UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Câmpus de São José do Rio Preto

Thatiane Kawabata

Detecção de *outliers* espaciais: refinamento de similaridade e desempenho

São José do Rio Preto
2015

Thatiane Kawabata

Detecção de *outliers* espaciais: refinamento de similaridade e desempenho

Dissertação apresentada como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação, junto ao Programa de Pós-Graduação em Ciência da Computação, Área de Concentração – Computação Aplicada, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista "Júlio de Mesquita Filho", Campus de São José do Rio Preto.

Orientador: Prof. Dr. Carlos Roberto Valêncio

São José do Rio Preto
2015

Kawabata, Thatiane.
Detecção de outliers espaciais : refinamento de similaridade e desempenho / Thatiane Kawabata. -- São José do Rio Preto, 2015
79 f. : il., tabs.

Orientador: Carlos Roberto Valêncio
Dissertação (mestrado) – Universidade Estadual Paulista "Júlio de Mesquita Filho", Instituto de Biociências, Letras e Ciências Exatas

1. Computação. 2. Sistemas de informação geográfica. 3. Sistemas de dados espaciais. 4. Mineração de dados (Computação) 5. Algoritmos de computador. I. Valêncio, Carlos Roberto. II. Universidade Estadual Paulista "Júlio de Mesquita Filho". Instituto de Biociências, Letras e Ciências Exatas. III. Título.

CDU – 518.72:91

Ficha catalográfica elaborada pela Biblioteca do IBILCE
UNESP - Câmpus de São José do Rio Preto

Thatiane Kawabata

Detecção de *outliers* espaciais: refinamento de similaridade e desempenho

Dissertação apresentada como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação, junto ao Programa de Pós-Graduação em Ciência da Computação, Área de Concentração – Computação Aplicada, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista "Júlio de Mesquita Filho", Campus de São José do Rio Preto.

Comissão Examinadora

Prof. Dr. Prof. Dr. Carlos Roberto Valêncio
UNESP – São José do Rio Preto, SP
Orientador

Profa. Dra. Rogéria Cristiane Gratão de Souza
UNESP – São José do Rio Preto, SP

Prof. Dr. Enzo Seraphim
UNIFEI – Itajubá, MG

São José do Rio Preto
06 Março de 2015

AGRADECIMENTOS

Agradeço primeiramente a Deus, que sempre me protegeu e deu forças para superar todos os obstáculos.

Aos meus pais, Maria e Orlando, que me proporcionaram todo o suporte para que eu pudesse fazer a minha pós graduação, também pelo carinho e amor. Muito obrigada.

A minha irmã Priscilla pelo carinho e amor.

Ao meu noivo, André, pelo apoio, carinho, amor e por sempre acreditar que eu seria capaz de atingir meus objetivos.

Aos meus amigos que durante esses anos compartilharam suas conquistas e sempre torceram pelas minhas.

A todos do Grupo de Banco de Dados (GBD) que contribuíram para o meu aprendizado.

Ao meu orientador Prof. Dr. Carlos Roberto Valêncio, pela disposição e orientação ao incentivar meu trabalho.

À minha família

A diferença entre o possível e o impossível está na vontade humana.
Louis Pasteur

RESUMO

O avanço e desenvolvimento de tecnologias utilizadas na coleta de informações georreferenciáveis proporcionou um aumento na quantidade de dados espaciais armazenados nas bases de dados. Isso também acarretou muitos problemas, comuns em grandes bases de dados, tais como: redundância de dados, dados incompletos, valores desconhecidos e *outliers*. Com o objetivo de obter informações relevantes dos dados espaciais, a aplicação de algoritmos de prospecção de dados espaciais, principalmente os algoritmos de agrupamentos espaciais, tornou-se uma prática bastante recorrente em todo cenário mundial. Por outro lado, muitos algoritmos atuais desconsideram a presença de *outliers* locais em dados espaciais, ou apenas consideram a sua localidade em relação aos demais dados da base, o que pode gerar resultados inconsistentes e dificultar a extração de conhecimento. Dessa forma, com o propósito de contribuir nesse sentido, o trabalho visa elaborar um levantamento de informações relacionadas a prospecção de dados espaciais, detecção de *outliers* convencionais e espaciais, assim como, apresentar os principais trabalhos no estado da arte. Por fim, propõe-se disponibilizar uma abordagem configurável e portável aos resultados dos algoritmos de agrupamento espaciais, na qual inclui-se uma melhoria em um algoritmo de detecção de *outliers* espaciais, que visa a prospecção de informações no conjunto de dados.

Palavras-chave: detecção de *outliers* espaciais; prospecção de dados espaciais; agrupamento de dados espaciais*; spatial data mining*.

ABSTRACT

The progress and development of technologies used to collect spatial information resulted in an increase in the amount of spatial data stored in databases. This also caused many problems, common in large databases, such as data redundancy, incomplete data, unknown values and outliers. Aiming to obtain relevant information from spatial data, the application of algorithms for exploration of spatial data, especially spatial clusters of algorithms, has become a fairly common practice across the world scene. Moreover, many current algorithms ignore the presence of local outliers in spatial data, or just consider your location in relation to other data in base, which can cause inconsistent results and complicate the extraction of knowledge. Thus, in order to contribute to this, the work aims to develop a survey of information related to exploration of spatial data, detection of conventional and spatial outliers, as well as, present the main work in state of the art. Finally, we propose to provide a portable and configurable algorithms to the results of spatial clustering approach, which includes an improvement on an algorithm to detect spatial outliers, aimed at prospecting for information in the dataset.

Keywords: spatial outlier detection; spatial data mining; spatial clustering.

Sumário

Lista de Figuras x
Lista de Tabelas xi
Lista de Abreviaturas e Siglas xii
Capítulo 1 Introdução 1
1.1 Considerações Iniciais 1
1.2 Motivação e Escopo 2
1.3 Objetivos e Metodologias 2
1.4 Organização do Trabalho 3
Capítulo 2 Fundamentação Teórica 5
2.1 Considerações Iniciais 5
2.2 O processo de prospecção de dados 6
2.3 Dados Espaciais 7
2.4 *Spatial Data Mining* 8
2.4.1 Classificação espacial 8
2.4.2 Regras de associação espacial 9
2.4.3 Tendência espacial 9
2.4.4 Caracterização espacial 9
2.4.5 Agrupamentos de dados espaciais 10
2.5 *Outliers* 11
2.6 Detecção de *outliers* espaciais 12
2.7 Estratégia para melhorar o desempenho dos algoritmos 13
2.8 Considerações Finais 14
Capítulo 3 Métodos para Detecção de *Outlier* 15
3.1 Considerações iniciais 15
3.2 Métodos de visualização 16
3.3 Métodos Estatísticos 17
3.4 Métodos de Grafos 19
3.5 Detecção de *outliers* em agrupamentos de dados 20
3.6 Trabalhos no estado da arte 21
3.7 Algoritmo PCR-SCOD 22
3.8 Estudos Comparativos entre os algoritmos de detecção de *outliers* 23
3.9 Considerações Finais 24
Capítulo 4 Refinamento de similaridade e desempenho: Algoritmo de detecção de *outliers* espaciais 25
4.1 Considerações Iniciais 25
4.2 O algoritmo *PCR-SCOD* 26
4.3 Trabalho desenvolvido 29
4.3.1 A Estratégia para refinamento de similaridade nos agrupamentos 29
4.3.2 Otimização do desempenho do algoritmo *PCR-SCOD-MR* 30
4.3.3 O algoritmo desenvolvido *PCR-SCOD-MR* 32
4.3.4 Qualidade dos resultados 36
4.3.5 Ferramenta para agrupamento de dados espaciais 38

4.4 Considerações Finais 39
Capítulo 5 Experimentos e Resultados 40
5.1 Considerações Iniciais 40
5.2 Experimento da detecção de *outliers PCR-SCOD-MR* 40
5.2.1 Dois atributos não-espaciais 41
5.2.2 Três atributos não-espaciais 43
5.3 Experimento da abordagem 45
5.3.1 Abordagem com algoritmo *CHSMST+* 46
5.3.2 Abordagem com o algoritmo *k-means* 51
5.4 Comparação da abordagem com PCR-SCOD-MR em relação a outro algoritmo de detecção de *outliers* espaciais 53
5.5 Análise do Desempenho do algoritmo PCR-SCOD-MR 54
5.6 Considerações Finais 57
Capítulo 6 Conclusões 59
6.1 Considerações 59
6.3 Trabalhos Futuros 61
Referências Bibliográficas 62

Lista de Figuras

Figura 1 Processo KDD (CARREIRA, 2008, PIATETSKY-SHAPIRO, 2014) 6
Figura 2 Exemplos de tipos de dados espaciais (YEUNG; HALL, 2007, MEDEIROS, 2014) 7
Figura 3 - Exemplo de *outliers* extraído do livro de (HAN; KAMBER; PEI, 2011) 11
Figura 4 Modelo *MapReduce* para o exemplo de contagem de palavras 13
Figura 5 Um exemplo de visualização de *outliers* em *Scatterplot* (adaptada de SHEKHAR; LU; ZHANG, 2003) 16
Figura 6 Um exemplo de visualização de *outliers* em Moran Scatterplot (adaptada de SHEKHAR; LU; ZHANG, 2003) 17
Figura 7 Um exemplo de visualização de *outliers* utilizando-se o Algoritmo Z (adaptada de SHEKHAR; LU; ZHANG, 2003) 18
Figura 8 Resultado dos algoritmos POD e ROD (KOU; LU; SANTOS, 2007) 19
Figura 9 Diagrama do algoritmo de duas fases com a utilização da base de dados KDD Cup99 (adaptada de SONGMA et al.,2012) 20
Figura 10 Detecção de *outliers* categóricos (LIU; CHEN; LU, 2013) 22
Figura 11 Fluxograma do algoritmo *PCR-SCOD* 27
Figura 12 Distribuição do conjunto de dados utilizado no experimento (LIU; CHEN; LU, 2013) 28
Figura 13 Proposta de fluxograma para a abordagem do trabalho 30
Figura 14 Fluxograma do algoritmo *PCR-SCOD-MR* 36
Figura 15 Arquitetura da ferramenta *Web* 39
Figura 16 Consulta para verificação dos objetos modificados 42
Figura 17 Consulta do agrupamento para três atributos não-espaciais 44
Figura 18 Gráfico e tabela da comparação entre o resultado com e sem a abordagem para dois atributos não-espaciais 46
Figura 19 Agrupamentos com os *outliers* destacados para dois atributos não-espaciais 47
Figura 20 Agrupamento em José Bonifácio 48
Figura 21 Gráfico e tabela da comparação entre o resultado com e sem a abordagem para três atributos não-espaciais 49
Figura 22 Agrupamentos com os outliers destacados para dois atributos não-espaciais 50
Figura 23 Agrupamento de acidentes de trabalho em São José do Rio Preto 50
Figura 24 Gráfico e tabela da comparação entre o resultado com e sem a abordagem para *k-means* 51
Figura 25 Agrupamentos com os *outliers* destacados para o resultado do *k-means* .. 52
Figura 26 Agrupamento de acidentes de trabalho em São José do Rio Preto para o algoritmo *k-means* 53
Figura 27 Gráfico do tempo de execução 56
Figura 28 Gráfico da memória máxima utilizada durante a execução 57

Lista de Tabelas

Tabela 1 Estudos comparativos 23
Tabela 2 Número de objetos nos agrupamentos 41
Tabela 3 Número de objetos modificados nos agrupamentos 41
Tabela 4 Valor dos *outliers* 41
Tabela 5 Informação dos *outliers* nos agrupamentos para 2% de objetos modificados e dois atributos não-espaciais 42
Tabela 6 Informação dos *outliers* nos agrupamentos para 5% dos objetos modificados e dois atributos não-espaciais 43
Tabela 7 Informação dos *outliers* nos agrupamentos para 2% dos objetos modificados e três atributos não-espaciais 43
Tabela 8 Informação dos *outliers* nos agrupamentos para 5% dos objetos modificados e três atributos não-espaciais 43
Tabela 9 *Outliers* identificados 45
Tabela 10 Comparação da similaridade dos agrupamentos 54
Tabela 11 Tempo médio de execução 55
Tabela 12 Estudos comparativos com o algoritmo PCR-SCOD-MR 60

Lista de Abreviaturas e Siglas

API	*Application Program Interface*
AVF	*Attribute Value Frequency*
GPS	*Global Positioning System*
GPU	*Graphics Processing Unit*
KDD	*Knowledge Discovery in Databases*
PCR	*Pair Correlation Ratio*
POD	*Point Outliers Detection*
ROD	*Region Outlier Detection*
SGBD	Sistema de Gerenciamento de Banco de Dados
SCOD	*Spatial Categorical Outlier Detection*
SDM	*Spatial Data Mining*
SIVAT	Sistema de Vigilância de Acidentes de Trabalho
SPF	*Spatial Pair Function*
WDA	*Weighted Difference Algorithm*

Capítulo 1
Introdução

1.1 Considerações Iniciais

O surgimento de novas tecnologias voltadas aos estudos espaciais proporcionou um aumento dos dados georreferenciáveis armazenados nas bases de dados espaciais (MENNIS; GUO, 2009, JIN; MIAO, 2010). Dessa forma, o aprimoramento de técnicas para extração de conhecimentos, advindas das técnicas de *Data Mining* convencional, tornou-se uma atividade imprescindível e importante no cenário mundial (JIN; MIAO, 2010). Essa área é conhecida como *Spatial Data Mining* (SDM) ou *Knowledge Discovery from Spatial Database*, e possui a finalidade de descobrir padrões implícitos nas bases de dados espaciais (SHENGWU, 2011).

Contudo em grandes bases de dados existem alguns obstáculos que dificultam a extração de conhecimento como: redundância, valores não especificados, dados incompletos e *outliers* (BAKAR et al., 2006).

Em SDM, o objetivo de detecção de *outlier* é descobrir objetos cujos atributos não-espaciais se diferenciam dos outros que estão em sua vizinhança (SHEKHAR; LU; ZHANG, 2003, CHEN et al., 2007). Tal técnica tem obtido resultados importantes para diversas áreas e tem proporcionado uma intensificação nos estudos a fim de melhorar a extração de conhecimento significativo e passível de ser utilizado (XI, 2008).

1.2 Motivação e Escopo

A prospecção de dados espaciais, em busca de conhecimento útil, tem sido uma tarefa bastante difundida atualmente para prover informações consistentes e que auxiliem nas tomadas de decisões (MENNIS; GUO, 2009, JIN; MIAO, 2010).

Por outro lado por ser uma área recente e os dados possuírem características complexas, alguns pontos devem ser ressaltados, tais como (JIN; MIAO, 2010): a maioria dos algoritmos de *Spatial Data Mining* são derivados dos algoritmos convencionais de prospecção de dados e não consideram os atributos espaciais quanto os não-espaciais, a eficiência para tratar objetos com mais de uma dimensão ainda é baixa, não existe uma padronização da linguagem de manipulação de dados espaciais, a interação do usuário com o sistema para a descoberta de conhecimento ainda é bastante trabalhosa, e ainda, muitos trabalhos nessa área são realizados para um domínio específico de aplicação.

Dessa forma, diversas técnicas foram propostas para auxiliarem na tarefa de prospecção em dados espaciais e tentar suprir as dificuldades de se tratar dados espaciais. Uma das técnicas que se destaca é o de agrupamentos de dados, pela alta aplicabilidade em diferentes áreas e também por não ser necessário um conhecimento prévio da base de dados (JIN; MIAO, 2010). Tal técnica consiste em agrupar objetos próximos, por meio do cálculo da distância, e similares para a formação dos agrupamentos (HAN; KAMBER; PEI, 2011).

Apesar disso, os resultados obtidos podem ser incorretos ou de difícil compreensão quando existem *outliers* espaciais nos agrupamentos. Uma vez que os algoritmos de agrupamento espaciais não detectam *outliers* na formação dos agrupamentos ou apenas detectam os *outliers* pelo atributo espacial, conhecido como *outlier* global (CHAWLA; GIONIS, 2013).

Portanto, a aplicação de detecção de *outliers* espaciais nos agrupamentos gerados pelos algoritmos tendem a proporcionar resultados mais interessantes e mais fáceis de extração de conhecimento para os usuários finais diante do grande volume de dados.

1.3 Objetivos e Metodologias

Nesse trabalho é proposto um aprimoramento de um algoritmo de detecção de *outliers* espaciais com a estratégia de melhoria de desempenho *MapReduce*, de modo que seja adaptável a qualquer resultado de algoritmos de agrupamentos espaciais, com o objetivo de melhorar a similaridade dentro dos mesmos. Para isso será criada uma ferramenta portável e

adaptável, no qual será possível a execução dos algoritmos de agrupamento espaciais, aplicação da técnica de detecção de *outliers*, considerando tanto a sua localidade como os atributos não-espaciais, e a visualização dos objetos georreferenciados.

Com isso, pretende-se descobrir *outliers* locais que influenciam negativamente na formação do agrupamento e retirá-los dos agrupamentos de modo a melhorar a similaridade e refinar os resultados.

As hipóteses para o trabalho são:

- Os resultados obtidos pelos algoritmos de agrupamentos espaciais não são significativos quando se refere a *outliers* locais, ou seja, considerando a localidade e os atributos não-espaciais;

- A detecção de *outliers* espaciais dentro dos agrupamentos visa uma melhoria na similaridade para o refinamento dos resultados;

- O tempo de execução será menor com a utilização do modelo de programação *MapReduce*.

A metodologia de desenvolvimento utilizada nesse trabalho envolve as etapas de levantamento bibliográfico das técnicas de prospecção de dados espaciais, estudos comparativos de algoritmos de detecção de *outliers* espaciais no estado da arte e desenvolvimento do algoritmo aprimorado e da construção de uma ferramenta *Web*.

Para comprovar que a abordagem escolhida obteve os resultados esperados serão realizados experimentos com os resultados gerados pelos algoritmos de agrupamentos espaciais *k-means e CHSMST+,* no qual o primeiro utiliza o método de particionamento e o segundo o método de grafos para geração dos agrupamentos, de modo a verificar que a abordagem pode ser aplicada em qualquer técnica de agrupamentos espaciais. Por fim, serão realizados vários experimentos para validação da estratégia de desempenho do algoritmo de detecção de *outliers* espaciais escolhido.

1.4 Organização do Trabalho

A seguir é apresentada a estrutura deste trabalho:

- **Capítulo 1** Introdução – são apresentadas as principais motivações do trabalho, assim como os objetivos e metodologia.

- **Capítulo 2** Fundamentação teórica – são apresentados os principais conceitos envolvendo prospecção de dados espaciais e detecção de *outliers*, assim como as estratégias para melhoria de desempenho dos algoritmos.

- **Capítulo 3** Métodos para detecção de *outliers* espaciais – serão apresentados os principais métodos para detecção de *outliers* espaciais, bem como os algoritmos que envolvem essa área.
- **Capítulo 4** Refinamento de similaridade e desempenho: Algoritmo de detecção de *outliers* espaciais – são apresentadas as etapas que contemplam a abordagem proposta.
- **Capítulo 5** Experimentos e resultados – contempla os experimentos realizados para verificação das hipóteses geradas no trabalho.
- **Capítulo 6** Considerações finais – são apresentadas as considerações referentes ao trabalho apresentado.

Capítulo 2
Fundamentação Teórica

2.1 Considerações Iniciais

O progresso das tecnologias voltadas para coleta de informações georreferenciadas, tais como (MENNIS; GUO, 2009, SHENGWU, 2011): GPS, sistemas de localização espacial, sistemas de informações geográficas, sensoriamentos remotos, entre outros, ocasionou um aumento significativo na quantidade de dados espaciais armazenados (MENNIS; GUO, 2009, JIN; MIAO, 2010).

Com isso, o processo de prospecção de dados em busca de conhecimento útil, conhecida como *Spatial Data Mining*, tornou-se um processo imprescindível para obter padrões implícitos nas bases de dados espaciais (SHENGWU, 2011).

Para tal processo, a detecção de *outliers* é um importante passo para encontrar objetos distintos dos objetos em sua volta (HUANG; QIN, 2004).

Nesse capítulo, são apresentados os principais conceitos que envolvem o processo de *Data Mining*, dados espaciais, *Spatial Data Mining* e suas técnicas, assim como os conceitos de *outliers* e *outliers* espaciais.

Por fim, é apresentada estratégia para melhorar o desempenho de algoritmos e quais suas principais vantagens.

2.2 O processo de prospecção de dados

A prospecção de dados, ou do inglês K*nowledge Discovery in Databases (KDD),* pode ser entendida como um processo não trivial, para a descoberta de conhecimento passível de ser utilizado, por meio dos padrões implícitos contidos nas bases de dados (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996, PIATETSKY-SHAPIRO, 2014).

Este processo pode ser dividido em três grandes etapas: pré-processamento dos dados, *data mining* e pós processamento dos resultados, como apresentado na Figura 1.



Figura 1 Processo KDD (CARREIRA, 2008, PIATETSKY-SHAPIRO, 2014)

Na primeira etapa, conhecida como pré-processamento, ocorre a preparação da base de dados para a prospecção de dados. As técnicas mais comuns aplicadas nessa etapa são: limpeza das inconsistências e duplicações de dados, integração de fontes distintas, normalização e seleção de dados.

