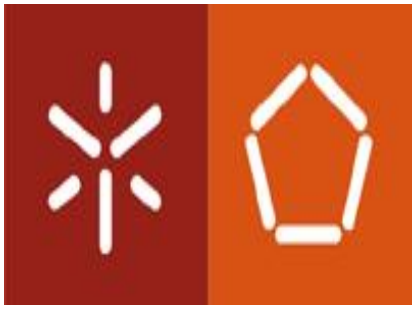


Universidade do Minho
Instituto Federal do Rio Grande do Norte

Orientador: César Analide

Doutorando: José Antônio da Cunha
E-mail:jose.cunha@ifrn.edu.br

Uma Ferramenta de Suporte Educacional, usando Data Mining

Braga, 01 abril de 2014

Sumário

1. Resumo

As escola tem armazenado grande massa de dados, obtidas através do rastreamentos de notas, frequência, grade escolar, compras de livros texto, dados sócio econômicos dos pais, preocupação com a saúde (física e mental) dos alunos, a escolaridade dos pais, dados governamentais e similares. Mas pouco tem sido feito com essas informações - devido a questões de privacidade ou capacidade técnica - para melhorar a aprendizagem dos alunos.

No entanto, com a adoção de mais tecnologias nas escolas e com o acesso mais fácil aos dados governamentais, há claramente uma maior oportunidade para uma melhor coleta de dados e análises desses dados em prol da educação. Infelizmente, muitas escolas e universidades fazem muito pouco com essa riqueza de dados, uma ou outra que possivelmente produz um relatório anual de seus perfis. Mesmo uma simples análise dos dados institucionais poderia levantar perfis dos alunos para um potencial atendimento padronizado ou indicar um atendimento individual ao aluno.

Segundo Rezende (2003), os avanços em hardware e software tem permitido que os computadores tenham aplicações em áreas não convencionais. Como por exemplo, os Sistemas Inteligentes, utilizam a tecnologia da informação para manipular conhecimentos especializados com benefícios qualitativos e quantitativos. Permitindo desta forma que, um maior número de pessoas tenha acesso ao conhecimento a partir da aquisição, sistematização, representação e processamento desse conhecimento.

Os Sistemas Inteligentes podem manipular símbolos que representam entidades do mundo real, e dessa forma, são capazes de trabalhar eficazmente com conhecimento.

A Inteligência Artificial, tem em suas pesquisas, o objetivo de capacitar o computador a executar funções que os seres humanos desempenham usando conhecimento e raciocínio. Então, para que possamos aspirar à ação inteligente, é preciso analisar todos os aspectos relativos ao desenvolvimento e uso da inteligência (Rezende, 2003). Dentro deste contexto, torna-se evidente que a incorporação de conhecimento é um requisito fundamental para a construção de sistemas computacionais inteligentes.

Neste trabalho pretende-se desenvolver uma solução tecnológica que permita ao professor fazer o acompanhamento e a avaliação do aluno a partir das interações destes com um ambiente de ensino. O foco principal do trabalho é usar o processo de descoberta de conhecimento em bases de dados, também conhecido *Knowledge Discovery in Databases* (**KDD**), afim de observar a viabilidade e aplicabilidade de um caso real de apoio a decisão. O estudo inclui também o uso de tecnologias de análise e recuperação de dados úteis ao processo decisório, conhecidas como OLAP e, da aplicação de técnicas e algoritmos de Data Mining para descoberta de novos conhecimentos e padrões nos dados. Este conhecimento adquirido poderá então ser usado na melhoria da educação.

Palavras-chave: Business Intellingence, Mineração de Dados, Descoberta de Conhecimento em Bases de Dados, Raciocínio Baseado em Casos, Redes Neurais, Big Data.

2. Abstract

Educational institutions, in general, have many accumulated data: tracking note, calls: library, cafeteria, scholarship programs, test results, economic data members, registration forms, and so on. But little has been done with the data stored - whether due to political reasons or through ignorance - to improve student learning. At most what you do is take some statistical reports such data.

However, with the increased adoption of technology in schools and greater access to data, including government, there is clearly a lot of opportunities for better data collection and data analysis to promote improved learning in teaching.

In this context, this work aims to develop a tool to assist both students and managers on improving learning in teaching. To this end, the tool must use advanced features of Data Mining and Artificial Intelligence, to diagnose problems related to evasion and failure in school, as well as diagnose problems related to poor performance of students in the disciplines. This tool to diagnose performance problems in the student's particular discipline, will suggest, as challenges, some tasks to the same address, in order to better your performance in that discipline.

Keywords: Business Intellingence, Data Mining, Knowledge Discovery in Databases, Case-Based Racionary, Neural Network, Big Data.

3. Motivação

Um dos grandes problemas dos IFS é identificar o porquê de tanta evasão e reprovações, nos diversos cursos da área tecnológica. Esta dificuldade se justifica por diversos motivos, dentre eles, a falta de estudos mais aprofundados sobre estas questões e a grande quantidade de variáveis envolvidas em tais questões. Sendo assim, faz-se necessário o desenvolvimento de modelos que representem o status de aprendizagem dos alunos. Seguindo esta linha de pesquisa, esta pesquisa propõe uma estratégia para o acompanhamento do aprendizado dos alunos baseada nas práticas de acompanhamento do ensino já utilizados nas escolas, acrescida da tática de análise de dados, onde fatores do acompanhamento podem ser relacionados para se verificar a aprendizagem de forma mais elaborada através da geração de um novo conhecimento descoberto com a utilização de ferramentas de Mineração de Dados.

Considerando esta problemática, esta pesquisa propões uma estratégia para o acompanhamento do aprendizado no contexto de um ambiente de suporte ao ensino. Tal estratégia contempla a modelagem de um conjunto de dados sobre os alunos do Instituto Federal, de forma que esta possa ser devidamente utilizada para a descoberta de conhecimento através de técnicas de Mineração de Dados, onde padrões comportamentais ou características interessantes à cerca do processo de ensino-aprendizagem possam ser encontrados.

O uso de Mineração de Dados é justificado segundo vários motivos:

1) O contingente de alunos é consideravelmente, e, portanto, a tarefa de acompanhar o aprendizado pode-se tornar árdua;
2) Normalmente, os dados são mantidos em bancos de dados e a natureza histórica destes dados pode ser útil para análises prospectivas;
3) Conforme Bernhardt (2001), as decisões baseadas em dados históricos ajudam os educadores a enxergarem quem são seus alunos e quais qualidades e dificuldades eles compartilham;
4) Johnson (2000) destaca que a implantação de um programa de coleta e análise de dados pode levar a melhorias na educação como nenhuma outra inovação o fez;
5) Pouco se tem feito em termos de sistemas de suporte a decisão em educação. A maioria dos ambientes existentes não oferece recursos sofisticados para apoiar decisões.

3.1 Acompanhamento do Aprendizado

Com base nos resultados obtidos pelo sistema, o professor deverá ter à sua disposição uma série de instrumentos de acompanhamento que numa ocasião ou noutra podem ser aplicados, onde os seus resultados podem prestar um grande serviço para a tomada de decisão. Para Linderman (1986), tais decisões podem representar uma **operação preditiva**, ou seja, com base no desempenho presente e passado, deve-se formar um juízo sobre o possível sucesso ou fracasso de um estudante em várias

atividades que ele empreenderá futuramente; ou uma **operação classificatória**, onde o professor classifica os alunos com base na consecução de certos objetivos escolares.

Adicionalmente, uma tendência no acompanhamento da aprendizagem é o processo ***Datadriven Decision Making*** (**D3M**), onde vários fatores são correlacionados para se verificar o aprendizado de maneira mais elaborada. O processo de D3M admite o uso de uma base de dados baseado no aluno, agregando informações sobre a sua vida escolar. Esta base de dados identifica quem cada estudante é (**dados demográficos** como idade, sexo, situação financeira, nível de escolaridade, região de procedência, etc; e **dados comportamentais**, como o número de reprovações, a situação no curso, etc) e o que eles sabem (**dados sobre avaliações**). Assim, o professor pode obter informações individuais, sumariar os resultados dos estudantes (agregação) e reorganizar as informações para entender resultados sob a óptica de diferentes grupos de estudantes (desagregação e análise das informações fazendo cruzamentos de dados).

4. Objetivos

4.1 Objetivos Gerais

Nesta pesquisa pretende-se desenvolver uma ferramenta para auxiliar o professor ou os gestores a diagnosticar problemas na aprendizagem de alunos, em relação ao ensino aprendizagem, tais como: alto índices de desistências e reprovações, bem como, identificar o mau desempenho do aluno nas disciplinas e sugerir soluções para os problemas encontrados. Para isto, será desenvolver uma solução tecnológica (Ferramenta de Suporte à Educação) que permita ao professor fazer o acompanhamento e a avaliação do aluno a partir dos dados coletados de diversas fontes, tais como sistema acadêmico e de formulários de pesquisas. O foco principal do trabalho é utilizar a mineração de dados, para descobrir padrões de comportamento dos alunos dentro do sistema, analisando como os alunos adquiriram conhecimento. Os padrões descobertos poderão então ser usado na melhoria do desempenho desses alunos. Portanto, o objetivo principal desse trabalho de investigação é responder as seguintes questões: **(1)** Quais são causas que implicam no alto índice de desistências e de reprovações dos alunos, **(2)** Porque existe hoje um certo desinteresse por parte dos alunos em alguns cursos ministrados pelo IFRN.

4.2 Objetivos Específicos

- Coletar os dados dos diversos sistema de informação (acadêmico, formulários de pesquisas do aluno, do professor e da instituição);
- Modelar uma estrutura multidimensional utilizando o modelo estrela ou floco de neve;
- Utilizar um aplicativo para extração, transformação e carga dos dados para o sistema multidimensional;
- Criar um Data Warehouse para organizar os dados para os algoritmos de Mineração de Dados;
- Definir os critérios e indicadores de avaliação do desempenho;
- Construir o processo de avaliação de desempenho da aprendizagem na educação (Data Mining);
- Interpretar os resultados obtidos;
- Criar a Base de Casos
- Desenvolver um Visualizador (protótipo) de dados que será utilizado pelo usuário final.
- Desenvolver a solução completa do sistema.

5. Metodologia

Esta sessão tem como finalidade descrever o modo como será desenvolvida esta pesquisa, em relação aos objetivos definidos. Primeiramente será feita uma pesquisa do referencial bibliográfico, que dará um embasamento teórico para o desenvolvimento do projeto (estudo de caso). Fazendo vir à tona os conceitos mais relevantes sobre Banco de Dados, Modelagem Multidimensional, **OLAP**, *Data Warehouse* e Descoberta de Conhecimento em Bases de Dados, mas conhecido como *Knowledge Discovery Data* (**KDD**).

Após o levantamento bibliográfico, será feita a coleta de dados nas diversas fontes de dados do Instituto Federal, desde a base de dados acadêmica até os relatórios estatísticos disponíveis pelas coordenações pedagógicas e administrativas do Instituto Federal, após isso, será dado início ao processo de descoberta de conhecimento em bases de dados.

Feita a coleta dos dados, então será criado o modelo multidimensionais do sistema, com base nos objetivos a serem alcançados, para em seguida, utilizar ferramentas de extração, transformação e carga de dados, para preparar e carregar os dados no modelo multidimensional.

Com o modelo multidimensional pronto, será criado o *Data Warehouse*, para se ter uma base de dados otimizada para se aplicar o(s) algoritmo(s) de Mineração de Dados para descoberta de novos conhecimentos.

Com o *Data Warehouse* criado, deve-se definir que técnicas e algoritmos de Data Mining devem ser utilizados e em seguida definir também as tecnologias a serem utilizadas no processo de descoberta de conhecimentos.

Considerando a complexidade normalmente inerente a processo de descoberta de conhecimento em bases de dado, a metodologia proposta utiliza como base princípios de planejamento de atividades. Dessa forma, em função dos objetivos de cada aplicação de KDD, os passos do processo de descoberta de conhecimento deverão ser planejados ante do início de sua execução. A aplicação da metodologia será dividida em quatro momentos.

A primeira etapa envolve a definição sobre "o que fazer" diante da base de dados apresentada. Nesta etapa, devem ser executadas as tarefas de "Levantamento Inicial" e de "Definição de Objetivos".

O levantamento inicial compreende um exame preliminar da base de dados, procurando obter informações sobre a natureza dos dados a serem analisados.

Na definição de objetivos, devem ser identificadas quais as tarefas de Mineração de dados são viáveis, para atender as expectativas e às necessidades do usuário do domínio da aplicação. Nesta fase, devem ser formulados alguns requisitos quanto ao modelo de conhecimento a ser produzido.

A partir da escolha de um objetivo, a abordagem é direcionada para a definição sobre "como fazer", que corresponde a etapa de "Planejamento de Atividades". Nesta

etapa devem ser definidas as alternativas de plano de ação associados ao objetivo escolhido. Os planos de ação devem ser constituído a partir de cada método de mineração de dados aplicável à tarefa de KDD associado ao objetivo selecionado.

Finalmente, a abordagem proposta é concluída pela etapa de "Avaliação de Resultados". Essa etapa corresponde à "analise do que foi feito". Neste momento, as características do modelo de conhecimento gerado devem ser confrontados com as expectativas quanto ao modelo formulados na etapa "Definição de Objetivos".

Este processo deve ser iterativo e interativo, de forma que, dependendo dos resultados obtidos, os analistas de KDD possam retornar a qualquer etapa realizada anteriormente em busca de melhores resultados. Para que isso seja possível, a metodologia requer uma documentação detalhada das ações realizadas e dos resultados produzidos.

Todo esse processo será englobado em uma ferramenta, onde o usuário possa manipular facilmente, para selecionar as informações utilizadas no processo e também visualizar os resultados obtidos.

6. Calendário de Atividades Programadas

Para o desenvolvimento do projeto, devem ser realizadas diversas atividades, tais como coletar dados de diversas fontes (sistema acadêmico, de formulários de pesquisas aplicados pelos pedagogos da instituição, contra alunos e professores), e dados de formulários aplicadas pela instituição para que os seus servidores avaliem a instituição quanto a infraestrutura, seu corpo docente, biblioteca, e assim por diante. Esta tarefa deve durar em torno de trinta dias.

De posse desses dados, a tarefa seguinte é modelar a base de dados multidimensional. Esse armazém de dados receberá os dados a partir de uma outra atividade, que é a extração, transformação e carga dos dados das diversas fontes de dados. Esta tarefa, por envolver fontes de dados de diversos formatos (planilhas, documentos texto, e banco de dados), ela deve gastar mais tempo, estimo, em torno de dois meses.

A etapa seguinte é a fase de definição dos indicadores de desempenho, que serão utilizados pelo sistema para identificar a real situação do aluno, em relação a sua aprendizagem. Estes indicadores vão ser definidos juntamente com a equipe pedagógica e, como trata-se de um assunto que é bastante conhecido pelos mesmos, acredito que, será definido em pouco tempo, estimo quinze dias aproximadamente.

Tem-se que criar uma Base de Casos. Esta Base de Casos será desenvolvida, juntamente com o(s) especialista(s) do setor educacional (pedagogos), pois os mesmos são detentores de conhecimentos dos reais problemas dos alunos e, de suas soluções aplicadas até o momento. Esta é uma tarefa que deve-se ter bastante atenção, principalmente na indexação dos casos, portanto, estimo em torno de dois meses, o tempo de conclusão dessa tarefa.

Em seguida, será realizado o processo de Mineração de Dados, e consequentemente, diante dos resultados, a interpretação dos mesmo. Este processo é bastante trabalhoso, pois temos de definir quais são as métricas a serem utilizados, em cada situação problema a ser analisado. Portanto, estimo aproximadamente dois meses, para esta tarefa.

E finalmente, o desenvolvimento de uma ferramenta para ser manipulada pelos usuários (Gestores, Professores e aluno). Uma ferramenta de suporte educacional, que é o objetivo deste trabalho. Devo aqui frisar que, o desenvolvimento da interface do sistema, ou seja, da ferramenta de suporte educacional, objetivo desse trabalho, vai ser realizado em paralelo com as demais atividades.

Deve-se também frisar que durante a fase de desenvolvimento, deverá também ser produzido artigos, a serem submetidos em congressos. Descrevo a seguir, alguns dos temas que são possíveis de serem desenvolvidos: Mineração de Dados Educacionais usando *Knowledge Discovery in Databases* - **KDD**, Big Data e Análise de Dados, Analisando Dados Educacionais usando Banco de Dados NoSQL e a Linguagem R.

