Extração de dados científicos para construção e análise de Redes Sociais

Alan de Paula Duque

JUIZ DE FORA

JUNHO, 2015

Extração de dados científicos para construção e análise de Redes Sociais

Alan de Paula Duque

Universidade Federal de Juiz de Fora

Instituto de Ciências Exatas

Departamento de Ciência da Computação

Bacharelado em Sistemas de Informação

Orientador: Victor Ströele de Andrade Menezes

JUIZ DE FORA

JUNHO, 2015

Extração de dados científicos para construção e análise de Redes Sociais

Alan de Paula Duque

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTEGRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE BACHAREL EM SISTEMAS DE INFORMAÇÃO.

Aprovada por:

Victor Ströele de Andrade Menezes
Dr. em Computação - UFRJ - RJ

Regina Maria Maciel Braga Villela
Dra. em Computação - UFRJ - RJ

Luciana Conceição Dias Campos
Dra. em Engenharia Elétrica - PUC - RJ

JUIZ DE FORA

24 DE JUNHO, 2015

Aos meus amigos e família, pelo apoio e dedicação.

Resumo

Este trabalho visa à aplicação de técnicas de Mineração de Dados a partir de uma base de dados relacional extraída da DBLP. Foi realizado um estudo do arquivo XML contendo os registros da base, elaborado um modelo entidade-relacionamento que a representa e, finalmente, o *parsing* deste arquivo, populando o BD elaborado. De posse desta base relacional, foi aplicada uma técnica de Mineração de Dados, visando descobrir redes sociais científicas baseadas em relacionamentos de coautoria entre pesquisadores de diferentes instituições pelo país.

Palavras-chave: DBLP; Mineração de Dados; Banco de dados; Redes Sociais; Extração de Dados

Abstract

This work targets the application of Data Mining techniques on a relational database built from the DBLP. A reserach was made on the XML containing the database registers, as entity-relationship model was built, representing the DBLP database and, finally, the XML was parsed, filling the created DB. Upon the posession of this relational database, a Data Mining technic was applied to unfold cientific social networks based on coauthorship relationships between researchers form different institutions countrywide.

Keywords: DBLP; Data Mining; Database; Social Network; Data Extraction

Agradecimentos

Agradeço a todas as pessoas que, de alguma forma, contribuíram para que esta realização fosse possível.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviações

DBLP *Digital Bibliography Library Project*

1 Introdução

1.1 Apresentação do tema e contextualização do problema

Com o aumento da interatividade e serviços disponíveis on-line, a Mineração de Dados tem se tornado de suma importância para a tomada de decisões estratégicas. Temos bases de dados cada vez maiores e precisamos de uma ferramenta eficiente para "minerar" essa informação bruta em busca de dados mais significativos para o usuário. Essas informações permitem que se tenha um conhecimento mais palpável sobre um determinado contexto.

Este trabalho se baseia no mapeamento de uma base de dados disponibilizada em XML através da construção de um *parser*[1] Java, que possibilitou a análise da organização dos dados, levando a sua modelagem em um modelo de BD relacional. Com este modelo, utilizamos este mesmo *parser* para a carga da base, possibilitando, então, a aplicação de técnicas de Mineração de Dados e a construção de uma rede social científica com base em relacionamentos de coautoria. O intuito é representar as diversas comunidades científicas presentes em algumas universidades do país. A Figura 1.1 traz uma representação do fluxo descrito acima.

Seguindo a explicação acima, na Figura 1.1 temos a representação do arquivo XML da DBLP(1), o *parser* Java(2), que possibilitou analisar as estruturas ali presentes e construir um modelo que as representasse(3), levando a construção de um Banco de Dados(4), onde aplicamos técnicas de Mineração de Dados(5) e construímos uma Rede Social Científica baseada em relacionamentos de coautoria(6).

1.2 Justificativa

O aumento da disponibilidade de dados sobre como as pessoas se conectam umas as outras, em suas diversas esferas de interação, criou a possibilidade de se estudar como e por qual

[1]Ferramenta que analisa uma cadeia de símbolos, de acordo com um gramática pré-estabelecida.

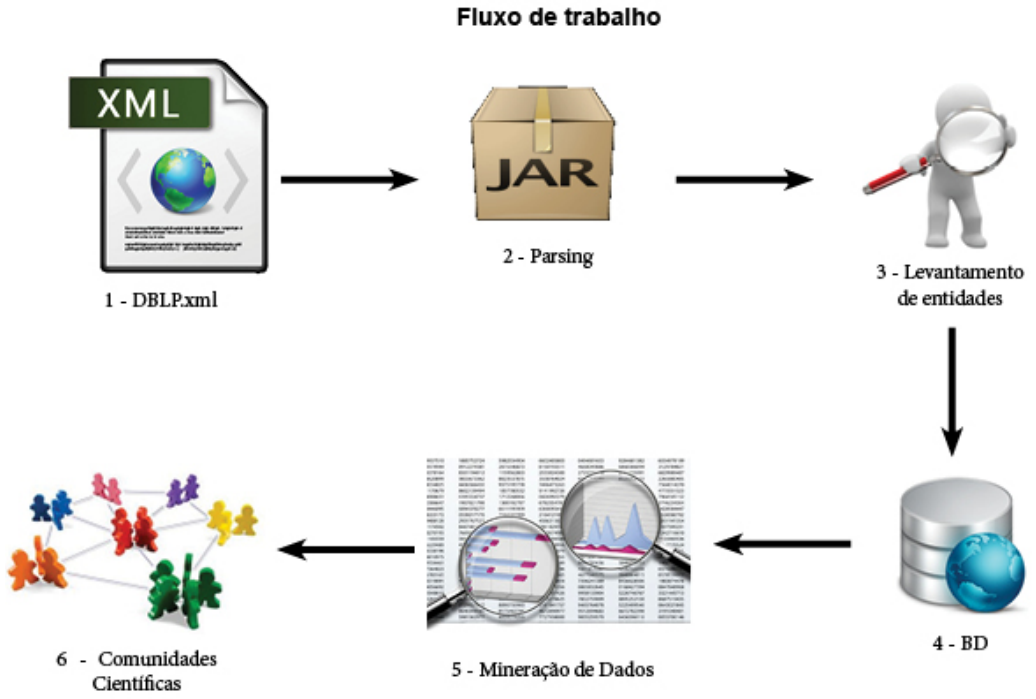

Figura 1.1: Representação gráfica do fluxo de trabalho realizado

motivo estas conexões estão sendo formadas. Tendo isso em mente, foi criado o conceito de Rede Social para facilitar a representação e análise destas relações (Ströele, 2012).

As Redes Sociais podem ser utilizadas também no meio científico no qual, ao analisar a distribuição de relações específicas, podemos identificar, por exemplo, em quais centros acadêmicos se encontra a maior concentração de conhecimento sobre determinada área e como esse conhecimento é difundido. A partir desta análise, poderíamos, por exemplo, sugerir maneiras de facilitar esta difusão, promovendo um maior contato entre pesquisadores de centros com conhecimentos complementares.

Com este pensamento, ter uma ferramenta que nos possibilite analisar e consolidar dados de pesquisas científicas de diversas fontes de forma fácil abriria um grande leque de possibilidades de estudo das relações entre pesquisadores, promovendo, inclusive, melhorias nestas relações e na qualidade da pesquisa no país.

1.3 Objetivos

1.3.1 Objetivo geral

Este trabalho visa à extração dos dados de um Banco de Dados científico, a DBLP, através da construção de uma ferramenta que realize o *parsing* desse arquivo. Ele possibilita, dessa forma, a modelagem e posterior carga de um Banco de Dados relacional no qual aplicaremos técnicas de Mineração de Dados, construindo uma representação das comunidades científicas presentes na base.

1.3.2 Objetivos específicos

Como a DBLP é disponibilizada em formato XML, foi desenvolvida uma ferramenta que realiza o parsing deste XML para a extração dos dados. A princípio esta ferramenta apenas listava as entidades presentes no arquivo. Assim pudemos construir um Modelo Entidade-Relacionamento que representasse corretamente estes dados e esse modelo foi, então, transformado em um banco de dados Relacional.

Em uma segunda etapa, a ferramenta de *parsing* foi evoluída de forma a realizar a carga dos dados neste banco. Com a base populada, para facilitar a validação, limitamos o número de pesquisadores-alvo e aplicamos técnicas de Mineração de Dados, gerando a Rede Social Científica correspondente àquele grupo de pesquisadores.

Finalmente, com esta aplicação criada, temos o objetivo de prover um meio fácil de manter uma base sempre atualizada. Já que a DBLP tem atualizações diárias, um pesquisador poderia estudar, com esse ferramental, a evolução de determinada rede social com o passar do tempo.

1.4 Metodologia

Para a elaboração deste trabalho, partimos do estudo realizado em (Ströele, 2012), onde foi feita uma análise de Redes Sociais Científicas baseadas na DBLP. Com base nisso, decidimos desenvolver uma aplicação que realizasse a extração dos dados dessa base e os representasse em um Banco de Dados relacional. O objetivo é facilitar a realização de

estudos baseados nessas informações, e, ao final, submeter esses dados a uma Mineração de Dados, formando uma Rede Social Científica alicerçada em relacionamentos de coautoria.

Para a extração dos dados, foi desenvolvida uma ferramenta em Java que realiza o parsing do XML utilizando a biblioteca *JAVA SAXParser*[2]. A princípio, esta ferramenta apenas listava as entidades presentes no arquivo, desta forma pudemos utilizar o *Astah Community*[3] para construir um Modelo Entidade-Relacionamento que representasse corretamente estes dados. Este modelo foi, então, transformado em um banco de dados PostGreSQL[4].

Em uma segunda etapa, a ferramenta de *parsing* foi evoluída de forma a realizar a carga dos dados para esta base. Com a base populada, utilizamos a aplicação desenvolvida no trabalho-base que aplica o algoritmo de agrupamento *k-medoids* para construir grupos e montar uma representação gráfica da Rede Social.

A fim de realizar uma validação da extração feita, reduzimos o grupo de estudo ao mesmo utilizado em Ströele (2012). Desta forma, pudemos, ao fim do trabalho, comparar os agrupamentos obtidos em ambos em busca de similaridades suficientes que validassem os dados exportados do XML da DBLP.

1.4.1 Estrutura do trabalho

No capítulo 2, começaremos falando sobre Redes Sociais Científicas e sua definição, partindo para a explicação do que é a base objeto do estudo, a DBLP, quais os tipos de dados contidos nela, sua organização e como mapeamos esses dados para uma estrutura de BD relacional. Falaremos a seguir sobre o sistema criado para mapear esses dados e a validação realizada sobre os resultados deste mapeamento.

No capítulo 3, traremos uma definição de Mineração de Dados e falaremos também sobre três técnicas existentes (classificação, predição e agrupamento). Teremos um foco especial nos métodos de agrupamento, por ser a técnica utilizada para este trabalho, e explicaremos também o algoritmo utilizado(*k-medoids*)

No capítulo 4, falaremos sobre o estudo de caso realizado e seus resultados, pas-

[2]https://docs.oracle.com/javase/7/docs/api/javax/xml/parsers/SAXParser.html
[3]http://astah.net/
[4]http://www.postgresql.org/

sando desde a definição dos grupos de estudo até as comunidades obtidas.

No capítulo 5, trataremos das considerações finais, com os objetivos alcançados, problemas encontrados e sugestão de trabalhos futuros.

