UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

ARILENE SANTOS DE FRANÇA

OTIMIZAÇÃO DO PROCESSO DE APRENDIZAGEM DA ESTRUTURA GRÁFICA DE REDES BAYESIANAS EM BIGDATA

DM 402/2014

UFPA/ITEC/PPGEE
Campus Universitário do Guamá
Belém-Pará-Brasil
2014

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

ARILENE SANTOS DE FRANÇA

OTIMIZAÇÃO DO PROCESSO DE APRENDIZAGEM DA ESTRUTURA GRÁFICA DE REDES BAYESIANAS EM BIGDATA

DM 402/2014

UFPA/ITEC/PPGEE
Campus Universitário do Guamá
Belém-Pará-Brasil
2014

UNIVERSIDADE FEDERAL DO PARÁ

INSTITUTO DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

ARILENE SANTOS DE FRANÇA

OTIMIZAÇÃO DO PROCESSO DE APRENDIZAGEM DA ESTRUTURA GRÁFICA DE REDES BAYESIANAS EM BIGDATA

Dissertação submetida à Banca Examinadora do Programa de Pós Graduação em Engenharia Elétrica da Universidade Federal do Pará para a obtenção do Grau de Mestre em Engenharia Elétrica na área de Computação Aplicada.

UFPA/ITEC/PPGEE

Campus Universitário do Guamá

Belém-Pará-Brasil

2014

Dados Internacionais de Catalogação-na-Publicação (CIP)

França, Arilene Santos de , 1988-
Otimização do processo de aprendizagem da estrutura gráfica de redes bayesianas em bigdata / Arilene Santos de França. - 2014.

Orientador: Ádamo Lima de Santana.
Dissertação (Mestrado) - Universidade Federal do Pará, Instituto de Tecnologia, Programa de Pós-Graduação em Engenharia Elétrica, Belém, 2014.

1. Teoria bayesiana de decisão estatística. 2. Mineração de dados. I. Título.

CDD 22. ed. 519.542

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

"OTIMIZAÇÃO DO PROCESSO DE APRENDIZAGEM GRÁFICA DA ESTRUTURA GRÁFICA DE REDES BAYESIANAS EM BIGDATA"

AUTOR: **ARILENE SANTOS DE FRANÇA**

DISSERTAÇÃO DE MESTRADO SUBMETIDA À BANCA EXAMINADORA APROVADA PELO COLEGIADO DO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA, SENDO JULGADA ADEQUADA PARA A OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA ELÉTRICA NA ÁREA DE COMPUTAÇÃO APLICADA.

APROVADA EM: 20 / 02 / 2014

BANCA EXAMINADORA:

Prof. Dr. Ádamo Lima de Santana

(Orientador – PPGEE/UFPA)

Prof. Dr. Marcos César da Rocha Seruffo

(Avaliador Externo ao Programa – UFPA/CASTANHAL)

Prof. Dr. Cláudio Alex Jorge da Rocha

(Avaliador Externo – IFPA)

VISTO:

Prof. Dr. Evaldo Gonçalves Pelaes

(Coordenador do PPGEE/ITEC/UFPA)

Dedico este trabalho:

Aos meus pais, Francisco e Maria, pelo investimento na base da minha formação.

Aos meus irmãos André e Adriano pelo apoio em muitos momentos.

AGRADECIMENTOS

Agradeço a Deus por estar comigo em todos os momentos, mesmo quando em silêncio, me iluminando e me impulsionando para sempre seguir em frente quando achei que não tinha mais forças para continuar esta jornada.

À minha família que, mesmo muitas vezes sem entender o real significado de tudo isto, não poupou esforços para investir tempo e, principalmente, dinheiro em minha formação, sempre preocupados com o meu futuro. Por serem a minha base. Por me apoiar sempre.

Às minhas amigas, pelos momentos de descontração ainda que não quantitativos, mas qualitativos, os quais contribuíram ainda que indiretamente para tornar esta jornada um pouco menos estressante.

Ao Haroldo Filho, pela amizade e dedicação, pelo apoio e incentivo nos momentos de desespero, pelos abraços apertados, pelas risadas “infinitas” e por não deixar que eu jogasse tudo pro alto. Por estar ao meu lado quando eu mais precisei.

Ao meu orientador, Prof. Dr. Ádamo Santana, pela paciência e compreensão, por entender as situações adversas que quase me fizeram parar pelo meio do caminho e pelo incentivo para que eu não desistisse. Pelas correções e orientações neste período de aprendizado.

Aos meus companheiros do laboratório de pesquisa, pelos momentos de descontração, que ajudaram tornar um período de longa dedicação em algo menos tedioso, e também pelos conselhos e ensinamentos que levarei por toda a vida. Em especial aos colegas J. Gabriel e ao Jacob Jr., pela colaboração direta que tiveram para que este trabalho fosse finalizado e o artigo submetido.

A todos que colaboraram de forma direta ou indireta para a realização deste trabalho.

À CAPES, pelo apoio financeiro.

“Nunca se é tarde demais para ser aquilo que sempre se desejou ser”

George Eliot.

SUMÁRIO

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS

BD	Banco de Dados
BDI	Base de Dados Intermediária
BI	*Business Intelligence*
BO	Boletim de Ocorrência
DAG	Grafo Dirigido Acíclico (do inglês *Directed Acyclic Graph*)
DM	Mineração de Dados
EM	*Expectation Maximization*
IOPS	*Input/Output Per Second*
KDD	*Knowledge Discovery in Database*
MGI	*McKinsey Global Institute*
MR	MapReduce
NoSQL	*Not only SQL* (Bases de Dados Não-Relacionais)
RB	Redes Bayesianas
TPC	Tabela de Probabilidade Condicional

RESUMO

A automação na gestão e análise de dados tem sido um fator crucial para as empresas que necessitam de soluções eficientes em um mundo corporativo cada vez mais competitivo.

A explosão do volume de informações, que vem se mantendo crescente nos últimos anos, tem exigido cada vez mais empenho em buscar estratégias para gerenciar e, principalmente, extrair informações estratégicas valiosas a partir do uso de algoritmos de Mineração de Dados, que comumente necessitam realizar buscas exaustivas na base de dados a fim de obter estatísticas que solucionem ou otimizem os parâmetros do modelo de extração do conhecimento utilizado; processo que requer computação intensiva para a execução de cálculos e acesso frequente à base de dados.

Dada a eficiência no tratamento de incerteza, Redes Bayesianas têm sido amplamente utilizadas neste processo, entretanto, à medida que o volume de dados (registros e/ou atributos) aumenta, torna-se ainda mais custoso e demorado extrair informações relevantes em uma base de conhecimento.

O foco deste trabalho é propor uma nova abordagem para otimização do aprendizado da estrutura da Rede Bayesiana no contexto de BigData, por meio do uso do processo de MapReduce, com vista na melhora do tempo de processamento. Para tanto, foi gerada uma nova metodologia que inclui a criação de uma Base de Dados Intermediária contendo todas as probabilidades necessárias para a realização dos cálculos da estrutura da rede.

Por meio das análises apresentadas neste estudo, mostra-se que a combinação da metodologia proposta com o processo de MapReduce é uma boa alternativa para resolver o problema de escalabilidade nas etapas de busca em frequencia do algoritmo K2 e, consequentemente, reduzir o tempo de resposta na geração da rede.

Palavras-chave: Redes Bayesianas, Mineração de Dados, Big Data.

ABSTRACT

Automation at data management and analysis has been a crucial factor for companies which need efficient solutions in an each more competitive corporate world.

The explosion of the volume information, which has remained increasing in recent years, has demanded more and more commitment to seek strategies to manage and, especially, to extract valuable strategic informations from the use of data mining algorithms, which commonly need to perform exhausting queries at the database in order to obtain statistics that solve or optimize the parameters of the model of knowledge discovery selected; process which requires intensive computing to perform calculations and frequent access to the database.

Given the effectiveness of uncertainty treatment, Bayesian networks have been widely used for this process, however, as the amount of data (records and/or attributes) increases, it becomes even more costly and time consuming to extract relevant information in a knowledge base.

The goal of this work is to propose a new approach to optimization of the Bayesian Network structure learning in the context of BigData, by using the MapReduce process, in order to improve the processing time. To that end, it was generated a new methodology that includes the creation of an Intermediary Database, containing all the necessary probabilities to the calculations of the network structure.

Through the analyzes presented at this work, it is shown that the combination of the proposed methodology with the MapReduce process is a good alternative to solve the scalability problem of the search frequency steps of K2 algorithm and, as a result, to reduce the response time generation of the network.

Key words: Bayesian Networks, Data mining, Big Data.

INTRODUÇÃO

1.1 CONSIDERAÇÕES INICIAIS

A quantidade de dados gerados nas mais diferentes áreas do conhecimento tem crescido de maneira espantosa. Este crescimento exponencial gera não somente o desafio de armazenamento e gerenciamento do grande volume de dados (“Big Data”), mas também de como analisá-los e extrair conhecimento relevante (BAKSHI 2012; DEMIRKAN et al., 2012; FAN et al., 2013).

Devido à limitação humana em analisar e entender grandes conjuntos de dados, cientistas e pesquisadores tem se engajado no estudo de novas abordagens e técnicas, com o propósito de tratar de maneira eficiente a grande quantidade de informações existentes (AGGARWAL, 2012).

Neste sentido, diversos modelos computacionais vêm sendo desenvolvidos com o intuito de simplificar o entendimento da relação entre as variáveis em grandes conjuntos de dados brutos. Algoritmos de Mineração de Dados podem auxiliar na descoberta de conhecimento, entretanto, estes algoritmos geralmente necessitam fazer uma leitura de toda a base de treinamento para obter as estatísticas necessárias para otimizar os parâmetros dos modelos, processo que requer computação intensiva e acesso frequente aos dados em larga escala (WU et al., 2014).

Dentre os modelos existentes, as Redes Bayesianas (RBs), devido a sua fácil interpretabilidade e tratamento de incerteza, têm sido amplamente utilizadas em diversos campos de conhecimento (PERRIER, 2008). Em especial, pode se destacar os campos de aprendizado de máquina e DM, onde tem recebido bastante atenção da comunidade científica (FANG, 2013).

O aprendizado da estrutura da Rede Bayesiana é um importante problema a ser estudado, pois, à medida que o volume de dados aumenta, torna-se cada vez mais difícil construir a estrutura da rede manualmente (CHEN et al., 2008).

Em um dos trabalhos principais nesta área, Cooper e Herskovits (1991) discutem sobre a complexidade em encontrar a estrutura mais provável de uma RB, e propõem a utilização de um método heurístico denominado K2, o qual tem por objetivo aprender a estrutura da rede de modo automático. O método discutido pelos

autores seleciona a estrutura da RB de maneira quantitativa com base em uma função de pontuação.

Entretanto, como citado anteriormente, enumerar todas as possíveis estruturas de rede torna-se uma tarefa dispendiosa à medida que o número de variáveis do domínio aumenta, pois o tamanho do espaço de busca tende a aumentar de modo exponencial de acordo com o número de variáveis do modelo, portanto, com o aumento do volume de dados, o tempo de processamento do algoritmo K2 tende a aumentar também, devido ao alto custo para calcular os parâmetros necessários do algoritmo (COOPER e HERSKOVITS, 1991; FANG et al., 2013).

1.2 OBJETIVO

O presente trabalho objetiva o estudo de técnicas que possam aperfeiçoar o processo de extração de conhecimento em bases de dados a fim de diminuir o tempo utilizado nas buscas realizadas em um domínio com grande quantidade de registros, ou seja, com ênfase na melhora do tempo de processamento, mas mantendo a qualidade do conhecimento extraído.

Para esse fim, propõe-se uma abordagem de implementação do MapReduce com vista à redução do tempo de processamento necessário para a aprendizagem da estrutura gráfica de RBs, a partir de grandes conjuntos de dados, utilizando o algoritmo K2. Através de dados experimentais, pretende-se mostrar que é possível essa otimização a partir de um conjunto de variáveis, mesmo que não haja uma ordenação prévia dos atributos especificada por um especialista no domínio, que é uma premissa conhecida para a utilização do algoritmo citado.

1.3 ESTRUTURAÇÃO

Este trabalho está dividido em 6 capítulos, cada um contendo tópicos específicos aos assuntos a que se referem e se apresenta organizado da seguinte forma. O capítulo 2 trata da fundamentação teórica, apresentando a base de conhecimento necessária ao entendimento do assunto discutido. O capítulo 3 apresenta os trabalhos correlatos que mostram a relevância deste estudo e demais trabalhos realizados na área; O capítulo 4 faz a apresentação do domínio de estudo e análise

dos dados, mostrando como se deu o processo de extração de conhecimento; O capítulo 5 apresenta e discute os resultados obtidos durante o processo, a fim de validar este estudo; O capítulo 6 apresenta as considerações finais; e, em seguida, são apresentadas as referências utilizadas.