Na Tabela. 61 apresento um quadro resumido cronológico de todas as atividades (calendário preliminar).

Tabela 6.1 Calendário de atividade de desenvolvimento do projeto

Atividade	Data inicial	Data final
Coletar dados	01/05/2014	30/05/2014
Modelar base de dados multidimensional	02/06/2014	30/08/2014
Preparar e carregar dos dados (ETL)	01/09/2014	28/11/2014
Criar os Cubos (Data Marts)	01/12/2014	15/12/2014
Definir os indicadores de desempenho	16/12/2014	30/12/2014
Modelar todo processo Data Mining e realizar testes	01/01/2015	31/03/2015
Interpretar os resultados	01/04/2015	30/04/2015
Criar a base Casos	01/05/2015	30/06/2015
Desenvolver a ferramenta proposta no projeto	Será desenvolvida em paralelo com as outras atividades	
Realizar teste para analisar os resultados	Será feito continuamente	
Artigos	Serão feitos em demanda.	

7. Business Intelligence

Segundo Rob (2011), o termo *Business intelligence* (**BI**) é utilizado para descrever um conjunto amplo, coeso e integrado de ferramentas e processos utilizados para captar, coletar, integrar, armazenar e analisar dados para a geração e a apresentação de informações que deem suporte à tomada de decisões de negócio. Como o próprio nome diz, **BI** trata da criação de inteligência sobre o negócio. Portanto, o **BI**, é um modelo que permite à empresa transformar dado em informação, informação em conhecimento e conhecimento em sabedoria.

O **BI** não é, por si só, um produto, mas um modelo de conceitos, práticas, ferramentas e tecnologias (*data warehouse*, *data mart*, **OLAP** e/ou ferramentas de mineração de dados) que auxiliam uma empresa a compreender melhor seus recursos centrais e identificam oportunidades fundamentais para criar competitividade (Rob, 2011). Em geral, o **BI** envolve as seguintes etapas:

- Coleta e armazenamento de dados operacionais.
- Agregação de dados operacionais em dados de suporte a decisões.
- Análise de dados de suporte a decisões para gerar informações.
- Apresentação dessas informações ao usuário final para dar suporte a decisões de negócios.
- Tomada de decisões de negócio, o que, por sua vez, gera mais dados que são coletados, armazenados etc. (reiniciando o processo).
- Monitoramento para avaliar os resultados das decisões de negócio (Rob, 2011).

7.1 Arquitetura de Business Intelligence

Segundo Rob (2011), o **BI** utiliza-se de tecnologias e aplicações para o gerenciamento de todo o ciclo de vida dos dados, da aquisição ao armazenamento, transformação, integração, análise, monitoramento e apresentação. Não existe uma arquitetura única de **BI**, no entanto, há alguns tipos gerais de recursos, que são compartilhados por todas as implementações de **BI**.

Para compreender a arquitetura de **BI**, será feita uma descrição dos componentes básicos que fazem parte de sua infraestrutura. Alguns desses componentes, possuem recursos adicionais. Porém, há quatro componentes básicos que todos os ambientes de **BI** devem fornecer, descritos as seguir (Rob, 2011):

- **Ferramentas de extração, transformação e carregamento (ETL) de dados**: esse componente é encarregado de coletar, filtrar, integrar e agregar dados operacionais a serem salvos em um armazém de dados otimizado para o suporte a decisões.
- **Armazenamento de dados**: o armazém de dados é otimizado para o suporte a decisões e costuma ser representado por um *data warehouse* ou data mart. Ele contém dados de negócios extraídos de bancos de dados operacionais e de

fontes externas. Esses dados são armazenados em estruturas otimizadas, com foco na velocidade de análise e consulta.

- **Ferramentas de consulta e análise de dados**: esse componente executa as tarefas de recuperação, análise e mineração, utilizando os dados no armazém de dados e os modelos de análise de dados de negócio. Tal componente é utilizado pelo analista de dados para criar as consultas que acessam o banco de dados. Essa ferramenta orienta o usuário sobre quais dados selecionar e como construir um modelo de dados confiáveis. Tal componente costuma aparecer na forma de uma ferramenta **OLAP**.
- **Ferramentas de apresentação e visualização de dados**: esse componente é encarregado de apresentar os dados ao usuário final de várias formas. É utilizado pelo analista de dados para organizar e apesentar os dados. Essa ferramenta ajuda o usuário final a selecionar o formato de apresentação mais adequado, como relatório resumido, mapa ou gráfico.

Deve-se ficar atento para o fato de que, os bancos de dados de suporte a decisões tendem a ser muito grandes. Muitos chegam à faixa dos gigabytes ou terabytes. Esses bancos de dados, demandam por análise sofisticada de dados e, incentivaram a criação de um novo tipo de armazém de dados. Esse armazém de dados, contém dados em formatos que facilitam sua extração, análise e a tomada de decisões. É conhecido como *data warehouse* e se tornou o fundamento de uma nova geração de sistemas de tomada de decisões (Rob, 2011).

7.2 Data Warehouse

Segundo Inmon (1994), o termo ***data warehouse*** é "um conjunto de dados integrado, orientado por assunto, variável no tempo e não volátil, que fornece suporte a tomada de decisões". A seguir, será detalhado cada um desse componentes (Inmon, 1994):

- **Integrado**. O *data warehouse* é um banco de dados consolidado e centralizado, que integra dados proveniente de toda a organização e de várias fontes, com diversos formatos.
- **Orientado por assunto**. Os dados do *data warehouse* são dispostos e otimizados de modo a fornecerem respostas a perguntas provenientes de diversas áreas funcionais da empresa. São organizados e resumidos por temas, contendo assuntos de interesse especifico – produtos, clientes, departamentos, regiões, promoções, e assim por diante.
- **Variável no tempo**. Os sistemas transacionais focam nas transações correntes, enquanto os sistemas de *data warehouse*, representam o fluxo de dados através do tempo. Ou seja, os dados são carregados periodicamente no *data warehouse*, e quando isso acontece, todas as agregações dependentes do tempo (ou se dependentes dessa carga de dados), são recalculadas. Por exemplo, se os dados de vendas da semana, são carregados no *data warehouse*, serão atualizadas todas as agregações dependentes dessa carga, ou

seja, os agregados semanais, mensais, anuais e de qualquer outras periodicidade que seja dependente dessa carga. Cada conjunto de dados, ao ser carregado em um *data warehouse*, fica vinculado a um rótulo temporal que o identifica dentre os demais. Cada rótulo temporal, fica portanto, associado a uma visão instantânea e sumarizada dos dados operacionais que corresponde ao momento de carga do *data warehouse*. Dessa forma, na medida que o data warehouse vai sendo carregado com tais visões, pode-se realizar análise de tendências a partir dos dados.

- **Não volátil**. Uma vez inserido um dado no *data warehouse*, ele nunca será removido. Uma vez que ele representa o histórico da empresa. Por este fato, o *data warehouse* está sempre crescendo. Portanto, o SGBD, que dá suporte a ele, deve ser capaz de suportar vários gigabytes de dados, ou até mesmo tera-bytes, operando com hardware com diversos processadores.

Resumindo, o *data warehouse* é um reposotório de dados semanticamente consistente, que serve como uma implementação física de um modelo de dados de apoio a decisões. Ele armazena as informações que uma empresa necessita para tomar decisões (Han & Kamber, 2011). Normalmente é um banco de dados apenas de leitura, otimizado para processamento de análises e consultas. Em geral, os dados são extraídos de diversas fontes e, em seguida, transformados e integrados, antes de serem carregados no *data warehouse* (Inmon, 1994). A Figura 2.2 ilustra como o data warehouse é criado a partir dos dados contidos em um banco operacional.

7.3 Arquitetura de Data Warehouse

Segundo Han & Kamber (2011), um **data warehouse** adota uma arquitetura em três camadas:

1. A camada inferior (*bottom*) é um servidor de *data warehouse*, que quase sempre, é um sistema de banco de dados relacional. Segundo Han & Kamber (2011), são usadas ferramentas de back-end e utilitários para extrair dados dessa camada e alimentar a camada superior. Ainda segundo Han & Kamber (2011), os dados são extraídos usando interface de programação de aplicativo, conhecidos como ***gateways***. Um *gateway* permite que clientes gerem código SQL, para ser executado no servidor. Pode-se citar como exemplos de *gateways* ODBC (*Open Database Connection*) e OLEDB (*Object Linking and Embedding Database*) da Microsoft e JDBC (*Java Database Connection*). Essa camada também contém um repositório metadata, o qual armazena informações sobre o *data warehouse* e seus conteúdos.
2. A camada intermediária (*middle tier*), segundo Han & Kamber (2011), é um servidor OLAP que geralmente é implementado usando (1) um modelo relacional **OLAP** (**ROLAP**) (fornece recursos de OLAP utilizando bancos de dados relacionais e ferramentas familiares de consulta relacional para armazenar dados multidimensionais; ou (2) um modelo multidimensional

OLAP (**MOLAP**) (amplia os recursos de OLAP para sistemas de gerenciamento de banco de dados multidimensionais (**SGBDM**s). O SGBDM utiliza técnicas especiais para armazenar dados em matrizes de n dimensões. O pressuposto do MOLAP é que os bancos de dados multidimensionais são os mais adequados para gerenciar, armazenar e analisar dados multidimensionais (Rob, 2011).

3. A terceira camada (top), segundo Han & Kamber (2011), é o fronte-end do cliente, a qual contém as ferramentas de consulta, de relatório, de análise e mineração de dados (por exemplo, análise de tendência, previsão, e assim por diante.).

Segundo Han & Kamber, do ponto de vista da arquitetura, há três modelos de data warehouse: o warehouse empresarial, o data mart e warehouse virtual.

7.4 Data Mart

De acordo com Inmon (1994), embora o *data warehouse*, seja uma proposta muito atraente, que traga muitos benefícios, os gerentes podem relutar em adotar essa estratégia, pelo fato de que, a criação de um *data warehouse* exige tempo, dinheiro e considerável esforço gerencial. Estes fatos, fazem com que muitas empresas iniciem na criação de *data warehouse*, focando em conjuntos de dados gerenciais, orientados a atender pequenas áreas de negócio, dentro da empresa. Esses armazenamentos menores são chamados de ***data marts***. Um ***data mart*** é portanto, segundo Inmon (1994), um pequeno subconjunto de um *data warehouse*, sobre um único assunto, que fornece suporte às decisões de um pequeno grupo de pessoas. No entanto, pode-se criar um *data mart* a partir de dados extraídos de um *data warehouse*, com a finalidade especifica de dar suporte a um acesso mais rápido a determinado grupo ou função. Dessa forma, os *data marts* e o *data warehouse* podem coexistir em um ambiente de business *intelligence* (Inmon, 1994).

7.5 Virtual Data Warehouse

De acordo com Han & Kamber (2011), um warehouse virtual é um conjunto de visões sobre bases de dados operacionais. Você pode materializar algumas visões operacionais, para obter um processamento de consultas eficientes. O warehouse virtual é o estado de visibilidade global de recursos, com base na aquisição e processamento de dados operacionais em tempo real. Informações disponíveis no armazém virtual tem o potencial de reduzir custos e melhorar o serviço ao cliente. A infraestrutura já está disponível para captura de dados em tempo real, e o custo de aquisição de dados continuará a reduzir.

7.6 Processamento Analítico On-Line

De acordo com Rob (2011), a necessidade de suporte a decisões mais intensivo, levou à introdução de uma nova geração de ferramentas. Tais ferramentas, foram denominadas de **processamento analítico on-line** (**OLAP** – *Online Analytical*

Processing). Essa nova ferramenta cria um ambiente avançado de análise de dados que dá suporte à tomada de decisões, modelagem comercial e pesquisa operacional. Ainda segundo Rob (2011), esses sistemas comportam quatro características principais:

- Utilizam técnicas de análise de dados multidimensionais.
- Proporcionam suporte avançado a bancos de dados.
- Fornecem interface fácil de utilizar para o usuário final.
- Dão suporte a arquitetura cliente/servidor

7.7 Técnicas de Análise de dados Multidimensionais

De acordo com Rob (2011), a característica mais evidente das modernas ferramentas **OLAP**, é a capacidade de análise multidimensional, onde, os dados são processados e visualizados como parte de uma estrutura multidimensional. Essas técnicas de análise de dados multidimensionais, utilizam as seguintes funções:

- Funções avançadas de apresentação de dados. Gráficos 3D, pivô, tabulações cruzadas, rotação de dados e cubos tridimensionais.
- Funções avançadas de agregação, consolidação e classificação de dados. Permitem que o analista de dados crie vários níveis de agregação, detalhamento de dados, *drill down* e *roll up* de dados em diferentes dimensões e níveis.
- Funções computacionais avançadas. Incluem variáveis orientadas para negócio (tais como participação de mercado, margem de vendas, etc.), relações financeiras e contábeis (tais como lucratividade, despesas gerais, alocação de custos e retorno), funções estatísticas e de previsão.
- Funções avançadas de modelagem de dados. Dão suporte para cenários de simulação, avaliação de variáveis, contribuições de variáveis para o resultado, programação linear, dentre outras, ferramentas de modelagem.

7.8 Modelagem Multidimensional

A modelagem multidimensional é uma forma de Modelagem de Dados voltada para concepção e visualização de conjunto de medidas que descrevem aspectos comuns de um determinado assunto. É utilizada especialmente para sumarizar e reestruturar dados, apresentando-os em visões que suportem a análise dos dados envolvidos (Passos & Goldschimdt, 2005).

De acordo com Han & Kamber (2011), o *data warehouse* e as ferramentas **OLAP** são baseadas em um **modelo de dados multidimensional**. Nesse modelos, os dados são visto na forma de um cubo de dados. Um modelo multidimensional possui três componentes básicos: Fatos (***facts tables***), Dimensões (***dimensions***) e Medidas (***measures***). E existem diversas formas de modelagem física de um data warehouse, incluindo esquema estrela (***star schema***), esquema floco de neves (***snowflake***) e

constelação de fatos (***fact constellation***). A seguir serão discutidas cada uma desses conceitos.

7.9 Esquema Estrela

O esquema estrela, segundo Rob (2011), é uma técnica de modelagem de dados multidimensionais de suporte a decisões em um banco de dados relacional. Ainda segundo, Rob (2011), o esquema estrela foi desenvolvido, pois as técnicas de modelagem relacional, entidade relacionamento (**ER**) e normalização existentes não produziam uma estrutura que atendesse às necessidades de análise avançada de dados.

O modelo estrela é fácil de implementar e, ao mesmo tempo em que preserva as estruturas relacionais, em que o banco operacional criado. O esquema estrela básico possui quatro componentes: fatos, dimensões, atributos e hierarquias de atributos.

7.10 Fatos

Um **fato** é uma coleção de itens de dados, composta de dados de medidas e de contexto. Representa um item, ou uma transação ou um evento associado ao tema da modelagem. São medidas numéricas (valores) que representam um aspecto ou atividade específica dos negócios. Os fatos normalmente utilizados em análise de dados comerciais são unidades, custos, preços e receitas. Os fatos são armazenados em tabelas de fatos que constituem o centro do esquema estrela. A **tabela de fatos** (***fact table***) contém fatos vinculados por meio de suas dimensões (Kimball, 2002, Passos & Goldschimdt, 2005).

Segundo Rob (2011), os fatos também podem ser computados ou derivados no momento da execução. Esses as vezes são chamados de métricas para diferenciá-los dos fatos armazenados.

7.11 Dimensões

Uma dimensão é um tipo de informação que participa da definição de um fato. As dimensões determinam o contexto do assunto. As **dimensões** são características de qualificação que fornecem perspectivas adicionais a um determinado fato. Os dados de suporte a decisões são quase sempre vistos relacionados a outros dados, por isso, as dimensões são interessantes. Por exemplo, pode-se, em um sistema de suporte a decisões, querer comprar as vendas de certos produtos, entre regiões e entre períodos. Nesse exemplo, teríamos as vendas, as dimensões produto, localização e tempo. Segundo Rob (2011), as dimensões ampliam a visão dos fatos. Essa dimensões são armazenadas em **tabelas de dimensões**.

7.12 Medidas

Uma medida é um atributo ou variável numérica que representa um fato. Exemplos: valor da ação, número de evasões escolares, quantidade de produtos vendidos, valor total de venda, etc.

7.13 Atributos

De acordo com Kimball (2002), cada tabela de dimensão contém atributos. Os atributos costumam ser utilizados para buscar, filtrar e classificar fatos. As dimensões fornecem características descritivas sobre os fatos por meio de seus atributos.