2 DBLP como uma Rede Social Científica

2.1 Redes Sociais Científicas

Redes sociais são, de acordo com Ströele *et al.* (2012), "estruturas sociais dinâmicas formadas por indivíduos ou organizações". Assim, uma lista de e-mails trocados entre amigos é uma rede social ou mesmo uma simples conversa entre indivíduos. A grande questão é que este conceito é tão simples e se mostra presente de forma tão natural em nosso cotidiano que muitas vezes nem nos damos conta do tamanho e do número de redes nas quais estamos inseridos.

> "Geralmente, essas redes são representadas por nós ligados por um ou mais tipos de relacionamentos. Embora sejam estruturas extremamente complexas, analisá-las nos permite detectar diversos tipos de conexões entre as pessoas dentro e fora de suas instituições"(Ströele *et al.*, 2012).

Pensando desta forma nos damos conta de que poderíamos atingir um nível de colaboração muito maior e produzir uma interação muito mais completa se conseguíssemos modelar todos esses relacionamentos formando uma Rede Social. Desta forma, teríamos uma consciência maior sobre pessoas extremamente interessantes para nós e que, muitas vezes, passam despercebidas simplesmente por não tomarmos conhecimento de que ela existe e de que compartilha diversos interesses conosco.

Levando esta ideia para o meio científico temos uma situação interessante na qual, muitas das vezes, a solução para um problema pode estar, com um outro pesquisador geograficamente distante, porém conectado a nós por ter a mesmo área de estudo (ele pode estar atacando o mesmo problema de forma diferente e, ao juntarmos estes resultados parciais, temos algo novo e completo).

O que se vê é que, muitas vezes, pesquisadores e cientistas se limitam a trabalhar com apenas alguns colaboradores pelo fato de desconhecer o que está sendo realizado em outros lugares indiretamente conectados a ele. Porém, a partir do momento em que se centraliza estes trabalhos científicos em bases comuns, possibilitando o desenvolvimento de ferramentas que representem toda a gama de relacionamentos possíveis entre estes

pesquisadores, vemos uma nova definição: as redes sociais científicas. Definidas como "tipos específicos de redes sociais que representam as interações sociais oriundas do meio acadêmico"(Ströele *et al.*, 2012), neste trabalho será construída e utilizada uma Rede Social Científica.

2.2 DBLP

Considerada ainda uma nova alternativa, a DBLP(*Digital Bibliography Library Project*) teve seu início em 1993, evoluindo de um pequeno servidor web experimental para um serviço popular para a comunidade de ciência da Computação (Ley, 2009). Hoje, esta que começou como uma pequena e limitada base de dados é uma biblioteca digital contendo trabalhos de quase todos os campos de estudo em computação e, segundo dados da própria instituição mantenedora, hoje conta com mais de 1.2 milhão de registros.

Segundo o pensamento de Ley & Reuther (2006), dois dos idealizadores da DBLP, publicações científicas são utilizadas para avaliar o currículo de uma pessoa e o nível de um departamento ou instituição. Para ambos os aspectos, necessita- se de fontes confiáveis que reúnam estes trabalhos de forma fácil e organizada. Para tal existem as bibliotecas online.

Ainda segundo Ley (2009), devido à limitação de recursos, a base foi projetada de forma a conter muitos registros, porém com um baixo nível de detalhamento, preocupando em padronizar os nomes de forma a facilitar a catalogação e pesquisa. Entretanto, este ainda é um problema existente devido a fontes de origem diversas destas informações (arquivos eletrônicos, algoritmos de busca ou, às vezes, até a própria digitação do conteúdo). Dados chegam com nomes de conferências e autores abreviados, entre outros problemas.

Uma versão em XML atualizada diariamente é disponibilizada de forma gratuita no endereço *http://dblp.uni.trier.de/xml*, esta versão será utilizada no desenvolvimento do trabalho.

Realizaremos a extração dos dados deste arquivo para um banco de dados relacional e, a partir daí, mapearemos relacionamentos de coautoria para montar o grafo social que representa a rede social científica.

2.3 Conteúdo e mapeamento dos dados da DBLP

Na DBLP encontramos uma vasta coleção de dados mundiais com todo tipo de informação relativa a publicações da área de ciência da computação, como artigos, *journals*, informações sobre congressos, livros e até algumas teses de mestrado/doutorado. Por esta razão, ela se torna uma excelente fonte de estudos sobre redes sociais científicas.

Como dito anteriormente, os dados estão disponíveis no formato XML. Por isso, inicialmente, tentamos uma abordagem utilizando o pacote DOM[1] (nativo do JAVA) para o *parsing*, porém ela cria uma relação hierárquica entre todo o XML, o que levou a um gasto de memória e tempo inviável devido ao grande volume de dados. Para um melhor entendimento do funcionamento desta abordagem, considere, a título de exemplo, o XML da Figura 2.1:

```
<XML>
    <congresso>
        <nome> Congresso UFJF </nome>
        <artigos>
            <artigo>
                <titulo>DBLP</titulo>
                <autor>Alan de Paula Duque</autor>
            </artigo>
            <artigo>
                <titulo>SEO</titulo>
                <autor>Joao Silva</autor>
            </artigo>
        </artigos>
    </congresso>
</XML>
```

Figura 2.1: Exemplo de um XML da DBLP

Utilizando o método implementado por essa API, teríamos uma estrutura de árvore com o formato descrito na Figura 2.2.

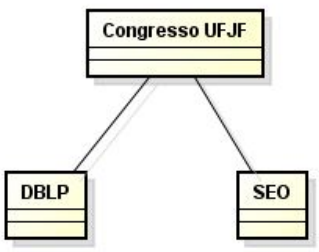

Figura 2.2: Estrutura gerada após o *parsing*

Apesar de, semanticamente, ser uma representação boa, devido ao grande número

[1]https://docs.oracle.com/javase/7/docs/api/org/w3c/dom/package-summary.html

de relacionamentos entre os diversos elementos, construir uma estrutura dessa forma se mostrou uma operação extremamente lenta e com um consumo alto de memória. É, portanto, inviável para o caso em questão (onde estávamos lidando com um arquivo XML de 2,5 GB que deveria ser completamente percorrido e, posteriormente, ter seus dados persistidos em um Banco de Dados).

Depois desta primeira tentativa, testamos a biblioteca *JAVA SAXParser* que funciona basicamente apoiada em três eventos: um na abertura de uma *tag*, um ao capturar o texto contido e outro ao fechar a *tag*.

Esta biblioteca se mostrou muito eficiente para nosso caso devido ao fato de não construir nenhum tipo de estrutura: ela apenas ativa os eventos e deixa o resto a cargo do usuário. Dessa forma, pudemos ter um maior controle sobre a complexidade e consumo de memória do algoritmo de *parsing*.

Com a utilização desta biblioteca, conseguimos dar início ao levantamento das informações contidas no arquivo XML. Esta etapa consistiu no desenvolvimento de um algoritmo que percorria todo o XML, construindo um *log* das diferentes *tags* encontradas, de forma que pudéssemos identificar a estrutura do documento.

Com estas informações em mãos, passamos para a fase de análise dessa estrutura. Para uma primeira análise, comparamos o *log* obtido com as informações contidas no artigo (Ley, 2009) publicado pela própria universidade que mantém a biblioteca, e identificamos que, em muitos casos, as *tags* existiam no XML, porém não estavam explicadas com clareza no documento. Para esses casos, resolvemos alterar o algoritmo inicial com a finalidade de que o *log* fosse gerado com as informações de trabalhos conhecidos, a fim de que pudéssemos, a partir do conhecimento prévio dos trabalhos, inferir sobre o que significariam as informações não listadas na documentação.

Passada esta fase, partimos para a elaboração do modelo entidade-relacionamento de nossa base de dados. Foi realizada uma modelagem contemplando os aspectos atuais da DBLP, mas também visando às possíveis implementações futuras com a utilização de outras bibliotecas que tenham informações complementares. O modelo obtido pode ser visto na Figura 2.3.

Dos dados contidos na DBLP, podemos extrair, por exemplo, informações sobre

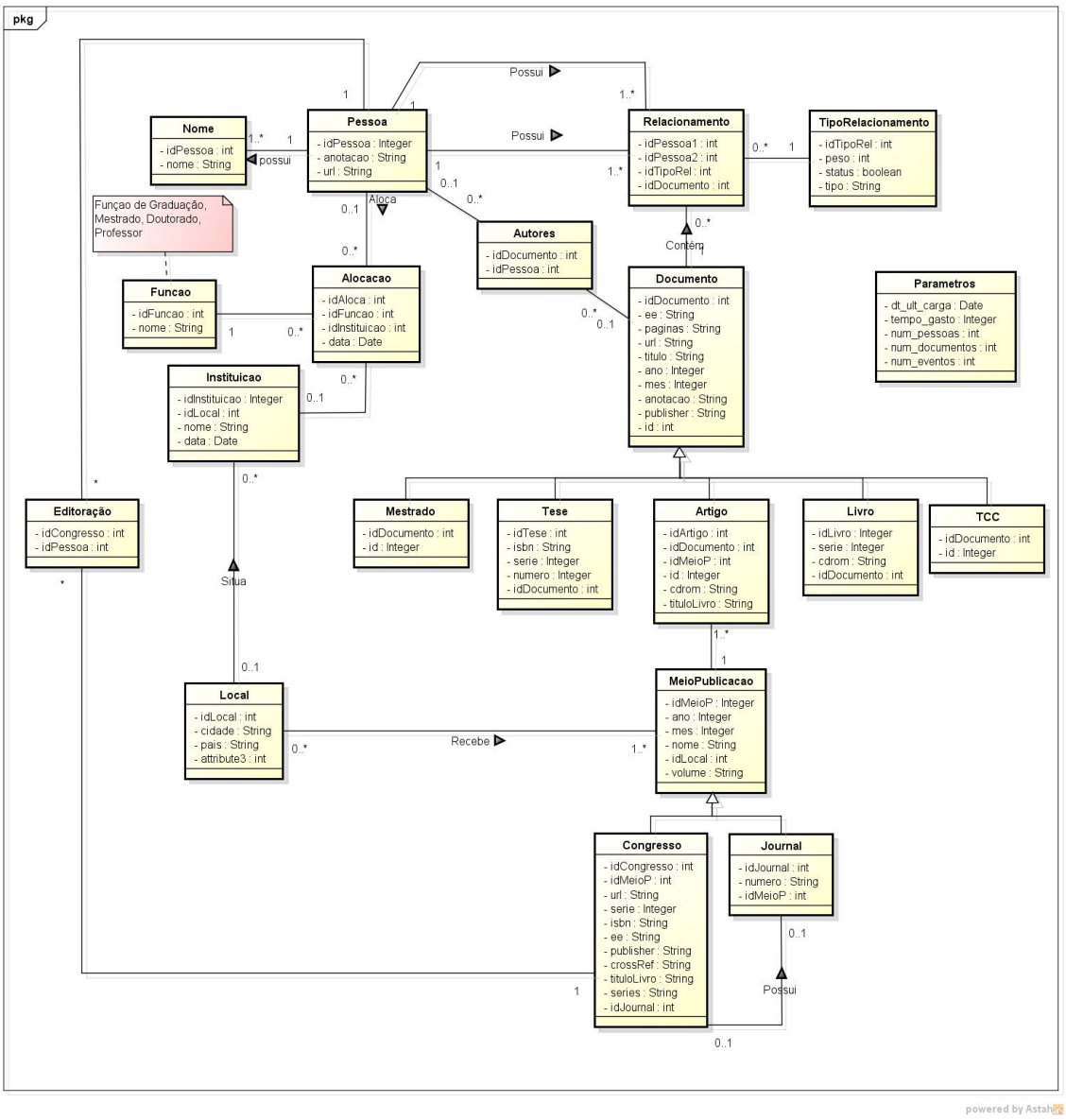

Figura 2.3: Modelo de banco de dados elaborado

autoria, em qual evento foram apresentados e quais autores publicaram juntos, porém não conseguimos estabelecer relacionamentos por citações, devido ao simples fato de estes não estarem representados no XML. Outro ponto é restringir com absoluta precisão todas as publicações de um pesquisador específico, devido ao fato de que um indivíduo pode ter seu nome escrito de diversas formas e não possuirmos nenhuma maneira de associar essas duas grafias. Os próprios criadores da DBLP declararam que isso é um problema em aberto (Ley, 2009). Ainda dentro das informações que não conseguimos extrair estão dados geográficos, pois não temos informações suficientes para associar os pesquisadores às suas respectivas instituições.

