Technology (EDBT), Espanha.