Segundo Rob (2011), conceitualmente, o modelo de dados multidimensional do exemplo de vendas, é melhor representado por um cubo tridimensional. Isso, não significa que haja um limite para o número de dimensões, que podem ser associadas a uma tabela de fatos. Não há limite matemático para o número de dimensões utilizadas. Usar um modelo tridimensional, torna mais fácil a visualização do problema.

7.14 Hierarquias de Atributos

De acordo com Kimball (2002), os atributos no interior de dimensões podem ser ordenados em hierarquias bem definidas. A hierarquia de atributos fornecem uma organização vertical utilizada para duas finalidades principais: agregação e análise de dados por ***drill down*** e ***roll up***.

7.15 Esquema Floco de Neve

Segundo Rob (2011), para facilitar a navegação do usuário final, utiliza-se a técnica de normalização das tabelas dimensionais. Esse esquema normalizado é conhecido como esquema floco de neves, que nada mais é do que, um tipo de esquema estrela, no qual as tabelas de dimensões podem ter suas próprias tabelas de dimensões. Resumindo, o esquema floco de neve resulta normalmente da normalização de tabelas de dimensão. Por exemplo, se a tabela da dimensão de localização contém dependência transitivas entre região, estado e cidade, é possível normalizar esta tabela dimensão para a 3FN (terceira forma normal).

Essa normalização, simplifica as operações de filtragem de dados relacionados a dimensão. No entanto, há um preço a pagar por ela, pois aumenta-se a complexidade das consultas SQL. Por exemplo, caso se deseje agregar os dados por região, deve-se utilizar uma junção de quatro tabelas.

De acordo com Passos e Goldschmidt (2005), existem diversos operadores OLAP que permitem acessar os dados em modelos multidimensionais. A seguir encontra-se indicados alguns deles:

- **Drill up/down** – Utilizado para amentar ou reduzir o nível de detalhe da informação acessada. Exemplo: Vendas por país, Vendas por estado, etc.
- **Slicing** – Utilizado para selecionar as dimensões a serem consideradas na consulta. Exemplo: Visualizar as vendas, separadas por país e por mês.
- **Dicing** – Utilizado para limitar o conjunto de valores a ser mostrado, fixando-se algumas dimensões. Exemplo: Vendas de um determinado estado, de um determinado produto em um determinado ano.
- **Pivoting** – Utilizado para inverter as dimensões entre linhas e colunas. Exemplo: Ao visualizar vendas por produto e por estado, aplicar o operador para visualizar as vendas por estado e por produto.
- **Data Surfing** – Executar uma mesma análise em outro conjunto de dados. Exemplo: Ao avaliar as vendas no Brasil, aplicar o operador para realizar a mesma consulta em Portugal.

Ainda de acordo com Rob (2011), quando um sistema de BI é implementado em áreas geograficamente dispersas, as técnicas de particionamento e replicação são especialmente importantes. O **particionamento** separa a tabela em subconjunto de linhas ou colunas e coloca esses subconjuntos próximos ao computador cliente, melhorando, dessa forma, o tempo de acesso, por outro lado, a **replicação** faz uma cópia da tabela e a coloca em uma localização diferente, também com a finalidade de aprimorar o tempo de acesso.

Resumindo, projetar um *data warehouse* significa receber a oportunidade de ajudar a desenvolver um modelo integrado que capture os dados considerados essenciais para a organização, tanto da perspectiva do usuário final, como da perspectiva dos negócios. Para tanto, um projeto de *data warehouse*, deve satisfazer:

- Critérios de integração e carregamento de dados.
- Recursos de análises de dados com desempenho aceitável de consulta.
- Necessidades de análises de dados do usuário final.

Segundo Rob (2011), a preocupação técnica mais evidente na implementação de um *data warehouse* é fornecer ao usuário final suporte a decisões com recursos avançados de análise de dados – no momento certo, no formato certo, com os dados certos e ao custo certo.

8. Mineração de Dados

O termo **Mineração de dados**, também conhecido como Descoberta de Conhecimentos em Bancos de Dados, ou **KDD** (do inglês, "*Knowledge Discovery in Databases*"), refere-se a disciplina que tem como objetivo descobrir "novas" informações através da análise de grandes quantidades de dados (Witten, 2005). O termo "novas informações" refere-se ao processo de identificar relações entre dados que podem produzir novos conhecimentos e gerar novas descobertas científicas.

As informações sobre a relação entre dados e, posteriormente a descoberta de novos conhecimentos, podem ser muito úteis para realizar atividades de tomada de decisão. Por exemplo, ao minerar os dados de um estoque de supermercado poder-se-ia descobrir que todas as sextas-feiras um determinado se esgota nas prateleiras e, portanto, um gerente com posse desta "nova informação" poderia planejar o estoque do supermercado para aumentar a quantidade desse produto específico as sextas-feiras. Analogamente, é possível minerar dados de alunos para verificar a relação entre uma abordagem pedagógica e o aprendizado do aluno. Através desta informação o professor poderia compreender se sua abordagem realmente está ajudando o aluno e desenvolver novos métodos de ensino mais eficazes. A Mineração de dados tem sido aplicada em diversas áreas do conhecimento, como por exemplo, vendas, bioinformática, e ações contra-terrorismo (Baker, 2009).

Recentemente, com a expansão dos cursos a distância e também daqueles com suporte computacional, muitos pesquisadores da área de Informática na Educação (em particular, Inteligência Artificial Aplicada à Educação) têm mostrado interesse em utilizar mineração de dados para investigar perguntas científicas na área de educação (e.g. quais são os fatores que afetam a aprendizagem? Ou como desenvolver sistemas educacionais mais eficazes?). Dentro deste contexto, surgiu uma nova área de pesquisa conhecida como "**Mineração de Dados Educacionais**" (do inglês, "*Educational Data Mining*", ou EDM). A EDM é definida como a área de pesquisa que tem como principal foco o desenvolvimento de métodos para explorar conjuntos de dados coletados em ambientes educacionais. Assim, é possível compreender de forma mais eficaz e adequada os alunos, como eles aprendem, o papel do contexto na qual a aprendizagem ocorre, além de outros fatores que influenciam a aprendizagem. Por exemplo, é possível identificar em que situação um tipo de abordagem instrucional (e.g. aprendizagem individual ou colaborativa) proporciona melhores benefícios educacionais ao aluno. Também é possível verificar se o aluno está desmotivado ou confuso e, assim, personalizar o ambiente e os métodos de ensino para oferecer melhores condições de aprendizagem (Baker, 2009).

8.1 Métodos para Mineração de Dados

Existem muitos métodos utilizados em Mineração de Dados (Witten, 2005). Em um modelo de Mineração de Dados Educacionais, a validação cruzada permite verificar a corretude (Witten, 2005) de um modelo gerado a partir da análise de dados de treinamento (*training data*). Essa validação oferece uma estimativa de como o modelo irá se comportar ao analisar um conjunto novo dados. Validação cruzada ao nível de aluno ou classe é fundamental em dados educacionais, pois existe uma grande quantidade de

dados por aluno e as conclusões obtidas ao utilizar métodos de mineração de dados precisam garantir que o modelo encontrado possa ser utilizados para inferir o comportamento ou a aprendizagem dos alunos e/ou classe.

Existem várias métodos utilizados na Mineração de Dados. Assim, nos parágrafos a seguir será feita uma breve introdução de alguns dos tópicos mais interessantes da área.

Uma taxonomia dos principais métodos de pesquisa em Mineração de Dados é apresentada em (Baker, 2009):

- Predição (*Prediction*)
    - Classificação (*Classification*)
    - Regressão (*Regression*)
    - Estimação de Densidade (*Density Estimation*)

- Agrupamento (Clustering)
- Mineração de relações (*Relationship Mining*)
    - Mineração de Regras de associação (*Association Rule Mining*)
    - Minieração de Correlações (*Correlation Mining*)
    - Mineração de Padrões Sequenciais (*Sequential Pattern Mining*)
    - Mineração de Causas (*Causal Mining*)

As sub–categorias de Predição: Classificação, Regressão e Estimação de Densidade estão diretamente relacionadas as categorias dos métodos de mineração de dados apresentados por Moore (2005).

Na área de **predição**, a meta é desenvolver modelos que deduzam aspectos específicos dos dados, conhecidos como variáveis preditivas (*predicted variables*), através da análise e fusão dos diversos aspectos encontrados nos dados, chamados de variáveis preditoras (*predictor variables*). A Predição necessita que uma certa quantidade dos dados seja manualmente codificada para viabilizar a correta identificação de uma ou mais variáveis predicionada previamente conhecidas (a codificação e a identificação das variáveis não precisam ser perfeitas). Como indicado na taxonomia, existem três tipos de predição: classificação, regressão, e estimação de densidade. Em classificação, a variável predicionada é binária ou categórica. Alguns algoritmos populares na EDM, disponíveis em ferramentas como o RapidMiner (Mierswa, 2006), incluem árvores de decisão, regressão logística (para predições binárias), e regressão step. Quando a variável predicionada é um número, os algoritmos de regressão mais populares incluem regressão linear, redes neurais, e máquinas de suporte vetorial. Para classificação e regressão, as variáveis preditoras podem ser categóricas ou numéricas; métodos diferentes ficam mais (ou menos) efetivos, dependendo das características das variáveis preditoras utilizadas (Baker, 2008).

Existem dois benefícios de se utilizar métodos de predição em modelos educacionais. Primeiro, métodos de predição são utilizados para estudar quais aspectos de um modelo são importantes para predição, dando informação sobre o construto sendo examinado (exemplos de constructo modelado incluem curvas de aprendizagem e representações de tipos variados de comportamento). Esta estratégia é frequentemente utilizada em pesquisas que tentam, de forma direta, predizer os benefícios educacionais para um conjunto de estudantes (Romero, 2008), sem primeiro predizer os fatores mediantes ou intermediários. Ou seja, o objetivo é verificar o quanto o aluno aprender sem considerar as diversas variáveis que influenciam a aprendizagem como, por exemplo, variáveis relacionadas ao comportamento do estudante (Pavlik, 2008). Segundo, os métodos de predição auxiliam a predizer o valor das variáveis utilizadas em um modelo. Essa abordagem é necessária, pois analisar *todos* os dados de um grande banco de dados para gerar um modelo é tipicamente financeiramente inviável, além de consumir muito tempo (Baker, 2008 – pg 38-47). Assim, o modelo pode ser construído utilizando parte dos dados e então ser aplicado para modelar dados mais extensos (Baker, 2008 – pg 287-314). Esse tipo de técnica pode auxiliar no desenvolvimento e uso de atividades instrucionais, pois consegue estimar os benefícios educacionais antes mesmo da atividade ser aplicada com os alunos.

Na área de **agrupamento**, o objetivo principal é achar dados que se agrupam naturalmente, classificando os dados em diferentes grupos e/ou categorias. Estes grupos e categorias não são conhecidos incialmente. Através de técnicas de agrupamento os grupos/categorias são automaticamente identificados através da manipulação das características dos dados. É possível criar esses grupos/categorias utilizando diferentes unidades de análise, por exemplo é possível achar grupos de escolas (para investigar as diferenças e similaridades entre escolas), ou achar grupos de alunos (para investigar as diferenças e similaridades entre alunos), ou até grupos de atos (para investigar padrões de comportamento) (Amershi, 2009).

Em **mineração de relações**, a meta é descobrir possíveis relações entre variáveis em bancos de dados. Esta tarefa pode envolver a tentativa de aprender quais variáveis são mais fortemente associadas com uma variável específica, previamente conhecida e importante, ou pode envolver as relações entre quaisquer variáveis presentes nos dados. Para identificar essas relações, existem quatro tipos de mineração: (a) regras de associação; (b) correlações; (c) sequências; ou (d) causas.

Na **mineração de regras de associação**, procura-se gerar/identificar regras do tipo *se-então* (*if-then*) que permitam associar o valor observado de uma variável ao valor de uma outra variável. Ou seja, caso uma condição seja verdadeira (e.g. variável *Y* possui valor 1) e uma regra associe essa condição ao valor de uma outra variável *X*, então podemos inferir o valor desta variável *X*. Por exemplo, ao analisar um conjunto de dados seria possível identificar uma regra que faz a associação entre a variável "*objetivo do aluno*", uma variável binária que pode ter os valores *alcançado* ou *não alcançado*, e uma outra variável binária "*pedir ajudar ao professor*" que pode ter os valores *sim* ou *não*.

Neste contexto, **se** o aluno tem como objetivo aprender geometria, mas está com dificuldade (i.e. a variável *objetivo do aluno* tem valor *não alcançado*), **então** é provável que ele peça ajuda do professor (i.e. a variável *pedir ajuda ao professor* tem valor positivo).

Em **mineração de correlações**, a meta é achar correlações lineares (positivas ou negativas) entre variáveis. Por exemplo, ao analisar um conjunto de dados, seria possível identificar a existência de uma correção positiva entre uma variável que indica a quantidade de tempo que um aluno passa externalizando comportamentos que não estão relacionados as tarefas passadas pelo professor (e.g. conversas paralelas, brincadeiras e outras perturbações que ocorrem em sala de aula) e a nota que este aluno recebe na próxima prova.

Em **mineração de sequências**, o objetivo principal é achar a associação temporal entre eventos e o impacto destes eventos no valor de uma variável. Neste caso, é possível determinar qual trajetória de atos e ações de um aluno pode, eventualmente, levar a uma aprendizagem efetiva. Dessa forma, é possível criar um conjunto de atividades instrucionais que podem melhorar a qualidade do ensino fazendo com que os alunos externalizem ações que vão ajudá-lo a construir seu conhecimento e desenvolver as habilidades necessárias para trabalhar com conteúdo sendo apresentado pelo professor.

Em **mineração de causas**, desenvolve-se algoritmos e técnicas para verificar se um evento causa outro evento através da análise dos padrões de covariância (uma sistema que faz isso é TETRAD (Scheines, 1994)). Por exemplo, se considerarmos o exemplo anterior onde um aluno externaliza comportamentos inadequados que não contribuem para resolver a tarefa dada pelo professor. Nesta situação o aluno, em muitos casos, recebe uma nota ruim na prova final. Nesta situação, o comportamento do aluno pode ser a causa dele não aprender e, assim, resultado em uma performance ruim na prova. Contudo, pode ser que o aluno externalize tal comportamento inadequado devido à dificuldade em aprender, e portanto, a causa da performance ruim na prova não é o comportamento em si, mas sim a dificuldade de aprendizagem do aluno. Analisando o padrão de covariância, a mineração de causa pode inferir qual evento foi a causa do outro.

8.2 Aplicações de Mineração de Dados

As tecnologias de Mineração de Dados podem ser aplicadas em grande variedade de contextos de tomada de decisão. Em particular, áreas de significativo retorno de investimento esperado incluem:

- ***Marketing*** – aplicações como análise de comportamento do consumidor baseados em padrões de consumo; a definição de estratégias de *marketing* incluem propaganda, localização de lojas e mala direta direcionada; segmentação de clientes, layouts de lojas e campanhas de publicidades.

- **Finanças** – aplicações incluem análise de crédito de clientes, segmentação de contas a receber, análise de performance de investimentos financeiros como ações, bonds e fundos mútuos e detecção de fraudes.
- **Produção**- aplicações envolvem otimização de recursos como máquinas, força de trabalho e materiais; projeto ótimo de processos de fabricação e projeto de produção de automóveis baseados nos requisitos dos clientes.
- **Saúde** – aplicações incluem descoberta de padrões em imagens radiológicas, análise de dados experimentais em microarray (gene-chip) para relação com doenças, análises de efeitos colaterais de remédios e efetividade de certos tratamentos, etc.

8.3 Conclusões

Nesta sessão foi apresentada a disciplina de Mineração de Dados, que utiliza tecnologias para descobrir conhecimento adicional ou padrões nos dados. Foram discutidas várias técnicas, priorizando os detalhes das regras de associação, a classificação e o agrupamento. Apresentou-se também, alguns algoritmos para algumas dessas áreas.

Como a Mineração de Dados é uma área em constante desenvolvimento, diversas dificuldades e desafios estão surgindo continuamente, as quais precisam ser atacadas. Portanto, nessa sessão foram apresentadas algumas das principais tecnologia que dão suporte ao processo de Mineração de dados, sem a pretensão de esgotá-las.