2.3.1 Tabelas e relacionamentos extraídos do XML

Nesta seção, será feita uma breve explicação das tabelas e relacionamentos obtidos a partir da análise descrita acima.

Uma das principais entidades da modelagem realizada é "pessoa". Ela representa os pesquisadores que são autores e/ou editores em quaisquer trabalhos ou eventos. Juntamente com a entidade “relacionamento”, ela irá compor a base de nossa análise dos relacionamentos entre autores na fase de mineração de dados[2].

A entidade “pessoa” é gerada basicamente pelos registros com tags <author> e <editor> dentro do XML. A marcação <author> aparece aninhada a outras, representando o autor de um artigo ou livro, e se repete no caso de mais de um autor. Os relacionamentos de coautoria surgem a partir desses artigos e livros que possuem mais de um autor.

Uma característica desta entidade é o seu relacionamento com a entidade "nome"(tal relacionamento se fez necessário em virtude da existência de pesquisadores com nomes distintos em diferentes publicações). Apesar de esta questão ainda estar pendente de uma resolução, o relacionamento já foi criado visando aos trabalhos futuros. Como citado anteriormente, da forma que é hoje, o XML não nos fornece informações suficientes para determinar quais nomes seriam de um mesmo autor de forma que isto tem que ser feito por um algoritmo independente. Os relacionamentos citados podem ser vistos na Figura 2.4.

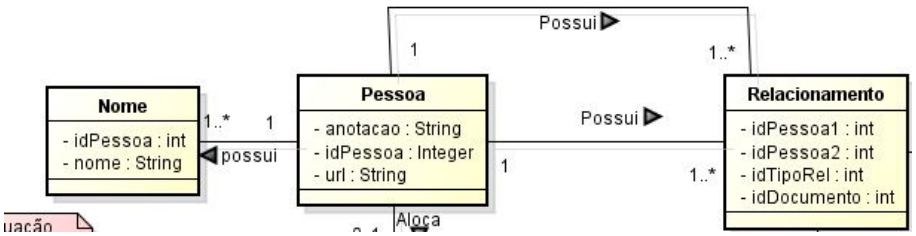

Figura 2.4: Relacionamento entre as tabelas Nome, Pessoa e Relacionamento

Outra entidade crucial para a base é "relacionamento", ela fará a correspondência entre coautores e, consequentemente, relacionará autores aos seus documentos. A carga dos dados dessa tabela é feita a partir de uma *stored procedure* que realiza a combinação

[2]A definição do termo mineração de dados será dada no capítulo 3

n a n dentre os autores de cada documento.

Ela está relacionada a "tipoRelacionamento" que, por sua vez, servirá para descrever cada tipo de relacionamento (a princípio teremos apenas coautoria), dar o seu peso (dado fundamental para a etapa de mineração) e também para podermos tornar certos relacionamentos inativos para pesquisas futuras. Esta parte do modelo está representada na Figura 2.5.

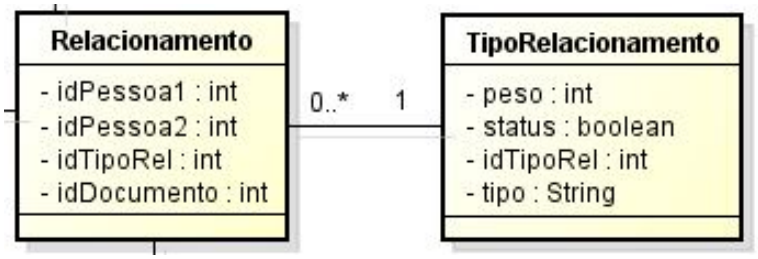

Figura 2.5: Relacionamento entre as entidades Relacionamento e TipoRelacionamento

Temos ainda, no diagrama, as tabelas relacionadas a publicações ("Mestrado", "Tese", "Artigo","Livro","TCC") responsáveis por armazenar dados das publicações em si. Como todos estes tipos possuem vários campos em comum, vimos que seria pertinente agrupá-los em um nível superior, daí surgiu a tabela "Documento" que será responsável pela ligação a um determinado relacionamento entre autores. Esta estratégia está representada na Figura 2.6

Apesar de a existência de todas essas categorias, no primeiro momento, nem todas serão utilizadas, visto que algumas destas informações não se encontram disponíveis ainda ou estão presentes de forma muito rudimentar. A tabela TCC é um caso que não está previsto na DBLP, então sua razão de ser é apenas uma previsão de expansão do trabalho a fim de analisar outras fontes de dados. Estas tabelas são populadas a partir das tags: <article>(tabela artigo), <phdthesis>(tabela tese), <masterthesis>(tabela mestrado),<book>(tabela livro).

As tabelas citadas acima, quando necessário, ligam-se aos eventos nos quais foram publicadas (congressos ou *journals*) através da tabela "Meio_Publicacao", que funciona também como uma superclasse, possuindo seus próprios campos que são comuns entre journal (tag <journal>) e congresso (tag <proceeding>). Esta modelagem foi pensada visando à inclusão futura de novas categorias e está evidenciada na Figura 2.7

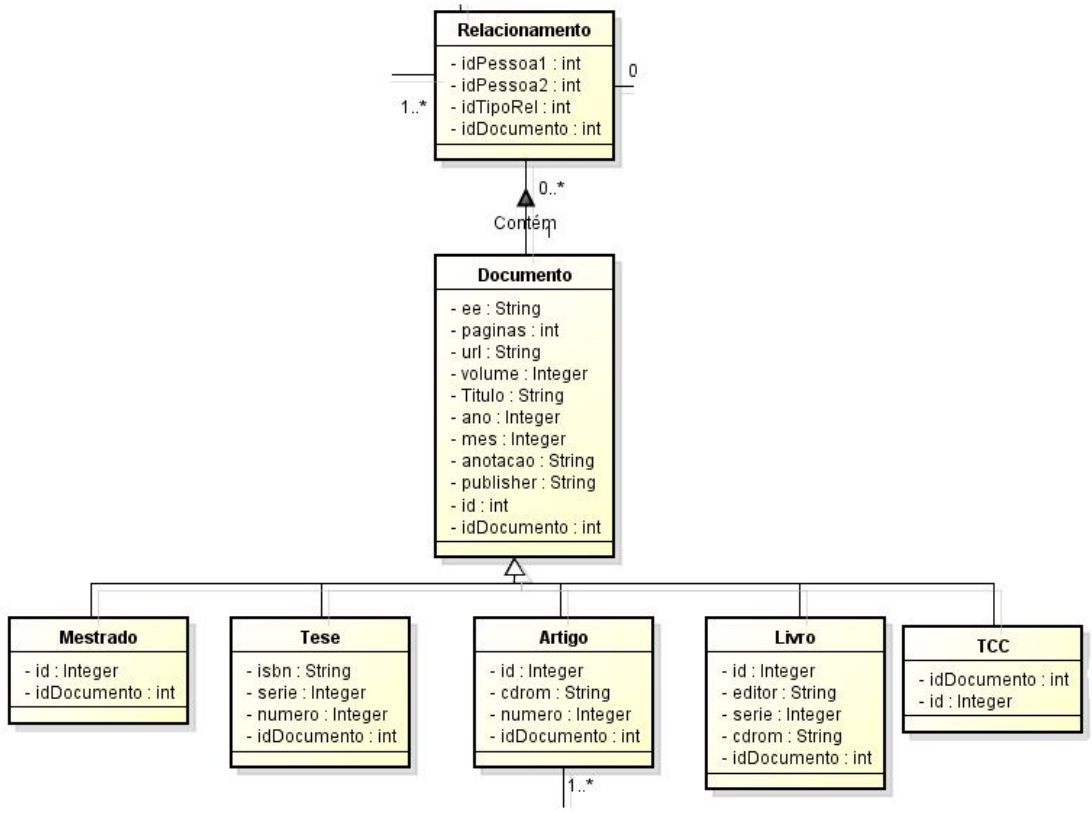

Figura 2.6: Representação dos diversos tipos de documentos

Para a explicação da próxima tabela, faz-se necessário um exemplo. Consideremos o XML da Figura 2.8. Supondo que o código correspondente ao autor "Alan de Paula Duque" seja 42, o tipo de relacionamento "autor" seja o de código 3, o documento de título "Trabalho de Conclusão de Curso" corresponda ao código 5 e que a chave desta tabela seja composta por todos os campos, este registro seria descrito na tabela relacionamento como o representado na tabela 2.1.

Tabela 2.1: Representação do relacionamento de autoria

Relacionamento			
idPessoa1	idPessoa2	idTipoRel	idDocumento
42	Null	3	5

Note que isto não seria possível, pois como não há outro autor para o documento em questão não existe a "Pessoa 2" e, portanto, o campo "idPessoa2" tem um valor nulo (levando a uma violação da chave primária que é composta também por esta coluna).

Este problema nos levou à criação da relação "autores", que modela todas as ligações entre eles e suas obras, e é a base para a carga da tabela "relacionamentos". Sua carga se origina também a partir das tags <author> contidas nos registros dos documen-

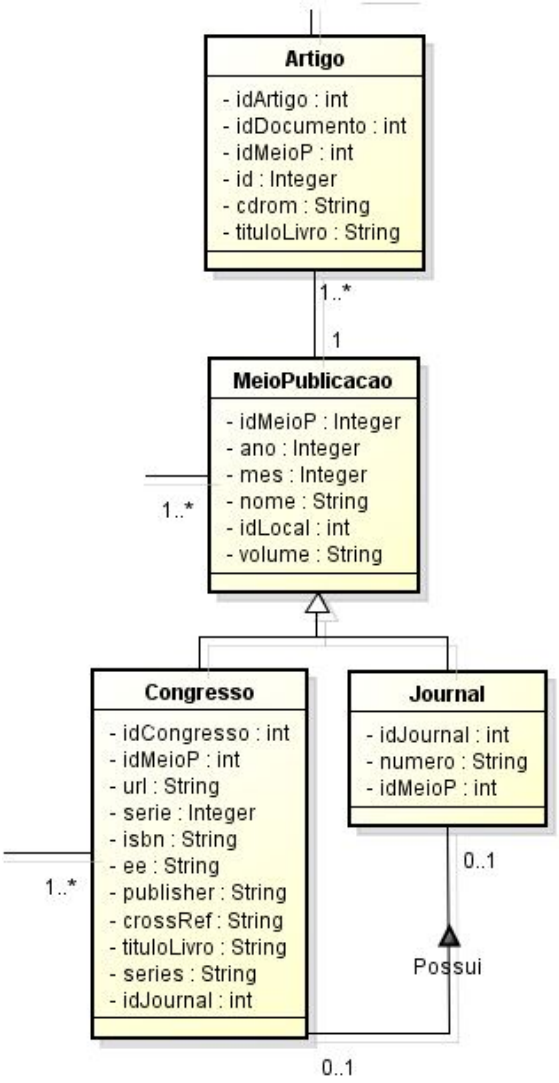

Figura 2.7: Relacionamento com a tabela Meio_Publicacao

tos. Assim, a carga dos relacionamentos é feita a partir da tabela autores. A tabela e seus relacionamentos podem ser vistos na Figura 2.9.