FUNDAMENTAÇÃO TEÓRICA

2.1 PROCESSO DE KDD

O interesse em solucionar o problema de transformar dados em conhecimento, de forma que o processo não se utilize de métodos eminentemente manuais, tem fomentado várias pesquisas em um campo emergente chamado Extração de Conhecimento de Bases de Dados (KDD - *Knowledge Discovery in Database*) (FAYYAD et al., 1996).

O processo de extração de conhecimento de bases de dados objetiva a compreensão dos dados, adquirindo relações de interesse não observadas pelo especialista do domínio, bem como auxiliando a validação do conhecimento extraído. Esse processo é bastante complexo, pois consiste de uma tecnologia composta de um conjunto de modelos matemáticos e técnicas de software, além de ser um processo centrado na interação entre usuários, especialistas do domínio e responsáveis pela aplicação do processo KDD (SANTANA, 2005 apud DECKER; FOCARDI, 1995).

Ainda não é consenso a definição dos termos KDD e Mineração de Dados. Para alguns autores estes termos são considerados sinônimos (REZENDE, 2005; WANG, 2005; HAN e KAMBER, 2006). Para outros autores o KDD refere-se a todo o processo de descoberta de conhecimento, e a Mineração de Dados a uma das atividades do processo (FAYYAD et al., 1996; CIOS et al., 2007).

Uma das definições mais utilizadas para o termo KDD é a apresentada por Fayyad (1996), que o define como “um processo não trivial de identificação de novos padrões válidos, úteis e compreensíveis”.

Segundo (FAYYAD et al., 1996), o processo de KDD pode ser dividido em cinco fases, conforme apresentado na (Figura 2.1), a saber: Seleção, Pré-processamento, Transformação, Mineração de Dados e Avaliação do Conhecimento. O principal objetivo deste processo é encontrar padrões válidos e potencialmente úteis nos dados.

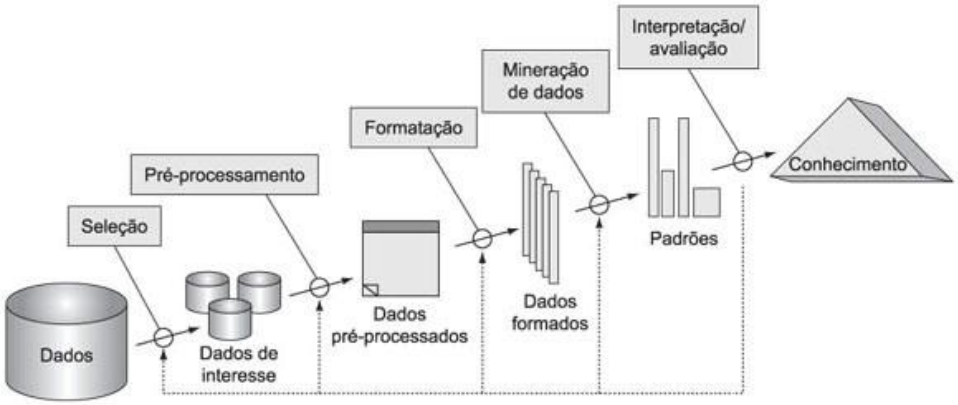

Figura 2.1 – Etapas do processo de KDD (FAYYAD et al., 1996)

A primeira etapa do processo de KDD consiste em selecionar um conjunto de dados relevante ao processo de extração do conhecimento, portanto, é importante conhecer o domínio dos dados a serem analisados.

Na segunda etapa, ocorre o tratamento da base de dado com relação a valores ausentes, ruídos, inconsistências, redundâncias ou quaisquer outros problemas específicos, de modo que seja possível a aplicação de algoritmos de mineração para a extração do conhecimento; Segundo (KLEMETTINEN et. al., 1994), esta é a parte mais demorada e consome cerca de 80% do esforço total do processo de KDD. Técnicas de pré-processamento de dados são frequentemente utilizadas para melhorar a qualidade dos dados por meio da eliminação ou minimização dos problemas citados, o que facilita o uso de determinados algoritmos e pode levar à construção de modelos mais eficientes, reduzindo assim a complexidade computacional associada ao problema. Deste modo, a próxima etapa consiste na organização do conjunto de dados segundo as requisições e particularidades da técnica que será aplicada na fase seguinte.

A etapa de Mineração de Dados é onde os dados são de fato analisados a fim de encontrar padrões consistentes que estabeleçam relações de dependência entre os dados do conjunto, permitindo assim que, na etapa seguinte, o usuário especialista do domínio possa interpretar e analisar as informações extraídas de modo a utilizá-las no processo de tomada de decisão.

2.2 MINERAÇÃO DE DADOS

Mineração de Dados, ou *Data Mining* (DM), é uma parte integrante do processo de busca por conhecimento em bases de dados (KDD) e consiste no processo de descoberta automática de informação útil em bases de dados de modo a encontrar novos padrões úteis que poderiam permanecer desconhecidos (FAYYAD et al.,1996; TAN et al., 2006).

Esta fase envolve a criação de modelos apropriados de representação dos padrões e relações identificados a partir dos dados. O resultado desses modelos, depois de avaliados pelo analista, especialista e/ou usuário final, são empregados para predizer os valores de atributos definidos pelo usuário final baseados em novos dados (FAYYAD et al.,1996).

Existem cinco tarefas gerais de DM que englobam todas as outras formas de apresentação e permitem uma visão mais global e apropriada ao assunto. São elas: a classificação, a estimativa, a previsão, a análise de afinidades e a análise de agrupamentos (AMORIM, 2006 apud CARVALHO, L., 2005).

Entretanto, com o aumento do volume de dados torna-se cada vez mais desafiador conseguir extrair informações de bases de conhecimento. (WU et. al., 2014), apresenta este desafio de maneira simples através do exemplo descrito abaixo:

"Imagine que alguns homens cegos estão tentando medir um elefante (BigData). O objetivo de cada homem cego é descrever o elefante de acordo com a parte da informação que foi coletada por ele durante o processo. Devido à análise limitada de cada um a uma determinada região, não é de se espantar que cada homem cego tenha uma perspectiva diferente, concluindo que o elefante pareça com uma corda, um cavalo, uma árvore ou uma parede, dependendo da região ao qual estivesse limitado." (Figura 2.2).

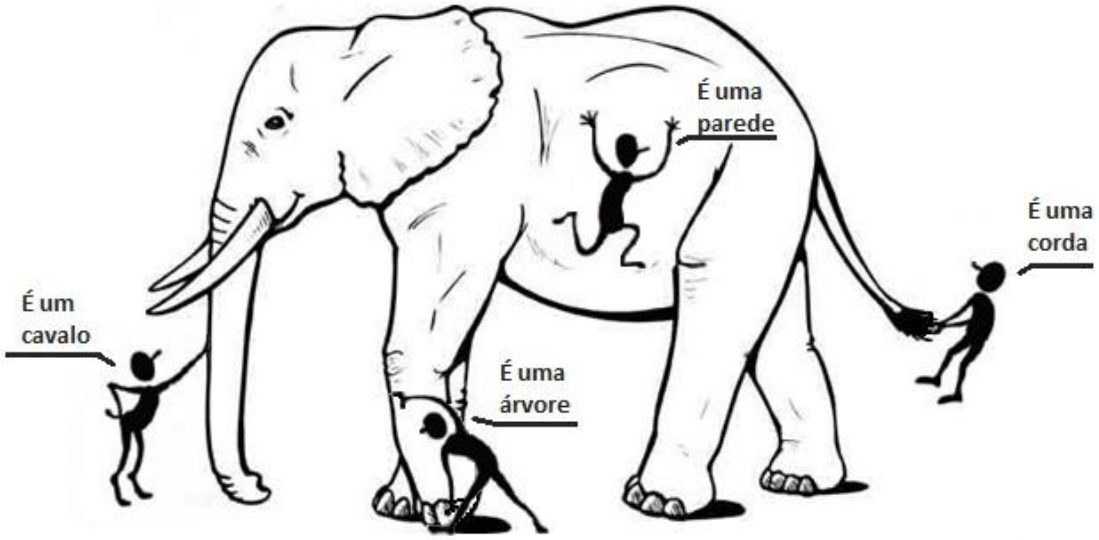

Figura 2.2 – Os homens cegos e o elefante gigante: a visão localizada (limitada) de cada homem cego leva a uma conclusão tendenciosa. Adaptado de: (WU et. al., 2014).

Ainda no exemplo utilizado por (WU et. al., 2014), considerando que o elefante esteja crescendo rapidamente e mude constantemente de posição, que cada cego tem sua própria fonte de informação a respeito do elefante e que essa informação possa ser compartilhada com outro homem cego, pode se dizer que é o equivalente em BigData a agregação de informações heterogêngeas de diversas fontes (cegos) de modo a conseguir descrever da melhor maneira possível uma imagem que represente a posição do elefante em tempo real.

Deste modo, com base no exemplo anterior, é possível concluir que o processo de amostragem, normalmente utilizado para representação de conhecimento de bases maiores a um reduzido número de informações, pode não representar o contexto real de um determinado cenário, visto que muitas informações podem ser perdidas neste processo.

2.3 BIGDATA

O termo “Big Data” pode ser caracterizado como um conjunto de dados que crescem exponencialmente e que são demasiadamente volumosos, brutos ou desestruturados para serem analisados por meio de técnicas tradicionais de bancos relacionais e de *Business Intelligence* (BI). Esta dificuldade pode estar relacionada com a captura de dados, armazenamento, pesquisa, compartilhamento, análise e visualização, etc. (D’ANDREA, 2010; SINGH e SINGH, 2012; MADDEN, 2012).

Nesse sentido, Big Data simboliza a aspiração de construir plataformas e ferramentas que possam solucionar estes problemas (CHAUDHURI, 2012).

Quando se fala em volume, os números são gigantescos. Olhando de maneira globlal, fala-se em zetabytes ou 10^{21} bytes. Grandes corporações armazenam múltiplos petabytes e mesmo pequenas e médias empresas trabalham com dezenas de terabytes de dados. Este volume de dados tende a crescer geometricamente e em mundo cada vez mais competitivo e rápido, as empresas precisam tomar decisões baseadas não apenas em palpites, mas em dados concretos (CHEDE, 2012).

Um documento publicado pelo *World Economic Forum* (2012), sobre os impactos do Big Data, mostra como é possível prever desde a magnitude de uma epidemia a sinais de uma provável seca em determinada região por meio da análise de padrões em grandes volumes de dados.

A complexidade do Big Data vem à tona quando lembramos que não estamos falando apenas de armazenamento e tratamento analítico de massivos volumes de dados, mas de revisão ou criação de processos que garantam a qualidade destes dados e de processos de negócio que usufruam dos resultados obtidos. Portanto Big Data não é apenas um debate sobre tecnologias, mas principalmente como os negócios poderão usufruir da montanha de dados que está agora à sua disposição (CHAUDHURI, 2012), através da análise e extração de valores significativos (BAKSHI, 2012).

De modo a facilitar as análises em grandes conjuntos de dados através de aplicações escaláveis, faz-se necessário a exploração de novas técnicas que permitam o armazenamento e gerenciamento eficiente de dados e redução dimensional. Este contexto contribuiu para o surgimento de novos paradigmas e tecnologias que tem por objetivo melhorar o desempenho das aplicações (SINGH e SINGH, 2012; LIU, 2012; ZHANG et al., 2012).

2.3.1 Bases de Dados Não relacionais (NoSQL)

O uso de bases de dados não relacionais surgiu como uma solução a grande parte dos problemas de armazenamento e gerenciamento do grande volume de dados por algumas razões, como ser facilmente distribuído, escalável, possuir um esquema flexível e suporte nativo para replicação (DIANA e GEROSA, 2010), além de outras

vantagens como a eficácia na manipulação de dados em massa, mesmo que não estruturados (provenientes de diversas fontes), especialmente em áreas como BI e Big Data Mining (RAUTENBERG, 2011; HU et. al., 2012).

A maioria dos bancos de dados NoSQL são *open source* e baseiam-se em diferentes tipos de modelos (baseado em colunas, documentos, tuplas, grafos e modelos híbridos)[1], dentre as diversas implementações de código aberto disponíveis, podemos citar: Cassandra, Hypertable, MongoDB, Redis, CouchDB, Dynamo, Neo4j e OrientDB (JAYATHILAKE et al. 2012). Para este trabalho optou-se pelo banco de dados não-relacional MongoDB.

2.3.2 MongoDB

Considerado um dos mais populares entre os bancos de dados baseados em documento, o MongoDB[2] foi escrito na linguagem de programação C++ e dentre suas principais características destacam-se o grande poder de indexação e facilidade de gerenciamento (BOICEA et al., 2012).

Uma única instância do MongoDB pode hospedar vários bancos de dados independentes, cada um dos quais pode ter suas próprias coleções e permissões, onde cada coleção é um conjunto de documentos, que é a unidade básica de dados para MongoDB, aproximadamente equivalente a uma linha de uma base de dados relacional. Os documentos podem ter diferentes esquemas, o que significa que um registro de um documento pode ter três atributos e o próximo registro possuir dez atributos (LIU et al., 2012; WEI-PING et. al., 2011).