9. Big Data

Big Data é um termo que vem chamando a atenção pela acelerada escalada em que volumes cada vez maiores de dados são criados pela sociedade. Fala-se comumente em petabytes de dados gerados a cada dia, e zetabytes começa a ser uma escala real e não mais futurista. A uma década atrás terabyres era uma quantidade futurista, agora temos em nosso próprios computadores. Muito tem sido escrito sobre Big Data e como ele pode servir como base para a inovação, diferenciação e crescimento da análise de dados em grandes massa de dados (Kolb, 2013).

De acordo com Raj (2013), as tecnologias que sustentam o Big Data, podem ser analisadas sob duas óticas: as envolvidas com análise de dados, tendo Hadoop e Map-Reduce como as principais e as tecnologias de infraestrutura, que armazenam e processam os dados. Neste aspecto, destacam-se os banco de dados NoSQL (Not Only SQL).

O termo Big Date está diretamente ligado a questões como volume, variedade, velocidade, complexidade e valor (Mayer-Schönberger, 2013).

- **Volume** – o volume está claro. Geramos petabytes de dados a cada dia. E estima-se que este volume dobre a cada 18 meses.
- **Variedade** - Variedade também, pois estes dados vêm de sistemas estruturados (hoje minoria) e não estruturados (a imensa maioria), gerados por e-mails, mídias sociais (Facebook, Twitter, YouTube e outros), documentos eletrônicos, apresentações estilo Powerpoint, mensagens instantâneas, sensores, etiquetas RFID, câmeras de vídeo, etc.
- **Velocidade** - De acordo com o Gartner, velocidade significa tanto o quão rápido os dados estão sendo produzidos quanto o quão rápido os dados devem ser tratados para atender a demanda. Etiquetas RFID e contadores inteligentes estão impulsionando uma necessidade crescente de lidar com torrentes de dados em tempo quase real. Reagir rápido o suficiente para lidar com a velocidade é um desafio para a maioria das organizações.
- **Valor** - E valor porque é absolutamente necessário que a organização que implementa projetos de Big Data obtenha retorno destes investimentos. Um exemplo poderia ser a área de seguros, onde a análise de fraudes poderia ser imensamente melhorada, minimizando-se os riscos, utilizando-se, por exemplo, de análise de dados que estão fora das bases estruturadas das seguradoras, como os dados que estão circulando diariamente nas mídias sociais.
- **Complexidade** - Quando você lida com grandes volumes de dados, eles vêm de diversas fontes. É um grande desafio vincular, correlacionar, limpar e transformar os dados de um sistema. No entanto, é necessário conectar e correlacionar interações, hierarquias e vínculos múltiplos de informação ou então os dados podem rapidamente sair de controle. Governança de dados pode ajudar a determinar como os dados díspares se relacionam com definições comuns e como integrar sistematicamente os ativos de dados

estruturados e não estruturados para produzir informações de alta qualidade, uteis, adequadas e atualizadas.

Em última análise, independentemente dos fatores envolvidos, acreditamos que o termo Big Data é relativo e se aplica (por avaliação do Gartner) sempre que a capacidade da organização de gerenciar, armazenar e analisar os dados exceder sua capacidade atual.

9.1 O Uso do Big Data

Os modelos relacionais, quando proposto por Edgar F. Codd, atenderam muito bem, a demanda era acessar dados estruturados, de acordo com (Elmasri & Navathe (2005), gerados pelos sistemas internos das corporações. Estes modelos não foram desenhados para tratar dados não estruturados e nem para volumes de dados na casa dos petabytes de dados.

Para tratar dados na escala de volume, variedade e velocidade do Big Data precisamos de outros modelos. Surgem os softwares de banco de dados NoSQL, desenhados para tratar imensos volumes de dados estruturados e não estruturados. Existem diversos modelos como sistemas colunares como o *Big Table* (Usado internamente pelo Google),o modelo Key/value como *DynamoDB da Amazon*, o modelo "*document database*" baseado no conceito proposto pelo Lotus Notes da IBM e aplicado em softwares como MongoDB, e o modelo baseado em grafos como o Neo4j, etc (Kolb, 2013).

Aplicações modernas de mineração de dados, frequentemente chamada "Big-Data Analytics", exigir-nos gerenciar grande quantidade de dados rapidamente e em muitas dessas aplicações, exige-se um amplo paralelismo (Kolb, 2013).

Para lidar com aplicações tais como essas, novos tipos de software tem surgido. Estes sistema de programação são projetados para obter o máximo do paralelismo. O novo tipo de software começa com uma nova forma de sistema de arquivos, chamada "Sistema de arquivos distribuídos", que contam com unidades muito maiores do que os blocos de disco dos sistemas operacionais convencionais. Além do mais, os sistemas distribuídos também fornecem replicação de dados ou redundância para proteger os dados, contra falhas frequentes de mídias, que ocorrem quando o dado é distribuído para milhões de nós de computadores (Kolb, 2013).

No topo destes sistemas de arquivos, diversos sistema de alto nível de programação foram desenvolvidos. No centro do novo software está o sistema de programação chamada ***Map-Reduce***. Implementações de ***Map-Reduce*** permite que os cálculos sob os dados em grande escala, sejam executados em clusters de computação de forma eficiente e tolerante a falhas de hardware (kolb, 2013).

9.2 Map-Reduce

Map-reduce não é um produto ou um software especifico, mas sim uma tecnologia desenvolvida pelo Google para lidar com grande quantidade de dados, cortando-os e combinando-os no final (Kolb, 2013).

A ideia básica é que os dados que precisam ser processados, entram no sistema, e é cortado em pedaços chamados de chunks. Essas peças de software que é responsável em fazer esses cortes é chamado de "***Mapper***". Os chunks são então enviados para outra peças de software para fazer o processamento requerido sobre eles, e então eles são ainda enviados para outra peça de software chamado "***Reducers***" que combina o resultado final para a saída (Kolb, 2013).

Todas essas peças de software - mapeador, processador, e redutor – tipicamente rodam no mesmo servidor de uma só vez. Dessa forma, a carga de processamento dos dados mapeados podem ser espalhados por todo servir disponível, e mais servidores podem ser adicionado em tempo real, se for necessário um resultado mais rápido (Kolb, 2013).

O importante aqui, é você entender que, a tecnologia de Map-Reduce é capaz de pegar uma grande quantidade de dados, que seria muito dispendioso rodar em apenas um servidor, e poder distribui-lo por vários servidores. Este é um novo paradigma de programação. Existem algumas ferramentas que implementam esse novo paradigma, entre elas cito, o **Hadoop** do *Apache Foundation* (Kolb, 2013)

9.3 As Tarefas de Mapeamento

De acordo com Kolb (2013), o arquivo de entrada para uma tarefa Mapeamento, consiste de elementos, que podem ser de qualquer tipo: uma tupla ou um documento, por exemplo. Um chunk é uma coleção de elementos, e nenhum elemento é armazenado em dois chunks. Tecnicamente, todas as entradas para as tarefas de mapeamento (The Map Tasks) e saídas para as tarefas Redução (The Reduce Tasks) são os pares na forma chave-valor (key-value), geradas por uma função hash. Essa forma de entradas e saídas são motivadas pelo desejo de permitir a composição de vários processos Map-Reduce (Kolb, 2013).

A função de mapeamento (Map) recebe um elemento com seus argumentos e produz zero ou mais pares chave-valor. Os tipos de chaves e valores são arbitrários. Mais, as chaves não são "chaves" no sentido usual; elas não precisam ser únicas. Mais uma tarefa de mapeamento pode produzir vários pares chave-valor com a mesma chave, mesmo a partir do mesmo elemento (Kolb, 2013).

Exemplo 4.1: Suponha que deseja-se contar o número de ocorrências para cada palavra em uma coleção de documentos. Neste exemplo, o arquivo de entrada é um repositório de documentos, e cada documento é um elemento. A função de mapeamento para este exemplo usa chaves que são do tipo String (a palavra) e valores que são inteiros. A tarefa

de mapeamento lê um documento e quebra ele em uma sequência de palavras w_1, w_2, w_3, ..., w_n. Ela então emite um sequência de pares de chave-valor onde o valor é sempre 1. Isto é, a saída da tarefa de mapeamento para este documento é a sequência de pares chave-valor:

$$(w_1,1),(w_2,1),....,(w_n,1)$$

Note que uma simples tarefa de mapeamento irá processar muitos documentos - todos os documentos em um ou mais chunks. Assim, a saída produzida será mais do que a sequência para o documento sugerida acima. Note também que se uma palavra w aparece m vezes entre todos os documentos atribuídos a esse processo, então haverá m pares chave-valor (w,1) entre sua saída. Uma opção para resolver esse problema é usar agrupamento e agregação, que é combinar esses m pares em um simples par (w, m), isso só é possível porque as tarefas Redução, aplica uma operação associativa e comutativa, para os valores.

9.4 Agrupamento e Agregação

O processo controlador mestre sabe quantas tarefas Reduce haverá, digamos r tarefas, pois o usuário normalmente informa ao sistema map-reduce quais são as r tarefas. Então o controlador mestre aplica uma função hash e produz uma tabela de chaves de números (códigos) de 0 até r-1. Cada chave produzida pela tarefa Map é um hash e seus pares chave-valor são colocados em um arquivo local. Cada arquivo é destinado para um das tarefas Reduce (Kolb, 2013).

Após todas as tarefas Map terem completadas com sucesso, o controlador mestre junta os arquivos de cada tarefa Map que são destinados para uma particular tarefa e alimenta o arquivo resultante com uma lista de pares chave-valor. Isto é, para chave k, a entrada para a tarefa Reduce que manipula a chave k é um par da forma (k, [v_1, v_2, ..., v_n]), onde (k, v_1), (k, v_2), ..., (k, v_n) são todos pares chave-valor e k, vindo de todas as tarefas Map.

9.5 As Tarefas de Redução

Os argumentos da função Reduce é um par consistindo de uma chave e sua lista de valores associados. A saída da função Reduce é uma sequência de zero ou mais pares chave-valor. Esses pares chave-valor pode ser de tipo diferente daqueles enviados das tarefas map para as tarefas Reduce, mais normalmente elas são do mesmo tipo. Referimo-nos a aplicação da função Reduce que reduz para uma simples chave e seus valores associados de redutor (Kolb, 2013).

Uma tarefa reduce recebe uma ou mais chaves e sua lista de valores associados. Isto é, uma tarefa reduce executa um ou mais redutores. As saídas de todas as tarefas reduce são juntas em um simples arquivo. Redutores podem ser divididos em tarefas

reduce menores e a função hash associa cada chave com um dos códigos da tabela hash (Kolb, 2013).

Exemplo 4.2: Vamos continuar com o exemplo conta palavras do Exemplo 4.1. A função Reduce simplesmente agrega todos os valores. A saída de um redutor consiste da palavra e da soma. Isto é, a saída de todos as tarefas Reduce é uma sequência de pares (w,m), onde *w* é uma palavra que aparece pelo menos uma vez entre todos os documentos e *m* é o total de ocorrências de *w* em todos os documentos.

9.6 Detalhes de Execução de Map-Reduce

A Figura 1 oferece um esboço de como processo, tarefas, e arquivos interagem. Aproveitando uma biblioteca fornecida por um sistema map-reduce tal como Hadoop, o programa do usuário bifurca o processo controlador mestre e alguns dos processos Worker para diferentes nós de computação. Normalmente, um Worker manipula suas tarefas Map (um Map worker) ou tarefas Reduce (um Reduce worker), mas não ambos (Kolb, 2013).

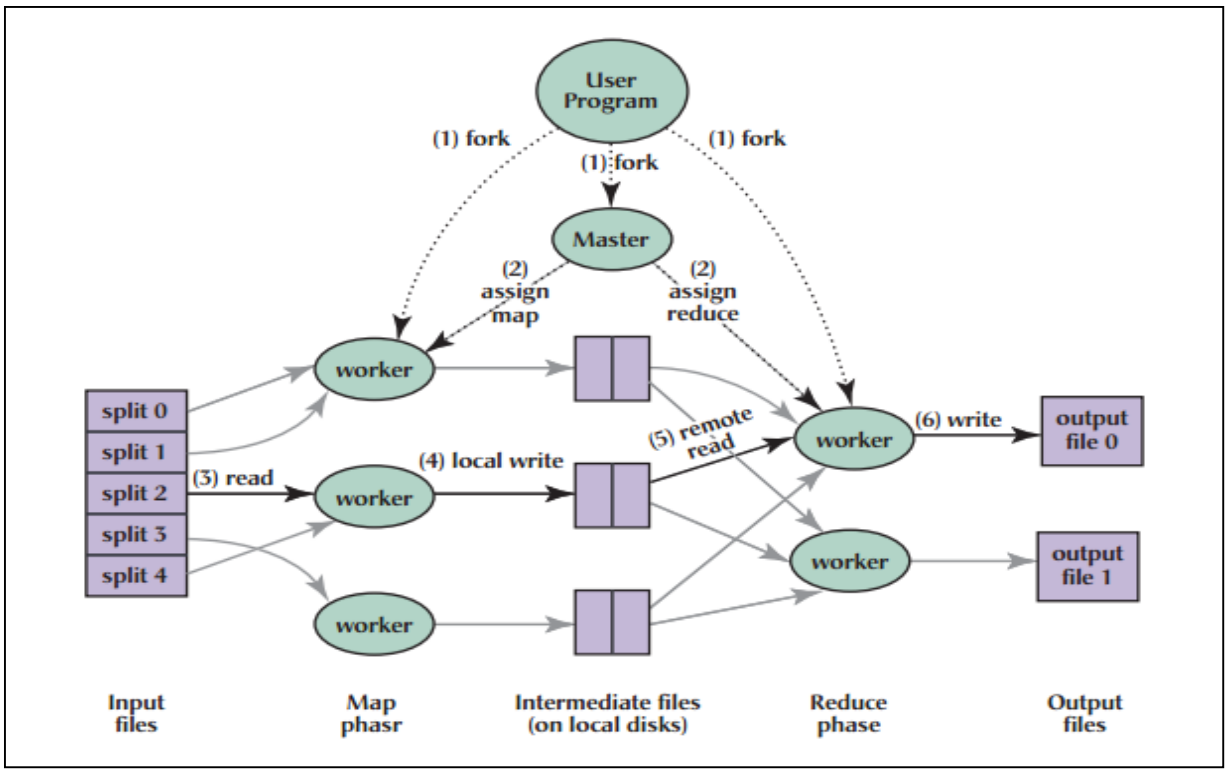


Figura 1 Esboço de Iteração de um Processo Map-Reduce. Fonte: Dean & Ghemawat (2008).

O mestre tem muitas responsabilidades. Uma é criar um certo número de tarefas Map e algumas tarefas Reduce, este número sendo selecionado pelo programa do usuário.

Estas tarefas serão atribuídas para o processo Worker pelo Mestre. É razoável criar uma tarefa Map para cada chunk de arquivo de entrada, mas pode-se desejar criar poucas tarefas Reduce. A razão para limitar o número de tarefas Reduce é que é necessário para cada tarefa Map criar um arquivo intermediário para cada tarefa Reduce, e se existe muitas tarefas Reduce o número de arquivos intermediários aumenta bastante (Kolb, 2013).

O Mestre (Master) se matem informado do estado de cada tarefa Map e Reduce (ocioso, executando um particular worker, ou concluído). Um processo Worker relata para o Mestre quando ele termina uma tarefa, e uma nova tarefa é agendada pelo Mestre para esse processo Worker (Kolb, 2013).

9.7 Conclusões

Concluindo, Map-Reduce é um modelo de programação, e framework introduzido pelo Google para suportar computações paralelas em grandes coleções de dados em clusters de computadores. Agora Map-Reduce é considerado um novo modelo computacional distribuído, inspirado pelas funções map e reduce usadas comumente em programação funcional. Map-Reduce é um "Data-Oriented" que processa dados em duas fases primárias: Map e Reduce. A filosofia por trás do Map-Reduce é: Diferentemente de data-stores centrais, como um banco de dados, você não pode assumir que todos os dados residem em um lugar central portanto você não pode executar uma query e esperar obter os resultados em uma operação síncrona. Em vez disso, você precisa executar a query em cada fonte de dados simultaneamente. O processo de mapear a requisição do originador para o data source é chamado de 'Map', e o processo de agregação do resultado em um resultado consolidado é chamado de 'Reduce'.

Hoje existem diversas implementações de Map-Reduce, como: Hadoop, Disco, Skynet, FileMap e Greenplum. Hadoop é a implementação mais famosa.

A tecnologia de big data não apenas suporta a habilidade de coletar grandes volumes de dados como também provê a habilidade de compreendê-los e tirar proveito de seu valor.