A tabela de parâmetros é a única do modelo que não se relaciona diretamente ao XML, ela tem por objetivo armazenar alguns dados de execução da aplicação a fim de termos estatísticas de desempenho e para o futuro desenvolvimento de formas de atualizar a base. Esta entidade pode ser vista na Figura 2.10.

Relacionando as tabelas “Congresso” e “Pessoa” temos a tabela “Editoração” que serve exclusivamente para criar o relacionamento dos congressos a seus editores. Apesar de ter a sua representação na base, esta relação não nos interessará para fins de análise das redes sociais. Ela se dá a partir da tag <editor> contida dentro da tag <proceeding> que define um congresso. Esta associação pode ser vista na Figura 2.11

```
<article>
    <title>DBLP</title>
    <author>Alan de Paula Duque</author>
</article>
```

Figura 2.8: Exemplo de XML representando um artigo publicado

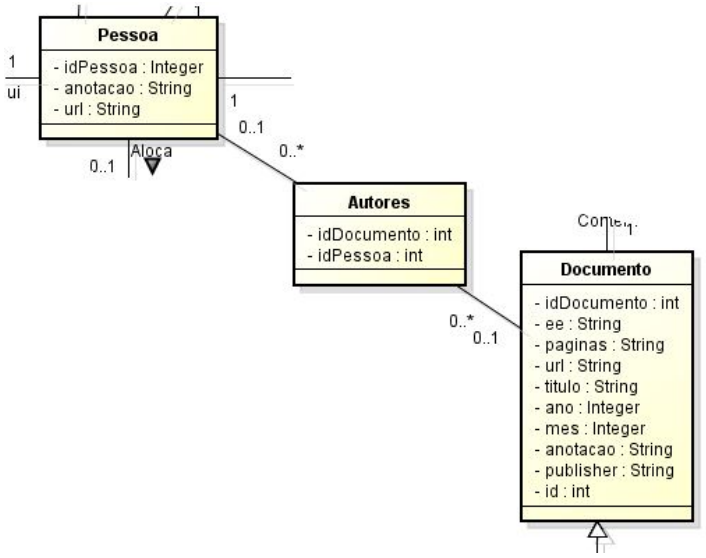

Figura 2.9: Tabela Autores e seus relacionamentos

O outro grupo de entidades, representado na Figura 2.12 ("Local", "Instituicao", "Alocacao", "Funcao"), foi feito para realizar a localização geográfica dos congressos e alocação dos pesquisadores nas suas instituições, porém estes dados ainda não são contemplados pela DBLP. A carga dos dados dessas tabelas depende de informações de outras fontes de dados, por isso, neste primeiro momento, essas tabelas não foram populadas.

2.4 Extração dos Dados

Com o modelo da base de dados pronta, tivemos que extrair os dados do XML, processá-los e inseri-los no banco. Para esta tarefa, utilizamos a base do algoritmo citado na seção anterior, que já fazia a varredura do XML. Para conseguir mapear estes dados em classes correspondentes ao modelo estabelecido foi elaborado um sistema, descrito a seguir.

O sistema é composto apenas de uma interface simples, apresentada na Figura 2.13. O campo “Arquivo da base (DBLP)”, possibilita a seleção do arquivo DBLP a ser processado. Logo abaixo está uma lista denominada “Pesquisadores Selecionados”, que

Figura 2.10: Tabela Parametros

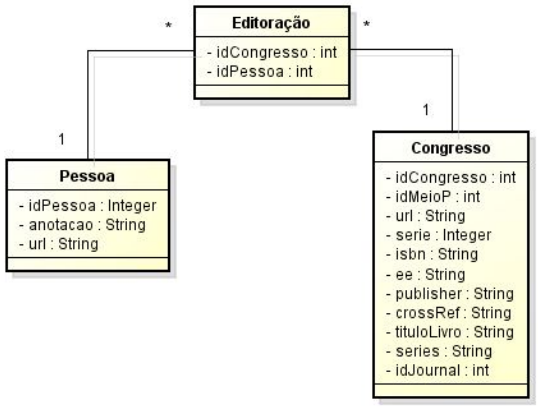

Figura 2.11: Tabela Editoracao e seu relacionamento com Pessoa e Congresso

será carregada a partir do arquivo "pesquisadores.txt", presente na pasta da aplicação. Caso este arquivo contenha dados, apenas serão considerados, durante o processamento, trabalhos de pesquisadores presentes na lista. Caso o arquivo esteja vazio, não será aplicado nenhum filtro e todos os dados serão importados. Logo abaixo temos o botão que aciona o procedimento da carga, entitulado de "Realizar carga da base" e a *label* "Tempo gasto" que mostra, ao fim do processo, o tempo total demorado para carregar todos os dados.

Para a explicação do funcionamento do algoritmo de carga, utilizaremos o XML de exemplo da Figura 2.8. Como foi mencionado anteriormente, a biblioteca utilizada trabalha apenas com três métodos, *startElement* (ativado quando uma *tag* é aberta), *characters* (ativado assim que termina a leitura dos caracteres contidos na *tag*) e *endElement* (ativado quando uma *tag* é fechada).

No XML de exemplo, o método *startElement* seria ativado pela *tag* <article>, verificaria que não existe nenhuma outra *tag* aberta (que estaria salva em um atributo da classe), e criaria um objeto do tipo "artigo". Ao ser chamado novamente, pela abertura da *tag* <title>, ele verificaria que o atributo que armazena a *tag* aberta atualmente contém <article> e então armazenaria <title> em outro atributo. O comportamento

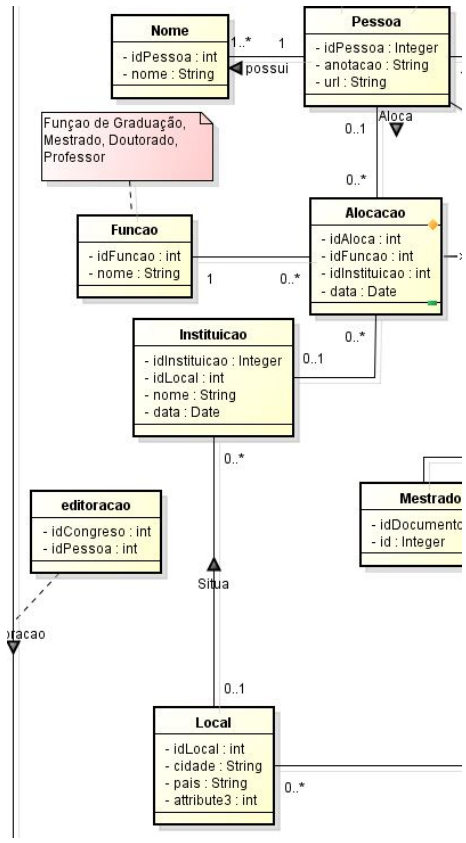

Figura 2.12: Tabelas Local, Instituicao, Alocacao e Funcao

deste método está esquematizado na Figura 2.14.

Cada vez que o método *characters* fosse chamado, concatenaria o texto em outro atributo da classe. O comportamento deste método está esquematizado na Figura 2.15.

Por sua vez, o método *endElement*, ao ser chamado pelo fechamento da *tag* <title>, iria verificar que a *tag* <article> está aberta, requisitaria o preenchimento do atributo "título" do objeto "artigo"corrente e passaria o atributo utilizado pelo método *characters* para branco novamente. Ao ser ativado pelo fechamento da *tag* <article>, ele verificaria que esta *tag* encontrava-se aberta e solicitará a persistência do objeto. O comportamento deste método está esquematizado na Figura 2.16

2.5 Validação

Para facilitar o processo de validação dos dados extraídos, utilizamos do trabalho Ströele (2012) no qual uma análise semelhante foi realizada para um grupo restrito de pesquisadores e, por se tratar de um trabalho já publicado e reconhecido, seria um bom parâmetro de comparação. Sendo assim, como realizamos a carga completa da DBLP, precisamos

Figura 2.13: Interface inicial do sistema de carga

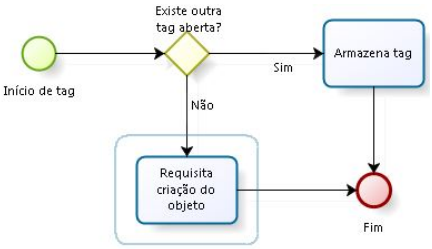

Figura 2.14: Representação do funcionamento do método *startElement*

reduzir a análise ao mesmo grupo de pesquisadores e nos utilizamos da mesma implementação de Mineração de Dados do trabalho original, de forma a igualar os ambientes para que os resultados finais pudessem ser comparados e pudéssemos validar nosso trabalho.

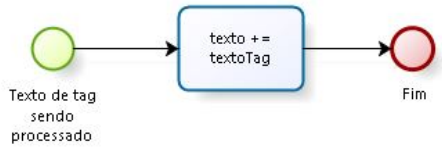

Figura 2.15: Representação do funcionamento do método *characters*

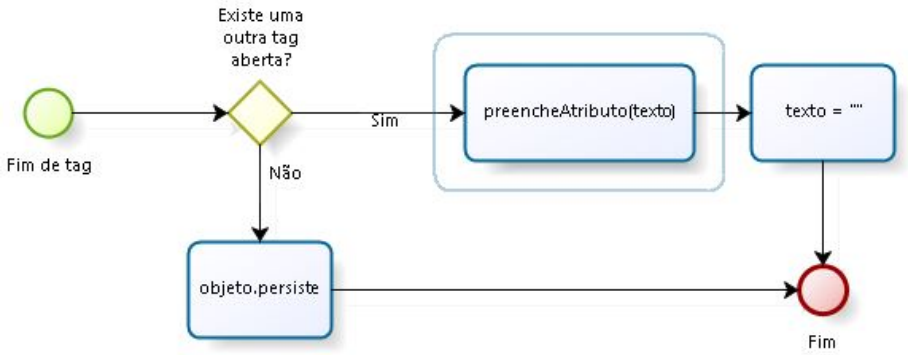

Figura 2.16: Representação do funcionamento do método *endElement*

3 Mineração de dados

3.1 Definição

Hoje em dia, o armazenamento de dados em bancos digitais é algo corriqueiro em todos os segmentos e cada vez mais o custo para tal se torna menor. Devido a esse fato, é comum ver bancos de dados gigantes nos quais, conforme afirmado por Rezende (2005), as "informações realmente novas e interessantes estão “embutidas” nessas Bases de Dados". De acordo com Lin & Cercone (1996), estas grandes bases são uma mina de ouro de informações, porém conforme descrito por Mitra *et al.* (2002) estes dados quase nunca são obtidos de forma direta e, geralmente, não são extraídos devido à falta de ferramentas apropriadas.

Neste sentido, conforme apontado por Rezende (2005), a Mineração de Dados surgiu no fim da década de oitenta para obter informações a partir de grandes volumes de dados. Apesar de ser uma área criada com o intuito de resolver problemas relacionados à computação, conforme o estudo realizado por Zhou (2003), ela não é isolada e tem relacionamentos bastante próximo a outras disciplinas. Ele provou que, entre outras, as áreas de Bancos de Dados, Aprendizado de Máquina, Estatística, Recuperação da Informação, Computação Paralela e Distribuída foram de grande valia para o desenvolvimento da Mineração de Dados.