MongoDB fornece flexibilidade durante o processo de desenvolvimento. Foi construído em suporte para escalabilidade horizontal utilizando *Sharding*, que é a abordagem MongoDB para atender as demandas de crescimento dos dados. Essa abordagem consiste no processo de armazenar registros de dados em várias máquinas a fim de suportar o crescimento dos dados e as demandas de leitura e

---

[1] http://nosql-database.org/

[2] http://docs.mongodb.org/manual

escrita (I/O), pois, com o aumento do volume de dados, uma única máquina pode se tornar insuficiente e, consequentemente, não apresentar rendimento aceitável para esses processos.

Além disso, é fácil de instalar e copiar dados de um servidor para outro usando ferramentas de exportação e importação. Permite armazenar dados complexos em um campo, como: um objeto, uma matriz ou uma referência em um campo. O Mongo mapeia facilmente alguns objetos de diferentes problemas de linguagem no banco de dados (como objetos *javascrip*t ou objetos python). Não precisa de nenhum tipo de conversão (BOICEA et al., 2012; LIU et al., 2012). Além disso, implementa nativamente tarefas de MapReduce como uma primitiva da interface de consulta dentro do sistema (VERMA et al., 2010; BONNET et al., 2011).

2.3.3 MapReduce

No que diz respeito a melhora do desempenho com relação ao tempo de processamento das aplicações, um paradigma que se tornou bastante popular foi o uso do MapReduce (BASAK et al., 2012a).

O MapReduce é um framework de programação e implementação para computação distribuída em grandes conjuntos de dados, popularizado pela Google em 2004. Tornou-se um paradigma popular principalmente entre grandes empresas de redes sociais e de compartilhamento de conteúdo, onde grandes quantidades de dados são geradas todos os dias por seus usuários (DEAN e GHEMAWAT, 2004). O MapReduce se divide basicamente em duas etapas (Figura 2.3):

1. Etapa de mapeamento (Map), que é aplicada a cada registro dos dados de entrada a fim de gerar um conjunto intermediário de pares do tipo <chave, valor>;
2. Etapa de redução (Reduce), a qual se inicia após a finalização do processo de mapeamento, agrupando os valores de acordo com as suas chaves.

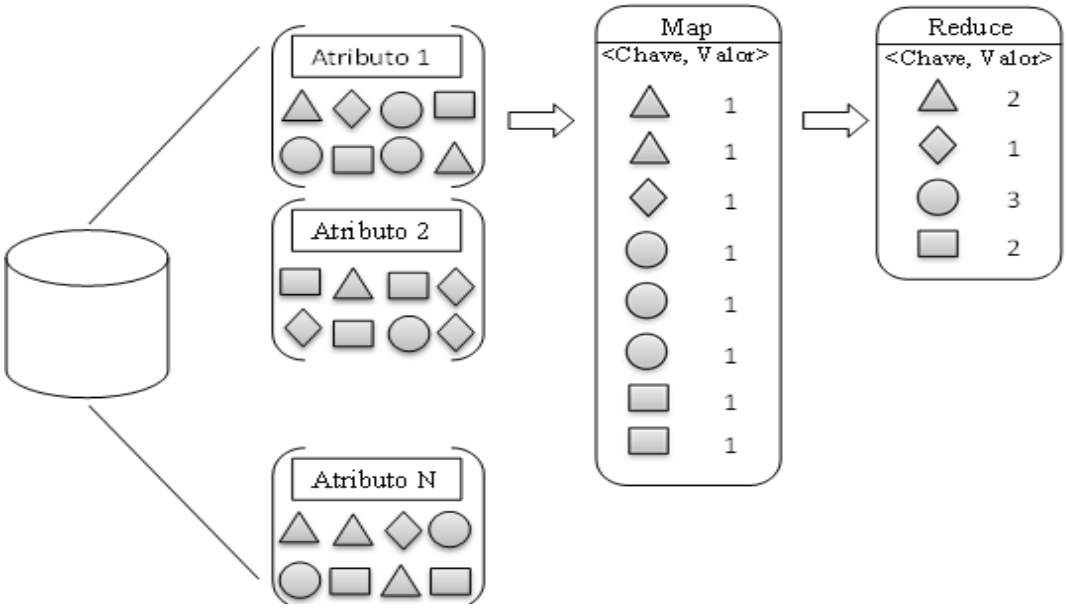

Figura 2.3 – Processo de contagem de frequências dos estados dos atributos em uma base de dados

Parte essencial do desempenho deste processo se baseia em um sistema de arquivos distribuídos capaz de realizar o processamento de forma paralela (BASAK et al., 2012a). Devido a essa característica, abordagens utilizando este modelo provaram ser eficazes para a análise de grandes quantidades de dados, permitindo a construção de sistemas de mineração distribuídos e redução no tempo de execução do processo de Mineração de Dados (PÉREZ et al., 2007).

2.4 MODELOS DE INTELIGÊNCIA COMPUTACIONAL

Diversos modelos de inteligência computacional podem ser aplicados às bases de dados, dependendo do tipo de análise que se pretende realizar. Basicamente, existem cinco técnicas gerais que englobam todas as outras formas de apresentação e permitem uma visão mais global e apropriada ao assunto, a saber: classificação, estimativa, previsão, análise de afinidades e análise de agrupamentos (AMORIM, 2006 apud CARVALHO, L., 2005).

Em geral, os modelos de Mineração de Dados podem ser obtidos a partir da aplicação de algoritmos, os quais comumente necessitam realizar uma varredura em toda a base de treinamento a fim de obter as estatísticas que solucionem ou otimizem os parâmetros do modelo utilizado. Este processo requer computação intensiva para acessar dados em larga escala de maneira frequente (WU et. al.,

2014). Etretanto, muitos destes algoritmos podem ser paralelizados utilizando o processo de MapReduce (GILLICK et al., 2006; CHU et al., 2006).

Para este trabalho, será utilizado o modelo Bayesiano, por ser um moledo que exige grande quantidade de operações matemáticas, o que o torna adequado a comprovação deste estudo, visto que há grande complexidade em utilizar este modelo a medida que aumenta a quantidade de variáveis e registros a serem analisados.

As Redes Bayesianas, também conhecidas como Rede de Crença, Rede Probabilística ou Rede Causal, podem ser vistas como um modelo que utiliza teoria dos grafos, condições de Markov e distribuição para representar uma situação, suas variáveis e estados; e, a partir disto, realizar inferências (GONÇALVES, 2008).

Três fatores têm motivado a utilização de RB no processo de Mineração de Dados (KORB e NICHOLSON, 2003): primeiro, a eficácia da manipulação de dados incompletos; segundo, a aprendizagem de relações causais entre as variáveis do domínio, o que facilita a sua análise; e, por fim, as redes bayesianas permitem a combinação de conhecimento *a priori* do domínio com os dados disponíveis.

A topologia da rede é formada por um conjunto de variáveis (nós) e arcos que ligam essas variáveis, formando um grafo dirigido acíclico (DAG – *Directed Acyclic Graph*) em que cada nó possui uma distribuição condicional $P(X_i|Pais(X_i))$ que quantifica o efeito dos pais sobre o nó. A Figura 2.4 mostra um esquema representativo em que os nós são definidos por A, B, C, D, E e F; e os arcos direcionais representam a relação de causalidade entre as variáveis. Segundo (CAMARINHA, 2009 apud CHARNIAK, 1991) a grande vantagem no uso deste tipo de estrutura está em conseguir representar incerteza de forma gráfica através de nós e grafos.

Basicamente, uma RB se constitui em dois componentes importantes: uma estrutura qualitativa, representando as dependências entre os nós (variáveis do domínio); e uma estrutura quantitativa (Tabelas de Probabilidades Condicionais - TPCs desses nós) para cada variável da rede, e que avalia, em termos probabilísticos, essas dependências (CHEN, 2001). Juntos, esses componentes propiciam uma representação eficiente da distribuição de probabilidade conjunta do grupo de variáveis X_i de um determinado domínio (PEARL, 1988).

Portanto, para a construção de uma Rede Bayesiana, são necessárias as dependências condicionais entre os atributos e suas TPCs. No caso das tabelas de

probabilidade, são necessárias apenas as probabilidades a *priori* de ocorrência para os atributos que não possuem nenhum nó pai relacionado a ele (atributos A e B), e para os que possuem, são necessárias as probabilidades do mesmo com relação aos seus respectivos pais (atributos C, D, E e F), como pode ser visto no exemplo dado a seguir (Figura 2.4).

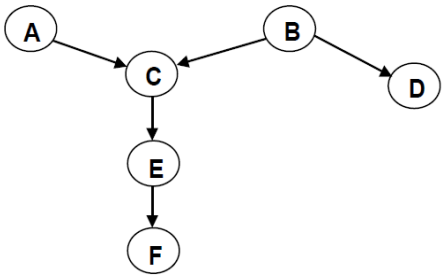

Figura 2.4 – Exemplo de uma Rede Bayesiana (SANTANA, 2008)

A partir da RB gerada é possível realizar inferências através de cálculos probabilísticos por meio da utilização da fórmula de probabilidade condicional do teorema proposto por Bayes (Equação 3).

2.4.1 Teorema de Bayes

Supondo dois eventos A e B, pelos axiomas básicos da probabilidade, sabe-se que a probabilidade de A acontecer dado que B ocorreu é dada por (Equação 1):

$$P(A|B) = \frac{P(A,B)}{P(B)} \qquad \text{(Equação 1)}$$

Onde:

$P(A|B)$ → Probabilidade de A dado que um evento B aconteceu;

$P(A,B)$ → Probabilidade de A e B terem acontecido,

$P(B)$ → Probabilidade do evento B ocorrer.

Uma vez que $P(A,B)$ é o mesmo que $P(B,A)$, onde ambos representam a probabilidade de que A e B tenham ocorrido, e haja vista que:

$$P(B|A) = \frac{P(B,A)}{P(A)} \qquad \text{(Equação 2)}$$

Então é possível igualar as equações 1 e 2, pela reordenação de seus termos, resultando no teorema de Bayes:

$$P(A,B) = P(B,A)$$

Portanto:

$$P(A|B)P(B) = P(B|A)P(A)$$

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \qquad \text{(Equação 3)}$$

De maneira geral podemos dizer que, para um evento A com n estados, a regra de Bayes é dada pela equação:

$$P(A = v_i|B) = \frac{P(B|A = v_i)P(A = v_i)}{\sum_{k=1}^{n_A} P(B|A = v_k)\mathrm{P}(A = v_k)} \qquad \text{(Equação 4)}$$

2.4.2 Inferência Bayesiana

Uma vez que construímos a RB, é possível realizar análises nas correlações entre os atributos da rede. Essa probabilidade *a posteriori* não é armazenada diretamente no modelo e, portanto, deve ser computada. De maneira geral, o cálculo de probabilidade de interesse, dado um modelo, é conhecido como inferência probabilística.

É possível inferir sobre as dependências condicionais que se estabelecem entre as variáveis com base no gráfico da rede, bastando evidenciar a ocorrência de um determinado estado em uma ou mais variáveis da rede, propagando, dessa forma, o efeito das observações pela rede (SANTANA, 2008).

O processo de inferência bayesiana é a base do uso de RBs e consiste no processo de obtenção da probabilidade *a posteriori* a partir da probabilidade *a priori*, ou seja, extrair o conhecimento representado em uma rede já definida. O cálculo da probabilidade pode ser representado pelo produto das probabilidades dos nós ou, quando possuírem pais, da sua probabilidade condicional. De maneira geral temos que (Equação 5):

$$P(X_1 = x_1,\ X_2 = x_2, \dots, X_{n-1} = x_{n-1}, X_n = x_n) =$$
$$P(X_n = x_n,\ X_{n-1} = x_{n-1}, \dots, X_2 = x_2, X_1 = x_1) =$$
$$P(X_n = x_n|X_{n-1} = x_{n-1}, \dots, X_2 = x_2, X_1 = x_1)P(X_{n-1} = x_{n-1}, \dots, X_2 = x_2, X_1 = x_1) =$$

$$P(X_n = x_n | X_{n-1} = x_{n-1}, \dots, X_2 = x_2, X_1 = x_1) P(X_{n-1} = x_{n-1} | X_{n-2} = x_{n-2}, \dots, X_1 = x_1)$$
$$P(X_{n-2} = x_{n-2}, \dots, X_1 = x_1) =$$
$$\vdots$$
$$\prod_{i=1}^{n} P(X_i = x_i | X_{i-1} = x_{i-1}, \dots, X_1 = x_1) = \prod_{i=1}^{n} P(X_i = x_i | Pais(X_i)) \qquad \text{(Equação 5)}$$

2.4.3 Aprendizagem Bayesiana

A aprendizagem da estrutura da RB pode se dar basicamente de duas formas: supervisionado e não-supervisionado. No primeiro caso, as informações são definidas manualmente por um especialista no domínio, com base no conhecimento pessoal. No segundo caso, a estrutura é dada através de algoritmos de aprendizagem aplicados a bases de dados e, diferentemente do primeiro caso, depende de cálculos sobre os dados armazenados (VEIGA e SILVA, 2002). Além disso, pode-se ainda utilizar uma combinação entre essas duas abordagens (NEAPOLITAN, 2004).