10. Raciocínio Baseado em Casos

Este capítulo apresenta uma introdução dos **Sistemas de Raciocínio Baseados em Casos** (**CBR** – *Case-based Reasoning*). No entanto, segundo (Watson, 2003), os **CBR**s e a Gestão do Conhecimento estão relacionados. Portanto, antes de discutir os sistema de Raciocínio Baseados em Caso, discute-se o que é Gestão do Conhecimento.

10.1 Gestão do Conhecimento

A função da Gestão do Conhecimento, segundo (Watson, 2003), é permitir que as organizações possam alavancar suas informações e conhecimento, através de experiências. Conhecimento, e consequentemente, sua gestão, está sendo alardeado como base da futura competitividade econômica, por exemplo.

De acordo com Watson (2003), o conhecimento, agora, passa a ser vista como um ativo, a criação e o compartilhamento tem se tornado um importante fator dentro e entre as organizações. Embora, muitos autores levantam a questão em relação ao "paradoxo valor" quando considera a natureza do conhecimento, em particular sua intangibilidade e inadequação como um ativo e a dificuldade de avaliar e proteger seu valor.

10.2 Definição de Gestão do Conhecimento

Muitos autores abordam o assunto de diferentes perspectivas. Eles, portanto, têm diferentes definições. A maioria da literatura sobre gestão do conhecimento, tratam o conhecimento de forma ampla, e usa-o para cobrir tudo o que a empresa necessita para realizar suas funções. Isto pode envolver o conhecimento formalizado, patentes, leis, programas, e procedimentos, bem como o mais intangível know-how, habilidades, e experiências das pessoas. Ele também inclui a maneira como as organizações funcionam, comunica-se, analisa situações, desenvolve novas soluções para os problemas, e desenvolve novas formas de fazer negócio. Mais ainda, pode envolver questões culturais, costumes, e valores, bem como os relacionamentos entre fornecedores e clientes (Watson, 2003).

Por outro lado, Gestão inclui todas as maneiras que o conhecimento ativo de uma organização é colhido, armazenado, transmitido, aplicado, atualizado, ou gerado (Watson, 2003). Neste trabalho, centra-se na gestão do conhecimento, por meio da aplicação de uma metodologia para implementar a solução da gestão do conhecimento, nomeadamente, *case-based reasoning* (**CBR**).

Portanto, a definição adotada é a apresentada por Watson (2003), que é "Gestão do Conhecimento envolve a aquisição, armazenamento, recuperação, aplicação, geração, e revisão do conhecimento ativo de uma organização de uma maneira controlada".

10.3 O que é Conhecimento?

De acordo com Watson (2003), o conhecimento não existe isolado. Não é algo que pode ser pego e trancado em um cofre de uma empresa. Na verdade, alguns filósofos acreditam que o conhecimento é uma construção humana que não existe fora da mente das pessoas. Vale a pena considerar o relacionamento entre dado e informação. Os computadores têm por décadas e, sempre manipulando dados (em sistemas de banco de dados).

Dados, informações e conhecimento pode ser considerado, não como entidades discretas, mas como algo continuo, como ilustrado na Figura 2. Eles exibem uma relação com seu contexto e quantidade de entendimento quer é requerido ou impactar (Watson, 2003).

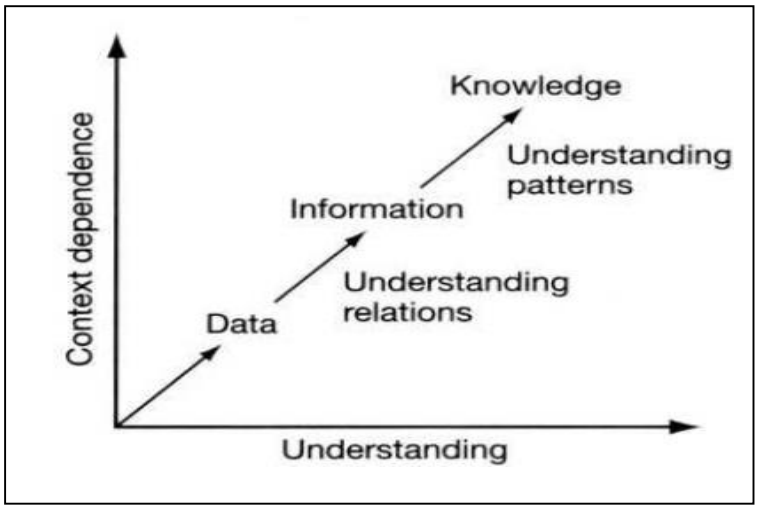


Figura 2 A relação de contexto para a compreensão. Fonte: Appling Knowledge Management: Techniques for Building Corporate Memories (Watson, 2003).

Uma noção importante a se observar na Figura 5.1 é que o conhecimento envolve o reconhecimento ou a compreensão de padrões. Isto envolve a criação de modelos mentais, exemplares, ou arquétipos (Watson, 2003).

10.4 Atividades da Gestão do Conhecimento

De acordo com Watson (2003), o ato de gerenciar conhecimento pode ser caracterizado por quatro atividades:

1. Adquirir conhecimento (aprendendo, criando ou identificando);
2. Analisando conhecimento (avaliar, validar ou valorar);
3. Preservar conhecimento (organizar, representar ou manter); e
4. Usar conhecimento (aplicar, transferir ou compartilhar).

Estas atividades não existem em isolado. Em vez disso, pode-se considera-las como um ciclo, como mostra a Figura 3. Você pode ver esta gestão do conhecimento ciclo (***KM-cycle***) com uma simplificação do mais detalhado Case-based reasoning-cycly (CBR-cycle) que será discutido em sessões seguintes (Watson, 2003). O elemento que liga o ciclo é o uso do conhecimento, uma vez que é provável que quando o conhecimento é usado, uma nova visão sobre o conhecimento pode ser criado. Este novo conhecimento, por sua vez, deve ser adquirido, analisado e preservado para uso futuro (Watson, 2003).

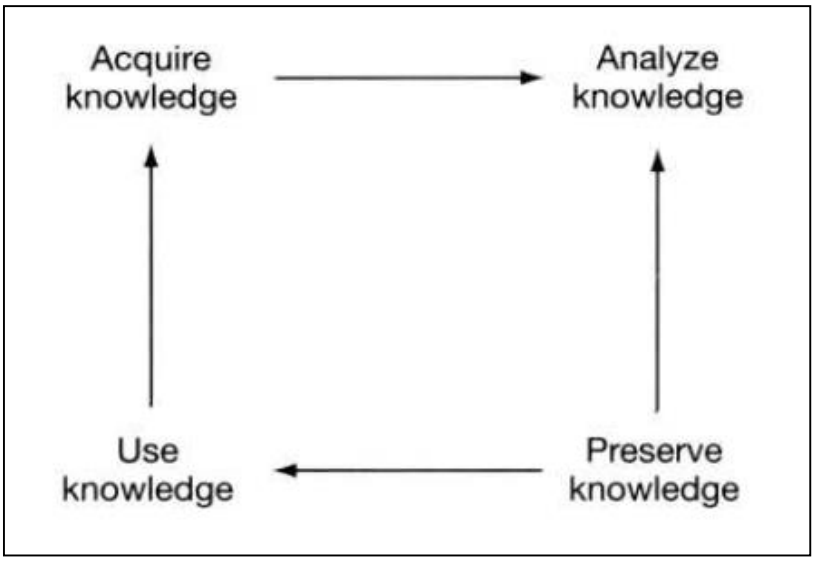


Figura 3 O KM-cycle. **Fonte**: Appling Knowledge Management: Techniques for Building Corporate Memories (Watson, 2003).

10.5 Uma Metodologia para o Conhecimento

Em recente workshop realizado na Universidade de Cambridge, na Inglaterra, um grupo de pessoas ativas gestão do conhecimento e Inteligência Artificial (IA) identificou as principais atividades necessárias por um conhecimento (Watson, 2003). Essa atividades estão ilustradas na Figura 4.

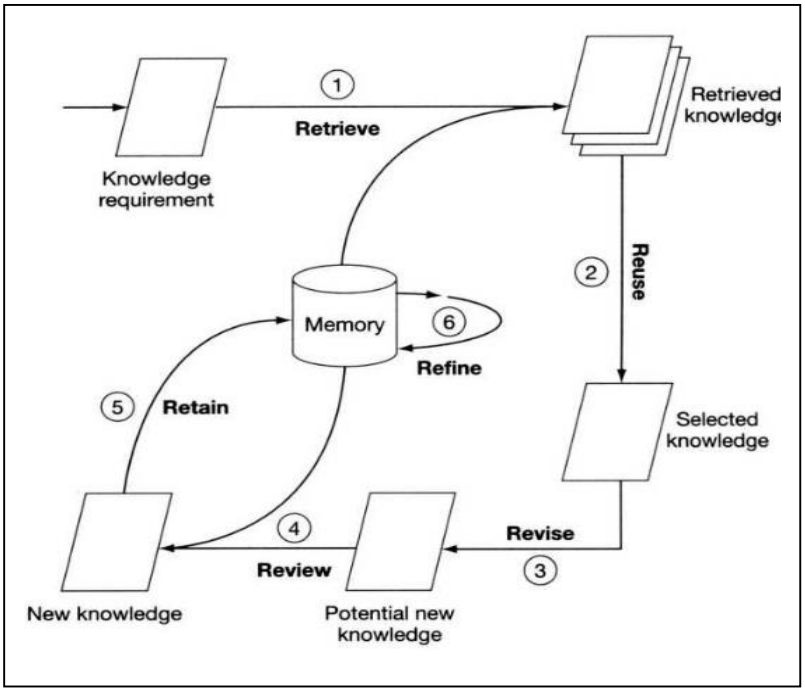


Figura 4 **O funcionamento do Raciocínio Baseado em Casos** (*CBR-cycle*). **Fonte**: Appling Knowledge Management: Techniques for Building Corporate Memories (Watson, 2003).

1. O processo de recuperação, reuso, e revisão suporta a aquisição de conhecimento.
2. O processo de revisão e refinamento suporta a análise do conhecimento.
3. A proporia memória (juntamente com a recuperação e o refinamento) suportam a preservação do conhecimento.
4. Finalmente, a recuperação, reuso, e revisão suportam o uso do conhecimento.

Os pontos chaves aqui discutidos foram (Watson, 2003):

- O conhecimento não é estático; isto é, um sistema de gestão do conhecimento deve poder suportar a aquisição, análise, preservação, e reuso do conhecimento como um processo continuo e cíclico.
- O conhecimento existe em duas formas: conhecimento explícito, que pode ser codificado e o conhecimento tácito, que nem sempre pode ser codificado. Se a representação do conhecimento for muito formalizado, muito conhecimento tácito pode ser perdido. Assim, a representação do conhecimento em sistemas de gestão do conhecimento, deve ser flexível e discursiva.

10.6 Introdução ao Raciocínio Baseado em Casos (em inglês *Case-Based Reasoning* – CBR)

Na sessão anterior foi introduzido o ***CBR-cycle*** e como ele satisfaz os requisitos de um sistema de gestão de conhecimento. Nesta sessão será detalhado cada processo do ***CBR-cycle***.

De acordo com Watson (2003), o **CBR** usa o conceito de similaridade para recuperar coisas (casos) de uma biblioteca (uma base de casos). Casos são usados em muitas situações; por exemplo, para fornecer informações de produtos para um cliente, resolver problemas em uma central de informações ao clientes, configurar equipamentos de manufatura, ou resolver problemas financeiros complexos.

De acordo com Watson (2003), nós resolvemos problemas usando experiências adquiridas e que podemos aprender novas experiências. O Raciocínio baseado em Casos pode ser descrito por seis atividades ocorrendo em ciclo, como discutido na sessão anterior.

Este ciclo é constituído de seis processos:

1. Recuperar
2. Reusar
3. Revisar
4. Avaliar
5. Manter
6. Refinar

10.7 Definição

Para Watson (2003), a ideia básica em um sistema CBR é que, para um domínio particular, os problemas a serem resolvidos tendem a ser recorrentes e repetir-se com pequenas alterações em relação a sua versão original. Dessa forma, soluções anteriores podem ser reutilizadas também com pequenas alterações.

Riesbeck e Schank (1996), definem **CBR** como "Um sistema de CBR resolve problemas por adaptar soluções que foram utilizadas para resolver problemas anteriores".

Em seguida será detalhado cada um desses processos, mas primeiro é necessário entender o que é recuperar, reusar, e revisar, e assim, casos.

10.8 Representação de Casos

De acordo com Watson (2003), casos são registros de experiências que contém conhecimento, que pode ser ambos explicito e tácito. Por exemplo, ele pode ser casos de históricos de pacientes no sentido médico, detalhes de empréstimos bancários, ou descrição de situações de erros de equipamentos. Cada um desses registros de casos compreende:

- Uma descrição
- O respectivo resultado ou solução

Assim, um caso tipicamente compreende um par problema e solução. Uma coleção de casos é chamado de uma base de casos, justamente como uma base de registros é chamado de banco de dados (Watson, 2003).

Uma forma de visualizar é em termos de espeço do problema e espaço de solução. Na Figura 5 vê-se que um caso individual é composto de dois componentes: uma descrição do problema e o armazenamento da solução. Estes residem respectivamente no espaço do problema e no espeço de solução. A descrição do problema a ser resolvido é colocado no espaço de problema. Recupera-se o caso mais similar a descrição do problema, e sua solução, armazenada encontrada. Se necessário, ocorrem adaptações, e uma nova solução é armazenada. Este modelo conceitual de **CBR**, assume que há um mapeamento direto de um-para-um entre o problema e os espaços de solução. (Watson, 1999).

Bases de Casos divide-se em duas grandes categorias (Watson, 2003):

- **Em bases de Casos homogêneas**: todos os casos compartilham os mesmos dados ou estrutura de registros; isto é, casos têm os mesmos atributos mas variando os valores.
- **Em Bases de Casos Heterogêneas**: casos têm estrutura de registros variados; isto é, casos podem ter diferentes atributos e valores variados.
- Um exemplo de caso homogêneo pode ser o caso de venda de casa, onde na base de casos de casas tem os mesmos atributos que são suficientes para descrever uma casa. Então, um corretor, que tenha acesso a essa base de casos, pode assumir que já tenha todas as informações para realizar a transação. Embora, se o corretor ainda não tiver esta propriedade na base de casos, ele pode facilmente criar um registro nessa base de casos.

Um exemplo de base de casos heterogênea poderia ser uma base de casos de diagnósticos de pacientes. Registros de pacientes contêm um lote de informações em comum, tais como idade, tipo sanguíneo, pressão sanguínea, mas também muitas informações que são únicas para cada paciente, por exemplo, histórico médico, tratamento, e prognósticos.

Quando desenvolve-se uma base de casos heterogênea, os desenvolvedores nunca pode assegurar que ele tenha um conjunto completo de características (Watson, 2003). Por exemplo, uma base de dados de diagnósticos de pacientes, os desenvolvedores poderia não listar todas as possíveis condições médicas, sintomas, e testes que uma pessoa poderia ter.

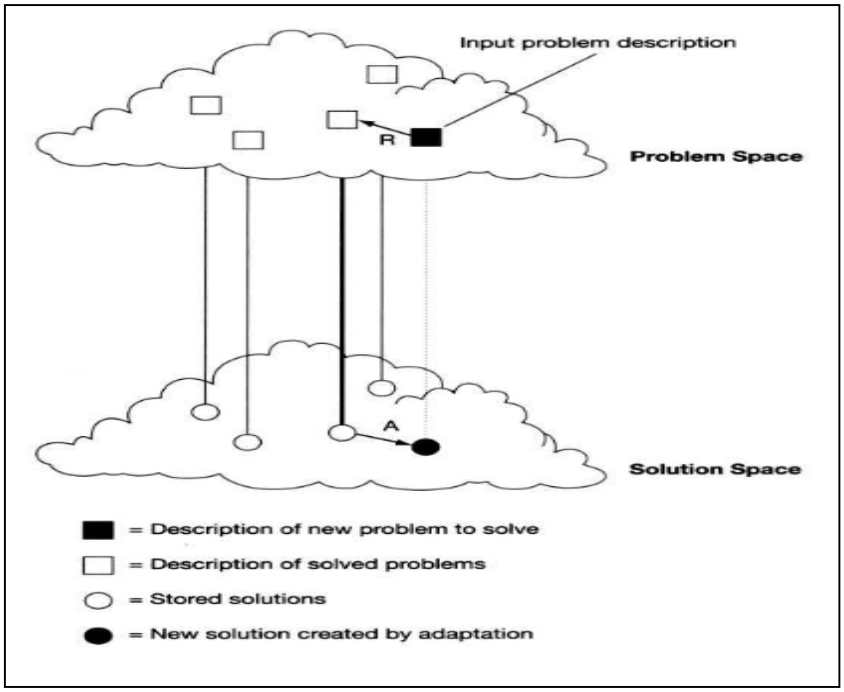


Figura 5 Os espaços de problema e de solução. **Fonte**: Appling Knowledge Management: Techniques for Building Corporate Memories (Watson, 2003).