Em seu trabalho, ele exemplifica como algumas destas contribuições ocorreram. Em primeiro lugar, Banco de Dados se concentrou na eficiência, visando à descoberta em grandes volumes de dados. O Aprendizado de Máquina prioriza a efetividade, visto que se baseia em heurísticas efetivas de análise desses dados, enquanto a Estatística nos traz a ideia de validade através da matemática que prova as técnicas de Mineração de Dados.

O reflexo desta variedade de áreas que abordam a Mineração de Dados é que encontramos várias definições na literatura, dependendo da perspectiva da área de interesse do autor. Citaremos aqui três caracterizações diferentes, mencionadas por Zhou (2003), retiradas de três livros que tratam do tema. Segundo Han & Kamber (2006),

da área de Banco de Dados, Mineração de Dados é o processo de descoberta de conhecimento interessante em grandes quantidades de dados armazenados em Bases de Dados, *Data Warehouses*[1] ou outros repositórios de dados. Na área de Aprendizado de Máquina, Witten & Frank (1999) a definem como extração de conhecimento implícito, previamente desconhecido e potencialmente útil a partir de dados. Por fim, sob a perspectiva Estatística, temos Hand *et al.* (2001) caracterizando-a como a análise de conjuntos de dados supervisionados, normalmente em grandes quantidades, para encontrar relacionamentos inesperados e resumir os dados em novas formas que são compreensíveis e úteis para o proprietário dos dados.

Como o foco de nosso trabalho é a análise de um grande volume de dados em um Banco de Dados, adotaremos a definição de Han & Kamber (2006), onde os "dados interessantes" são as comunidades científicas implícitas nas redes sociais de coautoria e a "grande quantidade de dados armazenados em Bases de Dados" são os relacionamentos científicos contidos na DBLP.

Para Rezende (2005), a Mineração de Dados tem como objetivo extrair informações "escondidas" nas bases de dados e obter formalismos que as representem, pois, da mesma forma que existem informações que podem ser obtidas de forma direta (sem nenhuma técnica especial), existem aquelas que precisam de estratégias mais elaboradas para que se tornem visíveis, daí a importância da Mineração de Dados.

3.2 Técnicas Existentes

Quando falamos de Mineração de Dados várias técnicas surgem, porém nos ateremos a citar e comentar apenas três: Classificação, Predição e Agrupamento (*Clustering*). Um dos grandes problemas da atualidade é a grande disponibilidade de informações contra a inabilidade do ser humano em analisar e processar todos os dados quando da tomada de uma decisão, muitas vezes, crucial. Visando a este cenário, veremos um pouco sobre classificação e predição de dados que têm por objetivo ajudar nestes casos.

[1] *Data Warehouse* é um grande banco de dados exclusivo de consulta que contém dados correntes e históricos, oriundos de diversos outros bancos de dados menores(Laudon & Laudon, 2011).

3.2.1 Classificação

Classificação é uma técnica de mineração de dados onde são construídos modelos, ou classificadores, para a criação de rótulos por categoria (Han & Kamber, 2006). Utilizando dos exemplos citados em Han & Kamber (2006), um gerente de banco precisa saber se determinado empréstimo é arriscado ou não; um vendedor de informática precisa saber se um cliente com o perfil "x" precisará de um novo computador em breve; ou ainda, um médico precisa analisar um quadro de câncer para saber qual o melhor tratamento. Em todos esses casos, trata-se de problemas de classificação de dados: "arriscado" ou "não arriscado" para o empréstimo, "sim" ou "não" para a necessidade de um computador novo e "tratamento A" ou "tratamento B" para o médico.

A classificação consiste em gerar conhecimento a partir da análise de exemplos, utilizados na classificação dos itens. Ela pode ser dividida em duas etapas (Han & Kamber, 2006):

- A etapa de aprendizado, ou treinamento, é responsável por analisar os exemplos fornecidos e construir um modelo classificador. Esse modelo pode ser um conjunto de regras de classificação, árvores de decisão, Máquinas de Vetores Suporte etc. Esta etapa está representada na Figura 3.1.

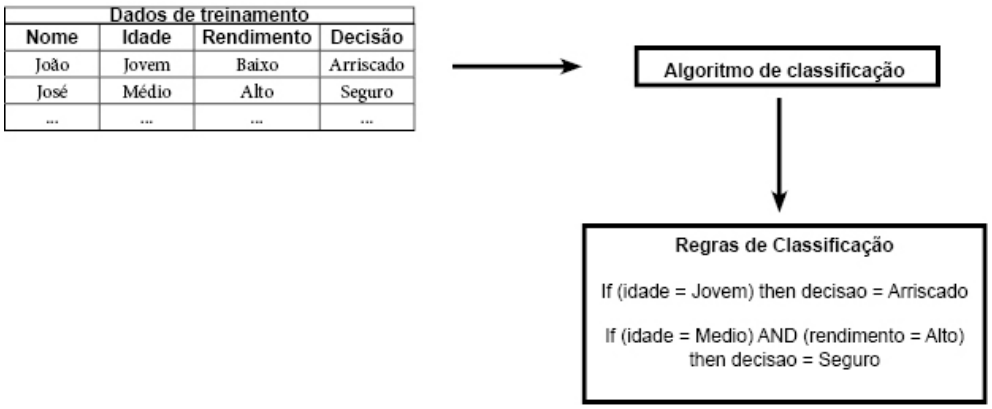

Figura 3.1: Método de classificação - etapa de aprendizado

- A etapa de classificação é focada em utilizar o modelo classificador para distribuir os elementos desconhecidos em seus grupos predefinidos (classes). Esta etapa está exemplificada na Figura 3.2.

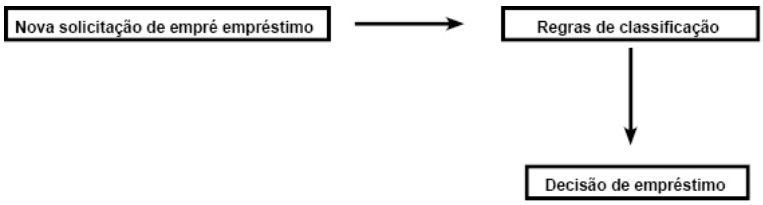

Figura 3.2: Método de classificação - etapa de classificação

De acordo com o proposto por Han & Kamber (2006), as principais características de um modelo de classificação são:

- **Precisão:** capacidade de associar corretamente um item a sua classe;
- **Velocidade:** esforço demandado para realizar a associação;
- **Robustez:** capacidade de acerto frente a dados incompletos;
- **Escalabilidade:** capacidade de manter o desempenho de forma proporcional ao volume de dados.

3.2.2 Predição

Suponha que precisamos prever o volume de vendas de um determinado produto ou calcular a variação futura de uma ação da Bolsa de Valores. Neste caso, a classificação não nos traz benefício nenhum. Para tal, existe a abordagem de predição que consiste em prever valores contínuos ou ordenados para um determinado parâmetro. A técnica mais usada é a regressão que se apresenta, basicamente, em duas técnicas distintas, a regressão linear e a regressão não linear (Han & Kamber, 2006).

A regressão linear envolve uma variável resposta "y", associada a uma única variável de predição "x", sendo assim a forma mais simples de regressão. A regressão pode ser representada por uma função linear da forma $y = b + kx$, na qual assumimos a variância de "y" como constante e "b" e "k" são coeficientes de regressão, podendo ser vistos também como pesos (Han & Kamber, 2006). Após calcular essa função, é possível prever um valor futuro. O gráfico de uma regressão linear pode ser visto na Figura 3.3.

Na grande maioria dos casos, o comportamento dos dados não se dará de forma linear. Para tal, existe a regressão não linear, que é utilizada quando temos apenas uma variável de predição e um modelo que nos permite uma maior precisão ao utilizar uma

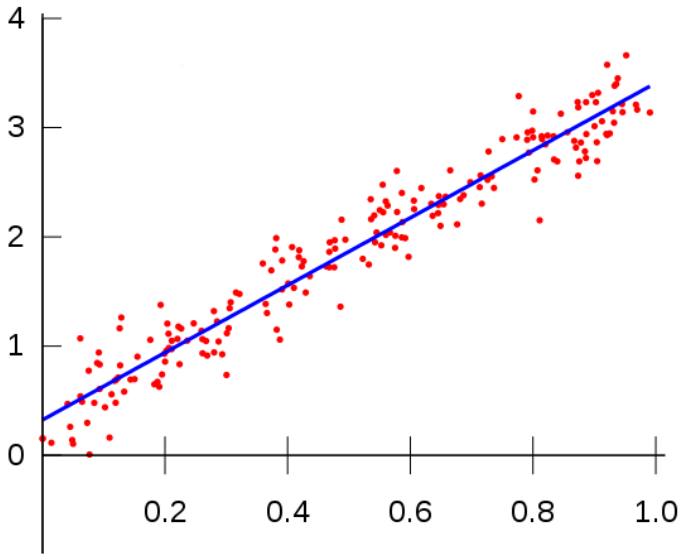

Figura 3.3: Gráfico de uma regressão linear (Fonte: http://pt.wikipedia.org/wiki/Regressão_linear)

função polinomial (Han & Kamber, 2006). O gráfico de uma regressão não linear pode ser visto na Figura 3.4.

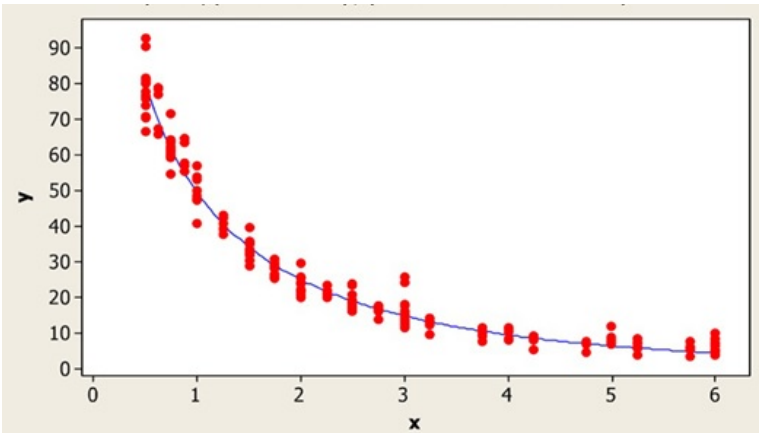

Figura 3.4: Gráfico de uma regressão não linear (Fonte: http://www.minitab.com/pt-br/Case-Studies/Mercer-Consulting)

3.2.3 Agrupamento

A terceira técnica mencionada aqui é o agrupamento. Esta será a técnica abordada na execução do trabalho, a ser discutida no capítulo 4. Sendo assim, iremos nos aprofundar mais em suas características.

Primeiramente, antes de definirmos agrupamento, devemos nos atentar a um conceito que será muito mencionado. Definiremos aqui um grupo (ou *cluster*) pela abordagem

de Han & Kamber (2006) como sendo um conjunto de elementos que são similares entre si e dissimilares aos elementos pertencentes a outros *clusters*.

Agrupamento é também chamado de segmentação de dados, pois leva a uma divisão em diversas porções de acordo com sua similaridade. Esta forma de mineração também pode ser usada para detectar elementos fora do padrão. Por exemplo, ao aplicar um agrupamento a um conjunto de dados que contém informações sobre compras realizadas com cartão de crédito por um cliente, podemos identificar valores incompatíveis com as compras usualmente feitas por ele, a fim de descobrir uma possível fraude. Também pode ser usada como preparação para um algoritimo de classificação, identificando os grupos a serem utilizados. O método de agrupamento é mostrado na Figura 3.5.