Após o pré-processamento dos dados, a etapa de *Data Mining* é realizada. As técnicas de *Data Mining* automatizam a tarefa de descobrir padrões implícitos por meio da execução dos algoritmos, que podem ser divididos em: agrupamento, classificação e regressão, predição e regras de associação.

Por fim, os resultados obtidos na fase anterior são apresentados na etapa de pós-processamento. Essa etapa tem como objetivo disponibilizar os resultados ao usuário de forma que sejam possíveis a extração de conhecimento e as tomadas de decisões. Uma das técnicas que pode ser utilizada é a de visualização dos dados, no qual os dados são apresentados graficamente (QIANG; WEI; HANFEI, 2010).

2.3 Dados Espaciais

Um dado espacial é um objeto que possui atributos espaciais e não-espaciais. Os atributos espaciais podem ser a localização, uma propriedade topológica, um ponto, uma reta ou um polígono, como apresentadas nas ilustrações da Figura 2. Já os atributos não-espaciais podem ser do tipo alfanumérico, como por exemplo: tamanho, peso, nome, entre outros (LU; CHEN; KOU, 2003b).

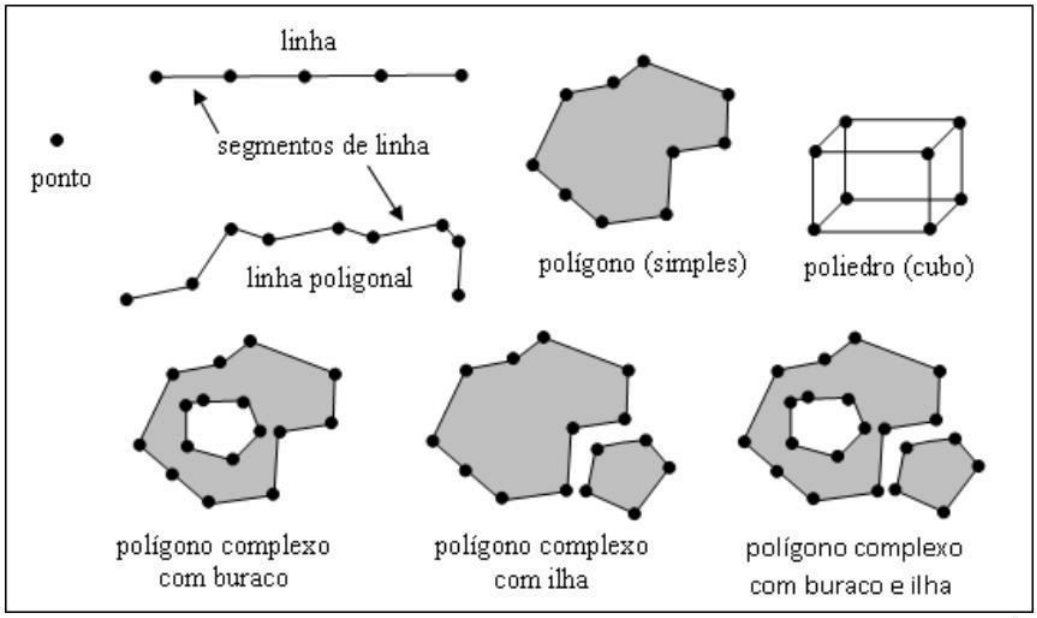


Figura 2 Exemplos de tipos de dados espaciais (YEUNG; HALL, 2007, MEDEIROS, 2014)

Além disso, entre os próprios dados espaciais pode existir uma relação, como por exemplo: distância espacial com outros objetos adjacentes, relação topológica e relação de sentido (JIN; MIAO, 2010).

As características principais dos dados espaciais e que também acarretam dificuldades de manipulações são (JIN; MIAO, 2010, SHENGWU, 2011): grande quantidade de cálculos para aplicação de algoritmos; relação não linear entre os atributos espaciais; características escalares; múltiplas dimensões espaciais; ambiguidade de dados e informações; e dificuldade de obtenção dos dados espaciais, o que ocasiona dados incompletos armazenados nas bases de dados espaciais.

Dessa forma, para possibilitar a manipulação desses dados, alguns sistemas gerenciadores de banco de dados relacionais (SGBDs) criaram extensões específicas, como o *PostGis* que é uma extensão do SGBD *PostgreSQL* (POSTGRESQL, 2014). Essa extensão

disponibiliza funções de topologia, transformação de coordenadas e validação dos dados espaciais.

2.4 *Spatial Data Mining*

Para suprir a necessidade de extração de conhecimento das grandes bases de dados espaciais foi criado o processo de *Spatial Data Mining*, cujo objetivo principal é encontrar padrões implícitos e relações entre os atributos espaciais e não-espaciais (JIN; MIAO, 2010, (PATTABIRAMAN et al., 2009).

No entanto, esse grande volume apresenta desafios para a prospecção de dados espaciais, de tal modo que os resultados obtidos sejam significativos (JIN; MIAO, 2010):

- A complexidade das estruturas dos dados espaciais e seus relacionamentos dificulta a armazenagem, processamento e manipulação;
- Os algoritmos de prospecção de dados espaciais são derivados das técnicas convencionais de análise de dados e não tratam as peculiaridades dos dados espaciais, o que ocasiona uma baixa eficiência nos resultados;
- Os bancos de dados não possuem uma linguagem padronizada para efetuar consultas dos dados espaciais;
- Os processos de extração de conhecimento por meio dos resultados obtidos são muito dependentes de uma análise humana;
- As ferramentas, são geralmente, focadas para um domínio específico de aplicação.

Dessa forma, as técnicas para extração de conhecimento no âmbito espacial tem sido intensificadas de modo a considerar os pontos citados anteriormente. Essas técnicas podem ser classificadas em: classificação espacial, regras de associação espacial, tendência espacial, caracterização espacial e agrupamento de dados espaciais.

2.4.1 Classificação espacial

A classificação espacial tem como objetivo definir um modelo de classificação de dados de acordo com as características espaciais e não-espaciais. Para isso, é necessário

primeiramente um conjunto de dados para treinamento, no qual podem ser utilizados modelos de redes neurais, redes artificiais, árvores de decisão entre outros (HAN; KAMBER; PEI, 2011).

Exemplos de áreas de aplicação dessas técnicas são: área de sensoriamento remoto (CLEVE, 2008, MENNIS; GUO, 2009), classificação de objetos no espaço sideral (KOPERSKI; HAN, 1997) e classificação de vegetação (CLEVE et al., 2008).

2.4.2 Regras de associação espacial

A técnica de regra de associação espacial consiste em criar regras que determinam o quanto um item está correlacionado a outros. Essa técnica visa também a descoberta de quais padrões são frequentes nas bases de dados.

As regras de associação são formadas por predicados espaciais e não-espaciais. Estes predicados espaciais podem ser: relação de topologia, orientações espaciais, informações de distância entre outros. A representação de uma regra de associação, portanto é: S $\rightarrow$ N ou N $\rightarrow$ S, no qual S é o conjunto de predicados espaciais e N é o conjunto de predicados não espaciais (MILLLER; HAN, 2009; SANTOS; AMARAL, 2004).

2.4.3 Tendência espacial

A técnica de tendência espacial tem como objetivo encontrar padrões de mudança de atributos não-espaciais a partir de um objeto espacial. Para analisar a tendência espacial é necessária a aplicação da regressão linear (JIN; MIAO, 2010).

Uma das aplicações que utilizam a tendência espacial é no âmbito econômico, no qual pode-se verificar a tendência econômica de regiões próximas.

2.4.4 Caracterização espacial

A técnica de caracterização espacial consiste em identificar, por meio de predicados espaciais e não espaciais, características de um grupo e seus vizinhos.

Para isso, é selecionado um conjunto de dados que possuam a mesma característica não espacial, após isso, verifica-se quais vizinhos possuem uma diferença significativa da frequência relativa de todo o conjunto de dados. Por fim, é gerada a regra de caracterização que descreve o conjunto de dados selecionados (MILLER; HAN, 2009).

2.4.5 Agrupamentos de dados espaciais

Por fim, a técnica de agrupamento espacial visa agrupar objetos similares e próximos, ou seja, os objetos do agrupamento formado possuem alta similaridade entre si e baixa similaridade em relação aos demais agrupamentos (HAN; KAMBER, 2011).

Essa técnica destaca-se no *Spatial Data Mining*, visto que não é necessário um conhecimento prévio da base de dados e também pela alta aplicabilidade em diferentes domínios (PATTABIRAMAN et al., 2009). Esta pode ser subdividida em:

1. Métodos hierárquicos - os objetos são agrupados em uma estrutura de árvores, no qual cada nó filho possui objetos comuns ao nó pai. A cada iteração do algoritmo a base de dados é dividida em conjuntos menores até a formação dos agrupamentos. Os principais algoritmos desse método são: *AGNES*, *DIANA*, *BIRCH* (HAN; KAMBER, 2011) e *CURE* (GUHA; RASTOGI; SHIM, 1998).

2. Métodos de particionamento - particionam o conjunto de objetos em *k* agrupamentos, no qual cada objeto pertence a um agrupamento. Os objetos que compõe o agrupamento possuem atributos não-espaciais semelhantes entre si e diferentes dos outros agrupamentos. Os principais algoritmos desse método são: *k-means*, *k-medoide, PAM* (HAN; KAMBER, 2011) e *CLARANS* (NG; HAH, 2002).

3. Métodos baseados em densidade - os agrupamentos são formados de acordo com a densidade dos objetos e dos seus vizinhos. Uma vantagem desse método é a formação de agrupamentos com formas arbitrárias. Os principais algoritmos desse método são: *DBSCAN*, OPTICS e *DENCLUE* (HAN; KAMBER, 2011).

4. Métodos baseados em grade - utilizam a estrutura em grade, no qual divide o espaço em células e dessa forma são gerados os agrupamentos. Os principais algoritmos desse método são: *STING, CLIQUE* (HAN; KAMBER, 2011) e *WaveCluster* (SHEIKHOLESLAMI; CHATTERJEE; ZHANG, 1998).

5. Métodos baseados em grafos - para modelar as estruturas complexas dos dados espaciais a estrutura de grafos é utilizada e dessa forma é possível agrupar os objetos próximos com a aplicação da árvore geradora mínima. Os principais algoritmos desse método são: *AMOEBA* (ESTIVILL-CASTRO; LEE, 2000) e *CHSMST*+ (MEDEIROS, 2014).

2.5 *Outliers*

Outliers ou anomalias são objetos que se diferem dos outros por suas características ou seu comportamento. Um exemplo é ilustrado na Figura 3 - Exemplo de *outliers* extraído do livro , onde pode-se observar que os objetos contidos em R são *outliers*, pois não possuem o mesmo padrão de distribuição dos demais objetos do conjunto de dados (HAN; KAMBER; PEI, 2011).

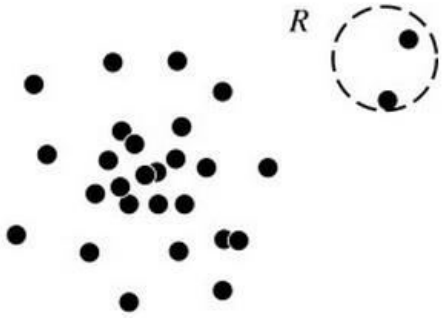


Figura 3 - Exemplo de *outliers* extraído do livro de (HAN; KAMBER; PEI, 2011)

Os *outliers* espaciais são objetos que possuem os atributos espaciais, referentes à sua localização, e os atributos não-espaciais diferentes de sua vizinhança. Um exemplo clássico de *outlier* espacial é uma casa construída recentemente em um bairro com construções antigas em uma área metropolitana, em que o atributo não-espacial "ano que a casa foi construída" é diferente dos seus vizinhos (SHEKHAR; ZHANG; HUANG, 2003).

Em comparação entre os *outliers* tradicionais e os espaciais é que os últimos não precisam ser diferentes de todo o conjunto de dados, conhecidos como *outliers* globais, mas diferente apenas de um local de comparação, conhecido como *outlier* local (CHEN et al., 2007).

Destaca-se ainda que *outliers* são diferentes de *noises*, uma vez que *noises* são dados que contem erros ou uma distorção de um parâmetro medido (SHEKHAR; ZHANG; HUANG, 2003). Na maioria dos casos, inclusive na detecção de *outliers*, *noises* não possuem informações relevantes e geralmente são removidos das tarefas de análises de dados (SHEKHAR; ZHANG; HUANG, 2003).

2.6 Detecção de *outliers* espaciais

A detecção de *outliers* espaciais tem como objetivo identificar objetos que possuam os valores de seus atributos muito diferentes dos objetos vizinhos de modo que a correção espacial e a continuidade sejam rompidas (KOU, 2006).

Dessa forma, os algoritmos de detecção de *outliers* convencionais não são eficientes para lidar com os *outliers* espaciais, já que não conseguem manipular as relações espaciais entre as variáveis de entrada. Por isso é importante a criação de métodos e algoritmos que consigam tratar separadamente os atributos espaciais dos atributos não-espaciais (KOU, 2006, JIN; MIAO, 2010).

Outro ponto a ser destacado na detecção de *outliers* espaciais é a diversidade de áreas de aplicação para tomadas de decisão e descoberta de conhecimento útil. Algumas áreas de aplicação são: detecção de ações maliciosas (CHANDOLA; BANERJEE.; KUMAR, 2009), diagnósticos médicos (WONG et al, 2003, ONG et al, 2012), controle de acidentes de trânsito (FIGUERA et al, 2011), detecção de erosão (HOSSEINALIZADEH; RIVAZ; HEDAYATIZADEH, 2011), detecção de danos ambientais (TORRES et al, 2011), entre outras áreas.

A aplicação dos métodos de detecção de *outliers,* é uma área bastante desafiadora por diversos fatores descritos a seguir (CHANDOLA; BANERJEE.; KUMAR, 2009):

- **Comportamento normal e anormal -** é difícil definir se um objeto é ou não um *outlier* quando está próximo da fronteira de uma região com comportamentos considerados normais;

- **Ações maliciosas** - adaptações para que um objeto considerado *outlier* tenha comportamento normal, tem se tornado cada vez mais frequentes com o aumento de ações maliciosas;

- **Domínio de aplicação** - um *outlier* é diferente para cada domínio de aplicação. Um exemplo é um pequeno desvio, na temperatura do paciente pode ser uma anomalia, já no valor de uma mercadoria pode ser considerado normal;

- **Disponibilidade de dados** - disponibilidade de dados para validar as técnicas de detecção de *outliers* nem sempre é fácil de encontrar;

- ***Noises*** - muitas vezes o *noise* contém dados semelhantes aos *outliers* e são removidos do conjunto de dados.

2.7 Estratégia para melhorar o desempenho dos algoritmos

Com o aumento na quantidade e diversidade dos dados, muitas técnicas para melhoria do desempenho dos algoritmos foram propostas como por exemplos: técnicas de paralelização, técnicas de *multithreads*, utilização de unidade de processamento gráfico (GPU), entre outros.

Uma das estratégias bastante utilizada atualmente para melhoria do desempenho dos algoritmos é a abordagem *MapReduce* (KIM et al, 2014), que é um modelo de programação desenvolvido pela multinacional *Google* em 2003. A finalidade é padronizar o desenvolvimento das aplicações paralelas que tenham um grande volume de dados a serem analisados utilizando-se do processamento eficiente de uma rede de computadores (KIM et al, 2014).

Os algoritmos são desenvolvidos em termos das funções baseadas nas primitivas *map* e *reduce* das linguagens funcionais como por exemplo *Lisp* e *Hashkell*.

Primeiramente, os dados de entrada são distribuídos em pequenas tarefas entre os computadores, após isso os valores são processados na função *map* e é gerado um conjunto de pares (chave, valor). Em seguida na função *reduce*, uma função definida pelo usuário é aplicada de modo a combinar os valores que possuem a mesma chave, e assim é gerado o resultado final. Na Figura 4 é apresentado o modelo *MapReduce* para o exemplo de contagem de palavras.

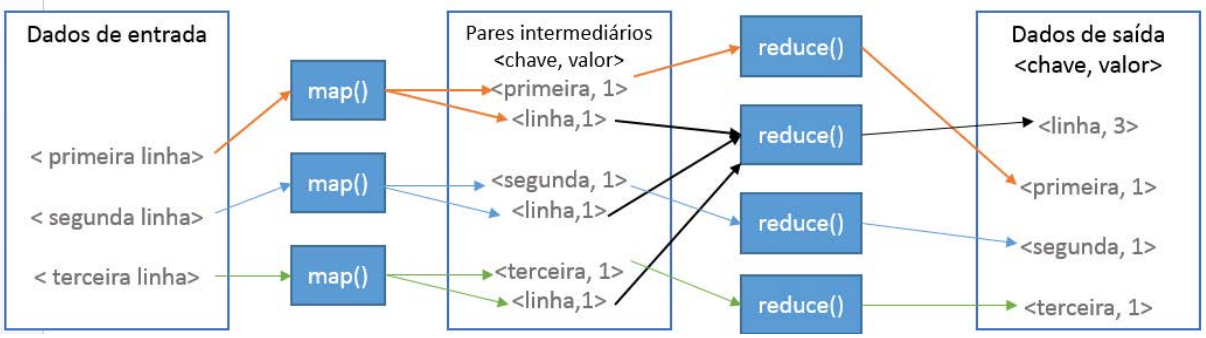


Figura 4 Modelo *MapReduce* para o exemplo de contagem de palavras

Uma das implementações mais conhecidas do *MapReduce* é *Hadoop*, que é um *framework* em *Java* mantido pela *Apache Software Foundation*. A principal finalidade é o desenvolvimento de software livre para computação distribuída, escalável e confiável. Isso permite que as aplicações em execução trabalhem com vários computadores independentes simultaneamente e com um grande número de dados.

As principais vantagens da utilização do *Hadoop* são (GOLDMAN et al, 2012):

- Código aberto - o projeto *Hadoop* é composto por várias empresas e programadores que compartilham conhecimento de desenvolvimento, funcionalidades e documentação. Outro ponto, é que por ser um *software* de código aberto, os programadores podem modificar o programa afim de adaptá-lo as suas necessidades;

- Robustez - o *Hadoop* considera a possibilidade de falhas nos equipamentos e oferece estratégias como replicação de dados, armazenamento de metadados, entre outros;

- Escalabilidade - o *Hadoop* permite aumentar a escalabilidade de maneira bastante simples e sem a necessidades de reescrever todo o código fonte, para isso é preciso apenas modificar um arquivo de configuração;

- Simplicidade - os detalhes de tolerância a falhas, escalonamento e balanceamento de carga são transparentes ao usuário, o que simplifica o desenvolvimento na abordagem *MapReduce* no *Hadoop*.

2.8 Considerações Finais

Neste capítulo foram apresentados os principais conceitos que envolvem as áreas de prospecção de dados convencionais e espaciais e apresentadas as principais técnicas para a extração de conhecimento em *spatial data mining*.

Outro conceito introduzido foi a definição de *outliers*, assim como, os desafios encontrados ao se trabalhar com detecção de *outlier*.

Por fim, foi abordada a estratégia de melhoria de desempenho de algoritmos, por meio da programação paralela com a utilização do paradigma *MapReduce*, que visa otimizar o tempo de execução do algoritmo de detecção de *outliers* espaciais.

Capítulo 3
Métodos para Detecção de *Outlier*

3.1 Considerações iniciais

A detecção de *outliers* espaciais visa descobrir objetos espaciais que se diferem dos outros objetos levando em consideração a sua localização e os atributos não-espaciais. O atributo espacial define a proximidade dos objetos com seus vizinhos enquanto o atributo não-espacial define a discrepância do atributo em relação aos demais (LIU; CHEN; LU, 2013). Dessa forma, essa técnica tem se mostrado importante no processo de descoberta de conhecimentos em dados espaciais, uma vez que não se limita a um domínio específico de aplicação, melhora a qualidade dos resultados e auxilia nas tomadas de decisões e descoberta de conhecimento (LIU; CHEN; LU, 2013). Com isso, muitas pesquisas têm sido feitas em relação a melhoria dos algoritmos de detecção de *outliers* espaciais, visando uma melhoria nos resultados e desempenho a fim de melhorar os resultados para a extração de informações úteis.

Os algoritmos de detecção de *outliers* convencionais não conseguem extrair os padrões anormais dos dados espaciais devido à complexidade na estrutura dos dados e suas correlações (SHEKHAR; XIONG, 2008). Dessa forma foi necessária a criação de algoritmos que considerem tanto o atributo espacial quanto o atributo não-espacial, os quais podem ser divididos em três grupos: métodos de visualização, métodos estatísticos e métodos de grafos. Cada um desses métodos é detalhado nas próximas seções, assim como os principais algoritmos utilizados, sendo realizado, por fim, um estudo comparativo desses algoritmos a fim de verificar suas características.

3.2 Métodos de visualização

O método de visualização baseia-se na ilustração da distribuição dos dados espaciais em um gráfico, e a partir disso os pontos discrepantes são destacados e identificados como *outliers* espaciais (SHEKHAR; XIONG, 2008).