Dentro de um **Caso** pode-se armazenar muitos tipos de dados, tais como nomes, identificação de produto, valores como custo, temperatura, e notas textuais. Algumas ferramentas de CBR também suportam dados com características de multimídia, tais com imagens, sons e vídeo (Watson, 1999, 2003).

Não há um consenso por parte da comunidade de CBR, que informações exatamente, poderia ser um Caso. Embora, duas medidas pragmáticas poderia ser tomadas para se decidir o que poderia ser representada em Casos: a funcionalidade da informação e a facilidade de aquisição da informação (Watson, 1999).

10.9 Indexação

Muitos sistemas de Banco de Dados utilizam-se de índices para agilizar a recuperação de dados. Um índice é computacionalmente, uma estrutura de dados que pode ser realizada em memória, tornando a localização da informação, muito rápida, sem ter que fazer a busca do(s) registro(s) no disco. O CBR também faz uso de índice para agilizar a recuperação de Casos. A informação dentro de um Caso, é de dois tipos (Watson, 1999):

1 Informação Indexadas, que á usada para recuperar um Caso.
2 Informação não indexada, que fornecem informações contextuais para o usuário, mas que não são usadas diretamente para recuperação de Caso.

Por exemplo, em um sistema médico, pode-se usar as informações do paciente, tais como idade, sexo, altura, e peso como características a serem indexadas, que pode ser usada para recuperação do Caso e, outras informações, tais como nome, endereço e fotografia como informações contextuais, ou seja, não indexadas, que não podem ser usadas para recuperação de Casos. A Figura 6 ilustra este exemplo.

Como diretrizes, os índices devem:

- Ser preditivo.
- Indicar o propósito em que o Caso será usado.
- Ser abstrato o suficiente para permitir ampliar a base de casos e seu uso no futuro.
- Ser concreto o suficiente para ser reconhecido em futuras situações.

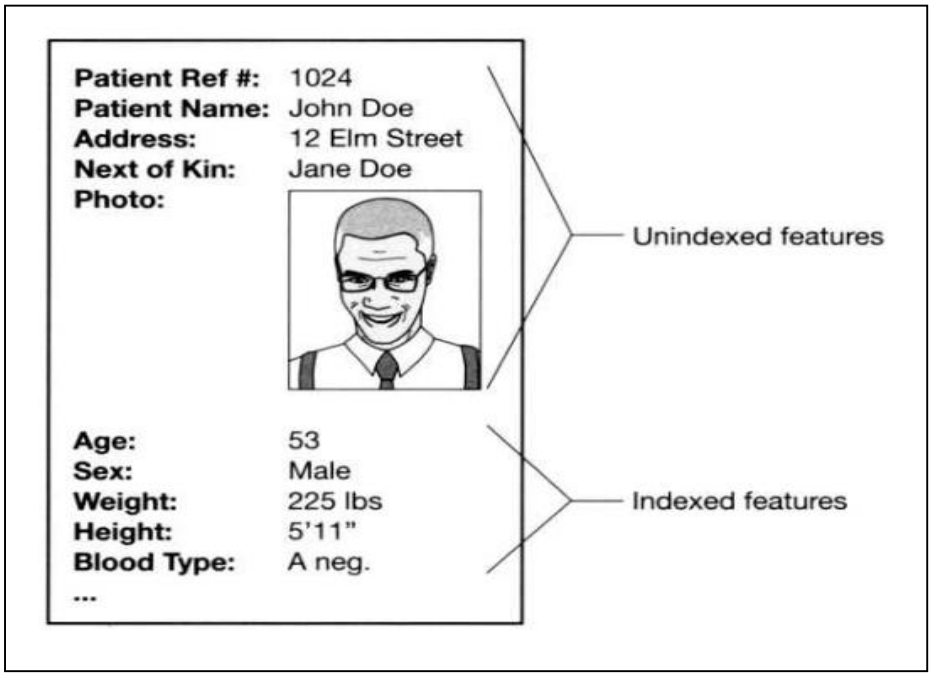


Figura 6 Informações indexadas e não-indexadas. **Fonte**: Appling Knowledge Management: Techniques for Building Corporate Memories (Watson, 2003).

Se tomarmos como exemplo um sistema bancário. Informações dos clientes, tais como nome e telefone, são claramente não preditivas, você não decidir emprestar dinheiro a um cliente como base em seu nome e telefone. No entanto, informações tais como renda, e seus compromissos financeiros, tais como empréstimos habitacionais, pagamentos de carros, e seguro de vida, e assim por diante, são claramente preditivos. Dessa forma,

informações como renda e compromissos financeiros podem ser escolhidos como índices e nome e telefone como informações contextuais (Watson, 1999).

A escolha do índice, tanto pode ser manual com automatizada. Escolher um índice manualmente, envolve decidir o propósito do Caso em respeito ao objetivo do sistema e decidir em que circunstancias o Caso vai ser útil (Watson, 1999).

Existem um número crescente de métodos de indexação automática na literatura, incluindo: MEDIATOR, CHEF e CYRUS, etc.

Várias ferramentas de CBR presentes no mercado suportam a identificação de índices de casos automaticamente, para aplicações práticas, índices pode ser escolhido automaticamente, manualmente ou ambas técnicas (Watson, 1999, 2003).

10.10 Aquisição (Storage)

A representação do Caso é um importante aspecto no projeto de sistemas CBR, no que se refere a visão conceitual do que é representado no Caso e levando-se em conta os índices que caracterizam os casos. A base de Casos poderia ser organizada em uma estrutura gerenciável que suporte pesquisas e métodos de recuperação eficientes. Deve-se ser encontrado um equilíbrio entre os métodos de armazenamento que preserve a riqueza de casos e seus índices e métodos que simplifique o acesso e recuperação de casos relevantes (Watson, 1999). Este métodos são usualmente chamados de modelos de memória de casos (*case-memory models*). Os dois modelos de memória de casos mais influentes na academia são o modelo de memória dinâmica de Schank e Kolodner, e o modelo categoria exemplares de Porter e Bareiss. Estas técnicas ainda são bastantes utilizadas pela comunidade de Ciência Cognitiva, mas nenhuma ferramenta comercial de CBR usam essas técnicas. Ao invés disso, muitas base de casos utilizam estruturas simples de arquivos planos (flat files), ou estruturas de bancos de dados relacionais e, usa índices para se referir aos casos (Watson, 1999).

10.11 Recuperação

Dada uma descrição de um problema, um algoritmo de recuperação deveria encontrar os casos mais similares à situação atual, utilizando-se dos índices da memória de casos. Os algoritmo baseiam-se nos índices e na organização de memória para guiar a busca dos casos potencialmente úteis.

De acordo com Watson (2003), a recuperação de casos está diretamente relacionado e dependente ao método de indexação usado. Em geral, duas técnicas são correntemente usadas pelas ferramentas de **CBR** comerciais: algoritmo de vizinhança (***Nearest-Neighbor***) e Indutivo.

10.12 Algoritmo de Vizinhança

Esse método, segundo Watson (2003), baseia-se na comparação entre um novo caso e aqueles armazenados na base de casos, utilizando uma soma ponderada das suas características. Para isso é necessário atribuir um peso a cada uma das características que descrevem o caso e que serão utilizadas na recuperação.

Na prática, a similaridade (isto é, a proximidade) do caso destino para o caso fonte para cada atributo é determinado. Esta medida é multiplicado por um fator peso. Então a soma da similaridade de todos os atributos é calculada. Esta pode ser representada por uma equação relativamente simples

(8.1)

$$\boldsymbol{Similarity(T,S) = \sum_{i=1}^{n} f(T_i, S_i)\, x\, w_i}$$

Onde

T é o caso destino

S é o caso fonte

n é o número de atributos em cada caso

i é um atributo individual de 1 até n

$\boldsymbol{f}$ é a função de similaridade para atributo i nos casos T e S

w é o peso do atributo i

Algoritmos de similares a este são usados por muitas ferramentas CBR para realizar recuperação do caso mais similar. Similaridade são normalmente para cair dentro da faixa de 0 para 1 (onde o significa totalmente dissimilar e 1 exatamente similar) ou usando um percentual, onde 100% é totalmente similar (Watson, 1999; 2003).

10.13 Algoritmo de Indução

Indução é uma técnica desenvolvida por pesquisadores de Aprendizado de Máquinas para extrair regras ou construir de dados passados. Em sistema CBR, a base de casos é analisada por algoritmo de indução para produzir uma árvore de decisão que classifica (ou indexa) os casos. O algoritmo de indução foi amplamente usado pela ferramenta CBR chamada ID3.

10.14 Adaptação

A tarefa final do Sistema CBR é adaptar a solução associada a um caso recuperado para as necessidades do problema corrente. Quando uma situação é fornecida, o algoritmo de recuperação traz o melhor caso que ele encontrar para a memória. Normalmente, o

caso selecionado não atende perfeitamente coma descrição do problema do usuário. Ou seja, existem diferenças entre o problema do usuário e o caso contido no banco de casos que devem ser levadas em conta. Então, o processo de adaptação procura por diferenças salientes entre as duas descrições e aplica regras de forma a compensá-las. Em geral, existem dois tipos de adaptação em CBR (Watson, 2003):

- **Adaptação Estrutural -** as regras de adaptação são aplicadas sobre a solução armazenada junto aos casos.
- **Adaptação Derivacional** – o algoritmo reusa os algoritmos, métodos ou regras que geraram a solução que consta no banco de casos para gerar uma nova solução para o problema corrente. Neste método, a sequência que construiu a solução original deve ser armazenada juntamente com o caso na memória de casos. O algoritmo de adaptação derivacional exige uma perfeita compreensão dos casos armazenados e da forma como as soluções foram geradas.

Segundo Watson (2003), várias técnicas tem sido usadas em sistema CBR. Incluindo as seguintes:

- **Adaptação nula** – ele simplesmente aplica a solução recuperada ao problema corrente sem modificação. Adaptação nula é útil para problemas envolvendo raciocínio complexo mais com solução simples. Por exemplo, em um sistema para concessão de crédito, embora seja necessário coletar muitas informações do cliente, a solução final de conceder ou rejeitar o crédito é direta.
- **Ajuste por parâmetros** – é uma técnica de adaptação estrutural que compara parâmetros específicos entre o caso recuperado e o novo para modificar a solução armazenada na direção apropriada. Esta técnica foi usada no sistema CBR chamado JUDGE, que recomenda sentenças mais curtas para crimes menos violentos.
- **Reinstanciação** – instancia uma nova solução para um caso recuperado do banco de casos com novas características adequadas ao problema do usuário. Por exemplo, o sistema CBR CHEF, que a partir de uma receita existente, criar uma nova receita.
- **Substituição derivacional** – repete o método, ou parte do método que gerou uma solução armazenada em um caso similar de forma a obter a solução para o novo caso, substituindo os atributos distintos. Como no sistema BOGART que reaplica os planos de geração de projetos para novos problemas.
- **Repara guiado por modelos** – utiliza um modelo casual para adaptar as soluções armazenadas ao problemas do usuário. O sistema CBR, CELIA, utiliza-o para aprendizado e diagnóstico de problemas mecânicos de automóveis.

Finalizando, de acordo com Watson (2003), a adaptação é útil em muitas situações. Mais não significa que seja essencial. Muitos dos sistema CBR comerciais não usam adaptação para tudo. Eles simplesmente reusam a solução sugerida para o melhor caso correspondente (i.e., adaptação nula) ou eles deixam a adaptação para as pessoas.

10.15 Conclusões

Pode-se concluir que, O **Raciocínio Baseado em Casos** é um método em que problemas novos são resolvidos através de soluções adaptadas que foram usadas para resolver problemas mais antigos.

Um **Caso** é uma pedaço contextualizado de conhecimento que representa uma experiência. Ao se analisar cada caso, se tem a descrição do problema e a solução armazenada. Caso já exista um problema semelhante já anteriormente armazenado no banco de dados, a solução será recuperada. Porém, se não existir um caso similar, a descrição desse novo problema será enviado ao espaço de problemas, recuperando o caso com o problema mais similar possível, criando uma nova solução (Watson,1997).

Devido a essas características, o **RBC** é indicado para tarefas de segmentação e categorização.

A recuperação de casos é profundamente relacionada e dependente do método de categorização utilizado. As duas técnicas utilizadas atualmente são a de **recuperação por vizinho mais próximo** e **recuperação indutiva**. Na recuperação por vizinho mais próximo, deverão ser definidos os índices dos casos e os seus respectivos pesos, dependendo do problema a ser resolvido. Esses índices deverão ser previsíveis, identificar o propósito em que os casos serão utilizados, serem abstratos o bastante para permitir o uso posterior da base de dados, e serem concretos o bastante para serem reconhecidos no futuro (Watson, 1997). Como explicitado anteriormente, o caso similar será utilizado como solução do problema. No caso de recuperação por indução, uma árvore de decisão é utilizada, classificando os casos.

11. Redes Reurais Artificiais

As Redes Neurais Artificiais (RNA), também conhecidas como métodos conexionistas, são modelos matemáticos que se assemelham às estruturas biológicas e que têm capacidade computacional adquirida por meio de aprendizado e generalização (Haykin, 2001; Negnevitsky, 2005).

Inicialmente será discutido nesta sessão a representação de conhecimento utilizada pelas Redes Neurais Artificiais, para depois tentarmos analisar a parte referente ao aprendizado. A princípio, é importante salientar que existem diferentes tipos de Redes Neurais Artificiais (**RNA**s) e que cada uma delas tem características próprias em relação a sua representação e aquisição de conhecimento (ou aprendizado) (Osório, 1999).

11.1 Definição

Segundo Negnevitsky (2005), uma Rede Neural pode ser definida como um modelo de raciocínio baseado no cérebro humano. O cérebro é composto por um conjunto densamente interligado de células nervosas, ou unidades básicas de processamento, chamada de neurônios. O cérebro humano incorpora cerca de 10 bilhões de **neurônios** e 60 trilhões de conexões, **sinapses**, entre eles (Shepherd e Koch, 1990; citado por Negnevitsky, 2005). Por usar múltiplos neurônios simultaneamente, o cérebro pode realizar suas funções muito mais rapidamente do que o computador mais rápido existente hoje.

Embora cada neurônio tenha uma estrutura muito simples, um exército desses elementos possui um tremendo poder de processamento. Um neurônio consiste de um corpo celular, **soma**, um número de fibras chamadas de **dendritos**, e uma fibra longa chamada de **axônio**. Enquanto os dendritos formam uma rede em torno do corpo celular (soma), o axônio se estende para os dendritos e soma de outros neurônios (Negnevistsky, 2005).

Os sinais são propagados de um neurônio para outro através de complexas reações eletroquímicas. As substancias químicas liberadas das sinapses causam mudanças no potencial elétrico do corpo celular. Quando o potencial atinge seu limite, um pulso elétrico, é enviado através do axônio. O pulso se espalha e eventualmente atinge as sinapses, causando o aumento ou diminuição de seu potencial. No entanto, a descoberta mais interessante é que a rede neural exibe plasticidade (Negnevistky, 2005).

Portanto, de acordo com Negnevitsky (2005), pode-se definir as redes conexionistas ou Rede Neuronal Artificial (RNAs), como sendo formadas por um conjunto de unidades elementares de processamento de informações fortemente conectadas, que denomina-se neurônio artificial. Uma RNA é constituída por um grafo orientado e ponderado. Os nós do grafo são autômatos simples, os chamados neurônios artificiais, que formam através de suas conexões um autômato mais complexo, a rede neural. A Figura 7 ilustra uma rede neural artificial.

Segundo Negnevitsky (2005), cada unidade da rede é dotada de um estado interno, que geralmente é denominado de estado de ativação. As unidades podem propagar seu estado de ativação para outras unidades do grafo, passando pelos arcos ponderados, que é chamado de conexões, ligações sinápticas, ou simplesmente de pesos sinápticos. A regra que determina a ativação de um neurônio em função da influência transmitidas de suas entradas, ponderadas pelos seus respectivos pesos, se chama regra de ativação ou função de ativação. E a Tabela 1 faz uma analogia entre uma rede biológica e uma rede conexionista.