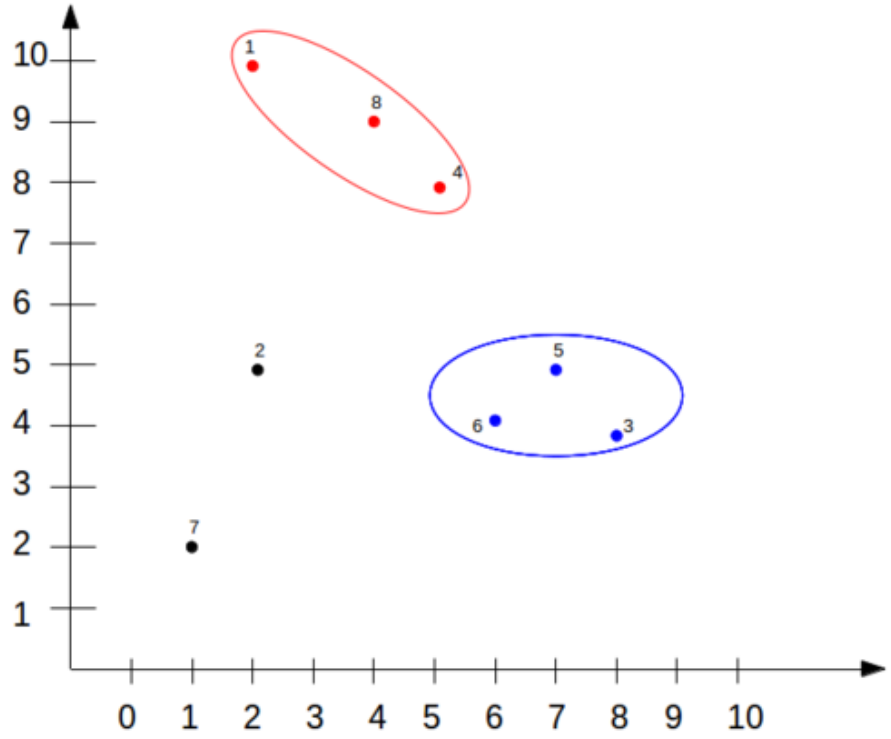

Figura 3.5: Representação da formação de agrupamentos (Fonte: http://pt.wikibooks.org/wiki/Processamento_de_Dados_Massivos/Projeto_e_implementação_de_aplicações_Big_Data/Agrupamento_baseado_em_densidade)

Segundo Han & Kamber (2006) são requisitos básicos para um algoritmo de agrupamento:

- **Escalabilidade:** os algoritmos devem ser capazes de serem executados em grandes bases de dados, pois se rodarmos em apenas uma amostra desta base, isso pode resultar em dados parciais.

- **Habilidade de lidar com diversos tipos de dados:** muitos algoritmos são feitos para trabalharem com dados numéricos, entretanto, pode ser necessário, em alguns casos, trabalhar com dados de outros tipos.

- **Descoberta de grupos com formatos distintos:** muitos algoritmos se baseiam em distâncias entre nós, por isso tendem a construir classes esféricas e com densidade semelhante. Um algoritmo deve ser capaz de construir grupos com qualquer forma. Um exemplo desta abordagem são os algoritmos baseados em densidade, que continuam populando os grupos enquanto a densidade não exceder um número mínimo preestabelecido. Este tipo de abordagem ajuda a eliminar ruídos (Han & Kamber, 2006). Esta habilidade está representada na Figura 3.6, na qual o mesmo algoritmo baseado em densidade foi aplicado em três bases diferentes, apresentando grupos com formatos arbitrários.

Figura 3.6: Execução de um algoritmo baseado em densidade em três bases distintas (Fonte: http://slideplayer.com.br/slide/359184/)

- **Conhecimento mínimo do domínio para determinar os valores de *input*:** o algoritmo deve exigir o mínimo de parâmetros do usuário, pois estes parâmetros geralmente influenciam muito no resultado e são difíceis de se determinar, além de serem um fardo para o usuário.

- **Habilidade de lidar com dados ruidosos:** a maioria das bases reais contém dados incompletos. Alguns algoritmos são menos sensíveis a estes problemas e, portanto, apresentam resultados melhores quando os dados possuem ruídos.

- **Possibilidade de incrementar dados às classes já existentes e insensibilidade a ordens dos *inputs*:** alguns algoritmos não conseguem adicionar novos dados às classes já existentes, construindo tudo do zero sempre. A sensibilidade

quanto à ordem dos parâmetros de entrada leva a resultados muito diferentes a cada maneira como eles são inseridos, isso é um defeito a ser evitado.

- **Grande dimensionalidade:** bases de dados ou *data warehouses* podem conter muitas dimensões. Muitos algoritmos são capazes de lidar com apenas duas ou três dimensões. O grande desafio é descobrir grupos em dados multidimensionais, principalmente porque estes dados podem estar esparsos.

- **Agrupamento baseado em constantes:** é importante realizar o agrupamento considerando constantes especificadas, pois é o que as aplicações reais precisam fazer. Por exemplo, se queremos encontrar o melhor lugar para instalar um caixa eletrônico em uma cidade devemos considerar a malha rodoviária da cidade e o tipo e número de clientes por classe, por exemplo.

- **Interpretabilidade e usabilidade:** o que se espera ao final de um agrupamento é que a informação produzida seja compreensível e útil. Deve-se atentar para a finalidade da aplicação quando da seleção de um método de agrupamento.

3.3 Métodos de agrupamento

Segundo Han & Kamber (2006), os métodos de agrupamento podem ser separados em cinco categorias: métodos hierárquicos, métodos baseados em densidade, métodos baseados em tabelas, métodos baseados em modelos e métodos de partição.

Métodos hierárquicos consistem em uma série de sucessivos agrupamentos, ou sucessivas divisões de elementos, nos quais os elementos são agregados ou desagregados. Eles são subdivididos em métodos aglomerativos (Figura 3.7), nos quais cada elemento começa representando um grupo e, a cada passo, um grupo ou elemento é ligado a outro de acordo com sua similaridade; e métodos divisivos (Figura 3.8), nos quais, inicialmente, todos os elementos formam um único grupo e, a cada iteração estes grupos vão sendo subdivididos (Doni, 2004).

Métodos baseados em densidade utilizam um método de formação de grupos que se utilizam da densidade de pontos em um local, ao invés da distância entre os pontos

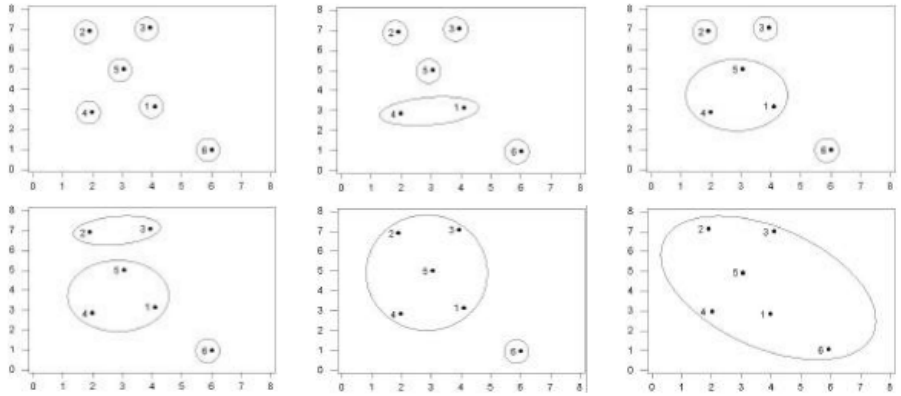

Figura 3.7: Métodos hierárquicos aglomerativos
(Doni, 2004)

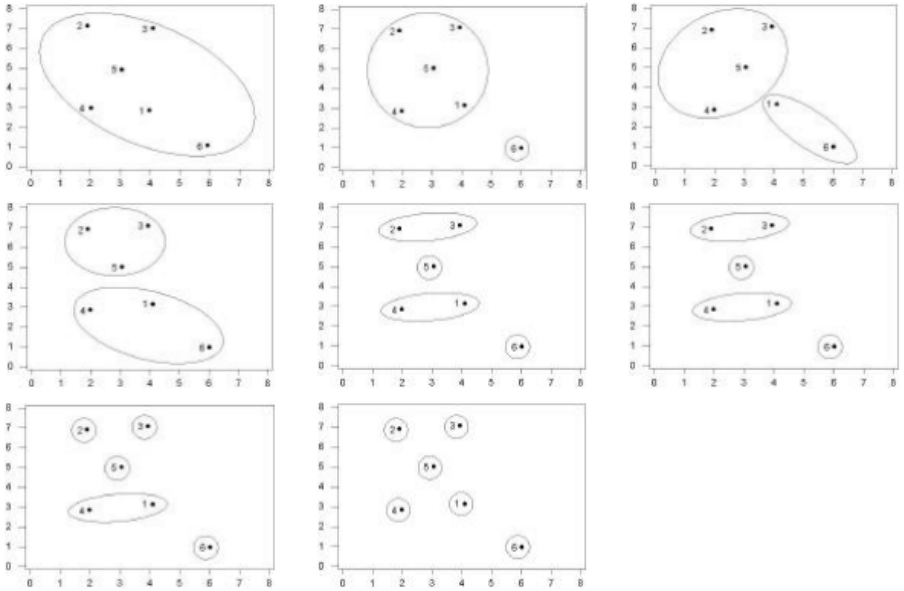

Figura 3.8: Métodos hierárquicos divisivos
(Doni, 2004)

(Zaki & Meira, 2014). Um exemplo de um grupo de dados formado por este método pode ser visto na Figura 3.9.

Métodos baseados em modelos constroem modelos hipotéticos para cada um dos grupos e descobrem em qual deles cada elemento melhor se encaixa. A localização de um grupo pode ser feita construindo uma função de densidade que reflete a distribuição espacial dos pontos (Han & Kamber, 2006). Um exemplo de grupos formados por estes métodos é a Figura 3.10.

Métodos de partição são aqueles que, dado uma quantidade de objetos n, ele constrói k partições, onde cada partição representa um grupo e $k \leq n$. Isto leva a uma classificação que atende a dois requisitos: (1) cada grupo tem pelo menos um objeto e (2)

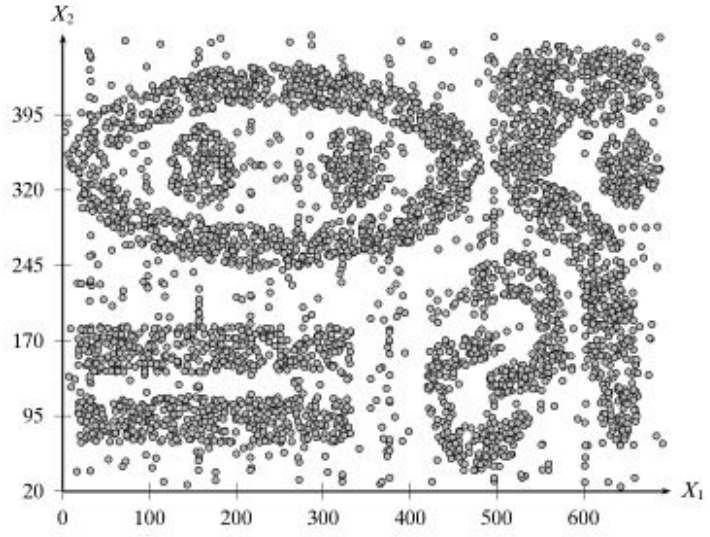

Figura 3.9: Grupo de dados baseado em densidade
(Zaki & Meira, 2014)

cada objeto deve pertencer a um só grupo.