Um dos mais conhecidos algoritmos é o *Scatterplot*, que é um gráfico no qual o eixo X representa os valores dos atributos enquanto o eixo Y representa a média dos valores dos atributos vizinhos. Os *outliers* são identificados utilizando-se a linha de regressão dos mínimos quadrados (SHEKHAR; XIONG, 2008, LIU; CHEN; LU, 2013). Na Figura 5 é apresentado o *Scatterplot,* no qual os pontos destacados P, Q e S são identificados como *outliers* espaciais.

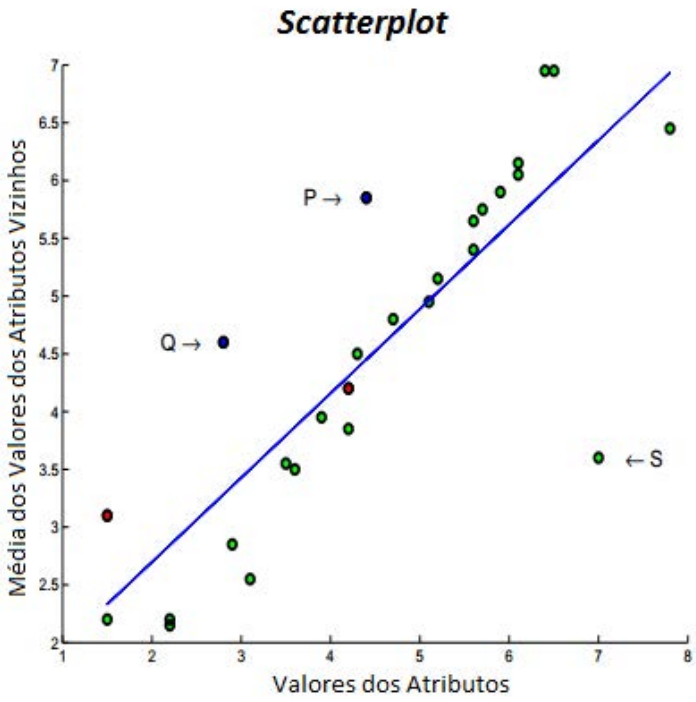


Figura 5 Um exemplo de visualização de *outliers* em *Scatterplot* (adaptada de SHEKHAR; LU; ZHANG, 2003)

O algoritmo *Moran Scatterplot* é formado por quatro quadrantes, nos quais os objetos localizados nos quadrantes superior esquerdo e inferior direito são considerados *outliers*. (SHEKHAR; XIONG, 2008, LIU; CHEN; LU, 2013). Na Figura 6 os pontos P e Q estão no quadrante superior esquerdo e o ponto S está no quadrante inferior direito e por isso são considerados *outlier*.

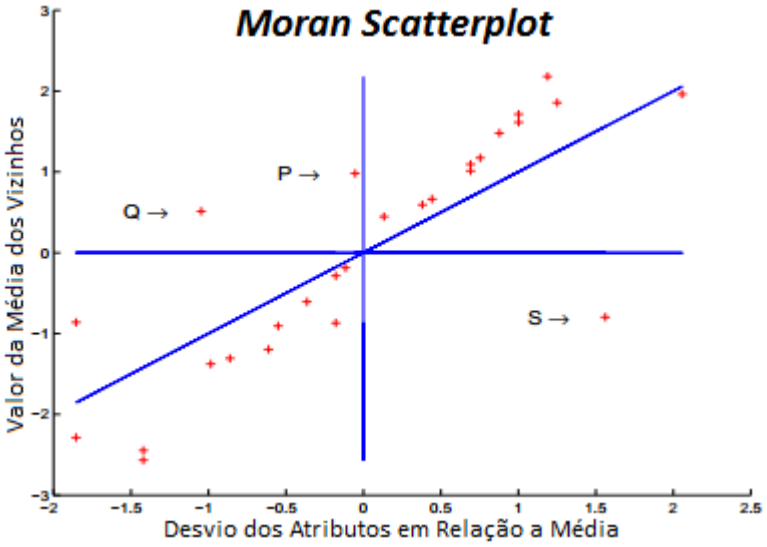


Figura 6 Um exemplo de visualização de *outliers* em Moran Scatterplot (adaptada de SHEKHAR; LU; ZHANG, 2003)

Um dos pontos negativos desses algoritmos é quando os valores de atributo dos vizinhos de um objeto são muito altos ou muito baixos, estes objetos podem ser detectados como *outliers* erroneamente. Dessa forma, os valores discrepantes que deveriam ser detectados como *outliers* são ignorados. Outro ponto é que esses algoritmos trabalham com um único atributo numérico não-espacial dos objetos, o que não demostra a realidade da maioria das bases de dados reais, que possuem vários atributos para um objeto e muito desses atributos são nominais (LU; CHEN; KOU, 2003a, LIU; CHEN; LU, 2013).

3.3 Métodos Estatísticos

Os algoritmos estatísticos para detecção de *outliers* espaciais utilizam testes estatísticos para detectar valores locais que são discrepantes de seus vizinhos.

O algoritmo Z utiliza a diferença normalizada entre o objeto e sua vizinhança, caso o valor resultante for maior que o limiar, o objeto é considerado um *outlier* espacial. O valor é calculado pela Equação (1), onde $S(x)$ é a diferença entre o valor do atributo não-espacial do objeto x, μ_S é o valor médio dos atributos não-espaciais e seus vizinhos e σ_S é o padrão de desvio de $S(x)$ para todo conjunto (SHEKHAR; XIONG, 2008).

$$Z_{S(x)} = \left|\frac{S(x) - \mu_S}{\sigma_S}\right| \quad (1)$$

Um exemplo de resultado desse algoritmo é apresentado na Figura 7, no qual o eixo X representa o local do objeto em uma dimensão e o eixo Y o valor da Equação 1. Na Figura 7 é

possível analisar que os objetos S, P e Q são identificados como *outliers*, uma vez que o objeto S possui o valor de $Z_{S(x)}$ superior a 3, já os objetos P e Q possuem valores próximos a -2 e são identificados como *outliers* devido aos objetos de sua vizinhança.

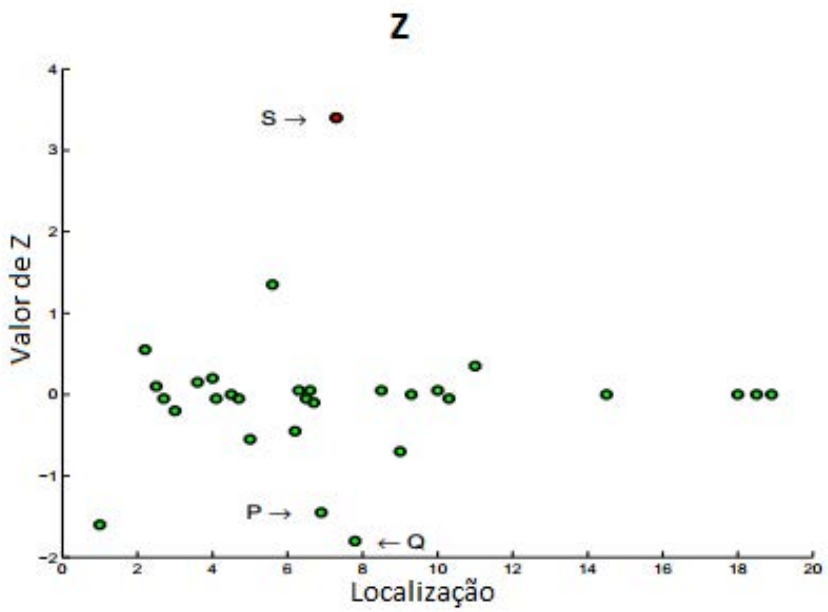


Figura 7 Um exemplo de visualização de *outliers* utilizando-se o Algoritmo Z (adaptada de SHEKHAR; LU; ZHANG, 2003)

Os algoritmos *interative-Z*, *interative-R* detectam *outliers* a partir de iterações, em que cada iteração apenas um *outlier* é detectado. Ambos os algoritmos, ao se detectar um *outlier* espacial o valor deste é substituído pela média de seus vizinhos. Dessa forma, nas próximas iterações o valor do *outlier* não irá ter um grande impacto, e assim, os objetos vizinhos não serão detectados como falsos *outliers* (LU; CHEN;KOU, 2003a).

Outra proposta foi o algoritmo da *mediana*, no qual não há necessidade de substituir o valor do atributo do *outlier* identificado. Esse algoritmo utiliza a mediana dos objetos vizinhos, em vez da média, pois a mediana estima o centro de uma amostra. Esse algoritmo também reduz o impacto causado pelos *outliers* nos objetos vizinhos (LU; CHEN;KOU, 2003a).

Os algoritmos *GLS-SOD* (CHEN; LU; BOEDIHARDJO, 2010) buscam encontrar *outliers* espaciais em conjuntos de dados que apresentam tendência tanto linear quando não linear.

3.4 Métodos de Grafos

O método baseado em grafos utiliza a sua conectividade para determinar se o objeto é um *outlier* ou não com relação aos objetos da sua vizinhança (KOU, 2006, SHEKHAR; LU; ZHANG, 2001).

O algoritmo proposto no trabalho por Shekhar, Lu e Zhang (SHEKHAR; LU; ZHANG, 2001) utiliza esse método, no qual faz a comparação entre o valor do atributo do objeto analisado com a média dos valores de seus vizinhos para identificar os *outliers*. O algoritmo proposto foi desenvolvido para investigar *outliers* espaciais com um único atributo não-espacial (SHEKHAR; LU; ZHANG, 2001).

Outros algoritmos são o *Region Outlier Detection* (ROD) e o *Point Outliers Detection* (POD), que utilizam o método de grafos para detecção de regiões atípicas ou *outliers* espaciais no conjunto de dados. Esses algoritmos baseiam-se na construção do grafo da dissimilaridade dos vizinhos mais próximos e a cada iteração faz o particionamento desse grafo até encontrar os *outliers* ou as regiões que contêm os valores discrepantes. Na Figura 8 é apresentado o resultado dos algoritmos POD e ROD, na qual é possível verificar que os objetos S1 e S2 são identificados como *outliers* isolados e uma região que contém os objetos S3 e S4 (KOU; LU; SANTOS, 2007).

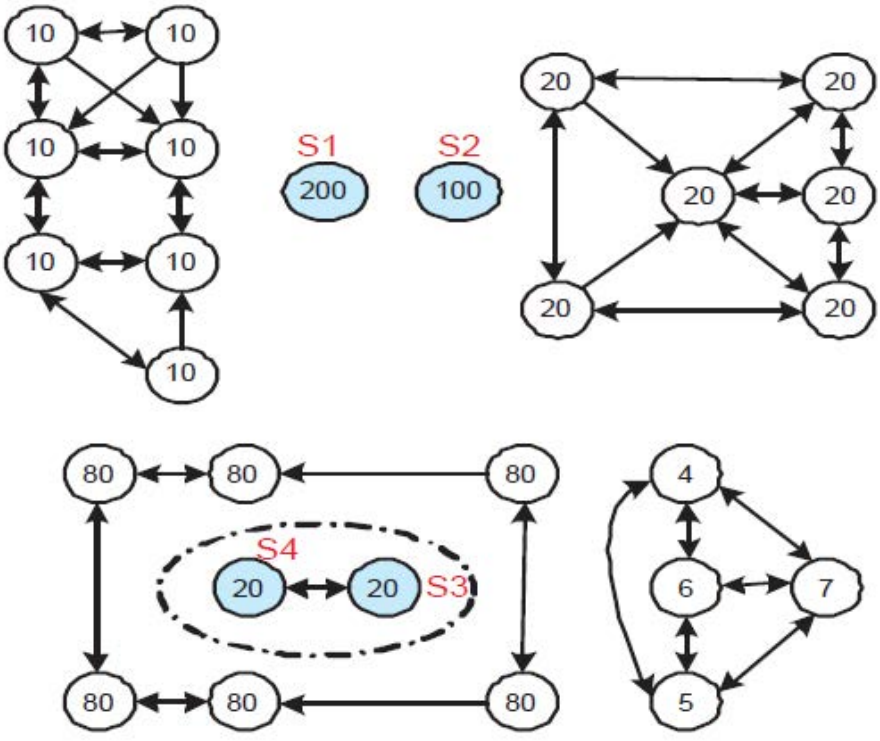


Figura 8 Resultado dos algoritmos POD e ROD (KOU; LU; SANTOS, 2007)

3.5 Detecção de *outliers* em agrupamentos de dados

Uma das técnicas que se destaca no SDM, devido a sua alta aplicabilidade em diferentes áreas, é o agrupamento de dados espaciais. Contudo, os algoritmos que utilizam essa técnica não detectam *outliers* na formação dos agrupamentos ou apenas detectam os outliers pelo atributo espacial - *outlier* global. Dessa forma, os resultados obtidos por esses algoritmos podem conter agrupamentos incorretos e de difícil entendimento (CHAWLA; GIONIS, 2013).

Dessa forma, o trabalho desenvolvido por Songma, Chimphlee, Maichalernnukul e Sanguansat (SONGMA et al.,2012) propõe um método de duas fases para detecção de *outliers*, o diagrama desse trabalho pode ser visualizado na Figura 9.

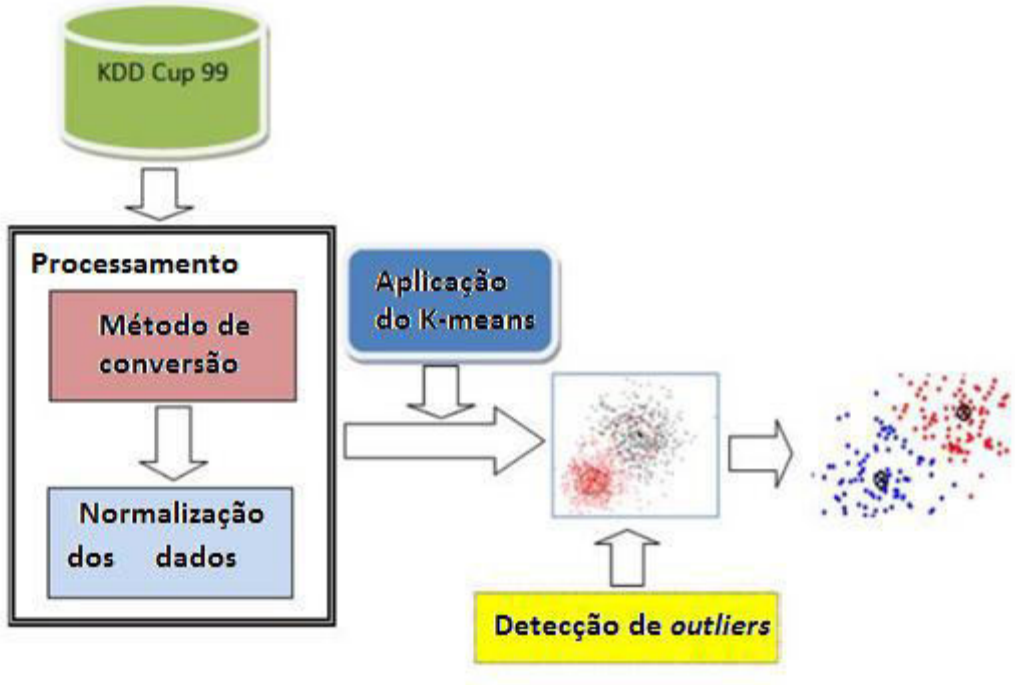


Figura 9 Diagrama do algoritmo de duas fases com a utilização da base de dados KDD Cup99 (adaptada de SONGMA et al.,2012)

A primeira fase consiste na aplicação do algoritmo de agrupamento *k-means* para formação dos agrupamentos de dados. Na segunda fase é utilizada a técnica de detecção de *outlier* baseado em distância. Os experimentos foram aplicados na base de dados *KDDCup 99* e obteve um bom resultado para detecção de intrusão.

Outro trabalho é o algoritmo *k-means-* - (CHAWLA; GIONIS, 2013) que é uma extensão do algoritmo *k-means*. O algoritmo funciona da seguinte maneira: primeiramente são encontrados os *k* agrupamentos de acordo com a menor distância entre os objetos e o centro, após isso são identificados os objetos que possuem as maiores distâncias em relação ao centro

do agrupamento. Esses objetos são identificados como *outliers* e os centros dos agrupamentos são recalculados até que o algoritmo convirja em uma solução ótima.

3.6 Trabalhos no estado da arte

Atualmente os trabalhos na área de detecção de *outliers* espaciais tem despendido esforços para melhoria dos resultados obtidos de forma a suprir os desafios discutidos anteriormente.

O algoritmo *ISOMRD* (CAI; HE;MAN, 2013) propõe a utilização de uma rede neural auto-organizável, do inglês *self-organizing map* (SOM), para detecção de *outliers* espaciais. A abordagem é a utilização do SOM para preservar as relações topológicas e métricas intrínsecas dos dados na formação dos agrupamentos espaciais e da aprendizagem iterativa para detectar os objetos com valores atípicos.

Uma variância do algoritmo *Z* é o trabalho de (SURI, et al, 2013), no qual o algoritmo original calcula o valor Z para cada observação e compara com o valor do limiar para detectar se uma observação é *outlier* ou não. A abordagem proposta nesse trabalho é obter o valor corrente dos valores de Z junto com o valor do limiar. Assim, em vez de calcular o valor de Z para cada observação, o valor corrente é calculado apenas uma vez durante a iteração.

O algoritmo *Weighted Difference Algorithm* (WDA) é baseado na abordagem de considerar os atributos espaciais como pesos e combiná-los com o algoritmo *Z*. Esse algoritmo resolve o problema da diferença entre os valores dos objetos vizinhos e o objeto analisado, encontrado no algoritmo *Z*, o que resulta em uma detecção mais precisa dos *outliers* (CAO et al., 2013).

Para suprir a grande quantidade de dados a serem processados nos algoritmos de detecção de *outliers* o *MR AVF* foi proposto, no qual utiliza a linguagem *MapReduce* no algoritmo *Attribute Value Frequency* (AVF). Um ponto a ser destacado nesse algoritmo é a análise de dados categóricos, uma vez que a maioria dos algoritmos, tanto tradicionais quando os espaciais, trabalham com dados numéricos ou ordinais (KOUFAKOU et al., 2008).

Por fim, o algoritmo *PCR-SCOD* (LIU; CHEN; LU, 2013), que é base do trabalho desenvolvido , é apresentado na próxima sessão com mais detalhes.

3.7 Algoritmo PCR-SCOD

O algoritmo PCR-SCOD apresenta uma nova métrica para detecção de *outliers* categóricos espaciais denominada *Pair Correlation Ratio* (PCR). O PCR consiste em medir para cada par de objetos espaciais a frequência da coocorrência dos atributos para os seus vizinhos mais próximos em uma determinada distância em um espaço de duas dimensões. Dessa forma os objetos que possuírem maiores pontuações são detectados como *outliers* espaciais categóricos, ou do inglês, *spatial categorical outliers*.

A equação para detectar os *outliers* é descrita na Equação (2), onde $PCR(r_i.A, r_{i_j}.A, d^E\ (r_i, r_{i_j}))$ denota a frequência da coocorrência dos pares de atributo $r_i.A, r_{i_j}.A$ com distância entre esses objetos $d^E\ (r_i, r_{i_j})$ e k é o número de vizinhos definido como parâmetro de entrada (LIU; CHEN; LU, 2013).

$$\boldsymbol{OutScore(r_i) = -\frac{\sum_{j=1}^{k} PCR(r_i.A, r_{i_j}.A, d^E\ (r_i, r_{i_j}))}{k}} \quad \textbf{(2)}$$

Outro ponto destacado no trabalho é a diferenciação da detecção dos *outliers* espaciais no domínio numérico e no domínio categórico, como pode ser analisado na Figura 10. O objeto A seria considerado *outliers* no domínio numérico, uma vez que o valor do atributo T se difere da sua vizinhança que possui o valor do atributo F. Por outro lado ao analisar o conjunto de dados é possível verificar que o par de atributos <T,F> ou <F,T> é frequente em uma distância d, o que implica que o objeto A não é um *outlier* em um domínio categórico com a técnica PCR (LIU; CHEN; LU, 2013).

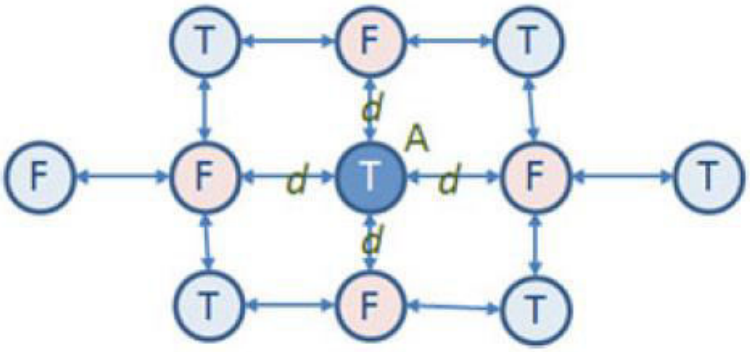


Figura 10 Detecção de *outliers* categóricos (LIU; CHEN; LU, 2013)

3.8 Estudos Comparativos entre os algoritmos de detecção de *outliers*

Foi realizado um estudo comparativo dos algoritmos de detecção de *outliers* com foco em dados espaciais, que envolveu trabalhos consagrados na literatura, assim como os trabalhos no estado da arte. Na Tabela 1 estão listados os algoritmos e as principais características analisadas para geração de resultados satisfatórios.