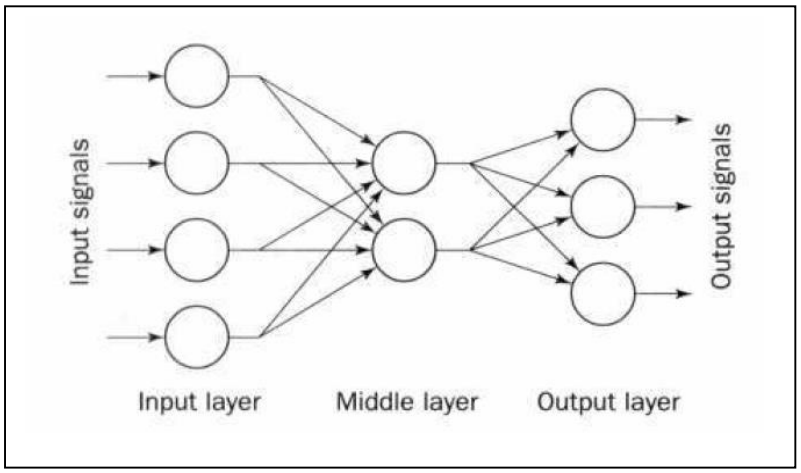


Figura 7 Arquitetura de uma típica Rede Neural Artificial. **Fonte**: Artificial intelligence: a guide to intelligence systems/Michael Negnevitsky, 2005).

Tabela 1 Analogia entre Rede Neural biológica e Artificial

Rede neural biológica	Rede neural artificial
Soma	Neuron
Dendrite	Input
Axon	Output
Synapse	weight

O processamento da informação em RNAs é realizado por meio de estruturas neurais artificiais em que o armazenamento e o processamento da informação são realizados de maneira paralela e distribuída por elementos processadores relativamente simples. Cada elemento processador corresponde a um neurônio artificial (Negnevistsky, 2005).

Segundo Negnevistsky (2005) para se construir uma RNA, deve-se decidir primeiro como os neurônios serão usados e como eles serão conectados na rede. Em outras palavras, deve-se escolher a arquitetura da rede. Então, decide-se que algoritmo de

aprendizado usar. E finalmente, treina-se a rede, isto é, inicializa-se os pesos da rede e atualiza-se os pesos para o conjunto de treinamento.

11.2 O Neurônio

Um neurônio recebe vários sinais através de seus links de entrada (input), calcula um novo nível de ativação, e envia-o como sinal de saída através dos links de saída. O sinal de entrada pode ser uma linha de dados ou a saída de outro neurônio. O sinal de saída pode ser a solução para o problema ou uma entrada para outro neurônio. A Figura 8 mostra um típico neurônio.

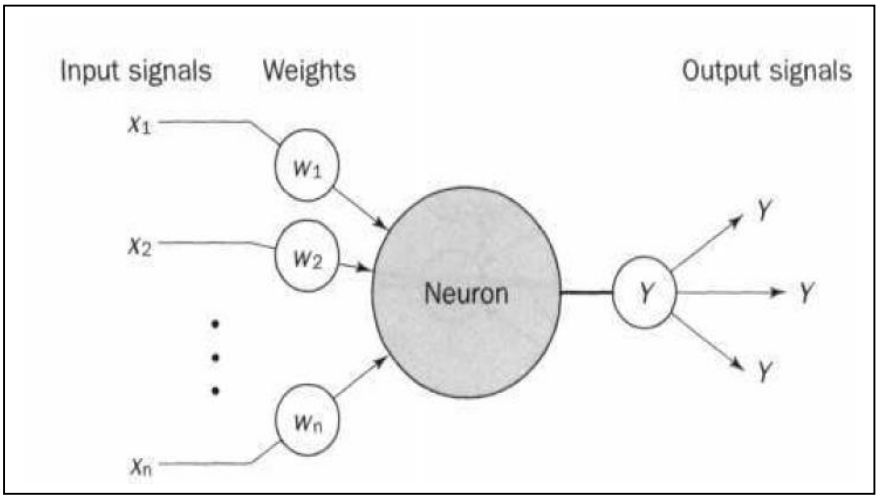


Figura 8 Diagrama de um Neurônio. **Fonte**: Artificial intelligence: a guide to intelligence systems/Michael Negnevitsky, 2005).

As mudanças realizadas nos valores dos pesos sinápticos ou na estrutura de interconexão das unidades de uma rede, são responsáveis pelas alterações no comportamento de ativação desta rede. Estas alterações nas conexões e na estrutura da rede é o que nos permite realizar o aprendizado de um novo comportamento. Desta maneira, pode-se modificar o estado de ativação da saída da rede em resposta a uma certa configuração de entradas. Dessa forma, a rede é capaz de estabelecer associações de entrada-saída (estímulos e respostas) a fim de se adaptar a uma situação proposta. O método utilizado para modificar o comportamento de uma rede é denominado de *regra de aprendizado* (Osório, 1999).

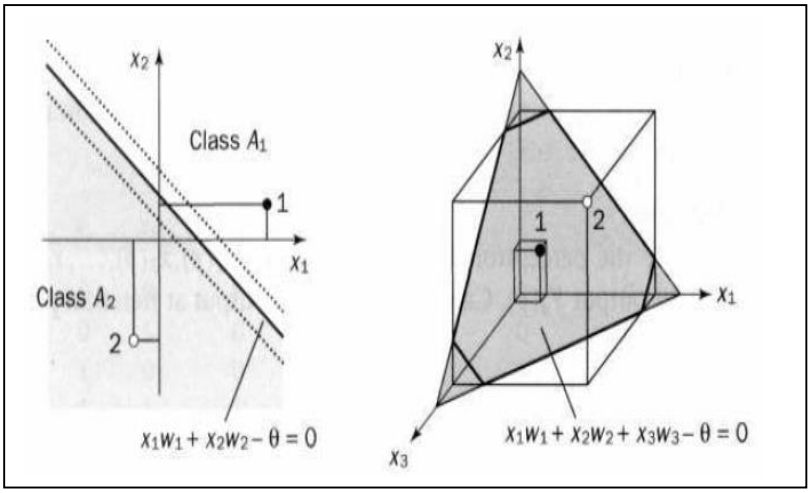
O primeiro modelo matemático do neurônio foi o modelo de proposto por McCulloch e Pitts, em 1943. Mais tarde Rosenblatt (1957) criou o modelo do perceptron. O modelo consiste de um modelador linear seguido de limitador. Um perceptron modela um neurônio tomando uma soma ponderada de suas entradas e compara essa soma com o limiar, o que produz uma saída +1 se suas entradas são positivas e -1 se elas são negativas. O objetivo do *perceptron* é classificar as entradas, ou em outras palavras,

aplicar externamente x_1, x_2, ..., x_n estímulos entre umas das classes, ditas A_1 e A_2. Assim, no caso de um *perceptron* elementar, o espaço n-dimensional é dividido por um hiperplano em duas regiões. O hiperplano é definido por função linearmente separáveis.

(9.1)

$$\sum_{i=1}^{n} X_i W_i - \theta = 0$$

Para o caso de duas entradas x_1 e x_2, o limite de decisão leva a forma de uma linha reta como ilustrada na Figura 9(a). O ponto, o qual se encontra acima da linha limite, pertence a classe A_1; e o ponto 2, o qual se encontra abaixo da linha limite, pertence à classe A_2.



(a) (b)

Figura 9 Separabilidade linear no perceptrons: (a) duas entradas; (b) três entradas. **Fonte**: Artificial intelligence: a guide to intelligence systems/Michael Negnevitsky, 2005).

O neurônio de McCulloch e Pitts, usa a seguinte transferência ou função de ativação:

(9.2)

$$X = \sum_{i=1}^{n} X_i W_i$$

$$Y = \begin{cases} +1 \; Se \; X \geq \theta \\ -1 \; Se \; X < \theta \end{cases}$$

Onde X é a soma ponderada das entradas do neurônio, e x_i é o valor da entrada **i**, w_i é o peso da entrada **i**, **n** é o número de neurônios de entrada e **y** é o número de neurônios de saída.

Este tipo de função de ativação é chamada de função sinal (*sign function*).

Assim, a saída do neurônio para uma função sinal, pode ser representado como

(9.3)

$$Y = sign \left[\sum_{i=1}^{n} X_i W_i - \theta \right]$$

Segundo Negnevistsky (2005) muitas funções de ativação têm sido testadas, mais somente umas poucas tem tido aplicações práticas. Quatro delas são a função, step, sign, linear e sigmoide.

- As funções de ativação step e sign – também chamadas de ***hard limit functions***, são frequentemente usadas em neurônios para tomada de decisões para tarefas classificação e reconhecimento de padrões.
- A função sigmoide – transforma as entradas, que pode ter qualquer valor entre mais ou menos infinito, em um valor razoável na faixa de 0 e 1.
- A função linear – neurônios com função de ativação linear são frequentemente usadas para aproximações lineares.

11.3 Classificação e Propriedades

A grande variedade de modelos existentes nos leva a um estudo ou análise das principais propriedades das redes neurais, que nos permita compreender melhor as vantagens e/ou inconveniências da escolha de um modelo em detrimento de outro. Considere-se que para essa análise sejam avaliados um grupo de atributos tais como: tipo de aprendizado, arquitetura de interconexões, forma interna de representação das informações, tipo de aplicação da rede, etc (Rosa, 1999).

11.4 Aprendizado RNA

Uma RNA aprende geralmente de forma gradual, onde os pesos são modificados várias vezes, paulatinamente, seguindo-se uma regra de aprendizado que estabelece a forma como estes pesos são alterados. Utiliza-se no aprendizado um conjunto de dados de aprendizado disponível (base de exemplos). Cada iteração deste processo gradativo de adaptação dos pesos de uma RNA, é chamada de época de aprendizado. Os métodos de aprendizado neural podem ser divididos em três grandes classe, segundo a grau de controle dado ao usuário (Rosa, 1999; Rezende, 2003):

- **Aprendizado supervisionado**: o usuário dispõe de um comportamento de referência preciso que ele deseja ensinar a rede. Sendo assim, a rede deve ser capaz de medir a diferença entre seu comportamento atual e o comportamento de referência, e então corrigir os pesos de maneira a reduzir este erro (desvio de comportamento em relação aos exemplos de referência).
- **Aprendizado semi-supervisionado**: o usuário possui apenas indicações imprecisas (por exemplo: sucesso/insucesso da rede) sobre o comportamento final desejado. As técnicas de aprendizado semi-supervisionado são chamadas também de aprendizado por reforço (reinforcement learning) [Sutton 98].
- **Aprendizado não-supervisionado**: os pesos da rede são modificados em função de critérios internos, tais como, por exemplo, a repetição de padrões de ativação em paralelo de vários neurônios. O comportamento resultante deste tipo de aprendizado é usualmente comparado com técnicas de análise de dados empregadas na estatística (*clustering*).
- **Aprendizado instantâneo**: o conjunto de dados de aprendizado é analisado uma única vez e com isto o conjunto de pesos da rede é determinado de maneira imediata em uma única passagem da base de exemplos. Este modo de aprendizado também é conhecido como: *one single epoch learning* / *one shot learning*.
- **Aprendizado por pacotes:** o conjunto de dados de aprendizado é apresentado à rede várias vezes, de modo que possamos otimizar a resposta da rede, reduzindo os erros da rede e minimizando o erro obtido na saída desta. Este modo de aprendizado é caracterizado por trabalhar com uma alteração dos pesos para cada época, ou seja, para cada passagem completa de todos os exemplos base de aprendizado. O algoritmo de aprendizado deve reduzir pouco à pouco o erro de saída, o que é feito ao final de cada passagem (análise) da base de exemplos de aprendizado.
- **Aprendizado continuo:** o algoritmo de aprendizado leva em consideração continuamente os exemplos que lhe são repassados. Se o conjunto de dados é bem delimitado, chamamos este método de aprendizado ***on-line***, e caso o conjunto de dados possa ir aumentando (sendo adicionados novos exemplos no decorrer do tempo), então chamamos este método de aprendizado **incrementa**l. O aprendizado on-line se opõe ao aprendizado por pacotes, pois ao contrário deste, para cada novo exemplo analisado já se realiza uma

adaptação dos pesos da rede, com o objetivo de convergir na direção da solução do problema. O principal problema do aprendizado contínuo é a dificuldade de achar um bom compromisso entre a plasticidade e a estabilidade da rede. Uma rede com uma grande facilidade de adaptação pode "esquecer" rapidamente os conhecimentos anteriormente adquiridos e uma rede com uma grande estabilidade pode ser incapaz de incorporar novos conhecimentos.

- **Aprendizado ativo:** este modo de aprendizado assume que o algoritmo de adaptação da rede pode passar de uma posição passiva (apenas recebendo os dados do jeito como lhe são passados), para uma posição ativa. Sendo assim, assumimos que este algoritmo poderá vir a intervir sobre a forma como os dados lhe são repassados. Neste caso, a rede pode intervir e determinar assim quais dados que serão considerados e/ou desconsiderados, além também de determinar a ordem em que estes dados deverão ser considerados. A rede pode também vir a solicitar novos dados que julgue necessários para o bom aprendizado do problema proposto.

11.5 Tipos de Unidades (Nós)

Segundo Rosa (1999), as unidades de uma rede – os neurônios artificiais – podem ser de diferentes tipos, de acordo com a função interna utilizada para calcular o seu estado de ativação, ou seja, qual a função de ativação utilizada linear, gaussiana, sigmoide, assimétrica, etc. Um outro elemento que pode diferenciar uma unidade, diz respeito a forma como os neurônio armazenam as informações: unidades baseadas em protótipos, unidades do tipo Perceptron.

- **Redes à base de protótipos**: este tipo de rede utiliza neurônios que servem para representar protótipos dos exemplos aprendidos – as unidades tem uma representação interna que agrupa as características comuns e típicas de um grupo de exemplos (Orsier 95). As redes baseadas em protótipos tem normalmente um aprendizado não supervisionado (com um ou mais protótipos associados à cada classe). Uma das vantagens deste tipo de redes é a possibilidade de fazer um aprendizado contínuo e incremental, uma vez que não é muito difícil de conceber um algoritmo capaz de aumentar a rede neural através da adição de novos protótipos. Os protótipos são também denominados de clusters.
- **Redes à base de Perceptrons**: as unidades do tipo "*Perceptron*" foram criadas por Frank Rosenblatt em 1950. Este é um dos modelos de neurônios mais utilizados na atualidade. Ele é a base de diversos tipos de RNA com aprendizado supervisionado utilizando uma adaptação por correção de erros (usualmente baseada na descida da superfície de erro usando o gradiente). O modelo do Perceptron de múltiplas camadas (**MLP** – *Multi-Layer Perceptron*) tornou-se muito conhecido e aplicado, sendo na maior parte das vezes

associado a regra de aprendizado do *Back-Propagation* (Jodoin 94, Widrow 90, Rumelhart 86) (Negnevitsky, 2005).

11.6 Tipos de Arquiteturas de Conexões de Redes

As unidades de uma rede neural podem se conectar de diferentes modos, resultando em diferentes arquiteturas de interconexão de neurônios. A Figura 6.7 apresenta alguns exemplos de possíveis maneiras de conectar os componentes de uma RNA. As arquiteturas mais importantes são (Osório, 1999):

- **Redes com uma única camada**: as unidades estão todas em um mesmo nível. Neste tipo de arquitetura, as unidades são conectadas diretamente às entradas externas e estas unidades servem também de saídas finais da rede. As redes de uma única camada possuem normalmente conexões laterais (entre os neurônios de uma mesma camada). Um exemplo deste tipo de arquitetura de redes são as redes do tipo "*Self-Organizing Feature Maps*" (Kohonen,1987).
- **Redes com camadas unidirecional**: as unidades são organizadas em vários níveis bem definidos, que são chamados de camadas ou *layers*. Cada unidade de uma camada recebe suas entradas vindas à partir de uma camada precedente, e envia seus sinais de saídas em direção a camada seguinte. Estas redes são conhecidas como redes *feed-forward*.
- **Redes recorrentes**: as redes recorrentes podem ter uma ou mais camadas, mas a sua particularidade reside no fato de que temos conexões que partem da saída de uma unidade em direção a uma outra unidade da mesma camada ou de uma camada anterior à esta. Este tipo de conexões permitem a criação de modelos que levam em consideração aspectos temporais e comportamentos dinâmicos, onde a saída de uma unidade depende de seu estado em um tempo anterior.
- **Redes de ordem superior:** as unidades deste tipo de rede permitem a conexão direta entre duas ou mais de suas entradas, antes mesmo de aplicar a função de cálculo da ativação da unidade (Fiesler, 1994a). Este tipo de rede serve para modelar "sinapses de modulação", ou seja, quando uma entrada pode modular (agir sobre) o sinal que vem de uma outra entrada. Um modelo particular de rede de ordem superior são as redes tipo *Sigma-Pi* que foram apresentadas no livro PDP – *Parallel Distributed Processing* (Rumelhart 86).