Dado k, o número de partições a serem construídas, o algoritmo cria partições iniciais e, iterativamente, troca os objetos de partição utilizando como critério geral que objetos de um mesma partição são semelhantes e objetos de distintas são diferentes, lembrando que existem ainda vários outros critérios de julgamento.

Para que estes métodos encontrassem o particionamento ótimo seria necessária uma numeração exaustiva das partições, portanto, eles se utilizam de métodos heurísticos, tais como (1) o algoritmo *k-means* onde cada partição é representada por um ponto central do grupo e (2) o *k-medoids* que representa cada partição por um objeto localizado próximo ao centro do mesmo. Estes métodos funcionam bem para encontrar partições esféricas em pequenos a médios grupos de dados. Porém, para a sua aplicação em grandes volumes eles devem ser extendidos.

3.4 O algoritmo *k-medoids*

O algoritmo *k-medoids* tem como parâmetro de entrada um valor k e particiona os dados em k classes para que a semelhança intraclasse seja alta e a semelhança extraclasse seja baixa. A semelhança entre objetos da classes é medida pelo medóide da classe (Han & Kamber, 2006).

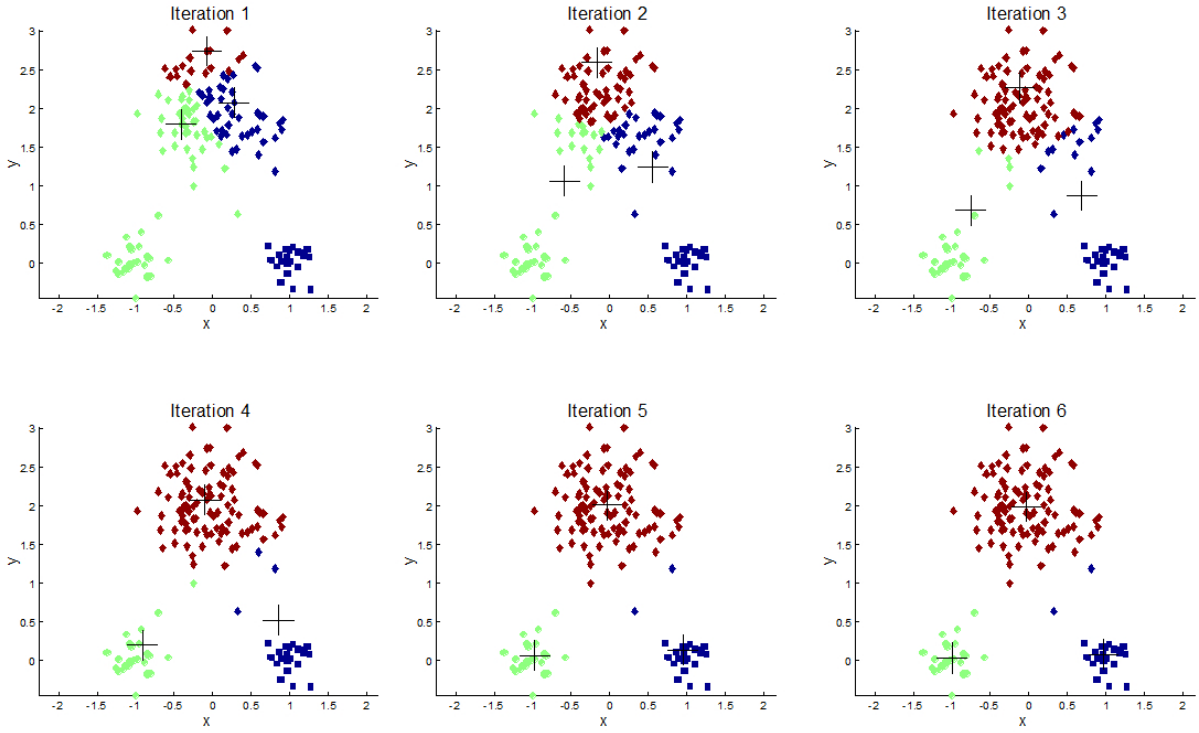

Figura 3.10: Exemplo de formação de grupos por algoritmo baseado em modelo (Fonte: http://datavisualization.blog.com/visible-data/cluster-analysis/)

Inicialmente, o algoritmo *k-medoids* seleciona k elementos aleatórios para serem os medóides dos grupos. Cada objeto restante é atríbuido ao grupo com qual mais se assemelha, esta semelhança é dada pela proximidade deste objeto ao medóide de cada grupo. Então um novo medóide para cada grupo é calculado. Este processo se repete iterativamente até a convergência da função de avaliação. Geralmente, a função utilizada é a função de erro quadrado (3.1), onde "E" é a soma do erro quadrado para todos os objetos na base; "p" é o ponto no espaço representando um dado objeto da classe c; e "m_i" é o centro da classe C. Em outras palavras, para cada objeto em cada classe, a diferença entre ele e o medóide é elevada ao quadrado e as distâncias são somadas. Este critério torna os k grupos o mais compactos e separados possível (Han & Kamber, 2006). O algoritmo está detalhado na Figura 3.11.

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} |p - m_i|^2, \tag{3.1}$$

Suponha um conjunto de objetos dispostos no espaço como o representado na Figura 3.12(a) e assumindo $k = 3$, isto é, teremos três partições ao final. De acordo com o algoritmo descrito anteriormente, arbitrariamente, selecionamos três objetos como

Algoritmo 1: Algoritmo *k-medoids*

Entrada: k: número de partições
Entrada: D: conjunto de dados
Saída: conjunto de k partições
início
  1 - Seleciona k objetos aleatórios em D como medóides
  repita
    1 - distribua cada objeto em uma classe de acordo com a sua similaridade ao medóide
    2 - encontre o medóide de cada classe
  até *sem alterações nos grupos*;
fim

Figura 3.11: Algoritmo *k-medoids*

medóide de cada grupo, marcados com um sinal de "+" e distribuímos os outros de acordo com a sua proximidade de cada um deles. Tal distribuição forma silhuetas arredondadas marcadas pelas linhas pontilhadas mostradas na Figura 3.12(a).

No próximo passo, os medóides são recalculados com base nos objetos já presentes nos grupos e a redistribuição dos elementos restantes ocorre da mesma forma, o que leva a novas silhuetas mostradas na Figura 3.12(b).

A iteração deste processo nos dá a Figura 3.12(c). Este processo de redistribuição dos objetos iterativamente para melhorar a partição é denominado realocação iterativa. Em dado momento, nenhum objeto muda de grupo e então o algoritmo termina, retornando os três grupos encontrados (Han & Kamber, 2006).

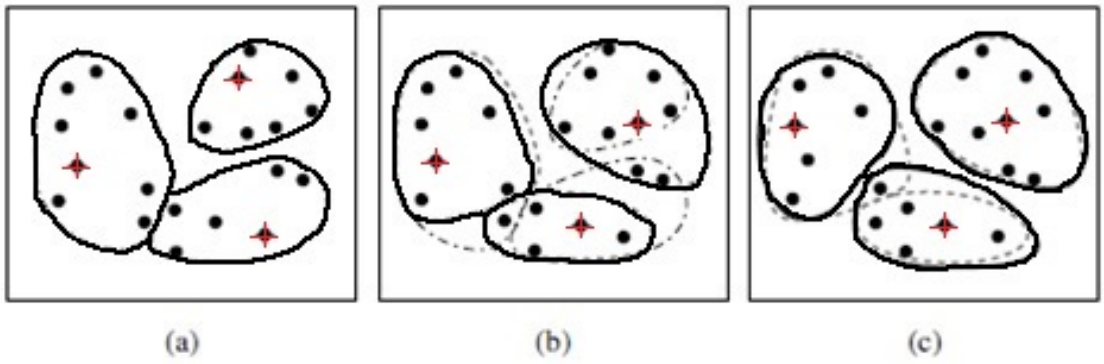

Figura 3.12: Formação de grupos pelo *k-medoids* (Han & Kamber, 2006)

O algoritmo tenta encontrar k partições para minimizar a função do erro quadrado(3.1) e funciona bem quando os dados são compactos e bem-separados uns dos outros. Este método é relativamente escalável e eficiente, pois sua complexidade é baixa O(nkt) com n sendo o total de objetos, k o número de classes e t o número de iterações (Han & Kamber,

2006).

Uma consideração é que ele só se aplica quando o objeto principal de uma classe é definido, o que pode não ser o caso para algumas aplicações. O algoritmo também não serve para identificar classes não convexas ou classes com tamanhos muito diferentes e o fato do número de classes ser um valor de entrada é uma desvantagem (Han & Kamber, 2006).

Apesar destes detalhes, este algoritmo foi escolhido para o trabalho em questão por seu desempenho razoável perante o tamanho da base escolhida e sua facilidade de implementação.

4 Estudo de caso

4.1 Introdução

Após o levantamento dos dados contidos na DBLP e modelagem de um banco de dados capaz de representar essas informações, passamos à fase de aplicação do algoritmo *k-medoids* sobre esta base. O objetivo desta aplicação foi construir uma rede social científica baseada nos relacionamentos de coautoria e encontrar comunidades científicas com interesses comuns de pesquisa.

Os grupos gerados por este processo foram representados em diversos arquivos texto e então submetidos a uma ferramenta de análise que nos permite uma visualização da rede através de diversos níveis (Ströele, 2012). O resultado do processamento feito pela aplicação é uma representação visual dos relacionamentos inter e intrauniversidades que será exibida na Figura 4.1.

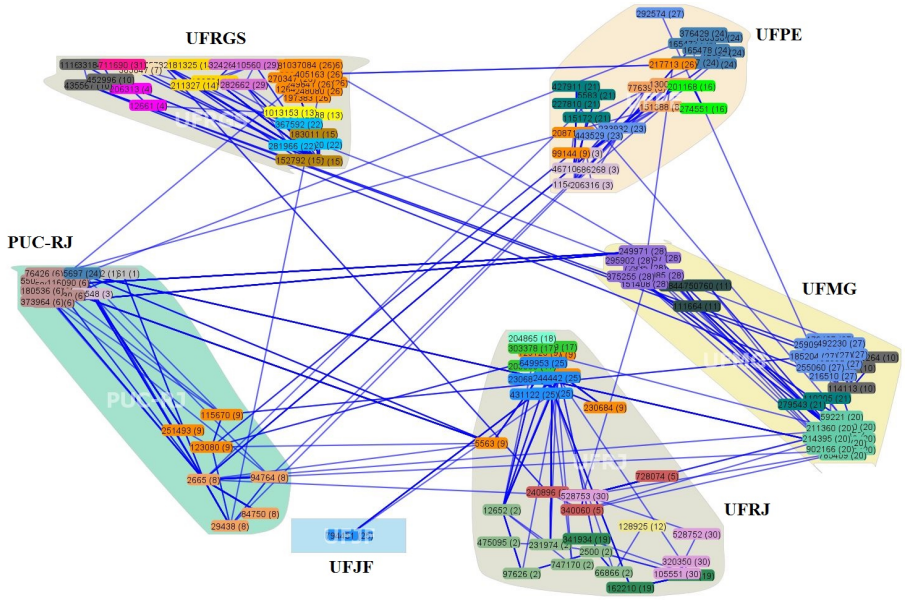

Figura 4.1: Relacionamentos intra e interuniversidades

De forma a possibilitar a validação do estudo, foi selecionado um conjunto de 150 pesquisadores entre todos os disponíveis na DBLP. O critério de seleção foi baseado em sua presença no trabalho realizado previamente por Ströele (2012), desta forma pudemos

comparar os resultados obtidos aos já existentes e comprovados. Com as publicações destes 150 disponíveis na DBLP pudemos extrair 2.622 relacionamentos entre 6 universidades, sendo elas: UFJF, UFMG, UFRJ, UFPE, UFRGS e PUC-RIO.