Como já foi dito em capítulos anteriores, à medida que o volume de dados aumenta, torna-se cada vez mais difícil construir a estrutura da rede manualmente (CHEN et al., 2008). Além disso, o tamanho do espaço de busca de possíveis estruturas tende a aumentar exponencialmente junto com o número de variáveis do modelo, como pode ser visto pela (Equação 6) (SANTANA, 2008 apud ROBINSON, 1976), que aponta o número de possíveis gafos acíclicos dirigidos G, que podem ser gerados com um número n de variáveis:

$$G(n) = \sum_{i=1}^{n} (-1)^{i+1} \binom{n}{i} 2^{i(n-i)} G(n-i) \qquad \text{(Equação 6)}$$

Tabela 2.1 – Número de possíveis grafos gerados de acordo com a Equação 6.

n	**Número de possíveis gafos acíclicos dirigidos**
1	1
2	3
3	25
4	543
5	29.281
6	3.781.503
7	1.138.779.265
8	783.702.329.343
9	1.213.442.454.842.881
10	4.175.098.976.430.598.100

Portanto, o aprendizado automático da estrutura da RB é um importante problema a ser estudado, a fim de otimizar este processo.

Os métodos de aprendizado Bayesianos a partir dos dados consideram, basicamente, dois aspectos. Primeiro, que a estrutura da Rede Bayesiana (com o conhecimento prévio do domínio) pode ou não ser fornecida a priori e, segundo, que os valores dos atributos da base de dados podem ser completos ou com valores ausentes (SANTANA et al., 2004).

Existem duas abordagens principais para o aprendizado da estrutura da RB de forma automática: Métodos baseados em análise por dependência, que utilizam testes estatísticos para encontrar a estrutura da rede de crenças; e métodos baseados em busca e pontuação, que se destacam pela redução da complexidade do tempo de processamento (PIFER e GUEDES, 2007).

Nos métodos baseados em análise de dependência, a estrutura qualitativa representa o conjunto de independência condicional associado aos nós da rede. A independência condicional na distribuição representada por uma RB é codificada na estrutura do grafo e pode ser encontrada usando o critério *d-separation* (PEARL, 1988). Exemplos de algoritmos deste método: PC, CDL, SGS, SRA e *Wermuth-Lauritzen*.

Os métodos de busca e pontuação começam por um grafo somente com os nós e vão sendo adicionados os arcos de acordo com o método de busca, então se pontua a nova estrutura e compara com a pontuação da estrutura anterior, selecionando aquela que apresenta melhor pontuação (VEIGA e SILVA, 2002). Dentre os métodos de busca e pontuação, o K2 é um dos mais conhecidos e utilizados (COOPER e HERSKOVITZ, 1992).

O K2 procura, dentre as $2^{n(n-1)/2}$ configurações possíveis, a que maximiza a função de pontuação, sendo n o número de variáveis e, permite encontrar a mais provável estrutura de rede de crença B's a partir de um determinado conjunto de dados *D* (HECKERMAN, 1997). O algoritmo K2 aplica a pontuação bayesiana segundo (Equação 7).

$$P(B_S \mid D) = \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(r_i)}{\Gamma(r_i + N_{ij})} \prod_{k=1}^{r_i} \Gamma(N_{ijk} + 1) \qquad \text{(Equação 7)}$$

Onde:

$n \rightarrow$ Número de nós;

$q_i \rightarrow$Número de configurações dos pais da variável X_i ;

$r_i \rightarrow$Número de possíveis valores do nó X_i ;

$N_{ijk} \rightarrow$Número de casos em D onde o atributo X_i é instanciado com o seu valor k, e a configuração dos pais de X_i é instanciada com o valor j,

$N_{ij} \rightarrow$Denota o número de observações em que a configuração dos pais de X_i é instanciada com o valor j, sendo $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$.

A partir do uso deste método, pode-se gerar uma RB para cada conjunto de dados e aplicar o algoritmo de propagação para efetuar inferências sobre essas redes.

O algoritmo utilizado apresenta algumas particularidades, dentre elas, a exigência de que o especialista informe a ordem de todas as variáveis, o que faz com que o algoritmo evite circularidade na rede ao inferir a orientação dos arcos que irão fazer parte da RB gerada, ou seja, a ordem de disposição das variáveis é um ponto crucial para o aprendizado adequado da estrutura da RB.

Como é possível perceber pela análise da (Equação 7), o processo de aprendizagem da estrutura de uma RB demanda grande quantidade de cálculos matemáticos. Parte do seu funcionamento consiste na busca pelas frequências das correlações entre os estados dos atributos do domínio, que é representado pelo N_{ij} e N_{ijk}, onde: i corresponde à variável estudada; j é a variável de correlação; e, k são os estados dos atributos que serão pontuados na RB. Esse mecanismo percorre todas as combinações dos registros dos atributos na base de dados, a fim de obter as frequências correspondentes de cada relação para cada estado de cada atributo da RB.

TRABALHOS CORRELATOS

Em (SIGH e SIGH, 2012), faz-se uma análise do crescimento da prática do uso do conceito de BigData, apresentando os benefícios de sua utilização, segundo estudos realizados pelo *McKinsey Global Institute* (MGI)[3] em cinco áreas de domínio. Os estudos mostram que a utilização de BigData pode aumentar a margem operacional de empresas e setores do governo impulsionando a eficiência e melhorando a qualidade.

Além disso, um levantamento realizado pela IBM® confirma que a maioria das organizações está atualmente nos estágios iniciais de esforços de desenvolvimento em BigData e visam melhorar a experiência de seus clientes, pelo entendimento de suas preferências e análise comportamental. Destaca-se a necessidade por técnicas capazes de encontrar padrões em grandes conjuntos de dados, de modo que seja possível extrair o máximo de conhecimento de grandes volumes de dados (IBM Corporation, 2012).

Neste sentido, o uso do MapReduce (MR) tem sido amplamente proposto, dentre outras aplicações, como uma alternativa que permite o desenvolvimento de aplicações de processamento paralelo escaláveis, capazes de lidar com grandes volumes de dados em grandes clusters (SHIM, 2012). Dentre os diversos trabalhos voltados a este processo, em especial os que tratam da melhora com relação a otimização do tempo e eficácia dos modelos, é possível destacar os seguintes:

Em (BASAK et al., 2012a), (BASAK et al., 2012b) e (REED e MENGSHOEL, 2012) propõe-se uma estrutura de computação distribuída baseada em MapReduce, Hadoop, para a aprendizagem de parâmetros (TPC) em RB utilizando o algoritmo de Expectation Maximization (EM).

(BASAK et al., 2012a), apresenta duas implementações para a aprendizagem de parâmetros clássica, que consiste em etapas de busca e contagem, uma para dados completos e outra para dados incompletos. Em ambas as análises o processo de busca em frequência foi substituído pelo processo de MR.

---

[3] http://www.mckinsey.com/features/big_data

Todo o processo foi implementado em um ambiente em nuvem (Amazon EC2) e o desempenho dos algoritmos testado em diversos benchmarks a fim de comprovar a eficácia dos resultados. Neste artigo, o autor apresenta análises que comprovam a redução significativa do tempo de processamento de uma média de 2h30min para apenas 15min.

Em (MA et al., 2012), investiga-se o uso de MapReduce para inferências exatas em Redes Bayesianas com plataformas *multicore*. Propõem-se algoritmos de propagação de evidências em árvores de junção utilizando MR, a fim de explorar a tarefa de paralelismo e solucionar o problema de dependência de dados baseados em restrições nos métodos transversais de árvores.

Em (FANG et al., 2013), o foco do trabalho consiste no aprendizado da estrutura da RB a partir de dados massivos; propõe-se uma abordagem baseada em MR, Hadoop, para a aprendizagem da RB pela aplicação do algoritmo tradicional de busca e pontuação K2. Além disso, apresenta-se uma função de pontuação que pode ser utilizada para computar de maneira fácil as TPCs para cada nó na RB, através do processo de MapReduce, que é usado para obter os parâmetros em paralelo. Entretanto, esta tarefa requer grande quantidade de interações na base de dados, o que demanda grande custo computacional devido à complexidade dos cálculos necessários para a execução do algoritmo.

A maioria dos estudos anteriormente citados possui a estrutura da Rede Bayesiana previamente definida por um especialista do domínio, o que reduz de maneira considerável o tempo de aprendizagem dos parâmetros da rede. Este trabalho propõe a otimização do algoritmo de aprendizagem da estrutura da RB em BigData usando o processo de MapReduce.

O MR é aplicado à base durante o processo de KDD, logo após a etapa de transformação, criando uma Base de Dados Intermediária (BDI), que nada mais é do que uma grande TPC contendo as probabilidades de todas as possíveis estruturas da RB. O intuito desta etapa consiste em reduzir o custo computacional associado ao cálculo das probabilidades necessárias para a aprendizagem da estrutura da rede por meio de simples consultas.

ESTUDO DE CASO

A análise das características presentes em um conjunto de dados permite a descoberta de padrões e tendências que podem fornecer informações valiosas que ajudem a compreender o processo que gerou os dados. Muitas dessas características podem ser obtidas por meio da aplicação de fórmulas estatísticas simples. Outras podem ser observadas por meio do uso de técnicas de visualização (FACELI et. al., 2011).

Conhecer o tipo dos dados com o qual se irá trabalhar também é fundamental para a escolha do(s) método(s) mais adequado(s). Pode se categorizar os dados em dois tipos: quantitativos e qualitativos. Os dados quantitativos são representados por valores numéricos. Eles ainda podem ser discretos e contínuos. Já os dados qualitativos contêm os valores nominais e ordinais (categóricos). Em geral, antes de se aplicar os algoritmos de mineração é necessário explorar, conhecer e preparar os dados (CAMILO e SILVA, 2009).

Para o desenvolvimento deste estudo, foi utilizado um conjunto de dados obtido através de uma parceria com o Governo do Estado. Os dados fazem parte dos registros do órgão de Segurança Pública do Estado do Pará, composto pelas informações obtidas a partir de Boletins de Ocorrência (BOs) registrados na capital do ano de 2002 até o ano de 2008. No total, o conjunto de informações contabilizava 965.530 registros e 27 variáveis armazenados em 73 planilhas de dados.

Devido ao processo manual aplicado para a coleta das informações, a base de dados obtida inicialmente apresentava grande quantidade de inconsistências como registros vazios, variáveis consideradas irrelevantes para o tipo de análise desejado, duplicação de valores, falta de padronização e notação incorreta de registros. Em virtude deste problema foi aplicada uma etapa de pré-processamento que consistiu na seleção, limpeza e padronização da base.

A base de dados final obtida foi utilizada para o desenvolvimento de uma ferramenta que permitisse aos gestores do sistema realizar uma análise que permitissem uma visão ampla e detalhada de cenários criminalísticos, por meio do uso de técnicas de inteligência computacional que usa métodos e modelos computacionais probabilísticos. Entretanto, apesar dos resultados obtidos e com o advento do conceito de BigData, percebeu-se a necessidade do estudo e aplicação de novas técnicas que permitissem a obtenção de respostas ainda mais rápidas.

Nesse sentido, este trabalho surge como uma complementação ao trabalho anterior (FRANÇA, 2011), não mais focado no sistema, mas na melhora do tempo de resposta dos algoritmos de inteligência computacional ao aumento significativo do número de registros na etapa de Extração de Conhecimento.

4.1 SELEÇÃO DOS DADOS

Os atributos foram selecionados de modo a melhor representar o cenário para o entendimento dos eventos criminosos, classificando-se assim em basicamente três itens centrais:

- **Data:** composta por ano e mês;
- **Local de Ocorrência:** composto pelo local onde o crime ocorreu (exemplo: bar, escola etc.), pela rua ou perímetro (logradouro), pelo bairro e pela unidade policial que registrou o BO.
- **Tipo de Crime:** composto pelo crime e a classificação em que ele se enquadra.

Tabela 4.1 - Seleção dos atributos do Boletim de Ocorrência.

Boletim de Ocorrência				
Ident BO	No. do BOP	Mês		Ano
Cód. Unidade	Sigla		Nome Unidade	
Dados da Ocorrência				
Cód. Bairro	Ident Bairro		Nome Bairro	
Cód. Localidade	Tipo Localidade		Nome Localidade	
Cód. Logradouro	Ident Logradouro		Nome Logradouro	
Compl Endereço	Rua Secundária	Perímetro 1	Perímetro 2	Fundos
Cód. Loc Ocorrência		Nome Loc Ocorrência		
Cód. Mot. Determinante	Nome Mot. Determinante	Cód. Classe Motivo	Nome Classe Motivo	

Devido às limitações iniciais apresentadas para análise realizada em (FRANÇA, 2011), a base sofreu uma redução no número de registros, tendo sido selecionadas aleatoriamente um total de 65.000 tuplas para cada um dos anos da série, com exceção do ano de 2002 por conter apenas registros referentes ao mês de dezembro, assim, para este ano em questão, foram utilizados todos os registros existentes (aproximadamente 3.000 registros).