As características analisadas nos algoritmos foram: suporta dados espaciais; suporta dados categóricos; utiliza alguma abordagem para melhoria de desempenho; considera tanto os atributos espaciais como os atributos não-espaciais na identificação do *outlier*; trabalha com mais de um atributo não-espacial e se possui algum parâmetro de entrada.

Ao analisar a Tabela 1 é possível verificar que nenhum trabalho apresenta todas as características selecionadas. A maioria suporta com eficiência a manipulação de dados espaciais e a identificação de *outliers* pelos atributos espaciais e não-espaciais, mas somente *PCR-SCOD* suporta dados categóricos e espaciais, o que garante uma maior abrangência em trabalhar com qualquer tipo de atributo nas bases de dados. Além disso, possui uma estrutura que contempla a adaptação para suportar vários atributos não-espaciais, e por isso foi selecionado para ser a base desse trabalho em relação ao algoritmo de detecção de *outliers* espaciais.

Tabela 1 Estudos comparativos

Algoritmo	Suporta dados espaciais	Suporta dados categóricos	Melhoria de desempenho	Considera proximidade e atributos não-espaciais	Multivalorado	Sem dados de entrada
Scatterplot (SHEKHAR; LU; ZHANG, 2003)	Sim	Não	Não	Sim	Não	Não
Moran (SHEKHAR; LU; ZHANG, 2003)	Sim	Não	Não	Sim	Não	Não
Z (SHEKHAR; LU; ZHANG, 2003)	Sim	Não	Não	Sim	Não	Não
Interative Z (LU; CHEN;KOU, 2003a)	Sim	Não	Não	Sim	Não	Não
Graph-based spatial outlier (SHEKHAR; LU; ZHANG, 2001)	Sim	Não	Não	Sim	Não	Não
Variância do algoritmo Z (SURI, et al, 2013)	Sim	Não	Sim	Sim	Não	Não
MR AVF (KOUFAKOU et. al., 2008)	Não	Sim	Sim	Não	Sim	Não

WDA (CAO et al., 2013)	Sim	Não	Não	Sim	Não	Não
ISOMRD (CAI; HE;MAN, 2013)	Sim	Não	Não	Sim	Sim	Não
PCR-SCOD (LIU; CHEN; LU, 2013)	Sim	Sim	Não	Sim	Não	Não

3.9 Considerações Finais

Nesse capítulo foram apresentados os métodos de detecção de *outliers* espaciais, bem como os detalhes dos principais algoritmos dessa área. Também foi descrito a importância da detecção de *outliers* em agrupamentos espaciais para melhoria dos resultados.

Por fim, foi realizado um estudo comparativos de algoritmos consagrados na literatura, bem como os algoritmos no estado da arte, que resultou na seleção do algoritmo *PCR-SCOD* para ser a base deste trabalho na detecção de *outliers* espaciais.

Capítulo 4
Refinamento de similaridade e desempenho: Algoritmo de detecção de *outliers* espaciais

4.1 Considerações Iniciais

Após a introdução dos conceitos relacionados à detecção de *outliers*, assim como os algoritmos envolvidos nessa área, é apresentado, nesse capítulo, o algoritmo desenvolvido, denominado *Pair Correlation Ratio – Spatial Categorical Outliers Detection – MapReduce* (*PCR-SCOD-MR)*, desde a sua concepção à sua implementação. Este algoritmo consiste na melhoria do algoritmo *PCR-SCOD* e foi adicionado na abordagem proposta para o aprimoramento dos resultados nos agrupamentos espaciais, por meio do refinamento de similaridade e desempenho.

Para isso, é detalhado o funcionamento do algoritmo *PCR-SCOD* por meio de pseudocódigo e do fluxograma. Em seguida, são detalhadas as estratégias para melhoria dos resultados nos agrupamentos e desempenho adotadas, por meio dos pseudocódigos dos principais métodos desenvolvidos, assim como o fluxograma que contém as modificações realizadas no algoritmo base.

Por fim, são apresentadas as estratégias utilizadas para verificar a eficiência da abordagem adotada, a fim de verificar se a detecção dos *outliers* dentro dos agrupamentos gerados aumentou a similaridade.

4.2 O algoritmo *PCR-SCOD*

O algoritmo *PCR-SCOD* (LIU; CHEN; LU, 2013), apresentado na seção 3.6, utiliza a métrica PCR para calcular se um objeto é *outlier* ou não. O algoritmo suporta a manipulação de dados espaciais e dados categóricos, porém só contempla um atributo não-espacial. Isso significa que os objetos com comportamento típicos terão uma alta frequência da coocorrência do par de atributo em relação a distância dos seus vizinhos.

Os parâmetros de entrada do algoritmo são: b número de *bins* para divisão dos valores da distância, n número de objetos espaciais no conjunto de dados, k número de vizinhos espaciais e l o número de *outliers* a ser encontrado. Os passos do algoritmo são os seguintes (LIU; CHEN; LU, 2013):

1. Para cada objeto, calcular a distância entre o objeto e o restante do conjunto de dados, de modo a verificar quais são os k vizinhos mais próximos;

2. Função *Spatial Pair Function* (SPF) para os objetos

   2.1 Identificar a maior distância (x,y) no conjunto de dados e dividir por b;

   2.2 Calcular a frequência dos valores do atributo e em seguida armazenar todas as possiblidades de pares entre os valores do atributo;

   2.3 Calcular SPF da seguinte forma - número pares de objetos que satisfazem a condição do atributo dividido pelo número de pares de objetos que satisfazem a condição de distância.

   - Condição do atributo é dada por: $(r_iA = A^l \; e \; r_jA = A^{l'}) \;\; ou$ $(r_iA = A^{l'} \; e \; r_jA = A^l)$, onde $r_i \; A \; e \; r_jA$ são os atributos dos objetos analisados, e $A^l \;\; e \; A^{l'}$ são os atributos da lista de possibilidades de pares encontrados na 2.2
   - Condição de distância é dada por: (c-1).d ≤ d < c.d, onde c varia de 1 até b, e d é a distância encontrada na 2.1, ou seja, para cada *bin* é formado um círculo menor com raio (c-1).d e um círculo maior com raio c.d.

   2.4 Ajuste de curva do polinômio: como os valores de SPF são discretos, é necessário fazer o ajuste de curva para encontrar os demais valores.

3. Calcular o PCR para cada par de objetos utilizando o ajuste de curva na 2.4, e fazer a média desses valores.

4. Por fim, os objetos são ordenados e os objetos com maiores pontuações obtidos na etapa anterior são identificados como *outliers* espaciais categóricos.

O algoritmo *PCR-SCOD* foi implementado no ambiente *Matlab* e foi utilizado uma função que retorna um vetor que contém os coeficientes do polinômio de ajuste (LIU; CHEN; LU, 2013).

Na Figura 11 é apresentado o fluxograma do algoritmo *PCR-SCOD*, no qual observa-se que no início do algoritmo é necessário entrar tanto com o conjunto de dados como com os parâmetros de entrada.

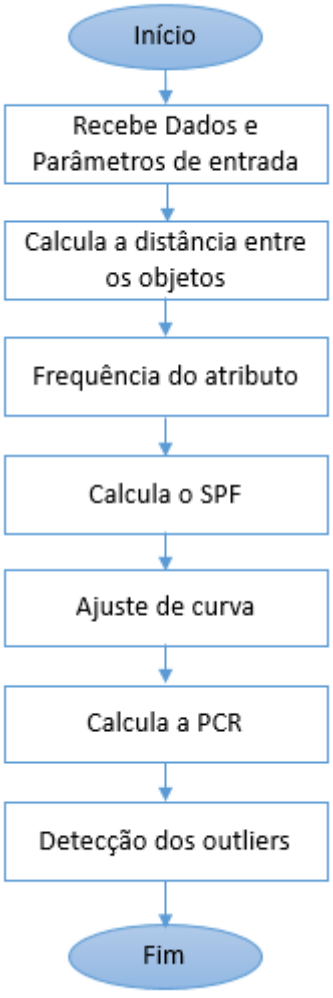


Figura 11 Fluxograma do algoritmo *PCR-SCOD*

Para validação do algoritmo *PCR-SCOD* o artigo (LIU; CHEN; LU, 2013) cita vários experimentos realizados comparando o algoritmo desenvolvido com outros algoritmos da área. As aplicações dos algoritmos foram realizadas em uma base sintética e três bases de dados reais com distribuições diferentes, como apresentado na Figura 12. Em cada base foram alterados 2, 3 e 5% dos dados e verificada a porcentagem que os algoritmos identificaram esses dados (LIU; CHEN; LU, 2013).



Figura 12 Distribuição do conjunto de dados utilizado no experimento (LIU; CHEN; LU, 2013)

Nos resultados do trabalho (LIU; CHEN; LU, 2013), vários experimentos foram realizados, com diferentes bases de dados espaciais, para auxiliar o usuário a escolher os valores dos parâmetros de entrada. Os experimentos consistiram em verificar o número de vizinhos "k" a serem analisados no algoritmo e o número de *bins* "b".

Dessa forma, após os experimentos, o número de vizinhos estava dentro de um intervalo e variavam de acordo com a distribuição dos objetos na base de dados utilizada, o que implicava na necessidade de conhecer a base de dados e sua distribuição para que o número de vizinhos seja definido.

Já o número ideal de *bins* "b" permaneceu constante, no valor de dez, para as diversas bases de dados testadas, independente da quantidade de objetos e distribuição. Portanto o valor será utilizado neste trabalho.

Outro ponto a ser destacado é que o algoritmo foi desenvolvido e testados para objetos de duas dimensões espaciais, devido ao embasamento teórico referente ao cálculo da SPF e, por esse motivo, este trabalho também seguirá a mesma base teórica.

4.3 Trabalho desenvolvido

Como descrito anteriormente, os algoritmos de agrupamento espaciais muitas vezes não detectam corretamente os *outliers* espaciais. Com isso os resultados obtidos podem conter agrupamentos com pouca similaridade e de difícil entendimento para os usuários finais (CHAWLA; GIONIS, 2013).

Por outro lado, os algoritmos de detecção de *outliers* espaciais são aplicados em todo o conjunto de dados, como é o caso do algoritmo *PCR-SCOD*. Esse algoritmo não possui todas as características desejadas em um algoritmo de detecção de *outliers* espaciais uma vez que possui parâmetros de entrada, não foi desenvolvido para suportar mais de um atributo não-espacial e não tem nenhuma abordagem para otimização do desempenho.

Nesse contexto o trabalho visa centrar-se nestas limitações do algoritmo *PCR-SCOD* e após isso aplicá-lo em cada agrupamento gerado pelo algoritmo, a fim de buscar uma melhora nos quesitos de similaridade e desempenho. Com as propostas de alterações implementadas, o novo algoritmo de detecção de *outliers* espaciais categóricos passa a ser denominado *PCR-SCOD-MR* e pode ser utilizado em qualquer resultado de algoritmos de agrupamentos espaciais.

Para contemplar as alterações no algoritmo base, o *PCR-SCOD* foi reimplementado em linguagem Java, a partir dos pseudo-códigos contidos no trabalho (LIU; CHEN; LU, 2013).

4.3.1 A Estratégia para refinamento de similaridade nos agrupamentos

A estratégia escolhida para o refinamento de similaridade dos agrupamentos, gerados pelos algoritmos, foi a detecção dos *outliers* espaciais categóricos locais. Dessa forma, os agrupamentos resultantes serão livres de *outliers* e, portanto, com maior similaridade.

Para facilitar o entendimento da abordagem é apresentado um fluxograma com as etapas ilustradas na Figura 13, no qual o algoritmo de agrupamentos espaciais recebe os dados e, se houver, os parâmetros de entrada do algoritmo. Em seguida, os agrupamentos são gerados, e a partir disto, o algoritmo *PCR-SCOD-MR* é executado em cada agrupamento. Com isso, os *outliers* espaciais locais são identificados e retirados do agrupamento, o que implica em um aumento na similaridade e na melhoria dos resultados finais.

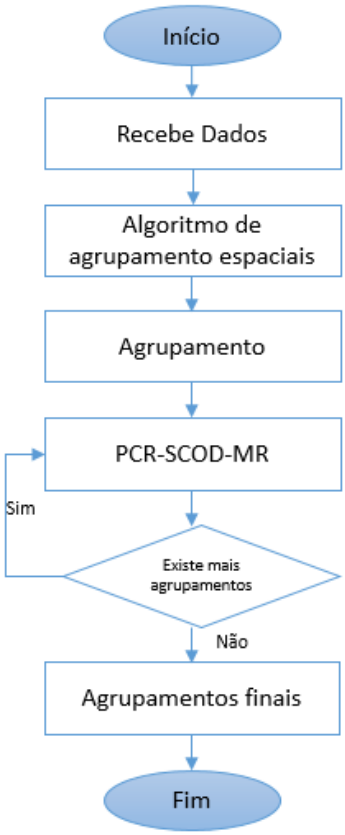


Figura 13 Proposta de fluxograma para a abordagem do trabalho

É importante ressaltar que o algoritmo de agrupamentos espaciais não é alterado nessa abordagem, dessa forma o algoritmo de detecção de *outliers* espaciais categórico somente utiliza as saídas geradas pelo algoritmo.

Outro ponto dessa abordagem é a fácil adaptação do algoritmo *PCR-SCOD-MR* a qualquer saída dos algoritmos espaciais, uma vez que é necessário somente saber quais objetos estão em quais agrupamentos.

4.3.2 Otimização do desempenho do algoritmo *PCR-SCOD-MR*

No sentido de proporcionar uma melhoria da capacidade de processamento de um grande volume de dados do algoritmo *PCR-SCOD-MR*, é proposta a alteração em dois pontos do algoritmo base *PCR-SCOD* ao se utilizar o *MapReduce*.

Primeiramente, foi verificado que a etapa mais lenta do algoritmo *PCR-SCOD* era o cálculo da distância e sua ordenação para verificar quais os objetos eram mais próximos. Dessa forma, o *MapReduce* foi inserido de modo que a lista de objetos foi passada para função *map* e na função *reduce,* no qual é efetuado o cálculo da distância entre os objetos. Na função *reduce,* ao se calcular a distância de um objeto com o outro, já é feita a ordenação,

dessa forma ao finalizar o *MapReduce* temos os k vizinhos mais próximos. A seguir é apresentado o pseudocódigo do *MapReduce* para a distância.

MapReduce para cálculo da distância

Entradas: Lista com os objetos e seus atributos.

Procedimento:

1. Função Map:
2. Para cada objeto
3. Cria os pares <objeto_atual, objetos_vizinhos>
4. Função Reduce:
5. Para cada objetos_vizinhos:
6. Calcula a distancia
7. Ordena a lista

Saídas: Lista de objetos com seus vizinhos ordenados pela distância.

Outra etapa do algoritmo que é beneficiada pelo *MapReduce* é o cálculo da frequência dos atributos. A função *map* recebe cada par de atributo e como valor a quantidade 1. Na função *reduce* as chaves iguais, ou seja, o par de atributos iguais, é somado o valor. A seguir o pseudocódigo do *MapReduce* para a frequência de atributo é apresentado.

MapReduce para cálculo da frequência

Entradas: Lista dos pares de atributos dos objetos.

Procedimento:

1. Função Map:
2. Cria os pares <par_de_atributo, 1>
3. Função Reduce:
4. Somar a quantidade de par_de_atributo iguais

Saídas: Lista com o par de atributo e a quantidade

4.3.3 O algoritmo desenvolvido *PCR-SCOD-MR*

A primeira alteração realizada no algoritmo *PCR-SCOD-MR* foi eliminar os parâmetros de entrada. Os parâmetros de entrada para o algoritmo são b – número de *bins*, k – número de vizinhos e *l* - número de *outliers* espaciais a serem detectados.

O parâmetro de entrada b foi inicializado com o valor dez, como explicado na seção 4.2. Esse valor foi definido após vários experimentos descritos no trabalho (LIU; CHEN; LU, 2013) e definido pelos autores como o valor ótimo para esse parâmetro de entrada.

Já o k vizinhos mais próximos foi definido seguindo-se uma regra simples descrita na Equação (3) (BICHLER; KISS, 2004), onde n é o número de objetos dentro dos agrupamentos.

$$\boldsymbol{k = \sqrt{n}} \quad \textbf{(3)}$$

Para verificar qual a quantidade de *outliers* a serem detectados foi utilizado a regra dos quartis (MASSART et al., 2005), ou seja, o valor de *l* não é definido incialmente e sim após os cálculos dos *scores* dos objetos. Dessa forma, os valores são ordenados em ordem crescente, e divididos em 3 quartis de distribuição dos dados.

Para ilustrar essa abordagem, considere um conjunto de dados com 7 observações.

Ordem	1	2	3	4	5	6	7
Valor	3,0	3,5	5,0	6,7	7,7	7,9	15,0

Para definir os quartis, primeiro é necessário encontrar a mediana, nesse caso como o número de observações é ímpar, a mediana é o valor da posição 4, ou seja, o valor é 6,7. Caso o número de observações seja par, a mediana é a média dos valores centrais.

O cálculo do primeiro quartil é feito calculando-se a mediana da metade inferior, no caso é 3,5. O terceiro quartil é calculado da mesma maneira, porém para a metade superior que é 7,9. Caso o número de observações seja par, é calculado a média dos valores centrais para cada metade.

Para definir quais são os valores *outliers*, foi utilizada somente os valores muito elevados, pois pela métrica PCR quanto menor o *score* do objeto maior a correlação e a similaridade em relação aos seus vizinhos. Assim, são *outliers* os objetos que seguem a Equação (4), sendo Q1 e Q3 os valores do primeiro e terceiro quartil respectivamente.

$$\boldsymbol{Outliers} > \textbf{Q3 + 1,5 (Q3 – Q1)} \quad \textbf{(4)}$$

No exemplo, são *outliers* os valores maiores que 14,5, ou seja, somente a observação da posição 7 é considerada *outlier*.

A segunda alteração realizada no algoritmo foi transformá-lo para suportar mais de um atributo não-espacial. Para isso, em vez de verificar se o atributo-espacial é igual aos seus vizinhos, é verificado se todos os atributos não-espaciais selecionados são iguais aos atributos não-espaciais de seus vizinhos. Outra alteração na estrutura do algoritmo, é armazenar todos os pares possíveis entre todos os atributos não-espaciais.

Com as modificações citadas, o algoritmo *PCR-SCOD-MR* precisa somente do conjunto de dados, uma vez que os parâmetros de entradas são definidos no próprio algoritmo.

A primeira etapa do algoritmo é fazer o cálculo das distâncias para encontrar os k vizinhos mais próximos, para isso foi utilizada as funções *map* e *reduce* para distribuir esses cálculos, como mostrado na seção 4.3.2, ao final da execução do *MapReduce* é obtido a lista dos objetos com os respectivos vizinhos ordenados pela menor distância.

Em seguida, é calculada a maior distância no conjunto de dados e dividida por b, que possui valor 10.

Após isto, é feito o cálculo das frequências utilizando também o *MapReduce*, como descrito no algoritmo da seção 4.3.2 e a listagem de todos os possíveis pares para o conjunto de dados.

Para calcular o SPF, primeiro é verificado para cada b os pares de pontos que satisfazem a condição de distância. Em seguida é calculado o número de pares de objetos que satisfaçam a condição de atributo, como mostrado a seguir.

A condição do atributo é dada por: $(r_i A = A^l \; e \; r_j A = A^{l'}) \;\; ou \;\; (r_i A = A^{l'} \; e \; r_j A = A^l)$, onde $r_i \; A \; e \; r_j A$ são a lista de atributos dos objetos analisados, e $A^l \;\; e \; A^{l'}$ são a lista de atributos das possibilidades de pares. Já a condição de distância é dada por: $(c-1).d \leq d < c.d$, onde c varia de 1 até b, e d é a distância encontrada anteriormente.

Algoritmo PCR-SCOD-MR

Entradas: D – Conjunto de dados.

Procedimento:

1. Receber o conjunto de dados *D;*
2. Chamar MapReduce para cálculo da distância *dist_matriz*;
3. Calcula *d* sendo a maior distância dividida por b=10;
4. Chamar MapReduce para cálculo da frequência *frequência_par*;
5. Listar todos os pares de atributos não-espaciais possíveis *lista_par*;
6. Chamar Calcular_SPF;
7. Ajustar o polinômio de curva;
8. Chamar Calcular _PCR;
9. Detecção dos *outliers* espaciais;

Saídas: Identificação dos *outliers* no conjunto de dados D

Algoritmo Calcula_SPF

Entradas: *n_lista_par* – Quantidade de pares de *lista_par*

d – distância

Procedimento:

1.Para cada c =1 até b

2. *Di* = número de pares de objetos que satisfaça a condição de distância
3. Para cada j = 1 até a *n_lista_par*
4. *Dj* = número de pares de objetos que satisfaça a condição de atributo
5. *SPF* = Dj / Di;
6. *HashMap <atributos dos dois objetos, SPF>*

Saídas: HashMap contendo a lista de atributos de ambos os objetos analisados e o valor de SPF.