4.2 Resultados do agrupamento

4.2.1 Definição dos grupos

Para o desenvolvimento do trabalho, como o número de grupos não é definido, precisamos estabelecer um critério para determinar o número ideal de agrupamentos, ou seja, definir em quantos grupos os dados ficam melhor distribuídos.

A análise de agrupamento visa apontar grupos homogêneos que façam com que as diferenças intragrupos sejam minimizadas e a soma das diferenças intergrupos seja maximizada (Aldenderfer & Blashfield, 1984).

Para a realização desta análise, optamos pela utilização do índice PBM, que é definido como se segue:

$$PBM(k) = \left(\frac{1}{k} \times \frac{E_1}{E_k} \times D_k\right)^2 \tag{4.1}$$

onde k é o número de grupos e

$$E_k = \sum_{i=1}^{k} E_i$$

$$E_i = \sum_{t=1}^{n} u_{ti} d(x_t, \bar{x_i}), \text{ tal que} \tag{4.2}$$

$$D_k = \max_{i,j=1}^{k} d(\bar{x_i}, \bar{x_j}) \tag{4.2}$$

Temos que **n** é o número de pontos no conjunto de dados, $U(X) = [u_{ti}]n \times k$ é a matriz que indica o grupo de cada elemento e $\bar{x_i}$ é o medóide do i-ésimo grupo. Queremos maximizar o índice PBM para obter o melhor número de grupos, o que significa que o valor máximo do índice indica o melhor particionamento possível (Pakhira, Bandyopadhyay and

Maulik).

O índice BPM se dá por três fatores: $\frac{1}{k}$, $\frac{E_1}{E_k}$ e D_k. O primeiro reduzirá o índice quando aumentarmos o número de grupos (Ströele, 2012).

Para o segundo fator temos que

> "O denominador do segundo fator é a soma dos desvios da posição de cada objeto no espaço de variáveis ao medóide do seu respectivo grupo. E o numerador E_1 é constante, sendo a soma dos desvios para todos os objetos alocados em uma única partição. Assim, $\frac{E_1}{E_k}$ é diretamente proporcional à homogeneidade dos grupos formados. Consequentemente, quanto menor E_k , maior a homogeneidade dos grupos e maior é o valor do índice"(Ströele, 2012).

O fator D_k representa "a distância máxima entre o medóide de dois dos k grupos formados. Quanto maior a distância entre os grupos, maior é o índice de qualidade"(Ströele, 2012).

A variação do índice para o trabalho realizado está representada na Figura 4.2. Analisando o gráfico, podemos verificar que o valor ideal de agrupamentos para o caso em questão é 31, pois este é o número que gera o maior valor de índice PBM e, portanto, o melhor particionamento.

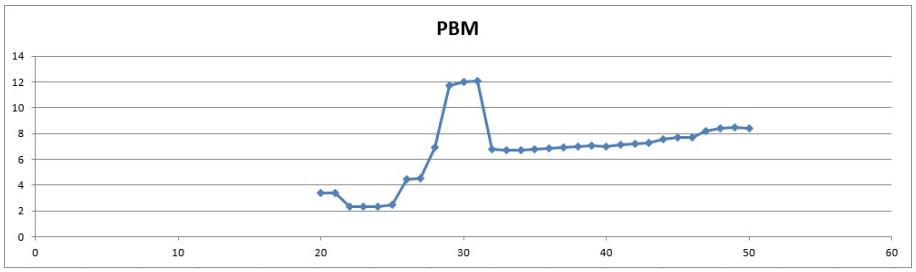

Figura 4.2: Variação do índice PBM

4.2.2 Comunidades científicas

Antes de falarmos dos grupos obtidos vale ressaltar que, a fim de realizar uma validação correta dos dados, optamos por utilizar um pequeno conjunto de pesquisadores previamente utilizados em Ströele (2012). Sendo assim, os resultados devem ser analisados considerando que estes representam uma parcela muito pequena do universo de pesquisadores disponíveis na DBLP.

Aplicando a Mineração de Dados a este reduzido grupo de pesquisadores, obtivemos uma representação visual de seus relacionamentos. Esta representação está eviden-

ciada na Figura 4.3, que mostra cada uma das seis universidades utilizadas no grupo de análise e os relacionamentos entre elas.

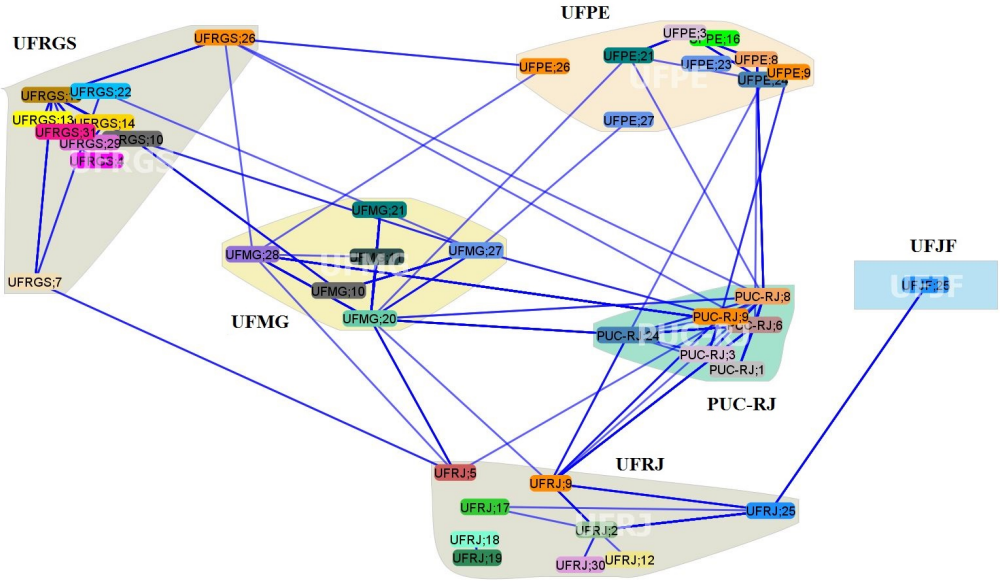

Figura 4.3: Relacionamentos entre universidades

Analisando a imagem, percebemos que a PUC-RJ se relaciona muito pouco com a UFRGS e, apesar da proximidade geográfica, quando comparamos o relacionamento dela com a UFRJ, com a UFPE e UFMG a quantidade é semelhante. A UFRGS se relaciona de forma fraca e parecida com todas as outras instituições, enquanto a UFPE tende a ter um maior relacionamento com a PUC-RJ e UFMG, não se relacionando com a UFRJ. A UFRJ tende a se relacionar com instituições mais próximas geograficamente (UFJF, PUC-RJ e UFMG). No caso da UFJF, temos apenas um relacionamento com a UFRJ, porém tínhamos apenas um pesquisador da instituição incluído no grupo, portanto isso não pode ser analisado de forma mais elaborada.

5 Considerações finais

5.1 Objetivos alcançados

Um dos objetivos do trabalho era realizar a extração de todos os dados da DBLP e modelá-los em um Bando de Dados relacional. Ao fim do trabalho, podemos dizer que este objetivo foi concluído, pois conseguimos realizar a carga do banco, inclusive nos possibilitando deixá-la preparada para uma expansão do trabalho.

Tendo realizado a extração e carga com sucesso, conseguimos alcançar nosso objetivo principal que é a geração da rede social composta pelos pesquisadores que foram alvo do estudo. Com a rede pronta, pudemos realizar uma análise superficial, visto que esse não era o foco do trabalho, sobre como se dão estas relações de publicação em seis universidades do país.

A próxima etapa concluída foi a validação da rede construída a partir do trabalho realizado em Ströele (2012). Esta validação foi feita comparando os agrupamentos gerados neste trabalho com os do trabalho prévio. Já esperávamos que os grupos não fossem iguais devido, principalmente, à diferença cronológica entre eles, porém, verificamos um alto grau de semelhança, o que foi o suficiente para concluirmos que a extração realizada estava correta.

Por fim, talvez o maior objetivo atingido por este trabalho tenha sido o desenvolvimento de uma ferramenta que possibilita a carga dos dados da DBLP, de forma que a base gerada possa ser atualizada constantemente para a realização de futuras pesquisas.

5.2 Problemas encontrados

Um dos maiores problemas encontrados durante a realização do trabalho, foi a falta de documentação disponível sobre a DBLP, basicamente o que tínhamos de informação era o trabalho Ley (2009), que notamos estar bastante incompleto.

Decorrente da falta de documentação, deparamo-nos com um outro problema ao

realizar a carga da base: a falta de informação sobre o tamanho máximo e tipo dos campos encontrados no XML. Por se tratar de um arquivo grande, o que elevava muito o tempo de processamento, tivemos que adotar uma postura de geração de *logs* e de tentativa e erro, até que conseguimos realizar uma carga completa.

Como tratamos uma quantidade muito grande de dados, notamos que realizar toda a carga de uma única vez seria algo extremamente demorado. Assim, adotamos a abordagem de colocar restrições no próprio algoritmo de carga, de forma a processar apenas os dados de um determinado ano por vez, desta forma conseguimos carregar todo o arquivo na base de forma incremental.

Outra dificuldade foram os casos em que o mesmo pesquisador era citado com dois nomes diferentes. Como limitamos a análise a um grupo restrito de pesquisadores, optamos por eliminar essas duplicidades de forma manual, porém a modelagem da base já prevê um relacionamento para representar estes casos, como citado anteriormente.

5.3 Trabalhos futuros

No presente trabalho, nós focamos apenas na construção da rede baseada em relacionamentos de coautoria, sendo assim, como trabalho futuro, sugerimos a expansão desta análise para outros tipos de relacionamento, por exemplo, relações de citação em trabalhos de terceiros.

Outra sugestão de trabalho futuro seria a integração de informações de outras bases para complementar os dados não presentes na DBLP, a fim de aperfeiçoar o trabalho realizado.

Referências Bibliográficas

M. S. Aldenderfer, R. K. B. **Cluster analysis**. 1984.

Doni, M. V. **Análise de cluster: Métodos hierárquicos de particionamento**. 2004.

David Hand, Heikki Manilla, P. S. **Principles of data mining**. 2001.

Jiawei Han, M. K. **Data mining: Concepts and techniques**. 2006.

Keneth Laudon, J. L. **Sistemas de informação gerenciais**. 2011.

Michael Ley, P. R. **Maintaining an online bibliographical database: The problem of data quality**. 2006.

Ley, M. **Dblp - some lessons learned**. 2009.

TsauYoung Lin, N. J. C. **Rough sets and data mining: Analysis of imprecise data**. 1996.

Sushmita Mitra, Sankar K. Pal, P. M. **Data mining in soft computing framework: A survey**. 2002.

Malay K. Pakhira, Sanghamitra Bandyopadhyay, U. M. **Validity index for crisp and fuzzy clusters. pattern recognition**. 2004.

Rezende, S. O. **Mineração de dados**. 2005.

Ströele, V. **Análise de redes sociais científicas**. 2012.

Victor Ströele, Geraldo Zimbrão, J. M. S. **Análise de redes sociais científicas: Modelagem multi relacional**. 2012.

Ian H. Witten, E. F. **Data mining: Practical machine learning tools and techniques with java implementations**. 1999.

Mohammed J. Zaki, W. M. **Data mining and analysis: Fundamental concepts and algorithms**. 2014.

Zhou, Z.-H. **Three perspectives of data mining**. 2003.