4.2 TRANSFORMAÇÃO

4.2.1 Limpeza e Padronização

Para o conjunto de dados em questão, foi aplicado um processo de remoção de valores nulos e correção da grafia incorreta, bem como a padronização de valores através de uma função de similaridade denominada *Levenshtein Distance*[4] e o uso de um dicionário de dados contendo os valores de referência a serem utilizados.

A aplicação deste processo permite que seja possível obter uma base consistente, na qual palavras de mesmo valor semântico passam a ter mesma grafia e valores incompletos ou nulos são removidos, melhorando assim a precisão e qualidade das análises.

Por exemplo, considerando que se deseja obter as freqüências de um determinado tipo de crime, sabemos que “Art. 157”, “Roubo”, “Art. 157 – Roubo” e “Roubo – Art. 157” representam o mesmo crime, entretanto, o computador analisará estas informações como sendo quatro tipos diferentes de crime, cada um com a sua freqüência associada. Nesse caso, se o usuário procurar por “Roubo”, o valor total de freqüência deste crime não será obtida, pois as outras variáveis, que também representam este crime, não foram contabilizadas.

4.2.2 Seleção e expansão da quantidade de registros

As análises realizadas neste estudo foram feitas utilizando-se 9 (nove) conjuntos de amostras, contendo: 10 mil, 50 mil, 100 mil, 500 mil, 1 milhão, 5 milhões, 10 milhões, 50 milhões e 100 milhões de registros, todos contendo oito atributos (ano, mês, local de ocorrência, logradouro, bairro, unidade policial, crime e classe do crime) discriminados a seguir (Tabela 4.2).

---

[4] http://www.cut-the-knot.org/do_you_know/Strings.shtml

Tabela 4.2 – Discriminação do número de atributos da base

Atributo	Quantidade de valores
Ano	7
Mês	12
Bairro	38
Local de Ocorrência	28
Logradouro	2.574
Unidade Policial	42
Crime	267
Classe do Crime	41

A seleção e expansão foram feitas de modo aleatório e visa mostrar a eficiência do algorito de acordo com a variação da quantidade de registros utilizada. O conjunto de dados escolhido serve, portanto, apenas para demonstrar a aplicabilidade em um cenário real, contendo grandes quantidades de atributos, conforme demonstrado na tabela acima (Tabela 4.2).

4.3 CRIAÇÃO DA BASE DE DADOS INTERMEDIÁRIA

O tempo de processamento dos dados aumenta de acordo com a complexidade da análise, a qual varia de acordo com o número de estados de cada atributo, a quantidade de possíveis combinações entre os atributos ou, de maneira geral, pensando no contexto de Big Data, a quantidade de registros existentes na base. Deste modo, não é indicado executar o processo de MapReduce durante a execução do algoritmo K2, pois o mesmo possui diversos cálculos combinatórios que tornariam a aplicação ineficiente.

Visando solucionar esta problemática, foi feita uma complementação do processo de KDD (Figura 4.1), logo após a etapa de transformação dos dados, na qual as informações já se encontram de forma organizada e prontas para o processamento.

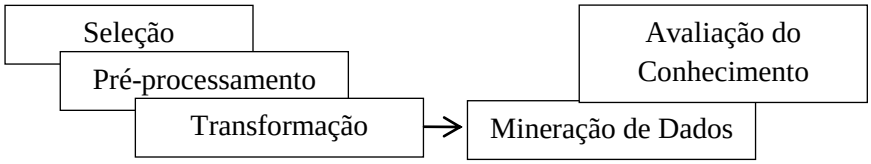

Figura 4.1 – Etapas do processo de descoberta de conhecimento em bases de dados

Esta complementação consiste na criação do que será chamado de “Base de Dados Intermediária” (Figura 4.2).

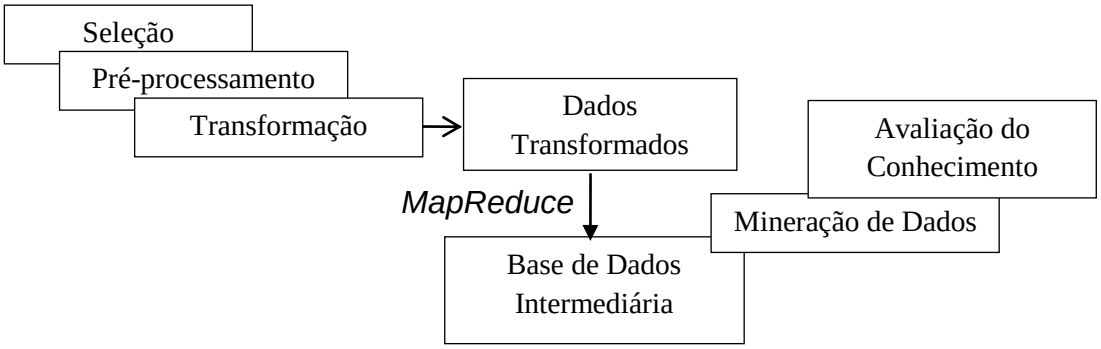

Figura 4.2 – Etapas do processo de KDD com a Base de Dados Intermediária.

A criação da BDI é feita antes que qualquer decisão sobre a ordem dos atributos seja definida, logo, não é possível prever como a BDI deve ser criada, portanto, sua criação será feita a partir de todas as combinações possíveis entre todos os estados de todas as variáveis presentes na base de dados. Sendo assim, essa base consiste no resultado do processamento da base original, utilizando a técnica de MapReduce e contém todas as informações necessárias para a execução do K2, abstraindo a complexidade das buscas em frequência utilizadas pelo algoritmo, substituindo-as por simples consultas na base de análise.

4.4 CENÁRIO DE ANÁLISES

4.4.1 Ambiente de Análises

As análises foram feitas utilizando o servidor *Amazon Elastic Compute Cloud* (Amazon EC2), que permite ao usuário criar instâncias de máquinas com uma variedade de sistemas operacionais de acordo com a necessidade.

Foram utilizadas instâncias de uso geral, as quais oferecem equilíbrio de recursos de computação, memória e rede. Deste modo, o ambiente utilizado para as análises possui configuração a seguir:

- **Máquina:** *m1.small* – 1,7 GiB de memória, 1 unidade de processamento EC2 (1 núcleo virtual com 1 unidade de processamento EC2), 160 GB de armazenamento de instância local, plataforma de 32 ou 64 bits. Processadores Intel Xeon.

- **Sistema:** Amazon Linux AMI 2013.09 (ESB) Linux 3.4; Kernel ID: Default; Ram ID: Default.
- **Banco de Dados:** MongoDB 2.2 com 2000 IOPS.

4.4.2 Cenários de arquitetura de banco de dados

Foram utilizados 5 (cinco) cenários para as simulações variando a quantidade de instâncias utilizadas para todos os conjuntos de amostras descritos no item 4.2.2. Assim tem-se que o cenário 1 possui uma única instância (*Standalone*) (Figura 4.3), e os demais cenários possuem duas, três, quatro e cinco instâncias, respectivamente, conforme apresentado na Figura 4.4.

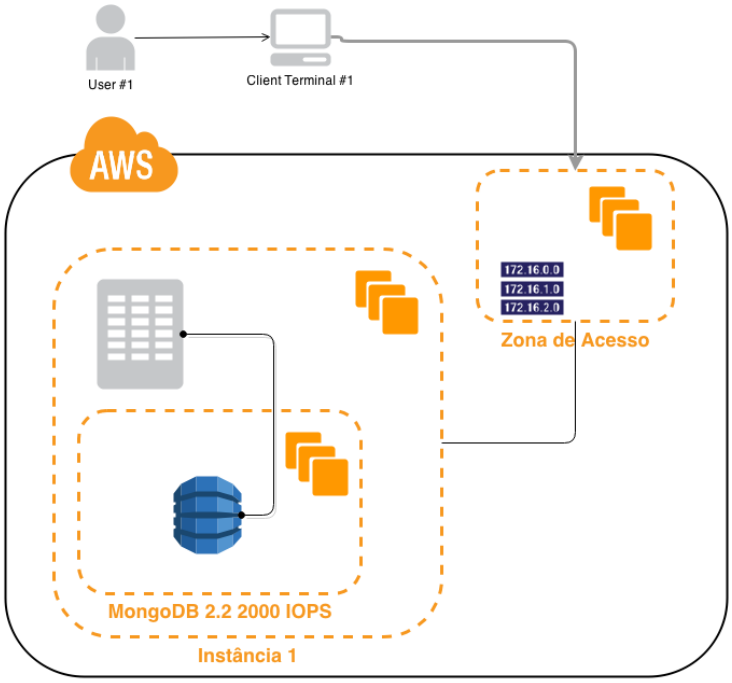

Figura 4.3 - Esquema representativo para uma única instância (cenário 1).

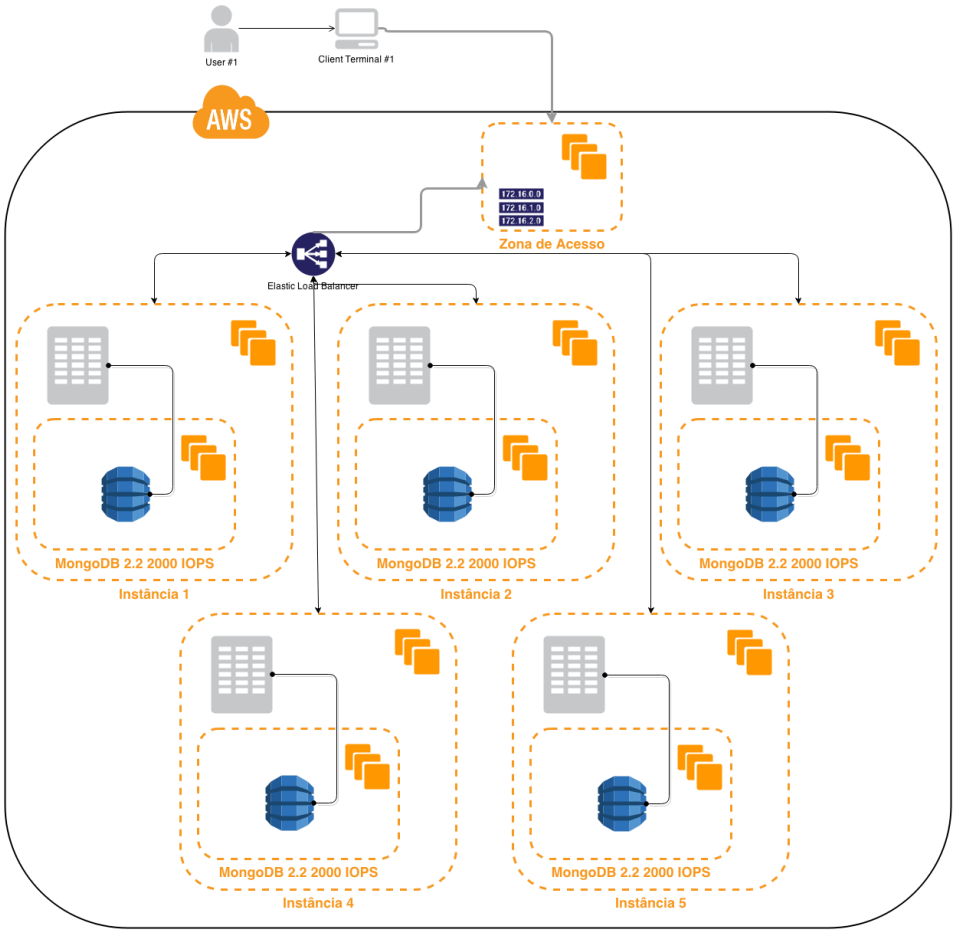

Figura 4.4 - Esquema representativo para mais de uma instância (cenários de 2 a 5).

De modo geral, há sempre uma instância com um banco MongoDB instalado, o qual fica em *sharding* na existência de outras instâncias, e um balanceador de carga (*Elastic Load Balancer*) que age equilibrando o tráfego das informações, de modo que, quando a distribuição de uma coleção em *sharding* de um cluster é desigual, o balanceador de carga migra fragmentos do bloco maior para os menores até que a coleção esteja equilibrada, além disso, atua na maneira como as instâncias se comunicam. A zona de acesso é o endereço público da cloud.

A diferença do primeiro cenário para os demais está na simplicidade, não havendo necessidade de balanceador de carga no tráfego interno, haja vista que não há demanda para a comunicação entre instâncias.

4.5 ANÁLISE DE RESULTADOS

4.5.1 Criação da TPC

A primeira análise foi feita com relação comportamento dos *jobs* de MapReduce, o intuito desta é demonstrar que com o aumento da quantidade de operações utilizadas há ganho no desempenho na criação da TPC da Rede Bayesiana (Tabela 4.3). Os resultados foram aplicados a um conjunto de dados contendo 1 milhão de registros e um atributo.