Logo, em seguida é feito o ajuste de curva do polinômio e calculado o PCR como descrito a seguir.

Algoritmo Calcula_PCR

Entradas: *HashMap do calcula SPF*

Lista de *frequência_par*

n – número de objetos do conjunto de dados

k – número de vizinhos

Procedimento:

1. Para cada i =1 até n
2. Para cada j = 1 até a *k*
3. *PCR(i,j)* = HashMap / (*frequência_par de i) x (frequência_par de j)*;

Saídas: Matriz PCR

Para finalizar, é feita a divisão dos quartis e verificado quais os objetos são detectados como *outlier* espaciais categóricos em cada agrupamento.

Com o objetivo de possibilitar uma visualização das etapas alteradas para o algoritmo *PCR-SCOD-MR* é apresentado a Figura 14 que contem o fluxograma, em que as áreas destacadas em verde representam a inclusão e/ou modificação do algoritmo original *PCR-SCOD*.

É possível analisar na Figura 14 que foram retirados os parâmetros de entrada no início do algoritmo, logo após o cálculo da distância e da frequência são feitas de forma distribuída por meio do *MapReduce*. O cálculo do SPF passa a considerar mais de um atributo não-espacial e a detecção de *outliers* é feita por meio dos valores atípicos calculados pelas regras do quartis.

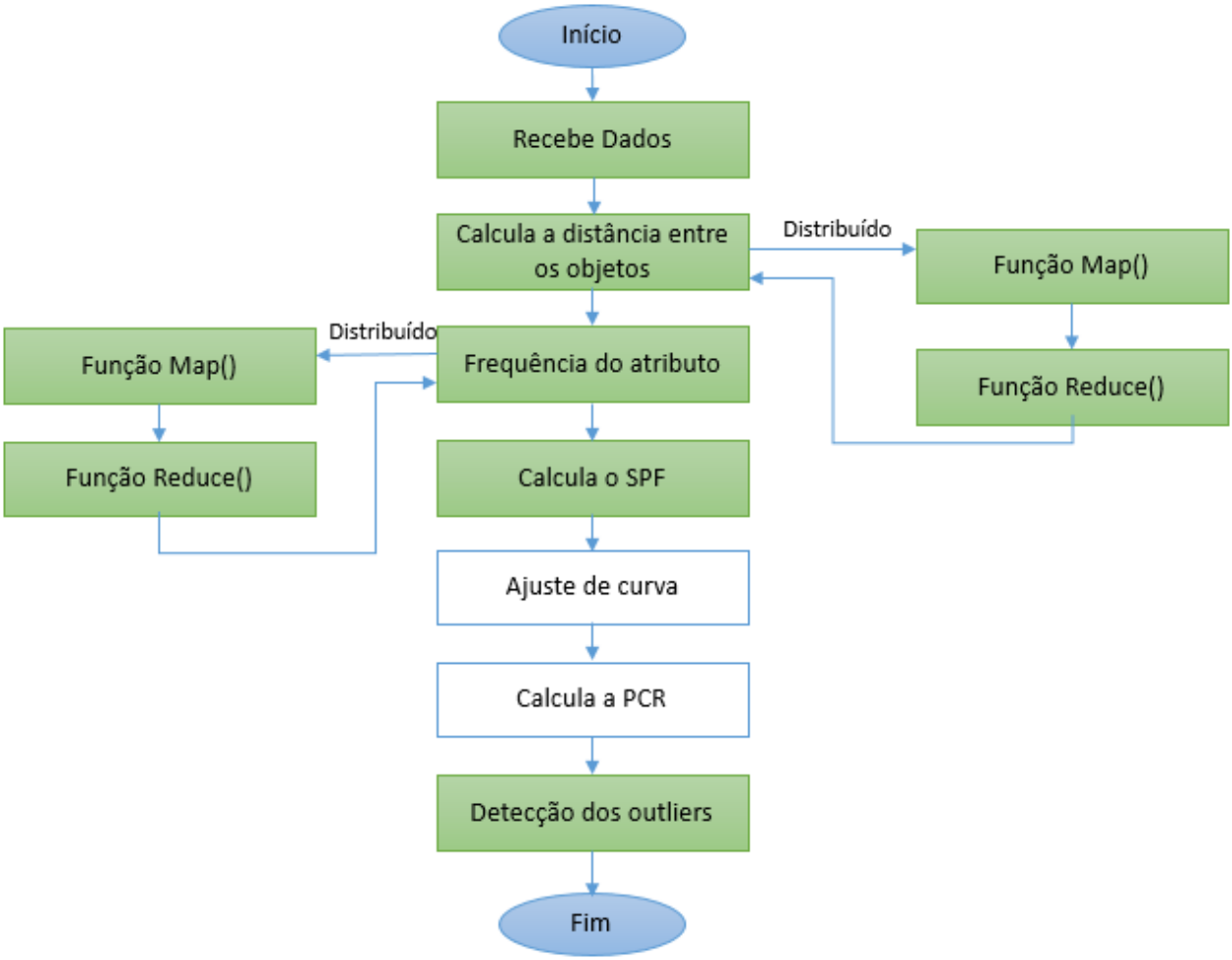


Figura 14 Fluxograma do algoritmo *PCR-SCOD-MR*

4.3.4 Qualidade dos resultados

Para comprovar que a abordagem proposta resultou em uma melhoria na qualidade dos resultados dos algoritmos de agrupamento espacial foram utilizados dois métodos, um que mede a qualidade dos agrupamentos ao verificar a distância entre os objetos e outro à similaridade dentro dos agrupamentos.

O primeiro método é um dos mais utilizados para verificação da qualidade dos agrupamentos em relação a distância do objeto com os outros e é conhecido como coeficiente de silhueta. O método consiste em calcular, para cada objeto do conjunto de dados, a média da distância do objeto para os demais do mesmo agrupamento; a distância média mínima do objeto para os outros agrupamentos; e, por fim, o coeficiente de silhueta (HAN; KAMBER, 2011).

Para isso, considere o objeto *o* pertencente ao agrupamento Ci, sendo Ci ∈ C={ C1, C2, C3, Ck}. Para o cálculo da média da distância do objeto *o* para os demais agrupamentos é utilizado a Equação (5) e dessa forma encontrado o a(*o*), ou seja, refere-se à coesão dentro do agrupamento.

$$\boldsymbol{a(o)} = \frac{\sum_{o' \in Ci\ o \neq o'} \boldsymbol{dist(o,o')}}{\boldsymbol{|C_i|-1}} \tag{5}$$

O cálculo da distância mínima do objeto *o* aos demais agrupamentos pode ser obtido pela Equação (6), no qual o valor de b(*o*) corresponde a separação entre o objeto *o* e os agrupamentos.

$$\mathrm{b}(\boldsymbol{o}) = \boldsymbol{min}_{C_j:1 \leq j \leq k, i \neq j} \left\{ \frac{\sum_{o' \in Cj} \boldsymbol{dist(o,o')}}{\boldsymbol{|C_j|}} \right\} \tag{6}$$

O coeficiente s(*o*) é obtido pela Equação (7), e pode variar entre -1 e 1. Sendo que os valores próximos a 1 indicam que o agrupamento Ci possui melhor coesão e separação entre os demais agrupamentos, e assim, a qualidade do agrupamento é alta.

$$\boldsymbol{s(o)} = \frac{\boldsymbol{b(o)-a(o)}}{\boldsymbol{max\{a(o),b(o)\}}} \tag{7}$$

O coeficiente de silhueta foi utilizado para medir a qualidade dos agrupamentos obtidos após a detecção de *outliers* em relação a distância Euclidiana.

Para calcular a similaridade dentro dos agrupamentos em dados categóricos foi utilizada a Equação (8), que verifica a coincidência dos atributos dos objetos o_1 e o_2. Onde p é o número total de atributos não-espaciais e m é o número de coincidências.

$$\boldsymbol{s(o_1, o_2)} = \frac{\mathrm{m}}{\mathrm{p}} \tag{8}$$

No trabalho utilizou-se a Equação (8), sendo considerado uma coincidência caso algum dos valores dos atributos não-espaciais forem iguais ao do outro objeto, uma vez que o algoritmo trabalha com múltiplos atributos. Por fim é feita a média de todos os valores calculados, para encontrar a similaridade do agrupamento.

4.3.5 Ferramenta para agrupamento de dados espaciais

Para facilitar a extração de conhecimento dos resultados gerados, foi construída uma ferramenta *Web* para suportar a abordagem desenvolvida neste trabalho.

Como ambiente de desenvolvimento da ferramenta foi utilizado o *NetBeans IDE*, que é executável em várias plataformas, e ainda possui código aberto. A linguagem escolhida foi o *Java* pela capacidade de conexão com vários Gerenciados de Banco de Dados – SGBS e por sua portabilidade (ORACLE, 2014).

Para visualização e análise de dados espaciais em mapa foram utilizadas as tecnologias *Geoserver*, *Openlayers* e o *Google Maps*.

O *Geoserver* é um servidor aberto de informações geoespacial nos padrões definidos pelo *Open Geospatial Consortium* (OGC). Utiliza a publicação de dados espaciais por meio de mapa e imagens (GEOSERVER, 2014).

O *Openlayers* é uma biblioteca *JavaScript*, também de código aberto, que permite a agilidade no desenvolvimento de aplicações *Web* e na integração com mapas (OPENLAYERS, 2014).

O *Google Maps JavaScript* é uma *Application Program Interface* (API) que possibilita a incorporação do mapa do *Google* em uma página *Web* ao se utilizar *JavaScript* (GOOGLE, 2014).

O sistema gerenciador de banco de dados utilizado foi o *PostgreSQL* (POSTGRESQL, 2014), que possui uma extensão própria para manipulação de dados espaciais, esta extensão é denominada *PostGIS*.

A arquitetura da ferramenta *Web* é a seguinte, primeiramente são configuradas as informações da base de dados espaciais, bem como a tabela alvo e seus respectivos atributos não espaciais escolhidos para execução do algoritmo de agrupamento de dados.

A partir disso, aplica-se o algoritmo de agrupamentos espaciais, como por exemplo os incorporados na ferramenta que são: o algoritmo *k-means* que possui como parâmetro de entrada o número de agrupamentos a ser gerado, e o algoritmo *CHSMST*$^{+}$. Após a execução do algoritmo de agrupamento é possível aplicar o algoritmo de detecção de *outliers* espaciais.

Para facilitar a análise dos resultados obtidos após o refinamento de similaridade, os objetos são projetados no mapa obtido do *Google Maps* a partir do *Geoserver* e do *OpenLayers*. A arquitetura da ferramenta é exibida na Figura 15, onde tem-se a interface *WEB*, a possibilidade de execução dos algoritmos de agrupamentos de dados espaciais e visualização dos resultados por meio do servidor de mapas.

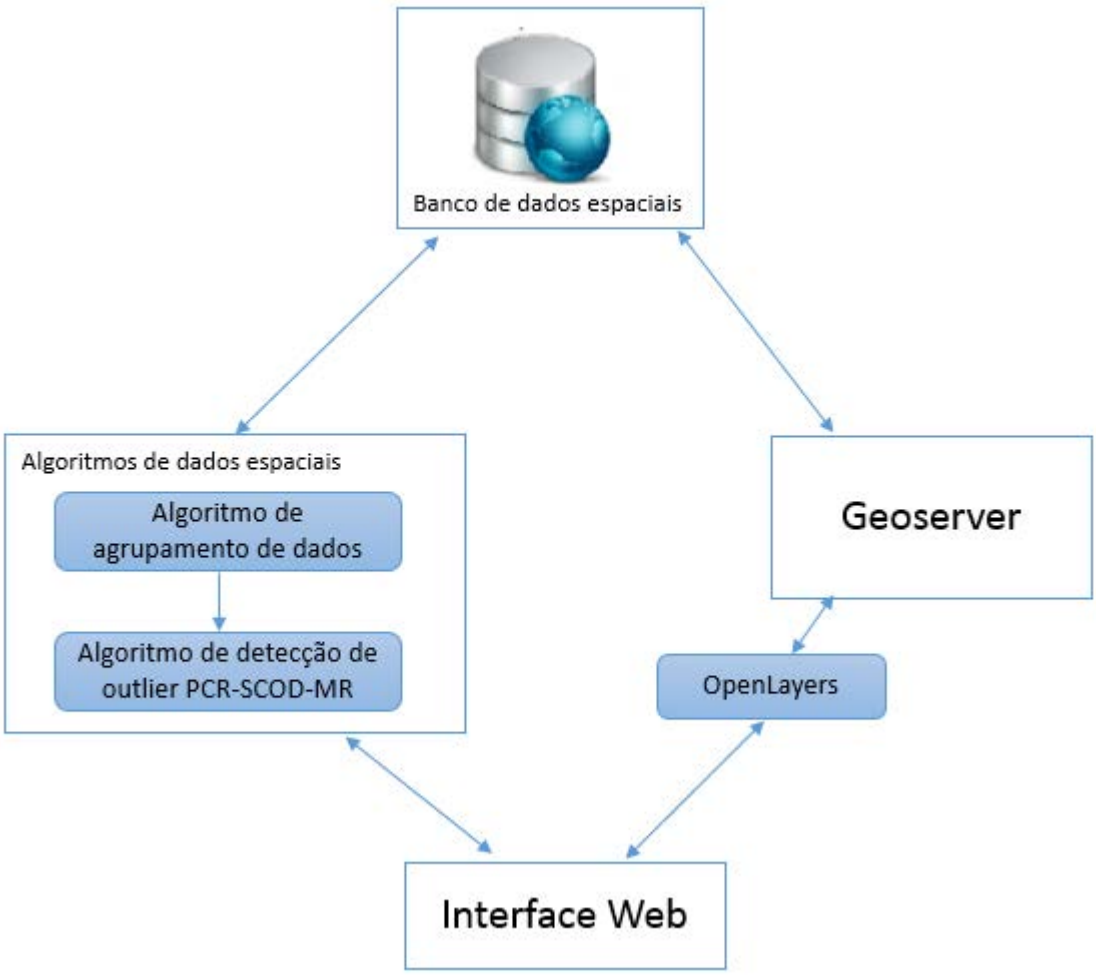


Figura 15 Arquitetura da ferramenta *Web*

4.4 Considerações Finais

Neste capítulo foi apresentado o desenvolvimento da abordagem, sendo descritas as modificações efetuadas no algoritmo *PCR-SCOD* que resultou no algoritmo denominado *PCR-SCOD-MR*, a fim de refinar a similaridade dentro dos agrupamentos como melhorar o desempenho do algoritmo original.

Para auxiliar na análise dos resultados foi construída uma ferramenta *Web* que permita ao usuário escolher o algoritmo de agrupamento de dados, executar o algoritmo de detecção de *outliers* espaciais e ainda identificar cada agrupamento junto ao mapa da região que está localizado.

Capítulo 5
Experimentos e Resultados

5.1 Considerações Iniciais

Nesse capítulo são apresentados os experimentos com o algoritmo proposto *PCR-SCOD-MR*, juntamente com a abordagem de detecção de *outliers* espaciais com o algoritmo proposto e outro algoritmo, a fim de verificar a melhoraria na similaridade dos resultados dos algoritmos de agrupamentos espaciais e por fim validar a estratégia para melhoria de desempenho do algoritmo.

5.2 Experimento da detecção de *outliers PCR-SCOD-MR*

Para validar os resultados obtidos por meio do algoritmo proposto *PCR-SCOD-MR* foi criada uma base de dados espaciais sintética. Dessa forma, é verificado em um ambiente controlado se os objetos alterados no experimento foram identificados pelo algoritmo como *outlier* espaciais.

Essa base sintética contém os agrupamentos e seus respectivos objetos, nos quais os atributos espaciais foram gerados randomicamente na região de São José do Rio Preto e os valores dos atributos não-espaciais são iguais para todos os objetos. Cada agrupamento possui 100, 500 e 800 objetos como pode ser analisado na Tabela 2.

Tabela 2 Número de objetos nos agrupamentos

Agrupamento	Número de objetos espaciais
Agrupamento 0	100
Agrupamento 1	500
Agrupamento 2	800

Em cada agrupamento foram alterados 2 e 5% dos objetos espaciais para execução dos experimentos, o número de objetos alterados pode ser analisado na Tabela 3. As alterações dos objetos foram realizadas nos atributos não-espaciais de maneira aleatória, alterando-se 1, 2 ou os três atributos não-espaciais. Os tipos do atributos não-espaciais foram texto, inteiro e booleano, o que comprova a utilização de dados categóricos no algoritmo.

Tabela 3 Número de objetos modificados nos agrupamentos

Agrupamento	Nº de objetos modificados (2%)	Nº de objetos modificados (5%)
0	2	5
1	10	25
2	16	40

5.2.1 Dois atributos não-espaciais

Neste experimento foram utilizados dois atributos não-espaciais com 2% dos objetos modificados e executado o algoritmo *PCR-SCOD*-MR.

Pela regra dos quartis, para o agrupamento 0, os valores acima de 0.001029 são considerados *outliers*. Dessa forma, apenas dois objetos apresentaram valores superior a 0.001 e são apresentados na Tabela 4.

Tabela 4 Valor dos *outliers*

Identificador	Valor
51	0.002323
77	0.002359

Ao fazer uma consulta na base foi verificado que esses dois objetos foram os modificados no agrupamento 0, como é destacado em amarelo na Figura 16.

	id integer	centroide integer	atributo1 text	atributo2 integer
49	49	0	atributo1	1
50	50	0	atributo1	1
51	51	0	modificado	2
52	52	0	atributo1	1
53	53	0	atributo1	1
54	54	0	atributo1	1
55	55	0	atributo1	1
56	56	0	atributo1	1
57	57	0	atributo1	1
58	58	0	atributo1	1
59	59	0	atributo1	1
60	60	0	atributo1	1
61	61	0	atributo1	1
62	62	0	atributo1	1
63	63	0	atributo1	1
64	64	0	atributo1	1
65	65	0	atributo1	1
66	66	0	atributo1	1
67	67	0	atributo1	1
68	68	0	atributo1	1
69	69	0	atributo1	1
70	70	0	atributo1	1
71	71	0	atributo1	1
72	72	0	atributo1	1
73	73	0	atributo1	1
74	74	0	atributo1	1
75	75	0	atributo1	1
76	76	0	atributo1	1
77	77	0	modificado	2
78	78	0	atributo1	1
79	79	0	atributo1	1
80	80	0	atributo1	1

Unix Ln 1, Col 118, Ch 118 100 rows.

Figura 16 Consulta para verificação dos objetos modificados

Para os demais agrupamentos foi realizado o mesmo procedimento e foi verificado que o algoritmo conseguiu tanto identificar os objetos mais dissimilares no agrupamento, quanto estimar o número de objetos a serem considerados como *outliers* por meio da regra dos quartis com 100% de precisão. Nas Tabela 5 é apresentado o valor para ser considerado *outlier*, o intervalo dos objetos que foram considerados *outliers* e o número de *outliers* encontrado pelo algoritmo.

Tabela 5 Informação dos *outliers* nos agrupamentos para 2% de objetos modificados e dois atributos não-espaciais

Agrupamento	Valor para ser outlier	Intervalo dos valores encontrados como outliers	Número de outliers encontrados
0	0.001029	[0.002323, 0.002359]	2
1	1.003564E-4	[1.195941E-4, 6.085251E-4]	10
2	4.583845E-5	[7.874449E-5, 1.139622E-4]	16

O mesmo experimento foi a realizado para dois atributos não-espaciais e 5% dos objetos modificados e o mesmo resultado foi obtido, o algoritmo identificou 100% dos objetos modificados. Nas Tabela 6 é apresentado esses dados.

Tabela 6 Informação dos *outliers* nos agrupamentos para 5% dos objetos modificados e dois atributos não-espaciais

Agrupamento	Valor para ser outlier	Intervalo dos valores encontrados como outliers	Número de outliers encontrados
0	0.001070	[0.001949, 0.002225]	5
1	9.595030E-5	[1.670175E-4, 1.902577E-4]	25
2	4.716217E-5	[8.462998E-5, 1.002804E-4]	40

5.2.2 Três atributos não-espaciais

Com intuito de verificar o algoritmo para vários atributos, o mesmo experimento foi realizado para três atributos não-espaciais com 2 e 5% dos objetos modificados.

Dessa forma, é possível verificar nas Tabela 7 e Tabela 8 que mesmo com o aumento no número de atributos não-espaciais o algoritmo identificou corretamente os objetos modificados.

Tabela 7 Informação dos *outliers* nos agrupamentos para 2% dos objetos modificados e três atributos não-espaciais

Agrupamento	Valor para ser outlier	Intervalo dos valores encontrados como outliers	Número de outliers encontrados
0	0.001029	[0.002323, 0.002359]	2
1	1.027638E-4	[1.729391E-4, 9.730153E-4]	10
2	4.575796E-5	[6.711457-5, 1.476753E-4]	16

Tabela 8 Informação dos *outliers* nos agrupamentos para 5% dos objetos modificados e três atributos não-espaciais

Agrupamento	Valor para ser outlier	Intervalo dos valores encontrados como outliers	Número de outliers encontrados
0	0.001070	[0.001949, 0.002225]	5
1	9.576293E-5	[1.662312-4, 2.042796E-4]	25
2	4.719056E-5	[8.456750-5, 1.002811E-4]	40

Outro ponto a ser observado é que no agrupamento 0 os valores de identificação de *outliers* são os mesmos, uma vez que foram modificados todos os atributos não-espaciais, e dessa forma a tendência a ser um *outlier* é a mesma. Já nos demais casos, os valores se alteram, pois nem sempre todos os objetos não-espaciais foram alterados.