A arquitetura de uma rede também pode ser classificada de acordo com a evolução desta no decorrer de sua utilização e desenvolvimento do aprendizado. Em relação a este critério pode-se ter os seguintes tipos (Osório, 99):

- **Redes com estruturas estáticas**: a rede tem a sua estrutura definida antes do início do aprendizado. A quantidade de neurônios, assim como a sua estrutura de interconexões, não sofrem alterações durante a adaptação da rede. As únicas mudanças se realizam à nível dos pesos sinápticos, que são modificados durante o processo de aprendizado (Osório, 98).

- **Redes com estruturas dinâmicas**: as redes que possuem uma estrutura dinâmica são redes onde o número de unidades e conexões pode variar no decorrer do tempo. Estas redes são também chamadas de *ontogênicas* (Fiesler, 1994). As modificações na estrutura da rede podem ser do tipo generativo (incremental) ou do tipo destrutivo (redutor por eliminação/simplificação). A escolha entre estes dois tipos de métodos é bastante polêmica: devemos começar com uma rede pequena e ir aumentando ela, ou devemos começar com uma rede bastante grande e ir reduzindo o seu tamanho posteriormente? Alguns autores defendem a ideia de uma criação construtiva de conhecimentos (Elman 1993, Osório 1999).

11.7 Tipos de Aplicações para Redes Neurais

De acordo com diversos autores, as RNAs podem ser aplicadas a diversos tipos de tarefas, tais como: o reconhecimento de padrões (e.g. reconhecimento de faces humanas), a classificação (e.g. reconhecimento de caracteres – **OCR**), a transformação de dados (e.g. compressão de informações), a predição (e.g. previsão de séries temporais, como as cotações da bolsa de valores, ou o uso para diagnósticos médicos), o controle de processos e a aproximações de funções (e.g. aplicações para área de robótica). Todas essas tarefas podem ser agrupadas em dois grandes grupos (Osório, 1999): **Redes para aproximações de funções**, **Redes para classificação de padrões**.

11.8 Vantagens das RNAs

De acordo com Osório (1999), as redes conexionistas, em particular aquelas comumente aplicadas na construção de sistemas inteligentes, apresentam as seguintes vantagens:

- **Conhecimento empírico**: em geral as redes aprendem mais fácil do que outros métodos de aquisição de conhecimento, pois o aprendizado acontece a partir de exemplos de maneira simples e permite um aquisição de conhecimento de forma automática.
- **Degradação progressiva**: apesar das redes serem menos sensíveis as perturbações, do que os sistemas simbólicos. As respostas dadas por uma rede se degrada progressivamente na presença de perturbações e distorções dos dados de entrada.
- **Manipulação de dados quantitativos:** as redes trabalham com a representação numérica dos conhecimentos e, isso implica que as redes são melhor adaptadas para a manipulação de dados quantitativos (valores contínuos). Isso pode ser considerado uma vantagem, uma vez que grande parte dos problemas do mundo real, manipulam valores contínuos.
- **Paralelismos em larga escala:** as redes neurais são compostas de um conjunto de unidade de processamento de informações que podem trabalhar

em paralelo. Apesar da maioria das implementações de RNAs serem feitas através de simulações em máquinas sequenciais, é possível de se implementar (softwares e hardwares) que possam explorar esta possibilidade de ativação simultânea das unidades de uma rede. A maior parte das implementações de redes neurais simuladas em máquinas sequenciais pode ser facilmente adaptada em uma versão paralela deste sistema.

11.9 Inconvenientes das RNAs

As redes apresentam alguns inconvenientes, do mesmo modo que outros tipos de métodos de aprendizado. As redes apresentam os seguintes inconvenientes (Osório, 1999):

- **Arquitetura e parâmetros**: a evolução do processo de aprendizado é bastante influenciado por estes dois parâmetros. Como não existe métodos totalmente automatizados para escolha correta da arquitetura para um problema, fica muito difícil de se encontrar uma boa topologia para a rede, bem como, bons parâmetros de regulagem para o algoritmo de aprendizado. O sucesso da rede depende bastante desses dois elementos, que variam muito de um problema para outro.
- **Inicialização e codificação**: uma má escolha dos pesos iniciais da rede, do método de codificação dos dados de entrada, ou mesmo da ordem de apresentação destes, pode levar ao bloqueio do processo de aprendizado, ou pode dificultar o processo de convergência da rede na direção de uma boa solução. Uma vez que, os algoritmos de aprendizado conexionistas são em geral muito dependentes do estado inicial da rede e da codificação dos dados da base de aprendizado.
- **Caixa preta**: as redes conexionistas são "caixas preta" onde os conhecimentos ficam codificados de tal forma que estes são inteligíveis para o utilizador ou até mesmo para um especialista. Isto pelo fato, de que, os conhecimentos adquiridos por uma rede estão codificados no conjunto de valores dos pesos sinápticos, e também pela maneira pela a qual as unidades se conectam.
- **Conhecimento teórico**: como as árvores de decisão, as redes neurais são orientadas para a aquisição de conhecimentos empíricos (baseados em exemplos). Um modo simplista de se aproveitar algum conhecimento teórico pré-existente, consiste em se converter regras em exemplos ("protótipos" representativos destas regras). Entretanto, este tipo de método não nos garante que a rede será capaz de aprender corretamente estes exemplos, sendo assim, não podemos garantir que ao final do aprendizado todos os conhecimentos teóricos disponíveis estarão bem representados internamente na rede.

11.10 Conclusões

Nesta sessão apresentou-se uma visão geral sobre os sistemas de I.A. e a necessidade do aprendizado para que um sistema inteligente possa ser considerado como tal. Dando ênfase ao aprendizado neural como sendo uma forma de aquisição de conhecimentos, que dadas as suas peculiaridades, possui um interesse particular na área de inteligência Artificial.

Considerando-se suas principais características: a representação de conhecimentos, o paralelismo inerente as unidades da rede, a sua capacidade de adaptação, entre outros aspectos. No entanto, observa-se que as redes neurais possuem ainda alguns pontos fracos a serem estudados, principalmente no que diz respeito a explicitação dos conhecimentos adquiridos e na dificuldade de convergência em relação a uma solução ótima.

12. Referência Bibliográfica

Agrawal, R.; IMIELINSKI, T.; SWAMI, A. Mining Association Rules Between Sets of Itens in Large Databases. ACM SIGMOD Conference Management of Data, 1993.

Amershi, S., Conati, C. Combining Unsupervised and Supervised Classification to Build User Models for Exploratory Learning Environments.*Journal of Educational Data Mining*,1(1):18-71. 2009.

Baker, R.S.J.d., Corbett, A.T., Roll, I., Koedinger, K.R. Developing a Generalizable Detector of When Students Game the System. *User Modeling and User-Adapted Interaction*, 18 (3): 287- 314. 2008.

Baker, R.S.J.d., de Carvalho, A. M. J. A. Labeling Student Behavior Faster and More Precisely with Text Replays. In *Proceedings of the International Conference on Educational Data Mining*. pagínas 38-47. 2008.

Dean, Jeffrey; Ghemawat, Sanjay. MapReduce: Simplified Data Processing On Large Cluster. CUMMUNICATIONS OF THE ACM, janeiro de 2008.

Dong, G. & J. LI. Interestingness of discovered association rules in terms of neighbordhood-based unexpectedness. Lecture Notes in Artificial Intelligence, pp. 72-86, 1998.

ELMAN, Jeffrey L. *Learning and Development in Neural Networks: The Importance of Starting Small*. Cognition, 48(1993), pp.71-99. 1993. Web: http://crl.ucsd.edu/~elman/
Ftp: ftp://crl.ucsd.edu/pub/neuralnets/cognition.ps.Z

Elmasri, Ramez; Navathe, Shamkant B. Sistemas de Banco de Dados. São Paulo: Addison Wesley, 2005.

Engels. R. Planning tasks for knowledge discovery in databases: Performing Task-Oriented User-Guidance. Proceeding of the International Conference on Knowledge Discovery and Data Mining. Portland: AAAI Press, 1996.

Engels, R.; LINDNER, G.; STUDER, R. A Guided Tour Through the Data Mining Jungle. Proceeding of the Third International Conference on Knowledge Discovery in Databases. Newport Beach, 1997.

Fayyad, Usama; PIATETSKI-SHAPIRO, Gregory; SMYTH, Padhraic (1996). The KDD Process for Extracting Useful Knowledge from Volumes of Data. In: Communications of the ACM, pp.27-34, Nov.1996.

Fiesler, E. Neural Networks Formalization and Classification. Computer Standard & Interfaces, Special Issue on Neural Networks Standards, John Fulcher (Ed.). V.16, N.3. Elsevier Sciences Publishers, Amsterdam, June, 1994. Web: http://www.idiap.ch/idiap-networks.html.

Freitas A. A. A multi-criteria approach for the evaluation of rule interestingness. Em Proceedings of the International Conference on Data Mining. Rio de Janeiro, RJ, pp. 7-20, 1998.

Freitas A. A. On rule interestingness measures. Knowledge-Based Systems 12(5-6), 309-315, 1999.

Goldschmidt, R.; Passos, E.; Vellasco, M.; Pacheco, M. Task Definition Assistence in KDD Applications. CLEI'03 – XXIX Conferência Latino Americana de Informática. La Paz, 2003.

Han, Jiawei; Kamber, Micheline. Data Mining: Concepts and Techniques. Second Edition. Elsevier. San Francisco, CA, 2006.

Han, Jiawei; Kamber, Micheline; Pei, Jian. Data Mining: Concepts and Techniques. Third Edition. Elsevier. San Francisco, CA, 2011.

Haykin, Simon. Redes neurais: princípios e práticas/Simon Haykin; trad. Paulo Martins Engel. – 2.ed. – Porto Alegre: Bookman, 2001.

Hussain F.; Liu H.; Suzuki E.; Lu H. EXCEPTION RULE MINING WITH RELATIVE INTERESTINGNESS MEASURE. PAKDD, 2000; pg 86-97.

Inmon, Bill & Chuck Kelly. The Twelve Rules of Data Warehouse for a Client/Server World, Data Management Review, 1994.

Kimball, Ralph. Data Warehouse toolkit: o guia completo para modelagem multidimensional /Ralph Kimball, Margy Ross; tradução Ana Beatriz Tavares, Daniela Lacerda. Rio de Janeiro: Campus, 2002.

Kolb, Jason; KOLB. Jeremy. The Big Data Revolution. The Tricks Tour Competitors Don't Want You To Know By Jason Kolb and Jeremy Kolb. AppliedData Labs. Plainfield, IL, 2013.

Kohonen, Teuvo. *Self-Organization and Associative Memory*. Springer-Verlag Series in Information Science. 1987.

Kolodner, J. L. Proceedings of the DARPA Case-Based Reasoning Workshop. San Francisco: Morgan Kaufmann Publishers, 1998.

Linderman, R. H. (1986) "Medidas Educacionais". Editora Globo. 1ª Edição. Rio Grande do Sul.

Liu, B. & W. Hsu. Post-analysis of learned rules. AAAI 1, 828-834, 1996.

Mayer-Schönberger, Viktor; Cukier, Kenneth. Big Data. A Revolution That Will Transform How We Live, Work and Things. First published in Greta Britain. John Murray (Publishers) an Hachette UK Compnay, 2013.

Mierswa, I., Wurst, M., Klinkenberg, R., Scholz, M., Euler, T.: YALE: Rapid Prototyping for Complex Data Mining Tasks. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2006). 935-940. 2006.

Moore, A. *Statistical Data Mining Tutorials.*Available online at http://www.autonlab.org/tutorials/2005.

Morik, K. The Representation Race- Preprocessing for Handling Time Phenomena. Proceedings of the European Conference on Machine Learning 2000, Lecture Notes in Artificial Intelligence 1810. Berlin: Springer Verlag, 2000

Negnevitsky, Michael. Artificial Intelligence: a guide to intelligent Systems/Michael Negvitsky. Pearson Education Limited. Edinburgh Gate, 2005.

Oliveira, C., EDACLUSTER: Um Algoritmo Evolucionário para Análise de agrupamentos Baseados em Densidade e Grade, Dissertação (Mestrado em Engenharia Elétrica), Universidade Federal do Pará, 2007.

Osório, Fernando. Redes Neurais – Aprendizado Artificial. Forum de I.A. "99 – pg.13". Rosa, João Luís Garcia. Fundamentos da Inteligência artificial /João Luís Garcia Rosa. Rio de Janeiro: LTC, 2011.

Passos, Emanuel; GOLDSCHMIDT, Ronaldo. Data Mining: Um guia prático. Editora Campos. Rio de Janeiro, 2005.

Pavlik, P., Cen, H., Wu, L. and Koedinger, K. Using Item-type Performance Covariance to Improve the Skill Model of an Existing Tutor. In *Proceedings of the International Conference on Educational Data Mining*, 77-86. 2008.

Pazzini, M. J. Knowledge discovery from data? IEEE Intelligent Systems, 10-13, 2000.

Piatetsky-Shapiro, G & C. J. Matheus. The Interestingness of deviations. Em Proceedings of the International Conference on Knowledge Discovery and Data Mining, pp. 23-36, 1994.

Pimentel, E.P., Omar, N. Descobrindo Conhecimentos em Dados de Avaliacao Aprendizagem com Tecnicas de Mineracao de Dado. Workshop sobre Informática na Escola. *Anais do Congresso da Sociedade Brasileira de Computação*, 147-155, 2006.

Raj, Subu. BIG DATA – AN INTRODUCTION. Kindle Ver 1.1, 2013.

REZENDE, Solange Oliveira. Sistemas inteligentes: fundamentos e aplicações. Editora Manole Ltda. Barueri, SP. 2003.

Riesbeck, C. K., and Schank, R. Inside Case-Based Reasoning. Northvale, NJ: Lawrence Erlbaum Associates, 1996.

Rob, Peter; Coronel, Carlos. Database Systems: Design, Implementation, and Management by Peter Rob and Carlos Coronel 8th Edtion. Thomson Place, Boston, Massachusetts, 2009.

Rob, Peter. Sistemas de Banco de Dados: Projeto, implantação e gerenciamento / Peter Rob, Carlos Coronel. São Paulo: Cengage Learning, 2011.

Romero, C., Ventura, S., Garcia, E. Data mining in course management systems: Moodle case study and tutorial, *Computers & Education*, 51: 368–384, 2008.

Scheines, R., Sprites, P., Glymour, C., Meek, C. Tetrad II: Tools for Discovery. Lawrence Erlbaum Associates: Hillsdale, NJ. 1994.

Silberschatz, A. & Tuzhilin. On subjective measures of interestingness in knowledge discovery. Proceeding of the First International Conference on Knowledge Discovery and Data Mining 1, 275-281, 1995.

Stonebraker, Michael, Abadi; Daniel, DeWitt; David J.; Madden, Sam; Paulson, Erik; Pavlo, Andrew; Rasin, Alexander. MapReduce complements DBMSs since databases are not designed for extracttransform-load tasks, a MapReduce specialty. COMMUNICATIONS OF THE ACM, pp 71. Publicado em Janeiro de 2001.

UTGOFF, P. Shift of Bias for Inductive Concept Learning. Machine Learning: an Artificial Intelligence Approach, v.3, São Francisco: Morgan Kaufmann, 1996.

Watson, Ian D. Applying case-based reasoning: techniques for enterprise systems. San Francisco, CA: Morgan Kaufmann Publishers, Inc, 1997.

Watson, Ian D. Applying Knowledge Management: Techniques for Building Corpoarte Memory. San Francisco, CA: Morgan Kaufmann Publishers, Inc, 2003.

Witten, Ian H.; Frank, Eibe; Hall, Mark A. Data Mining. Practical Machine Learning Tools and Techiniques. 2nd ed. Morgan Kaufmann Publishers is an imprint of Elsevier, 2005.

Witten, Ian H.; Frank, Eibe; Hall, Mark A. Data Mining. Practical Machine Learning Tools and Techiniques. Third Edition. Morgan Kaufmann Publishers is an imprint of Elsevier, 2011.