Tabela 4.3 - Análise de comportamento de jobs do MapReduce.

Quantidade de Jobs	Tempo de Map (segundos)	Tempo de Reduce (segundos)	Tempo Total de Execução (segundos)
1	1000	88	1200
2	580	160	650
10	220	190	360
20	150	200	355
50	80	187	320
100	81	220	410
500	90	301	490
1000	108	310	556

Apesar dos ganhos visíveis, foi possível observar também que o aumento da quantidade de jobs de MapReduce não pode ser feito de maneira indiscriminada, haja vista que o aumento da quantidade de operações de Map gera mais trabalho, tanto para particionar e mapear a base, quanto para reagrupar os dados na etapa de Reduce; causando queda no desempenho geral, dada a quantidade de registros analisada. Sendo assim, esta etapa deve ser feita para cada caso de aplicação em específico, visando otimizar o tempo desta operação para o conjunto estudado.

Essa conclusão é mais perceptível por meio da análise gráfica dos dados apresentados na Tabela 4.2, onde claramente percebe-se o aumento do tempo a partir da utilização de 50 jobs de MapReduce para o caso de estudo (Figura 5.1).

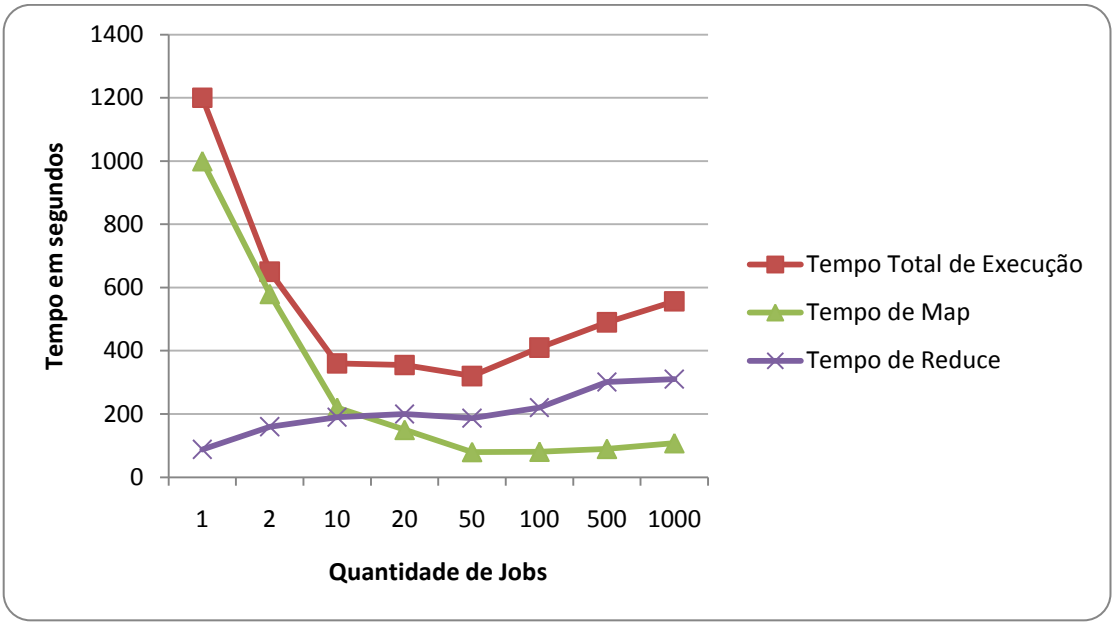

Figura 4.5 - Tempo médio de variação do MapReduce para 1 milhão de registros.

Outra análise realizada foi com relação ao tempo demandado para persistir os registros em um BD, com o intuito de demonstrar a evolução do custo computacional associado a esta tarefa (Figura 5.2). Para isso, foram utilizados os dados descritos na seção 4.2.2, conforme (Tabela 4.4).

Tabela 4.4 - Tempo de DUMP[5] em segundos.

Quantidade de registros	Tempo por atributo (segundos)	Tempo médio de todos os atributos (segundos)
10 mil	3	6
50 mil	7	11
100 mil	18	25
500 mil	31	40
1 milhão	42	59
5 milhões	98	146
10 milhões	114	199
50 milhões	165	241
100 milhões	213	298

[5] Exportação dos dados em um BD.

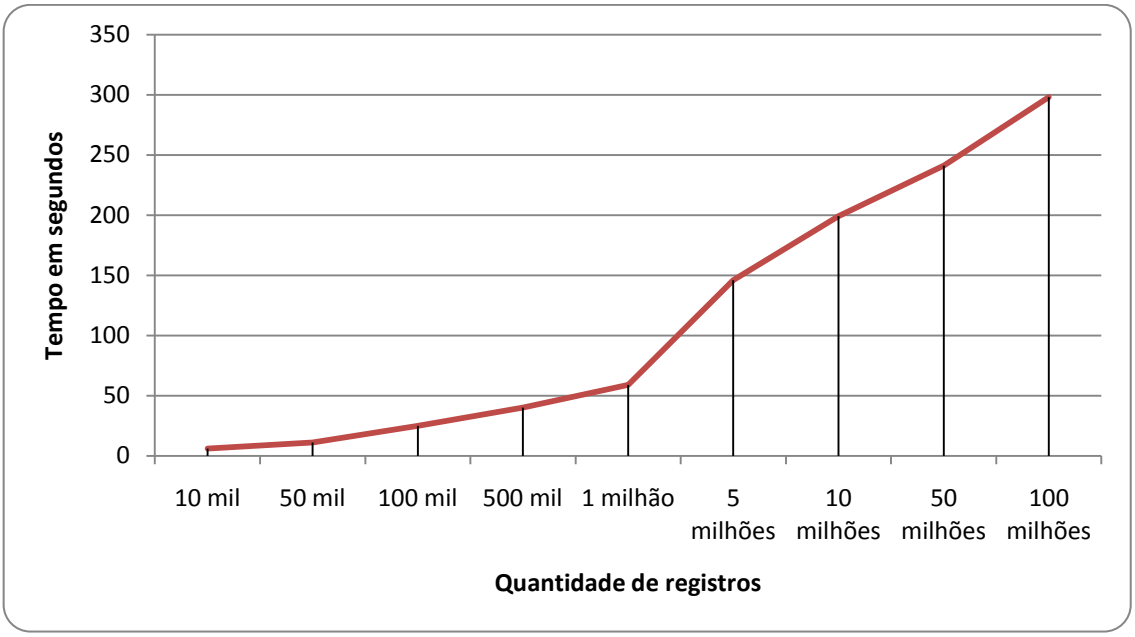

Figura 4.6 - Tempo médio de DUMP variando a quantidade de registros.

A terceira análise realizada avalia o desempenho das operações de MapReduce nos diferentes cenários descritos na seção 4.4 para a criação da TPC da RB. Desta análise percebe-se que o aumento do número de instâncias (*sharding*) há ganho no desempenho da geração das TPCs. As tabelas de 4 a 8 apresentam os resultados dessa análise, considerando dois aspectos: Tempo médio para um atributo; e tempo médio com todos os aributos.

Todos os valores de tempo descritos a seguir para as tabelas dos cenários de 1 a 5 foram apresentados na escala de segundos.

Tabela 4.5 - Análise do aumento da quantidade de registros utilizando uma instância *small*.

Cenário 1		
Quantidade de registros	**Tempo Médio/Atributo**	**Tempo Médio/Todos os Atributos**
10 mil	26.35	89.19
50 mil	65.76	156.32
100 mil	123.62	304.43
500 mil	231.23	567.45
1 milhão	346.10	879.87
5 milhões	897.98	1574.18
10 milhões	1023.12	2341.43
50 milhões	1327.54	3019.83
100 milhões	1987.24	3976.87

Tabela 4.6 - Análise do aumento da quantidade de registros utilizando duas instâncias *small.*

Cenário 2

Quantidade de registros	Tempo Médio/Atributo	Tempo Médio/Todos os Atributos
10 mil	19.62	81.43
50 mil	52.12	152.18
100 mil	103.09	298.90
500 mil	201.76	542.81
1 milhão	332.87	856.01
5 milhões	708.09	1542.87
10 milhões	998.17	2198.90
50 milhões	1198.11	2987.91
100 milhões	1799.98	3854.01

Tabela 4.7 - Análise do aumento da quantidade de registros utilizando três instâncias *small.*

Cenário 3

Quantidade de registros	Tempo Médio/Atributo	Tempo Médio/Todos os Atributos
10 mil	15.01	65.01
50 mil	47.62	142.23
100 mil	98.91	285.81
500 mil	178.29	538.98
1 milhão	308.78	798.64
5 milhões	689.01	1374.91
10 milhões	879.98	1976.93
50 milhões	1028.21	2830.11
100 milhões	1687.17	3659.03

Tabela 4.8 - Análise do aumento da quantidade de registros utilizando quatro instâncias *small*.

Cenário 4

Quantidade de registros	Tempo Médio/Atributo	Tempo Médio/Todos os Atributos
10 mil	8.0	43.1
50 mil	32.91	122.11
100 mil	76.88	265.00
500 mil	165.10	514.23
1 milhão	287.99	771.94
5 milhões	572.31	1203.67
10 milhões	765.45	1639.91
50 milhões	983.81	2357.64
100 milhões	1549.92	3321.82

Tabela 4.9 - Análise do aumento da quantidade de registros utilizando cinco instâncias *small*.

Cenário 5		
Quantidade de registros	**Tempo Médio/Atributo**	**Tempo Médio/Todos os Atributos**
10 mil	3.01	22.01
50 mil	25.88	101.17
100 mil	68.62	210.92
500 mil	124.61	441.12
1 milhão	210.01	653.01
5 milhões	439.11	1190.89
10 milhões	612.98	1437.32
50 milhões	843.78	1988.91
100 milhões	978.26	2989.11

Como na primeira análise realizada, percebeu-se que a partir de uma certa quantidade de instâncias pode haver queda na qualidade do tempo de resposta (Figura 5.3), já que devido a alta transmissão de dados poderá haver instabilidade na infraestrutura de rede, o que gera ruídos que podem afetar os valores analisados.

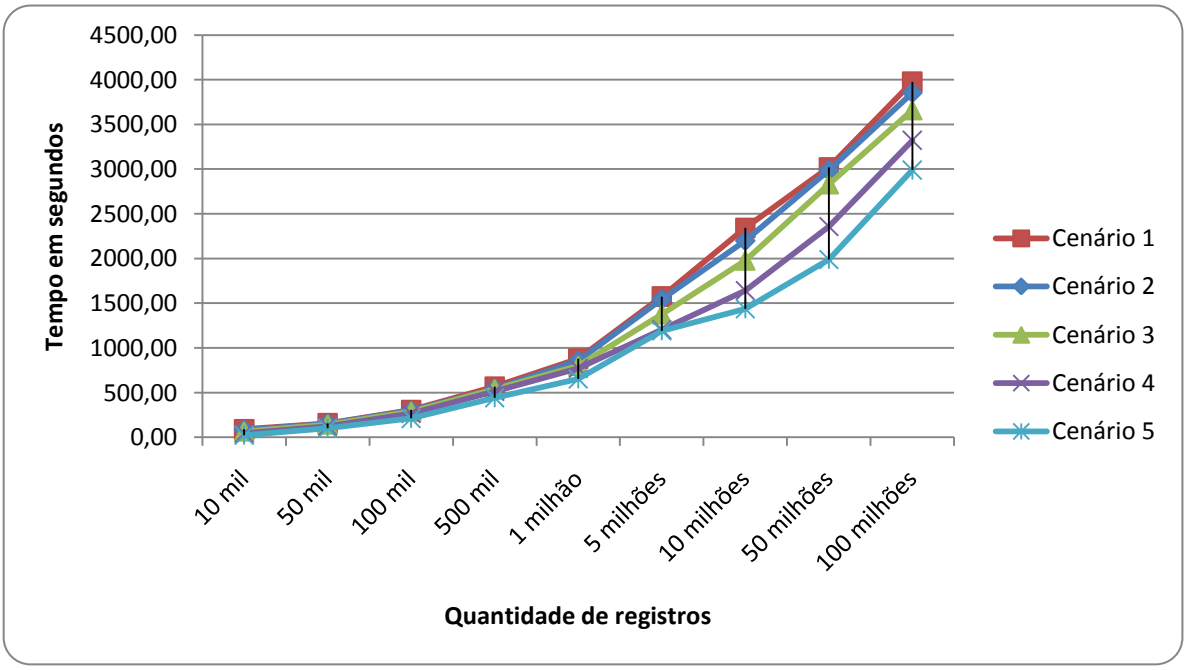

Figura 4.7 - Análise da variação do número de instâncias.