Para ilustrar o experimento com três atributos não-espaciais, foi realizado uma consulta no agrupamento 0 com 5% de objetos modificados, como é apresentado na Figura 17, onde tem-se destacados em amarelo os 5 objetos.

1	0	atributo1	1	t
2	0	modificado	2	f
3	0	atributo1	1	t
4	0	atributo1	1	t
5	0	atributo1	1	t
6	0	atributo1	1	t
7	0	atributo1	1	t
8	0	atributo1	1	t
9	0	atributo1	1	t
10	0	atributo1	1	t
11	0	atributo1	1	t
12	0	atributo1	1	t
13	0	atributo1	1	t
14	0	atributo1	1	t
15	0	atributo1	1	t
16	0	atributo1	1	t
17	0	atributo1	1	t
18	0	atributo1	1	t
19	0	atributo1	1	t
20	0	atributo1	1	t
21	0	atributo1	1	t
22	0	atributo1	1	t
23	0	atributo1	1	t
24	0	atributo1	1	t
25	0	atributo1	1	t
26	0	atributo1	1	t
27	0	atributo1	1	t
28	0	atributo1	1	t
29	0	atributo1	1	t
30	0	atributo1	1	t
31	0	atributo1	1	t
32	0	atributo1	1	t
33	0	atributo1	1	t
34	0	atributo1	1	t
35	0	atributo1	1	t
36	0	atributo1	1	t
37	0	atributo1	1	t
38	0	modificado	2	f
39	0	modificado	2	f
40	0	atributo1	1	t
41	0	atributo1	1	t
42	0	atributo1	1	t
43	0	atributo1	1	t
44	0	atributo1	1	t
45	0	atributo1	1	t
46	0	atributo1	1	t
47	0	atributo1	1	t
48	0	atributo1	1	t
49	0	atributo1	1	t
50	0	atributo1	1	t

51	0	atributo1	1	t
52	0	atributo1	1	t
53	0	atributo1	1	t
54	0	atributo1	1	t
55	0	atributo1	1	t
56	0	atributo1	1	t
57	0	atributo1	1	t
58	0	atributo1	1	t
59	0	atributo1	1	t
60	0	atributo1	1	t
61	0	atributo1	1	t
62	0	modificado	2	f
63	0	atributo1	1	t
64	0	atributo1	1	t
65	0	atributo1	1	t
66	0	atributo1	1	t
67	0	atributo1	1	t
68	0	atributo1	1	t
69	0	atributo1	1	t
70	0	modificado	2	f
71	0	atributo1	1	t
72	0	atributo1	1	t
73	0	atributo1	1	t
74	0	atributo1	1	t
75	0	atributo1	1	t
76	0	atributo1	1	t
77	0	atributo1	1	t
78	0	atributo1	1	t
79	0	atributo1	1	t
80	0	atributo1	1	t
81	0	atributo1	1	t
82	0	atributo1	1	t
83	0	atributo1	1	t
84	0	atributo1	1	t
85	0	atributo1	1	t
86	0	atributo1	1	t
87	0	atributo1	1	t
88	0	atributo1	1	t
89	0	atributo1	1	t
90	0	atributo1	1	t
91	0	atributo1	1	t
92	0	atributo1	1	t
93	0	atributo1	1	t
94	0	atributo1	1	t
95	0	atributo1	1	t
96	0	atributo1	1	t
97	0	atributo1	1	t
98	0	atributo1	1	t
99	0	atributo1	1	t
100	0	atributo1	1	t

Figura 17 Consulta do agrupamento para três atributos não-espaciais

Pode-se verificar que os mesmos 5 objetos encontrados como *outliers* pelo algoritmo na Tabela 9 foram os destacados na Figura 17.

Tabela 9 *Outliers* identificados

Identificador	Valor
2	0.002225
38	0.002213
39	0.002213
62	0.001959
70	0.001949

5.3 Experimento da abordagem

O segundo experimento visa validar a abordagem de melhoria na similaridade de agrupamentos espaciais resultantes de qualquer algoritmo de agrupamentos espaciais. Dessa forma serão realizados experimentos em dois tipos diferentes de algoritmos de agrupamentos espaciais, sendo o primeiro o *CHSMST+* e o segundo *k-means*.

O *CHSMST+* visa criar agrupamentos mais similares a partir dos atributos não-espaciais. Em contrapartida, foi executado também o *k-means*, que utiliza somente a distância entre os objetos para formação do agrupamento. O *CHSMST+* é um algoritmo baseado no *CHSMST* e utiliza o método de grafos para geração dos agrupamentos espaciais, já *k-means* é um algoritmo consagrado na literatura, tanto no domínio de dados convencionais quanto no domínio de dados espaciais, e utiliza o método de particionamento.

A base de dados a ser utilizada é uma base real referente ao Sistema de Vigilância de Acidentes de Trabalho - SIVAT, que armazena casos de acidentes de trabalho em São José do Rio Preto e região, totalizando mais de 100 municípios atendidos.

No ano de 2013 há mais 3600 registros de acidentes de trabalho georreferenciados quanto ao local do acidente e foram utilizados para os experimentos da abordagem. É importante ressaltar que como os registros são a respeito de casos de acidentes de trabalho, é

possível que tenha-se coordenadas geográficas iguais devido a acidentes na mesma empresa, por exemplo.

Para validação da qualidade dos agrupamentos com e sem a abordagem foram utilizadas as técnicas descritas na seção 4.3.4 referentes a distâncias entre os objetos e a similaridade dos atributos não-espaciais.

5.3.1 Abordagem com algoritmo *CHSMST+*

No primeiro experimento com o algoritmo *CHMST+* foram utilizados o atributo espacial referente ao local que ocorreu o acidente e dois atributos não-espaciais: ocorrência de afastamento do trabalho e tipo de acidente. O atributo não-espacial tipo de acidente é referente a doença de trabalho, acidentes típicos e acidentes de trajeto.

Foram gerados 68 agrupamentos com média de 34 registros cada agrupamento. Após isso, foi realizado os cálculos da silhueta e da similaridade de cada agrupamento com e sem a abordagem.

Na Figura 18 pode-se analisar, tanto no gráfico como na tabela, que houve um aumento na similaridade média dos agrupamentos e não houve perda no coeficiente da silhueta em relação à distância, mesmo com a identificação dos *outliers*. Assim o aumento da média de similaridade nos agrupamentos resultantes foi de 6,25%.

Foram encontrados *outliers* em 38 agrupamentos sendo a média de melhoria de similaridade desses agrupamentos de 11,50%.

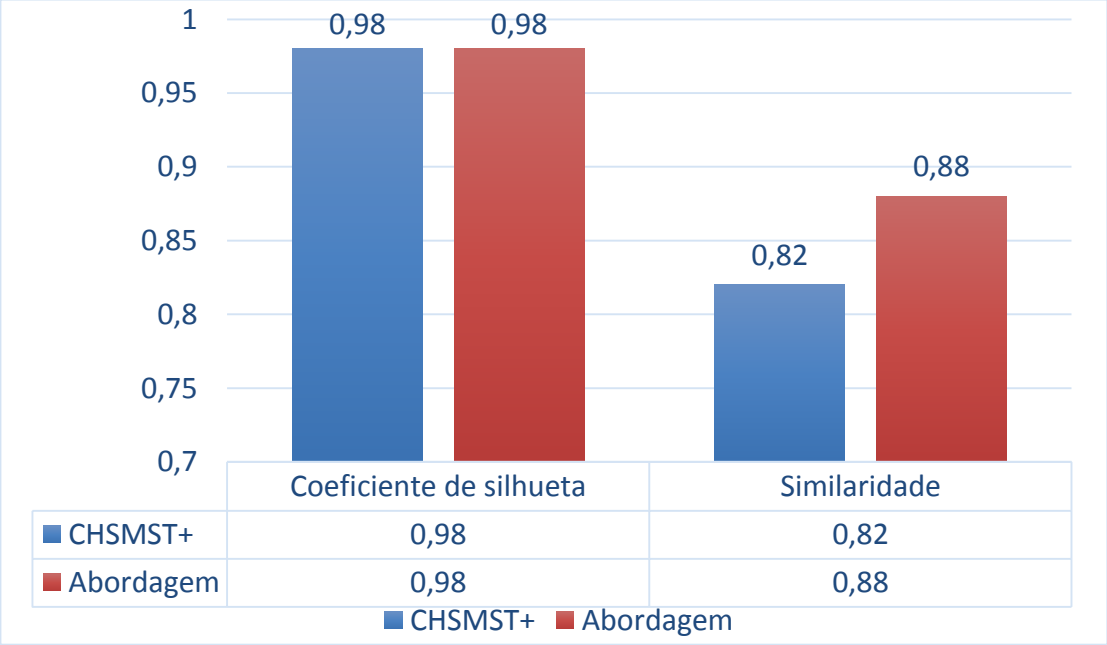


Figura 18 Gráfico e tabela da comparação entre o resultado com e sem a abordagem para dois atributos não-espaciais

Com a utilização da ferramenta é possível ainda analisar em que região de São José do Rio Preto os agrupamentos foram gerados por meio do mapa.

Na Figura 19 são apresentados os agrupamentos formados pelos registros que são os quadrados em azul juntamente com os quadrados vermelhos que representam os *outliers* detectados nos agrupamentos.

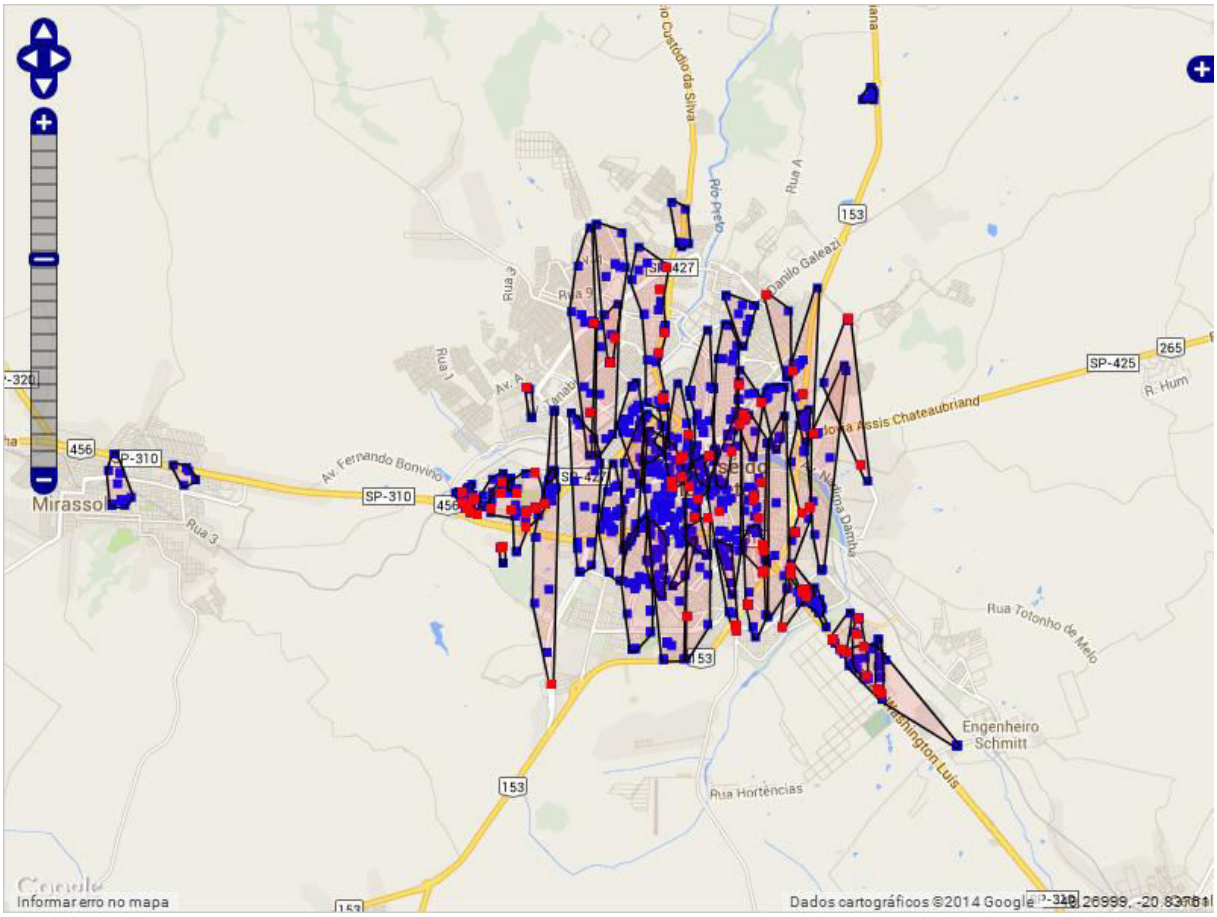


Figura 19 Agrupamentos com os *outliers* destacados para dois atributos não-espaciais

Para exemplificar a abordagem foi selecionado um agrupamento gerado na cidade de José Bonifácio que contem 234 acidentes de trabalho e foram detectados 7 *outliers*. O coeficiente de silhueta média do agrupamento se manteve em 0,99 enquanto a similaridade do agrupamento para o algoritmo CHSMST+ foi de 0,79 e com a abordagem 0,81. Na Figura 20 é possível analisar os registros dos agrupamentos e seus *outliers*.

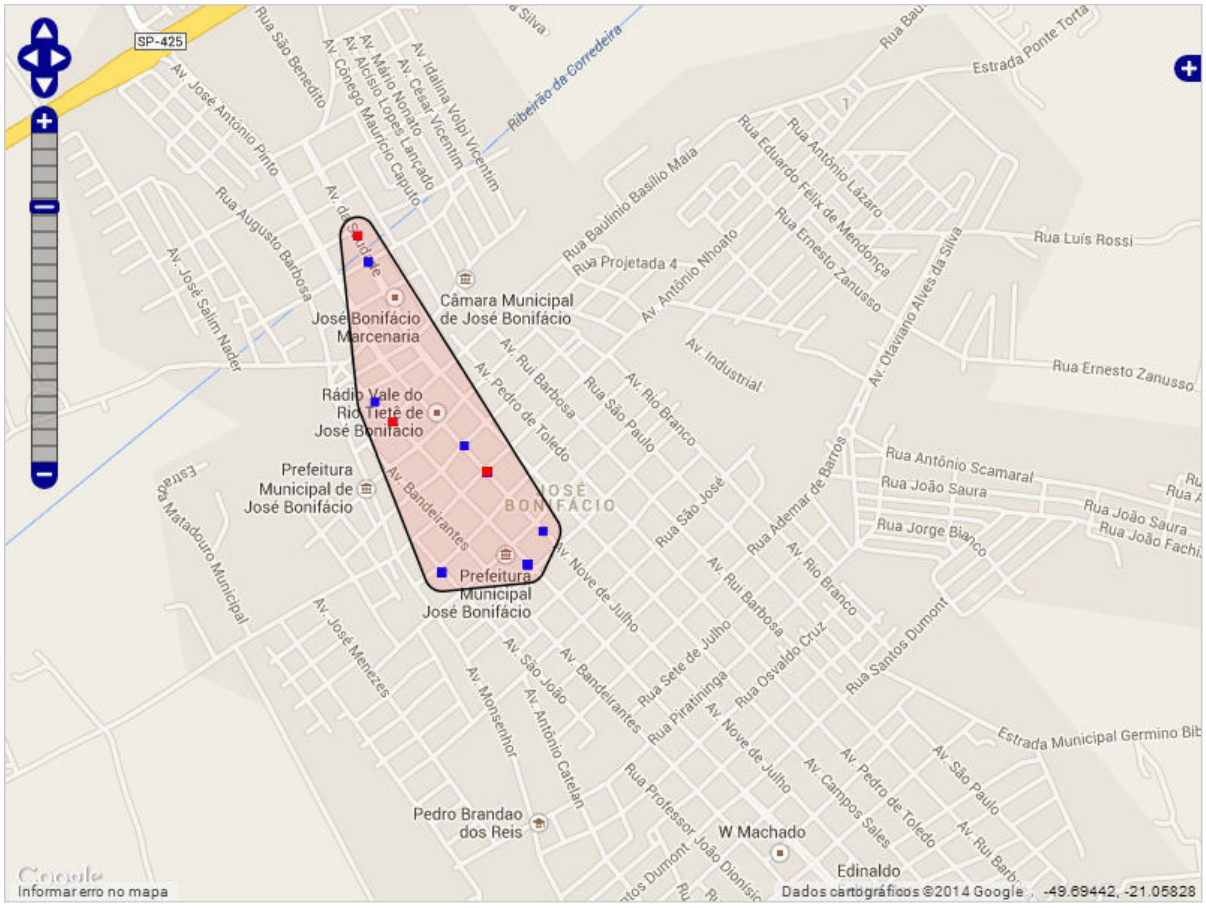


Figura 20 Agrupamento em José Bonifácio

Com esse agrupamento é possível fazer uma análise mais pontual dos casos de acidentes de trabalhos, e consequentemente tomar decisões que promovam melhorias nas condições de trabalho e na qualidade de vida dos trabalhadores. Neste caso, os acidentes de trabalho ocorreram em José Bonifácio, no qual os 227 acidentes foram típicos e 176 foram acidentes mais graves, no qual o trabalhador teve que se afastar das suas atividades. Ao verificar o ramo de atividade envolvida, foi verificado que dos 176 trabalhadores afastados, 140 trabalhava em uma empresa no ramo frigorífico, dessa forma os órgãos públicos podem tomar medidas para verificação e melhorias das condições de trabalho no local do acidente.

No segundo experimento com o algoritmo *CHSMST+* foram utilizados o atributo espacial, também referente ao local que ocorreu o acidente e três atributos não-espaciais: local do acidente de trabalho, se ocorreu internação, e se levou ao óbito do trabalhador. O atributo não-espacial local do acidente pode ser: instalação do empregador, via pública, instalação de terceiros, domicílio próprio, zona rural, outros ou não informado.

Foram gerados 70 agrupamentos com média de 41 registros cada agrupamento. Após isso, foi realizado os cálculos da silhueta e da similaridade de cada agrupamento com e sem a abordagem como no primeiro experimento.

Dessa forma, foi criado o gráfico juntamente com a tabela para melhor entendimento como é apresentado na Figura 21. A média do coeficiente da silhueta em relação à distância para os agrupamentos é de 0,97 tanto para o *CHSMST+* como para a abordagem e a similaridade teve um aumento de 2,11%. Desses 70 agrupamentos, 35 foram identificados contendo *outliers* e nesses teve um aumento de 4,22% na similaridade. Isso ocorre, devido a uma alta similaridade nos agrupamentos iniciais gerados pelo algoritmo *CHSMST+*.

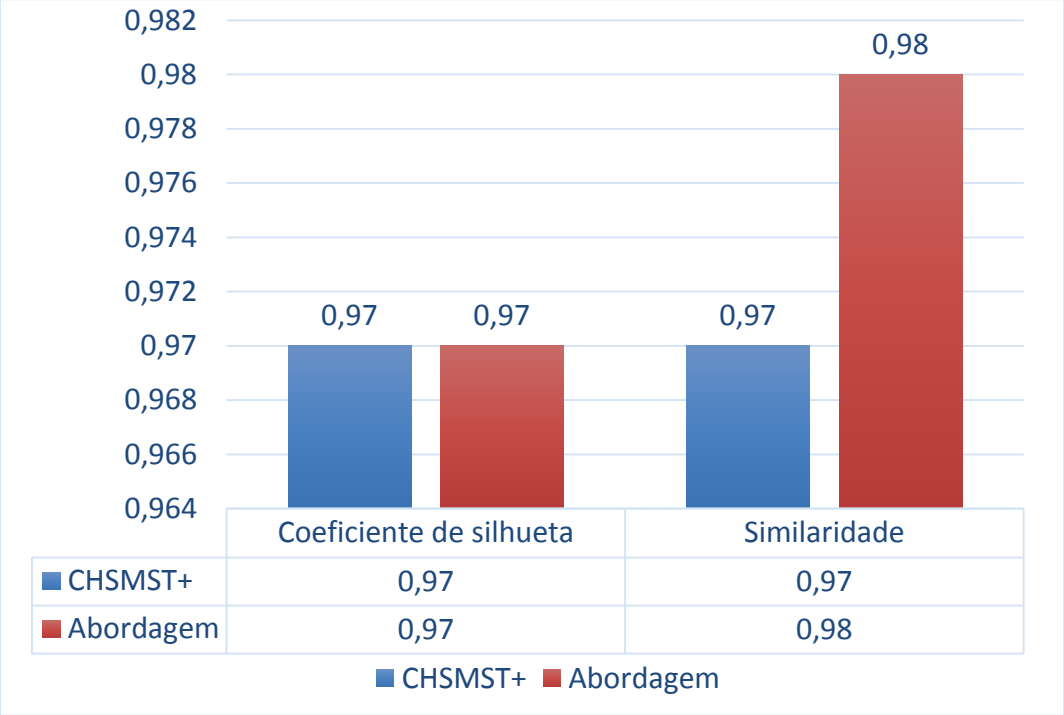


Figura 21 Gráfico e tabela da comparação entre o resultado com e sem a abordagem para três atributos não-espaciais

Na Figura 22 é apresentado o resultado da abordagem no mapa em relação aos agrupamentos formados a partir da similaridade e distância entre os registros por meio dos três atributos não-espaciais.