4.5.2 Aprendizagem da Estrutura da RB

As análises anteriores apenas discutiam a etapa de criação da TPC, o MapReduce foi aplicado à base de dados com a finalidade de encontrar todas as possíveis

combinações entre os estados dos atributos com os seus possíveis pais, com isso gerou-se uma BDI (em analogia ao que seria uma TPC de grande dimensão) onde serão feitas as consultas necessárias para a aplicação da fórmula do algoritmo de pontuação da qualidade da estutura da RB, neste caso o *score* bayesiano usado no algoritmo K2 (Equação 7).

Esta etapa já representa um enorme ganho no processo como um todo, no entanto, com a grande quantidade de dados estatísticos gerados devido ao número de possíveis combinações, faz-se necessário ainda otimizar o processo de busca para que a RB possa ser gerada com o melhor desempenho possível.

A solução proposta trabalha a indexação dos dados para garantir que o processo de consulta ocorra muito mais rápido no banco de dados. Nesse sentido, foram feitas análises que apresentam o tempo de aprendizagem da estrutura da RB usando o K2-MR nos cenários apresentados na seção 4.4.

Vale ressaltar que a quantidade de registros representa o tamanho da base de análise inicial e não o tamanho da BDI, ou seja, para cada base descrita na seção 4.2.2 foi gerada uma BDI correspondente a TPC da rede e, a partir dela, realizadas as análises de indexação (Tabela 4.10).

Tabela 4.10 - Análise do tempo para a aprendizagem da estrutura da RB.

Quantidade de registros	**Tempo Médio (segundos)**				
	Cenário 1	**Cenário 2**	**Cenário 3**	**Cenário 4**	**Cenário 5**
10 mil	218.09	178.94	102.45	94.09	43.01
50 mil	691.28	542.89	508.13	321.45	201.12
100 mil	876.26	819.23	672.86	409.87	342.56
500 mil	1176.54	989.28	763.04	634.91	487.32
1 milhão	1889.23	1324.65	871.20	710.98	510.53
5 milhões	1999.65	1498.19	976.90	798.64	598.76
10 milhões	2111.45	1786.90	1003.45	819.23	635.98
50 milhões	2256.84	1965.09	1109.67	993.09	784.87
100 milhões	2456.87	2019.21	1233.38	1109.25	987.64

A Figura 5.4 apresenta de maneira gráfica os resultados obtidos, facilitando o entendimento das análises, onde percebe-se claramente a redução do tempo de um cenário ao outro.

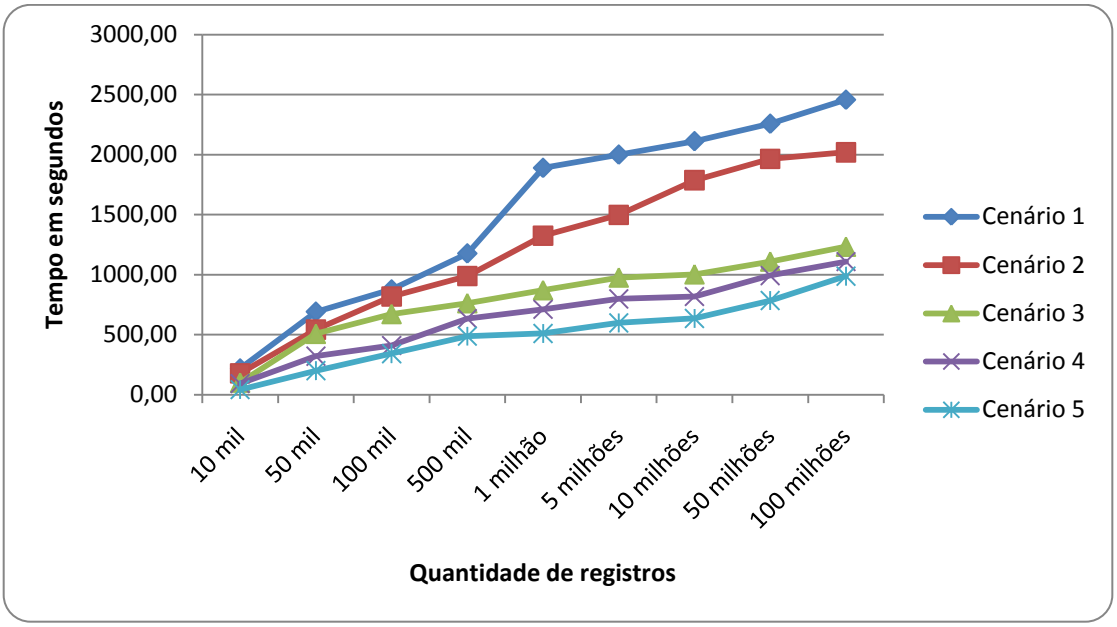

Figura 4.8 - Análise da variação do número de instâncias para a aprendizagem da estrutura da RB.

Por fim, visando ressaltar a influência do processo de indexação na base apresentado anteriormente (Figura 5.4), foi realizada uma comparação entre dois cenários (Tabela 4.11).

Tabela 4.11 - Análise comparativa do uso de indexação na base de dados.

Cenário 1		
Quantidade de registros	**Tempo Médio (segundos)**	
	Com Indexação	**Sem Indexação**
10 mil	218.09	432.09
50 mil	691.28	691.28
100 mil	876.26	1098.32
500 mil	1176.54	2451.79
1 milhão	1889.23	2981.02
5 milhões	1999.65	3897.34
10 milhões	2111.45	4182.23
50 milhões	2256.84	5623.10
100 milhões	2456.87	5987.23

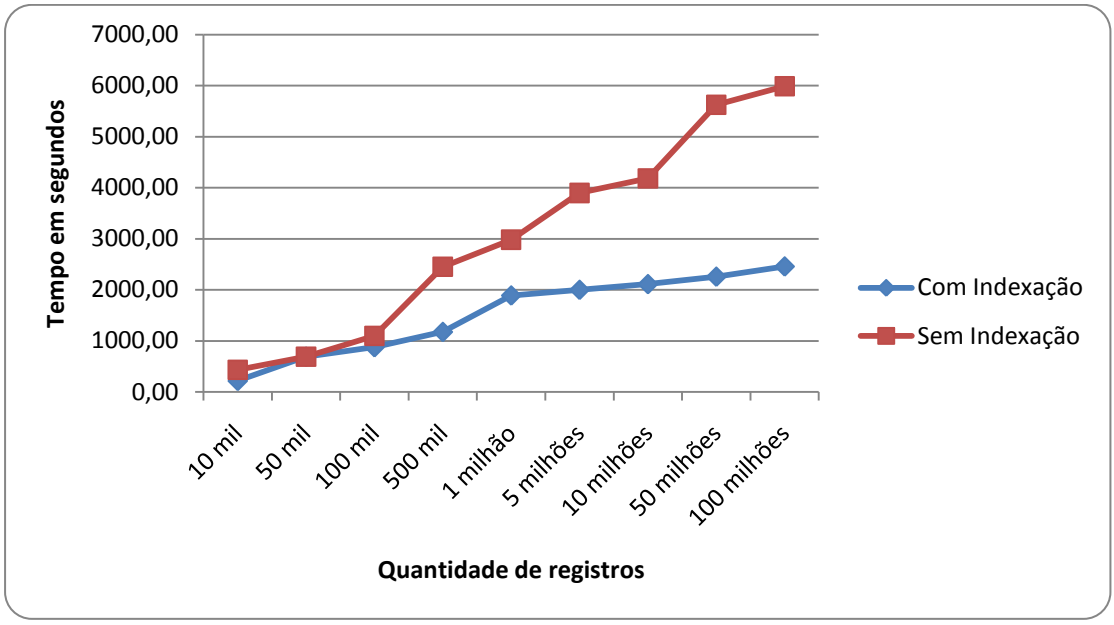

Figura 4.9 – Gráfico comparativo do uso de consulta indexada *versus* não-indexada.

Essa análise prova que o processo de busca simples na BDI é um processo custoso se comparado com o processo de busca indexado, como podemos perceber pela análise da Figura 5.5, que permite visualizar o comportamento dos dados descritos na Tabela 4.11.

CONCLUSÕES

Várias organizações estão preocupadas com o fato de que a quantidade de dados gerados está se tornando tão grande que se torna difícil encontrar a informação mais valiosa, mas a verdadeira pergunta é como estas empresas podem tirar vantagem dos dados que são relevantes e usar o conhecimento extraído para tomar melhores decisões e, consequentemente, tornarem-se mais competitivas no mercado.

Visando facilitar o entendimento das relações existentes entre as diversas variáveis que podem existir em um grande conjunto de dados, optou-se neste trabalho pelo uso de Redes Bayesianas para extração e representação do conhecimento. Diversos algoritmos podem ser usados para criar a estrutura da Rede Bayesiana, dentre os quais está o K2, que será usado neste trabalho para este propósito devido a complexidade associada.

O algoritmo K2 possui complexidade $O(m \cdot u^2 \cdot n^2 \cdot r)$, isso implica dizer que durante o processo de criação da estrutura da RB ocorrem várias interações na base de dados, as quais buscam pela frequência dos estados dos atributos. Devido a essa complexidade, quanto maior o número de registros existentes e, principalmente, a quantidade de atributos estudados, maior a complexidade associada.

A maioria dos algoritmos de aprendizado utiliza algum tipo de *score* ou análise probabilística, portanto, assim como o algoritmo analisado neste trabalho, demandam computação extensiva para os cálculos matemáticos associados.

Dado o exposto acima, o objetivo deste trabalho foi mostrar que o uso do processo de MapReduce é uma boa alternativa para resolver o problema de escalabilidade no processo de busca em frequência do algoritmo K2, bem como dos demais algoritmos de aprendizagem existentes, os quais podem usufruir dos aspectos encontrados neste trabalho. A busca em frequência, como apresentado nos capítulos anteriores, corresponde ao processo de maior custo computacional e requer maior tempo de processamento quanto maior o conjunto de dados.

Nesse sentido, foram realizados experimentos que buscassem comprovar este ganho de eficiência com relação ao tempo de processamento consumido para a aprendizagem da estrutura da rede. Os resultados apresentados mostram que é possível gerar a estrutura da Rede Bayesiana em tempo satisfatório com o algoritmo

K2 utilizando o processo de MapReduce para a criação de uma BDI e consultas indexadas a esta base.

5.1 DIFICULDADES ENCONTRADAS

Ao observarmos a enorme quantidade de interações e cálculos demandados pelos algoritmos de DM, percebemos que, de um modo geral, eles não foram feitos para trabalhar com BigData, portanto, escalá-los é uma tarefa árdua. Muito se fala sobre ferramentas, mas pouco sobre a interação delas com os algoritmos de DM, por isso, a falta de trabalhos que esclareçam de maneira mais objetiva como foram superadas as principais dificuldades com o BigData, foi uma das principais dificuldades encontradas no início deste trabalho.

O algoritmo escolhido apresenta alto custo computacional e a quantidade de simulações necessárias para a comprovação da proposta abordada gerou diversos retrabalhos.

Além disso, diversos problemas de ordem física contribuíram para atrasos no desenvolvimento geral deste trabalho, um dos fatores principais foram os problemas de infraestrutura local para trabalhar com a quantidade de dados utilizada, desde a montagem do ambiente até a configuração das máquinas. Muitas vezes os processos eram interrompidos antes da finalização, sendo necessário refazer todo o processamento. Esse fator foi o que impulsionou os estudos para a utilização de ambientes em cloud, entretanto, a falta de *know-how* nesta área dificultou bastante a montagem do ambiente de estudos.

5.2 PRINCIPAIS CONTRIBUIÇÕES

- Adaptação do algoritmo de aprendizagem da rede bayesiana K2 para trabalhar com grande volume de dados, como já foi supracitado, grande parte dos algoritmos de DM não estão prontos para trabalhar com BigData, devido seu processo iterativo;

- Redução de grande parte do custo computacional para executar o algoritmo bayesiano sobre uma base de dados extensa, através do uso da técnica proposta;

- Um avanço tencnológico em relação às técnicas de mineração de dados iterativas que necessitem de alto processamento sobre bases de dados extensas;

- Possibilidade de aplicação da técnica a problemas reais que envolvem análises complexas e grandes volumes de dados, haja vista que os experimentos foram realizados sobre uma base de dados de segurança pública real a fim de apresentar o potencial da técnica proposta.

5.3 ARTIGOS PUBLICADOS

FRANÇA, A. S. ; LIMA, J. G. ; JACOB JR, A. ; SANTANA, Á. L. Learning the Bayesian Structure in BigData using the K2 Algorithm with MapReduce. In: World Congress on Systems Engineering and Information Technology, 2013, Porto. Proceedings of the 2013 World Congress on Systems Engineering and Information Technology, 2013.

5.4 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Soluções como a proposta neste trabalho vêm a suprir uma grande demanda de mercado, que transborda os limites acadêmicos, onde há a necessidade soluções que ofereçam análises mais robustas para problemas com variáveis complexas.