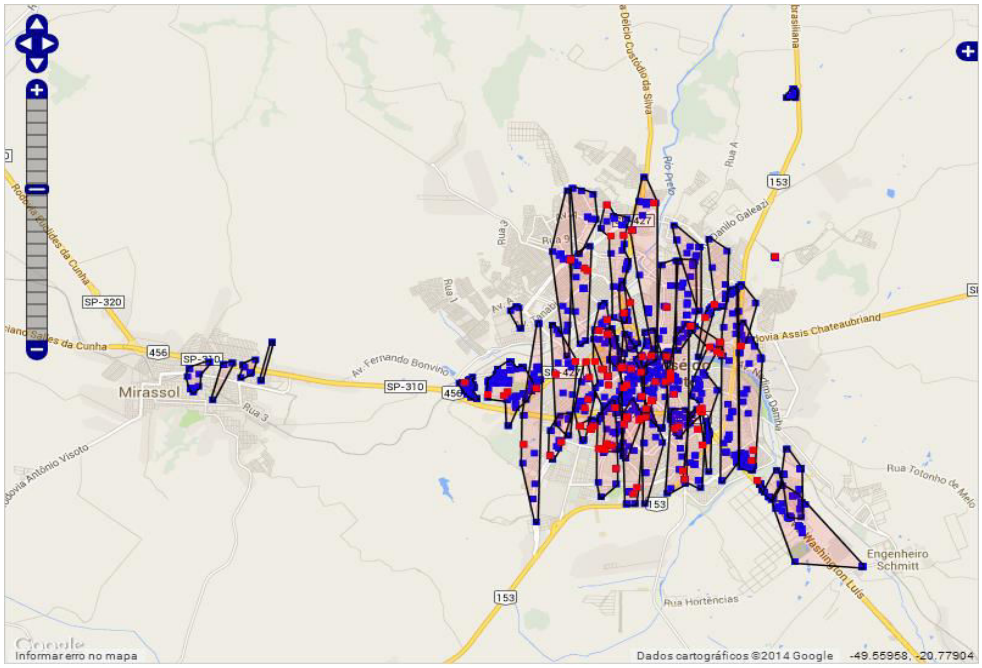


Figura 22 Agrupamentos com os outliers destacados para dois atributos não-espaciais

Foi selecionado um agrupamento gerado na cidade de São José do Rio Preto que contém 37 acidentes de trabalho e foram detectados 5 *outliers*. Na Figura 23 é possível analisar os registros do agrupamento e seus *outliers*, no qual o local do acidente é a Instalação do empregador, sendo que nenhum dos acidentes teve internação ou óbito do trabalhador. Outro ponto a ser destacado é que dentre os 32 acidentes, 25 são da área da saúde dos quais 22 são enfermeiros. Assim as autoridades podem fazer um acompanhamento dos funcionários que trabalham nessa área.

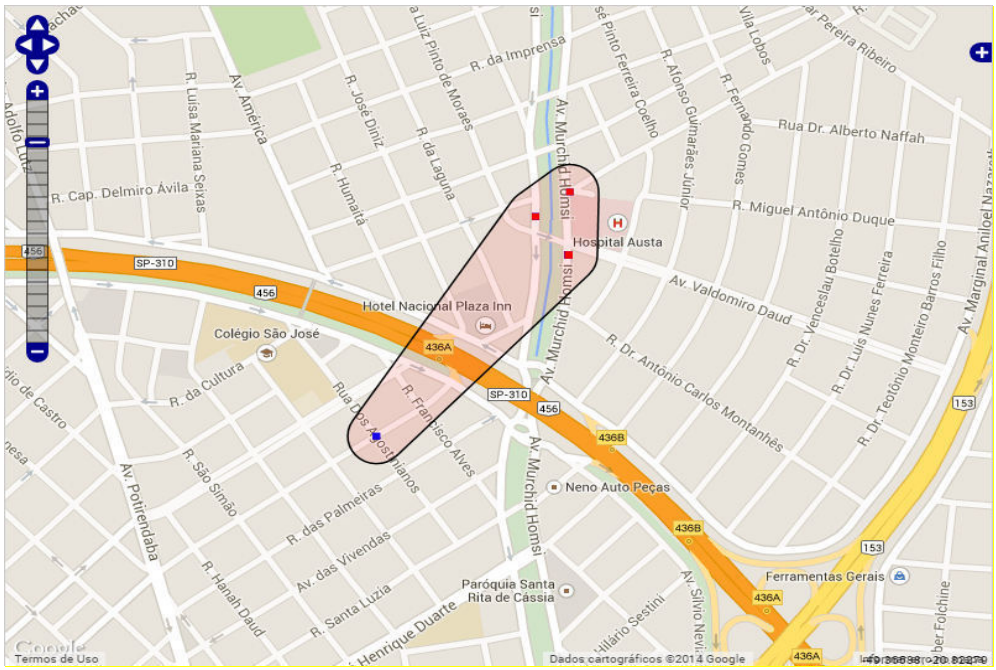


Figura 23 Agrupamento de acidentes de trabalho em São José do Rio Preto

5.3.2 Abordagem com o algoritmo *k-means*

Para o experimento do algoritmo *k-means* foi selecionado dois atributos não-espaciais: sexo e faixa etária, sendo a faixa etária: menor, jovem, adulto e idoso e o atributo espacial referente ao local que ocorreu o acidente de trabalho.

Como parâmetro de entrada o número de agrupamentos foi 30, porém ao final do algoritmo foram gerados apenas 28 agrupamentos. Isso pode ocorrer quando os objetos possuem uma distância menor em relação ao outro agrupamento do que foi inserido inicialmente devida as várias interações e os recálculos dos centróides.

Como o *k-means* não considera a similaridade dos atributos não-espaciais, a similaridade média dos agrupamentos é baixa ao comparado com o *CHSMST+*. Porém é importante verificar que a abordagem proposta para algoritmos de agrupamentos espaciais também aumentam a similaridade mesmo com estes resultados.

Foi construído o gráfico com a tabela apresentado na Figura 24, e analisado que mesmo com os agrupamentos obtidos houve um pequeno aumento em relação à similaridade e ao coeficiente de silhueta. Isso era esperado devido à baixa similaridade nos agrupamentos iniciais.

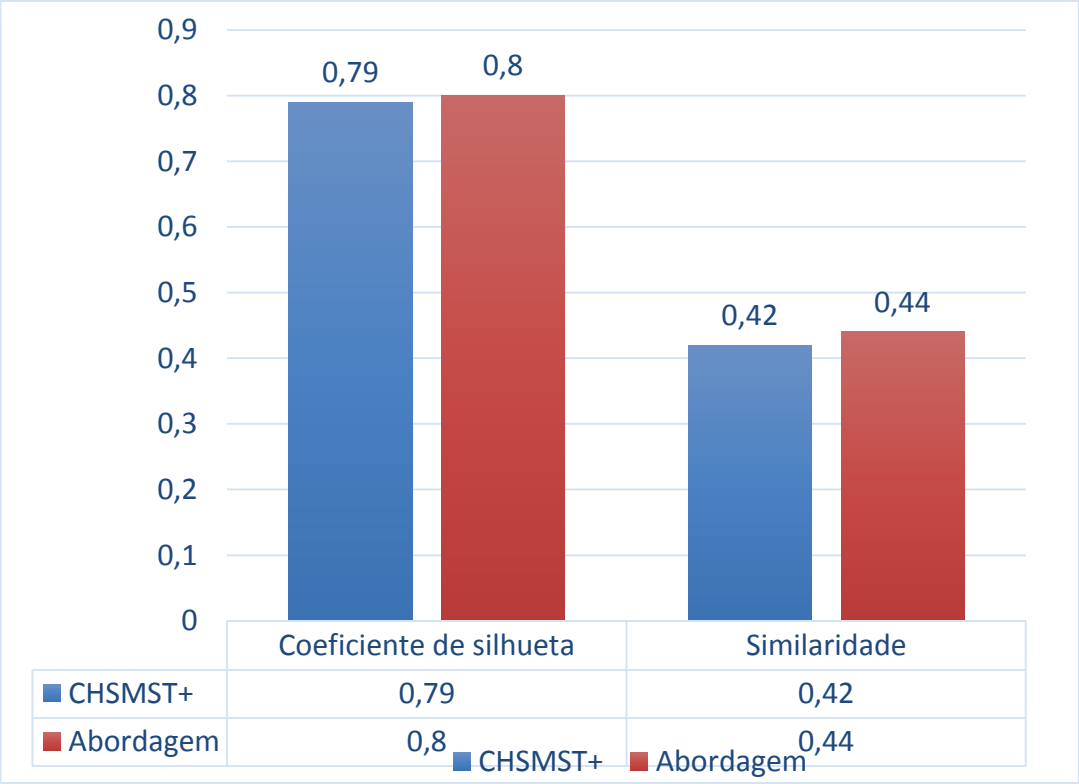


Figura 24 Gráfico e tabela da comparação entre o resultado com e sem a abordagem para *k-means*

Na Figura 25 é apresentado os agrupamentos gerados pelo algoritmo juntamente com os *outliers* destacados em vermelho.

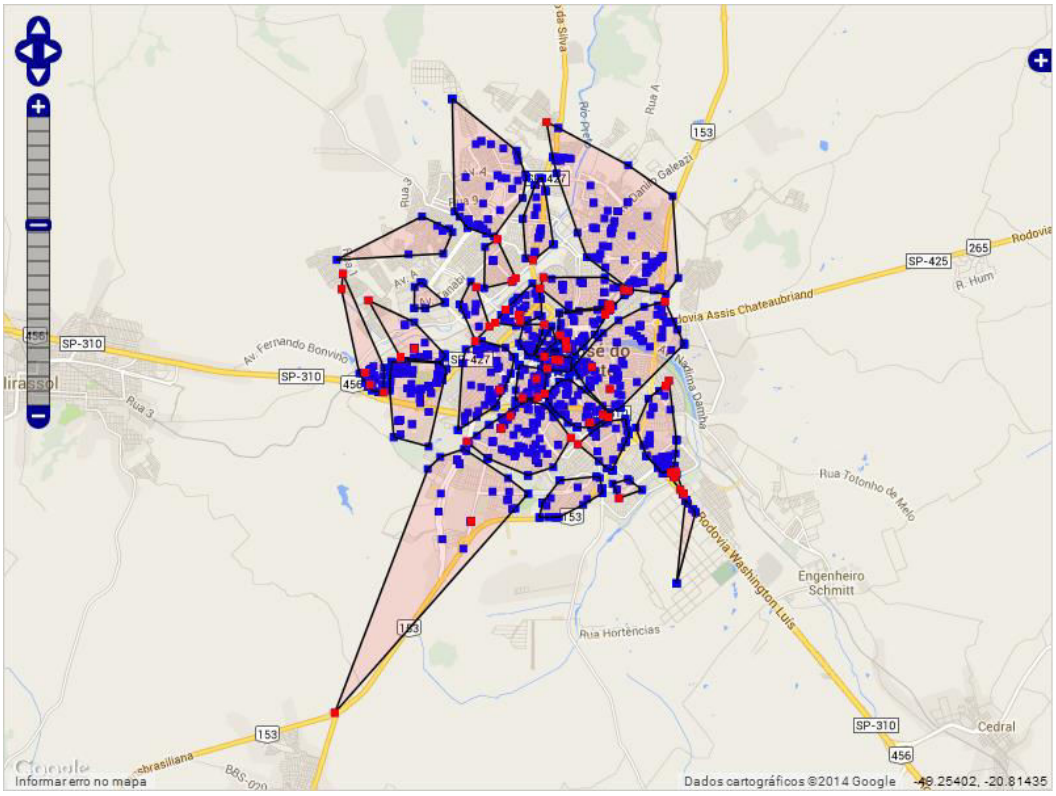


Figura 25 Agrupamentos com os *outliers* destacados para o resultado do *k-means*

Para exemplificar o experimento com o algoritmo k-means foi selecionado um agrupamento que contém 18 acidentes de trabalho, sendo 3 acidentes identificados como *outliers*. Além disso, é possível analisar que 12 ocorreram com trabalhadores do sexo masculino, dentre estes 9 são da faixa etária adulta, como é apresentado na Figura 26. Outra informação que pode ser extraída é que 12 acidentes ocasionaram o afastamento do trabalhador, o que demanda a verificação das autoridades em relação a gravidade desses acidentes.

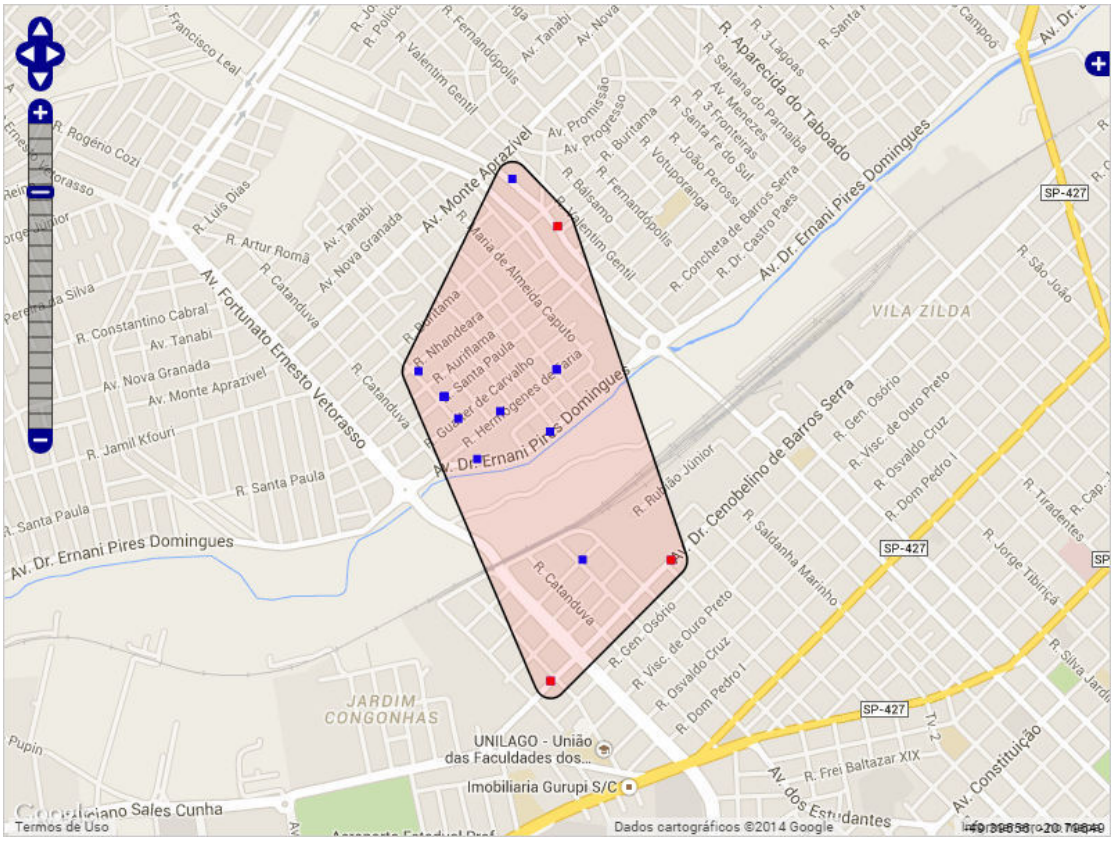


Figura 26 Agrupamento de acidentes de trabalho em São José do Rio Preto para o algoritmo *k-means*

5.4 Comparação da abordagem com PCR-SCOD-MR em relação a outro algoritmo de detecção de *outliers* espaciais

Para validar que a abordagem utilizando-se o PCR-SCOD-MR obtém resultados com maior similaridade do que outros algoritmos da literatura, foi implementado o algoritmo Z para fazer a comparação.

O algoritmo Z foi escolhido, pois é apresentado como base para testes em todos os trabalhos relevantes da área de detecção de outliers.

A partir disso, foi executado o algoritmo CHSMST+ em uma base real referente aos acidentes de trabalho (SIVAT). Foram utilizados como atributo não-espacial a cor do acidentado e o como atributo espacial o local do acidente.

O resultado dessa execução foi a geração de 90 agrupamentos com média de similaridade de 0,85.

Após isso, foram aplicados os algoritmos Z e PCR-SCOD-MR em cada um dos agrupamentos, que resultou em uma média de similaridade 0,86 e 0,92 respectivamente. Ou

seja, o PCR-SCOD-MR obteve uma melhora de similaridade de 7% em relação aos agrupamentos gerados pelo CHSMST+, em contrapartida o algoritmo Z obteve uma melhora de apenas 1%.

Outro ponto que deve ser destacado é que para alguns agrupamentos o algoritmo Z detectou *outliers* que diminuíram a similaridade dentro do agrupamento ou obtiveram um pequeno aumento na similaridade, enquanto o algoritmo PCR-SCOD-MR obteve para a maioria dos agrupamentos um aumento na similaridade e para o restante a similaridade foi mantida, como pode ser analisado na Tabela 10.

Tabela 10 Comparação da similaridade dos agrupamentos

Cluster	Similaridade do agrupamento com *outlier*	Similaridade após a execução do algoritmo PCR-SCOD-M	Similaridade após a execução do algoritmo Z
44	0,79	0,83	0,78
35	0,76	0,76	0,72
48	0,83	1	0,85

Dessa forma, foi possível verificar que a abordagem com o algoritmo PCR-SCOD-MR conseguiu aumentar a similaridade dos agrupamentos gerados ou mantê-la.

5.5 Análise do Desempenho do algoritmo PCR-SCOD-MR

O processamento do algoritmo *PCR-SCOD-MR* foi otimizado pela incorporação da abordagem *MapReduce*. Com o propósito de demonstrar a melhoria no desempenho foram realizados experimentos com e sem o *MapReduce* em um volume de dados maior.

Foi executado o algoritmo *k-means*, com parâmetro de entrada de 200 agrupamentos, para uma base de dados real que contém árvores do distrito de *Bronx* nos Estados Unidos. A base contém mais de 56 mil árvores georreferenciadas e foram escolhidos dois atributos não-espaciais. Assim foi realizada uma média com os números de objetos em cada agrupamento para cada 40 agrupamentos. Dessa forma, os agrupamentos possuem os seguintes volumes de dados: 51, 100, 500, 806 e 1014 objetos.

Para o teste que utiliza o *MapReduce* foi construído um *cluster* com três máquinas sendo 1 mestre e 2 escravos com as seguintes configurações:

- Máquina mestre - processador Intel(R) Pentium(R) CPU G2030 @ 3.00GHz, memória de 2GB, HD de 250GB e Sistema operacional Ubuntu Server 12.04.4 LTS 64 bits

- Máquina escravo 1 - processador AMD Athlon(tm) 64 X2 Dual Core Processor 5600+, memória de 4GB, HD de 320GB e Sistema operacional Ubuntu Server 12.04.4 LTS 64 bits

- Máquina escravo 2 - processador AMD Athlon(tm) 64 X2 Dual Core Processor 5000+, memória de 4GB, HD de 320GB e Sistema operacional Ubuntu Server 12.04.4 LTS 64 bits

Para os testes sem a abordagem, foi utilizada a máquina mestre do *cluster* criado por ser a melhor máquina dentre as três utilizadas em relação ao processador. Esses testes podem ser realizados em apenas uma máquina, uma vez que a execução do algoritmo sem a abordagem é sequencial.

Para a aferição do tempo de execução do algoritmo com e sem o MapReduce para os diferentes volumes de dados, foram realizadas três execuções e feita a média. A seguir, na Tabela 11 é apresentado o tempo médio de execução para cada caso e na Figura 27 o respectivo gráfico.