Estudos futuros consistem na aplicação das técnicas apresentadas neste trabalho em modelos de *benchmarks*, de modo a avaliar também a qualidade da estrutura da RB gerada. Além disso, é possível aprofundar os estudos em computação em nuvem, explorando arquiteturas de armazenamento de dados e processo de aprendizagem da estrutura da Rede Bayesiana.

REFERÊNCIAS

AGGARWAL, N.; KUMAR, A.; KHATTER, H.; AGGARWAL, V., “Analysis the effect of data mining techniques on database”, *Advances in Engineering Software*, v. 47(1), p. 164-169, 2012.

AMORIM, T. 2006. Conceitos, técnicas, ferramentas e aplicações de Mineração de dados para gerar conhecimento a partir de base de dados. Disponível em: < http://www.cin.ufpe.br/~tg/2006-2/tmas.pdf > Acesso em 2013.

BAKSHI, K. 2012. Considerations for big data: Architecture and approach. *Aerospace Conference, 2012 IEEE*, vol., no., pp.1,7, 3-10. doi: 10.1109/AERO.2012.6187357

BASAK, Aniruddha; BRINSTER, Irina; MA, Xianheng; MENGSHOEL, Ole J. 2012a. Accelerating Bayesian network parameter learning using Hadoop and MapReduce. *In Proceedings of the 1st International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications*, BigMine ’12, pages 101–108, New York, NY, USA, Agosto de 2012. ACM.

BASAK, Aniruddha; BRINSTER, Irina; MENGSHOEL, Ole J. 2012b. MapReduce for Bayesian Network Parameter Learning using the EM algorithm. *Proc. of Big Learning: Algorithms, Systems and Tools,* Dezembro de 2012.DEAN, J; GHEMAWAT, S. 2004. Mapreduce: simplified data processing on large clusters. In P*roceedings of the 6th conference on Symposium on Opearting Systems Design & Implementation - Volume 6*, pages 10–10, Berkeley, CA, USA. USENIX Association.

BOICEA, A.; RADULESCU, F.; AGAPIN, L.I.. 2012. MongoDB vs Oracle -- Database Comparison. *Emerging Intelligent Data and Web Technologies (EIDWT), Third International Conference*, pp.330,335, 19-21. doi: 10.1109/EIDWT.2012.32

BONNET, L.; LAURENT, A.; SALA, M.; LAURENT, B.; SICARD, N., 2011. Reduce, You Say: What NoSQL Can Do for Data Aggregation and BI in Large Repositories. *Database and Expert Systems Applications (DEXA), 22nd International Workshop*, pp.483,488, Aug. 29 2011-Sept. 2 2011. doi: 10.1109/DEXA.2011.71

CAMARINHA, M. M. O. *Auditoria na Banca Utilizando Redes Bayesianas*. Dissertação (Mestrado) – Universidade do Porto, 2009.

CHAUDHURI, S. 2012. How Different is Big Data?, *Data Engineering (ICDE), 2012 IEEE 28th International Conference on* , vol., no., pp.5,5, 1-5. doi: 10.1109/ICDE.2012.153

CHEDE, Cezar. 2012. Big Data = volume+variedade+velocidade de dados. *DeveloperWorks Brasil*, *IBM*. Disponível em < https://www.ibm.com/developerworks/mydeveloperworks/blogs/ctaurion/entry/big_data_volume_variedade_velocidade_de_dados?lang=en>. Acesso em Fevereiro/2013.

CHEN, Z. *Data Mining and Uncertain Reasoning - an Integrated Approach*, John Wiley Professional, 2001.

CHEN, X; ANANTHA, G.; LIN, X., "Improving Bayesian Network Structure Learning with Mutual Information-Based Node Ordering in the K2 Algorithm," *IEEE Transactions on Knowledge and Data Engineering,* v.20(5), p.628-640, 2008.

CHU, C., KIM, S.K., LIN, Y., YU, Y., BRADSKI, G., Ng, A., OLUKOTUN, K. 2006. Map-Reduce for Machine Learning on Multi-core, *Neural Information Processing Systems*, pp. 281–288, MIT Press.

CIOS, K. J; PEDRYCZ, W; SWINIARSKI, R. W; KURGAN, L. A. Data Mining – A Knowledge Discovery Approach. Springer, 2007.

COOPER, G. F.; HERSKOVITS, E. H., “A Bayesian Method for Constructing Bayesian Belief Networks from Databases”. In *Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence (UAI1991)*. Pp. 86, 94, Jul 13-15, 1991.

COOPER, G. F.; HERSKOVITZ, E. A Bayesian method for induction of probabilistic networks from data. Machine Learning, 9, 309-347, 1992.

D’ANDREA, Edgar. Big Data. 2010. *Revista Informationweek*, *pag. 38*, *coluna de segurança*, Brasil. http://www.pwc.com.br/pt/sala-de-imprensa/assets/artigo-big-data.pdf

DEMIRKAN, H.; DELEN, D., "Leveraging the capabilities of service-oriented decision support systems: Putting analytics and big data in cloud", *Decision Support Systems*, Available online 29 May 2012.

DIANA, M.; GEROSA, M. A. NOSQL in Web 2.0: A Comparative Study of Databases for Non-Relational Data Storage for Web 2.0 (in portuguese). Brazilian Symposion of Databases, 2010.

FACELI, K. et. al. Inteigência Artificial: Uma Abordagem de Aprendizagem de Máquina. Editora LTC, Rio de Janeiro, 2011.

FAN, J.; LIU, H., "Statistical analysis of big data on pharmacogenomics", *Advanced Drug Delivery Reviews*, Available online 17 April 2013.

FANG, Q.; YUE, K.; FU, X; WU, H.; LIU, W. A MapReduce Based Method for Learning Bayesian Network from Massive Data. *Lecture Notes in Computer Science*, v.7808, p. 697-708, 2013

FAYYAD, U; PIATETSKY-SHAPIRO, G; SMYTH, P. From Data Mining to Knowledge Discovery in Databases. American Association for Artificial Intelligence, 1996.

GILLICK, D., FARIA, A., DENERO, J. 2006. MapReduce: Distributed Computing for Machine Learning, Berkeley.

GONÇALVES, A. R. *Fundamentos e aplicações de técnicas de aprendizagem de máquina*. Trabalho de Conclusão de Curso (Graduação) – Universidade Estadual de Londrina, Londrina, 2008.

HAN, J; KAMBER, M. Data Mining: Concepts and Techniques. Elsevier, 2006.

HECKERMAN, D. *Bayesian networks for Data Mining*. Data Mining and Knowledge Discovery, Kluwer Academic Publishers, 1, 79-119, 1997.

HU, Hao; LU Kai; LI, Gen, WANG, Xiaoping; XU, Tianye. 2012. CAAC: A Key-value Database Performance Boosting Algorithm. *Fourth International Conference on Computational and Information Sciences*, China.

IBM Corporation. 2012. The Real World Use of Big Data. Disponível em <http://www-03.ibm.com/systems/hu/resources/the_real_word_use_of_big_data.pdf>. Acesso em 2013.

JAYATHILAKE, D.; SOORIAARACHCHI, C.; GUNAWARDENA, T.; KULASURIYA, B.; DAYARATNE, T. 2012. A study into the capabilities of NoSQL databases in handling a highly heterogeneous tree. *Information and Automation for Sustainability (ICIAfS), 2012 IEEE 6th International Conference*, pp.106,111, 27-29. doi: 10.1109/ICIAFS.2012.6419890

KLEMETTINEN, M.; MANILLA, H.; RONKAIEN, P.; TOIVONEN, H.; VERKAMO, A. I., *Finding Interesting Rules from Large Sets of Discovered Association Rules*, Proc. of the Third Int'l Conf. on Information and Knowledge Management, Maryland, 1994.

KORB, K. B.; NICHOLSON, A. E. *Bayesian Artificial Intelligence*. Florida, Chapman & Hall/CRC, 2003.

LIU, Y.; WANG, Y.; JIN, Y., 2012. Research on the improvement of MongoDB Auto-Sharding in cloud environment. *Computer Science & Education (ICCSE), 7th International Conference,* pp.851,854, 14-17. doi: 10.1109/ICCSE.2012.6295203

MA, N.; XIA, Y.; PRASANNA, V. K. Parallel exact Inference on Multicore using MapReduce. *IEEE 24th International Symposium Computer Architecture and High Performance Computing*, p. 187-194, 2012.

MADDEN, Sam. 2012. From Databases to Big Data. *Internet Computing, IEEE, vol.16, no.3, pp.4,6*. doi: 10.1109/MIC.2012.50.

NEAPOLITAN, R. E. Learning Bayesian Networks. New Jersey: Prentice Hall, 2004. 674 p.

PEARL, J. *Probabilistic Reasoning in Intelligent System*, Morgan Kaufmann Publishers, 1988.

PÉREZ, M. S.; SÁNCHEZ, A.; ROBLES, V.; HERRERO, P.; PEÑA, J. M. Design and implementation of a data mining grid-aware architecture. Future Gener. Comput. Syst., Amsterdam, The Netherlands, The Netherlands, v.23, n.1, p.42–47, 2007.

PERRIER, E.; IMOTO, S.; MIYANO, S. Finding Optimal Bayesian Network Given a Super-Structure, 2008. Journal of Machine Learning Research 9 (2008) 2251-2286.

PIFER, A. C.; GUEDES, L. A. 2007. Aprendizagem Estrutural de Redes Bayesianas Utilizando Métricas MDL Modificada. *Revista IEEE América Latina*, v. 5, p. 1-8.

RAUTENBERG, Phillip L. A Data System for Electrophysiological Data, LNCS Transactions on Large-Scale Data and Knowledge- Centered Systems, vol. 6990, Jun. 2011, pp: 1-14,doi:10.1007/978-3-642-23740-9.

REED, Erik B.; MENGSHOEL, Ole J. Scaling Bayesian Network ParameterLearning with Expectation Maximization using MapReduce. *Proc. of Big Learning: Algorithms, Systems and Tools* (2012).

REZENDE, S. O. Mineração de Dados. XXV Congresso da Sociedade Brasileira de Computação, 2005.

SANTANA, A. L.; FRANCÊS, C. R. L.; TARSO, P.; COSTA, C. W. A; ENDO, P. T; KLAUTAU, A. B. R. *Um Retrato da Aplicação de Recursos da Saúde e seu Impacto no IDH dos Municípios do Estado do Pará*. XXXI SEMISH, 2004.

SANTANA, Ádamo Lima de. Projeto e Implementação de um Sistema de Suporte à Decisão Para o Observatório de Saúde da Amazônia. 2005. Dissertação (Mestrado em Engenharia

Elétrica) – Universidade Federal do Pará, Instituto de Tecnologia, Programa de Pós-graduação em Engenharia Elétrica, Belém.

SHIM, Kyuseok. 2012. MapReduce Algorithms for Big Data Analysis. *Proceedings of the VLDB Endowment, Vol. 5, No. 12. 38th International Conference on Very Large Data Bases*, August 27th - 31st 2012, Istambul, Turquia.

SINGH, Sachchidanand; SINGH, Nirmala. 2012. Big Data Analytics. *International Conference on Communication, Information & Computing Technology (ICCICT)*, Oct. 19-20. Mumbai, India.

TAN, P.; STEINBACH, M.; KUMAR, V. *Introduction to Data Mining*. Boston, Pearson Education, Inc. /Addison Wesley, 2006.

VEIGA, Samuel Carvalho Alencastro; SILVA, Wagner Teixeira da. Redes Bayesianas: Uma visão geral. Universidade de Brasília, Departamento de Ciência da computação. Brasília, 2002

VERMA, A.; LLORA, X.; VENKATARAMAN, S.; GOLDBERG, D.E.; CAMPBELL, R.H., 2010. Scaling eCGA model building via data-intensive computing. *Evolutionary Computation (CEC), 2010 IEEE Congress*, pp.1,8, 18-23. doi: 10.1109/CEC.2010.5586468

WANG, J. editor. Encyclopedia of DataWarehousing and Mining. Idea Group Reference, 2005.

WEI-PING, Z.; MING-XIN, L.; HUAN, C., 2011. Using MongoDB to implement textbook management system instead of MySQL. *Communication Software and Networks (ICCSN), IEEE 3rd International Conference*, pp.303,305, 27-29. doi: 10.1109/ICCSN.2011.6013720

WORLD ECONOMIC FORUM. 2012. Big Data, Big Impact: New Possibilities for International Development. Disponível em <http://www3.weforum.org/docs/WEF_TC_MFS_BigDataBigImpact_Briefing_2012.pdf>. Acesso em Fevereiro/2013.

WU, Xindong; ZHU, Xingquan; WU, Gong-Quing; DING, Wei. Data Mining with Big Data. 2014. *IEEE Transactions on Knowledge and Data Engineering*, *vol.26*, *no 1*.

ZHANG, Guigang; LI, Chao; ZHANG, Yong; XING, Chunxiao. 2012. DataCloud: An Efficient Massive Data Mining and Analysis Framework on Large Clusters. *Web Information Systems and Applications Conference (WISA)* pp.198,203, 16-18. doi: 10.1109/WISA.2012.26.