Tabela 11 Tempo médio de execução

Número de árvores no agrupamento	Tempo sem MapReduce (milissegundos)	Tempo com MapReduce (milissegundos)
51	883	81244
100	5783,33	81313
302	301082,33	97599
500	2382104,5	124512
806	-	168053
1014	-	392900

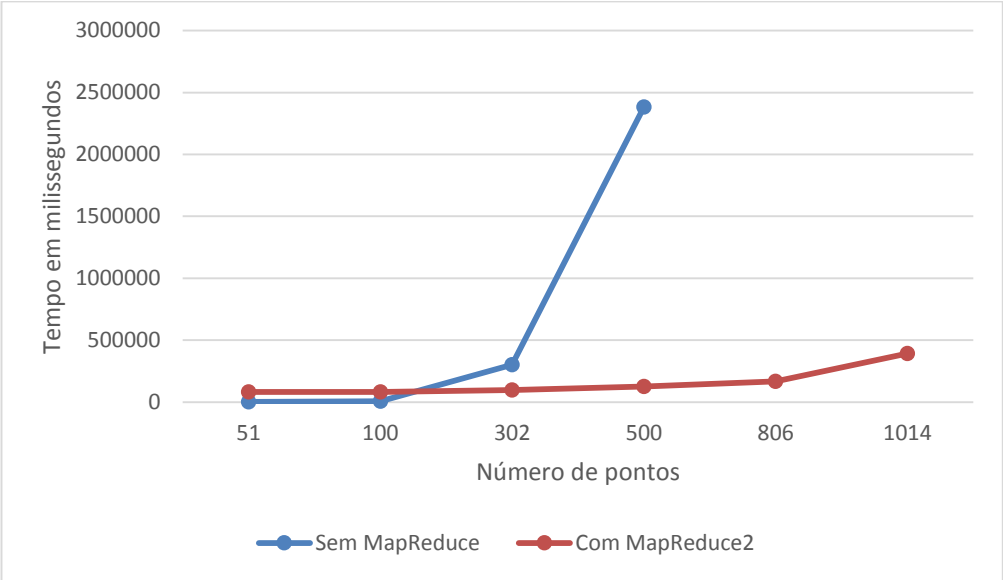


Figura 27 Gráfico do tempo de execução

Com isso é possível verificar que a abordagem com *MapReduce* para agrupamentos com pouco volume de dados como 51 e 100 árvores teve um desempenho inferior do que a sem a *MapReduce*, uma vez que essa técnica é para grandes volumes de dados o que é a tendência das bases de dados atuais.

Já para agrupamentos maiores a abordagem teve um grande desempenho em relação ao tempo de execução, e assim, houve uma redução significativa. Para os agrupamentos com 806 e 1024 a abordagem demorou em média 168053 e 392900 segundos, sem o MapReduce o tempo de execução passou de 1 hora e houve problema em relação ao esgotamento da memória não sendo finalizada a execução.

Para comprovar esse esgotamento de memória, foi verificado também a quantidade de memória utilizada pela aplicação em cada um dos experimentos realizados nessa sessão.

Como a aplicação foi desenvolvida em linguagem Java, a mesma é executada em uma JVM – *Java Virtual Machine*, ou seja, é disponibilizado um espaço determinado da memória RAM total para o uso da JVM. Dessa forma, a memória para a JVM das máquinas utilizadas são: Máquina mestre – 1007M; Máquina escravo 1 – 123M; Máquina escravo 2 – 123M. Assim o algoritmo sem a abordagem tem a disponibilidade de 1007M e o algoritmo com a abordagem tem a disponibilidade da máquina mestre mais as suas máquinas escravos contabilizando 1253M, dessa forma para o algoritmo com a abordagem, a memória utilizada é a soma da memória utilizada pelas três máquinas.

Na Figura 28 é apresentado o gráfico com a comparação entre a execução do algoritmo com e sem a abordagem. É possível verificar que para agrupamentos com 500 objetos sem a abordagem a aplicação já utilizou toda a memória, porém conseguiu finalizar a execução, para agrupamentos maiores, como é o caso do agrupamento com 806, não foi possível a finalização da execução por esgotamento da memória. Para o algoritmo com a abordagem, em um agrupamento com 1014 pontos, a soma da memória utilizada pelas três máquinas foi 405MB que é menos de um terço da memória disponível.

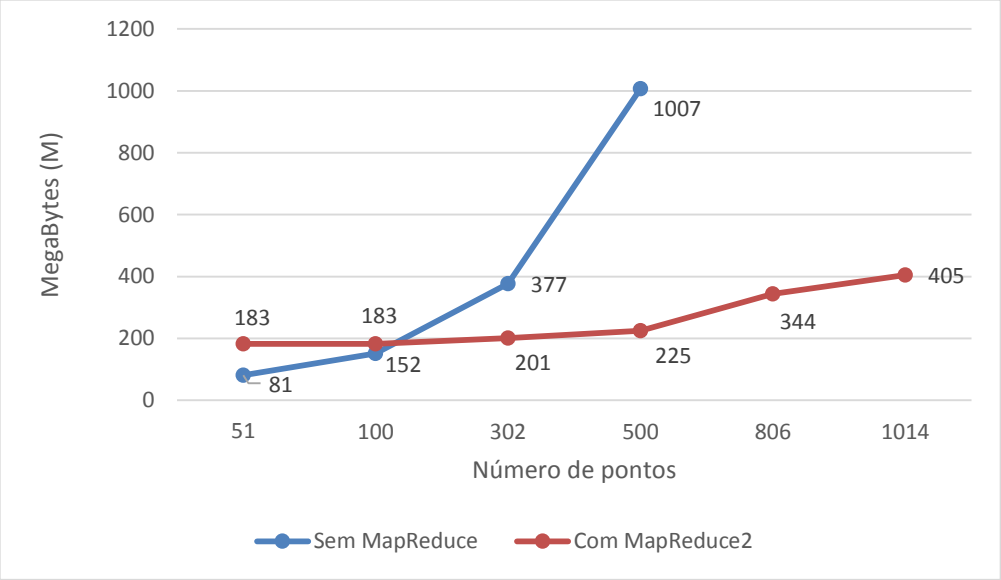


Figura 28 Gráfico da memória máxima utilizada durante a execução

5.6 Considerações Finais

Foram apresentados vários experimentos para validar tanto o algoritmo *PCR-SCOD-MR* como a abordagem proposta.

Primeiramente foi validado que não houve perda na identificação dos *outliers* espaciais devido as alterações realizadas no algoritmo base como os parâmetros de entradas que foram retirados e são definidos pelo próprio algoritmo e a execução para mais de um atributo não-espacial.

O segundo experimento verificou a abordagem proposta para dois tipos de algoritmos de agrupamentos espaciais, em condições de alta e baixa similaridade nos agrupamentos obtidos pelos algoritmos de agrupamentos espaciais. Dessa forma foi possível verificar que em todos

os casos houve uma melhoria na similaridade, porém obteve um resultado mais expressivo diante de agrupamentos que já possuíam uma similaridade média para alta. Isso ocorre devido ao algoritmo definir um padrão de atributos entre os objetos, ou seja, se os atributos tem baixa similaridade o algoritmo tem dificuldade em encontrar esse padrão para validar quais os objetos que são *outliers*.

No terceiro experimento foi analisado a abordagem posposta para o algoritmo PCR-SCOD-MR e o algoritmo Z. Dessa forma foi possível verificar que para o algoritmo PCR-SCOD-MR houve uma melhora significativa ao se utilizar essa abordagem, enquanto a melhoria dos resultados para o outro algoritmo foi menos expressiva.

Por fim, foi apresentada a análise do desempenho do algoritmo com e sem a abordagem de *MapReduce* que apresentou melhoras no tempo de execução do algoritmo principalmente para agrupamentos com um número elevado de dados.

Capítulo 6
Conclusões

6.1 Considerações

A detecção de *outlier* espaciais é uma técnica importante na tarefa de *Spatial Data Mining* e abrange diversos domínios de aplicação. No entanto, a área de detecção de *outliers* espaciais ainda é recente, e possui vários fatores limitantes.

Nesse cenário foi realizado um levantamento bibliográfico da área de detecção de *outliers* espaciais, bem como casos de sucesso ao aplicar os algoritmos de detecção de *outliers* em agrupamentos espaciais. Uma vez que, muitos algoritmos de agrupamentos espaciais não conseguem detectar os *outliers* locais nos agrupamentos, o que gera resultados com baixa similaridade e difíceis de serem analisados (CHAWLA; GIONIS, 2013).

Com isso, foi encontrada a motivação para a confecção desse trabalho, sendo as principais contribuições:

- Abordagem de detecção de *outliers* espaciais em agrupamentos espaciais - foi criada uma abordagem, no qual o algoritmo de detecção de *outliers* espaciais é aplicado nos resultados do algoritmo de agrupamentos espaciais, o que ocasiona o aumento na similaridade média e uma facilidade na extração do conhecimento;
- Algoritmo *PCR-SCOD-MR* – foram realizadas melhorias no algoritmo PCR-SCOD de forma a não possuir parâmetros de entrada e contemplar a utilização de mais de um atributo não-espacial;

- Incorporação do *MapReduce* - para redução do tempo de processamento e melhoria da utilização da memória na execução do algoritmo, foi utilizada a abordagem MapReduce.

Como contribuições secundárias foram efetuados: um estudo comparativo entre os principais algoritmos da área de detecção de *outliers*, reimplementarão do algoritmo *PCR-SCOD* em linguagem *Java*, além da visualização dos resultados de forma intuitiva por meio de mapas.

Por fim, foram analisadas e comprovadas as hipóteses descritas na introdução do trabalho por meio dos experimentos que validam as técnicas de melhoria do algoritmo *PCR-SCOD-MR*, a abordagem proposta em uma base real e o desempenho do algoritmo, com a redução do tempo de processamento e melhoria na utilização de memória.

6.2 Comparativo do PCR-SCOD-MR com demais algoritmos estudados

O desenvolvimento das estratégias utilizadas no algoritmo *PCR-SCOD-MR* possibilitam contemplar as demais características não consideradas no algoritmo base, conforme é apresentado na Tabela 12. Dessa forma, o algoritmo *PCR-SCOD-MR* passa a considerar todas as características importantes que os algoritmos de detecção de *outliers* espaciais deve conter.

Tabela 12 Estudos comparativos com o algoritmo PCR-SCOD-MR

Algoritmo	Suporta dados espaciais	Suporta dados categóricos	Melhoria de desempenho	Considera proximidade e atributos não-espaciais	Multivalorado	Sem dados de entrada
Scatterplot (SHEKHAR; LU; ZHANG, 2003)	Sim	Não	Não	Sim	Não	Não
Moran (SHEKHAR; LU; ZHANG, 2003)	Sim	Não	Não	Sim	Não	Não
Z (SHEKHAR; LU; ZHANG, 2003)	Sim	Não	Não	Sim	Não	Não
Interative Z (LU; CHEN;KOU, 2003a)	Sim	Não	Não	Sim	Não	Não
Graph-based spatial outlier (SHEKHAR; LU; ZHANG, 2001)	Sim	Não	Não	Sim	Não	Não
Variância do algoritmo Z (SURI, et al, 2013)	Sim	Não	Sim	Sim	Não	Não

MR AVF (KOUFAKOU et. al., 2008)	Não	Sim	Sim	Não	Sim	Não
WDA (CAO et al., 2013)	Sim	Não	Não	Sim	Não	Não
ISOMRD (CAI; HE;MAN, 2013)	Sim	Não	Não	Sim	Sim	Não
PCR-SCOD (LIU; CHEN; LU, 2013)	Sim	Sim	Não	Sim	Não	Não
PCR-SCOD-MR	Sim	Sim	Sim	Sim	Sim	Sim

6.3 Trabalhos Futuros

Com o objetivo de prosseguir nos avanços de detecção de *outliers* espaciais para o algoritmo PCR-SCOD-MR pretende-se aplicar ontologia para facilitar a extração de conhecimento.

Outra possível contribuição é a aplicação da abordagem em bases de dados com maior número de dados espaciais e computadores mais robustos para utilização do *MapReduce,* de modo que nos experimentos de desempenho possam ser verificados os valores máximos da memória *Heap* para finalização do algoritmo.

Para ampliar ainda mais a capacidade da abordagem sugere-se modificar o algoritmo PCR-SCOD para contemplar a execução em base de dados espaciais com mais de duas dimensões, além da utilização de estruturas para melhorar ainda mais o desempenho, como por exemplo, a *Quadtree*.

Por fim, ampliar a capacidade de análise dos dados por meio de gráficos com visualização em três dimensões e novos recursos gráficos.

REFERÊNCIAS BIBLIOGRÁFICAS

BAKAR, Z. A. et al. A comparative study for outlier detection techniques in data mining. In: IEEE CONFERENCE ON CYBERNETICS AND INTELLIGENT SYSTEMS, n. 24, 2006, Singapore. **Proceedings**… Singapore: IEEE, 2006. p.1-6.

BICHLER, M.; KISS, C. A comparison of logistic regression, k-nearest neighbour, and decision tree induction. In: AMERICAS CONFERENCE ON INFORMATION SYSTEMS, 2004, New York. **Proceedings**… New York, 2004. p.1917-1925.

CAI, Q.; HE, H.; MAN, H. Spatial outlier detection based on iterative self-organizing learning model. **Neurocomputing**, Nijmegen, v.117, p.161-172, 2013.

CAO, L. et al. WSO-based spatial outlier detection algorithms. **Journal of Networks**, v.8, n.7, p.1582-1588, 2013.

CARREIRA, J. A. **Ferramenta de apoio à limpeza de dados no processo de extração de conhecimento de bases de dados**. 2008. 83f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação)-Instituto de Biociências, Letras e Ciências Exatas, Universidade Estadual Paulista "Júlio de Mesquita Filho", São José do Rio Preto, 2008.

CLEVE, C. et al. Classification of the wildland-urban interface: a comparison of pixel and object-based classifications using high-resolution aerial photography. **Computers, Environment and Urban Systems**, New York, v.32, n.4, p.317-326, 2008.

CHANDOLA, V.; BANERJEE, A.; KUMAR, V. Anomaly detection: a survey. **ACM Computing Surveys**, New York, v.41, n.3, p.1-58, 2009.

CHAWLA, S. GIONIS, A. kmeans--: A unified approach to clustering and outlier detection. In: SIAM INTERNATIONAL CONFERENCE ON DATA MINING, 2013, Texas. **Proceedings**… Texas: Omnipress, 2013. p.189-197.

CHEN, D. et al. On detecting spatial outliers. **GeoInformatica**, Boston, v.12, n.4, p.455-475, 2007.

CHEN, F.; LU, C-T.; BOEDIHARDJO, A. P. Gls-sod: a generalized local statistical approach for spatial outlier detection. In: ACM SIGKDD INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING, n. 3, 2010, New York. P**roceedings**… New York: ACM, 2010. p.1069-1078.

ESTIVILL-CASTRO, V.; LEE, I. AMOEBA: Hierarchical clustering based on spatial proximity using Delaunay diagram. In: INTERNATIONAL SYMPOSION ON SPATIAL DATA HANDLING, 2000, Beijing. **Proceedings…** Beijing, 2000, p.26-41.

FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. The KDD process for extracting useful knowledge from volumes of data. **Communications of the ACM**, New York, v.39, n.11, p.27–34, 1996.

FIGUERA, C. et al. Multivariate spatial clustering of traffic accidents for local profiling of risk factors. In: INTERNATIONAL IEEE CONFERENCE ON INTELLIGENT TRANSPORTATION SYSTEMS, 2011, Anchorage. **Proceedings**… Anchorage: IEEE, 2011. p.740-745.

GEOSERVER. **Servidor de mapas para web**. Disponível em: <http://geoserver.org >. Acesso em: 29 jun. 2014.

GOLDMAN, A. et al. Apache Hadoop: conceitos teóricos e práticos, evolução e novas possibilidades. In: JORNADAS DE ATUALIZAÇÃO EM INFORMÁTICA, n. 1, 2012, Porto Alegre. **Anais**… Porto Alegre, 2012.

GOOGLE. **Visualizar dados geoespaciais**. 2013. Disponível em: <https://developers.google.com/maps/visualize?hl=pt-br>. Acesso em: 29 jun. 2014.

GUHA, S.; RASTOGI, R.; SHIM, K. CURE: An efficient clustering algorithm for large databases. In: ACM SIGMOD INTERNATIONAL CONFERENCE MANAGEMENT OF DATA, 1998, Washington. **Proceedings…** Washington: ACM, 1998. p. 73-84.

HAN, J.; KAMBER, M.; PEI, J. **Data mining concepts and techniques**. 3Rd ed. São Francisco: Elsevier, 2011.

HOSSEINALIZADEH, M.; RIVAZ, F.; HEDAYATIZADEH, R. Importance of outlier detection in spatial analysis of wind erosion. **Procedia Environmental Sciences**, Amsterdam, v.7, p.341-346, jan. 2011.

HUANG, T.; QIN, X. Detecting outliers in spatial database. In: INTERNATIONAL CONFERENCE ON IMAGE AND GRAPHICS, 2004, Hong Kong. **Proceedings**… Hong Kong: IEEE, 2004. p.556-559.

JIN, H.; MIAO, B. The research progress of spatial data mining technique. In: INTERNATIONAL CONFERENCE ON COMPUTER SCIENCE AND INFORMATION TECHNOLOGY, 2010, Chengdu. **Proceedings**… Chengdu: IEEE, 2010. p.81-84.

KIM, Y. et al. DBCURE-MR: an efficient density-based clustering algorithm for large data using MapReduce. **Information Systems**, v.42, p.15-35, jun. 2014.

KOPERSKI, K.; HAN, J. Discovery of spatial association rules in geographic information databases. In: SCHOLL, M.; VOISARD, A. (Ed.). **Advances in spatial databases**. Portland: Springer-Verlag, 1997. p.47-66.

KOU, Y. **Abnormal pattern recognition in spatial data**. 2006. 126f. Tese (Doctor of Philosophy in Computer Science and Applications)-Faculty of Virginia Polytechnic Institute e State University, Virginia, 2006.

; LU, C-T; SANTOS, R. F. Spatial outlier detection: a graph-based approach. In: TOOLS WITH ARTIFICIAL INTELLIGENCE, 2007, Patras. P**roceedings**… Patras: IOS, 2007. p.281-288.

KOUFAKOU, A. et al. Fast parallel outlier detection for categorical datasets using MapReduce. In: NEURAL NETWORKS, 2008. **Proceedings**… Prague: IEEE, 2008. p.3298-3304.

LIU, X.; CHEN, F.; LU, C-T. On detecting spatial categorical outliers. **GeoInformatica**, New York, v. 18, p.501-536, 2013.

LU, C-T.; CHEN, D.; KOU, Y. Algorithms for spatial outlier detection. In: INTERNATIONAL CONFERENCE ON DATA MINING**,** 2003, Washington. P**roceedings**… Washington: IEEE, 2003a. p.597-600.

_____; _____; _____ Detecting spatial outliers with multiple attributes. In: IEEE INTERNATIONAL CONFERENCE ON TOOLS WITH ARTIFICIAL INTELLIGENCE, 2003, Sacramento. **Proceedings**… Sacramento: IEEE, 2003b. p.122-128.

MASSART, D. L. et al. Visual presentation of data by means of box plots. **LCGC** Europe, v.18, p.215–218, 2005.

MEDEIROS, C. A. **Extração de conhecimento em bases de dados espaciais:** algoritmo CHSMST+. 106f. 2014. Dissertação (Mestrado em Ciência da Computação) -Instituto de Biociências, Letras e Ciências Exatas, Universidade Estadual Paulista "Júlio de Mesquita Filho", São José do Rio Preto, 2014.

MENNIS, J.; GUO, D. Spatial data mining and geographic knowledge discovery: an introduction. **Computers, Environment and Urban Systems**, New York, v.33, n.6, p.403-408, nov. 2009.

MILLER, H. J.; HAN, J. **Geographic data mining and knowledge discovery**. 2Nd ed. New York: CRC Press : Taylor & Francis, 2009.

NG, R.; HAN, J. CLARANS: A method for clustering objects for spatial data mining. **IEE Transactions on Knowledge and Data Engineering**, Washington, v.14, n.5, p.1003-1016, 2009.

ONG, K. H. et al. Automatic white matter lesion segmentation using an adaptive outlier detection method. **Magnetic Resonance Imaging**, Chicago, v.30, n.6, p.807-823, 2012.

OPENLAYERS. **OpenLayers**: free maps for the web. Beaverton, c2014. Disponível em: <http://openlayers.org>. Acesso em: 29 jun. 2014.

ORACLE. **O que é o Java?,** c2014 Disponível em: <http://www.java.com/pt_BR/download/whatis_java.jsp>. Acesso em: 29 jun. 2014

PATTABIRAMAN, V. et al. A novel spatial clustering with obstacles and facilitators constraint based on edge deduction and k- mediods. In: INTERNATIONAL CONFERENCE ON COMPUTER TECHNOLOGY AND DEVELOPMENT, 2009, Kota Kinabalu. **Proceedings**… Kota Kinabalu, 2009. v.1, p.402-406.

PIATETSKY-SHAPIRO, G. **KDNUGGETS**: data mining, web mining, text mining, and knowledge discovery. c2014. Disponível em: <http://www.kdnuggets.com/>. Acesso em: 29 jun. 2014.

POSTGRESQL. **About**. c1996-2014. Disponível em: <http://www.postgresql.org/about>. Acesso em: 29 jun. 2014.

QIANG, X.; WEI, Y.; HANFEI, Z. Application of visualization technology in spatial data mining. In: INTERNATIONAL CONFERENCE ON COMPUTING, CONTROL AND INDUSTRIAL ENGINEERING, 2010, Hong Kong. **Proceedings**… Hong Kong: IEEE, 2010. p.153-157.

SANTOS, M. Y.; AMARAL, L. A. Mining geo-referenced data with qualitative spatial reasoning strategies. **Computers & Graphics**, New York, v.28, n.3, p.371-379, 2004.

SHEIKHOLESLAMI, G.; CHATTERJEE, S.; ZHANG, A. Wavecluster: A multi-resolution clustering approach for very large spatial databases. In: INTERNATIONAL CONFERENCE ON VERY LARGE DATA BASES, 1998, San Francisco. **Proceedings**… San Francisco: ACM, 1998. p.428-439.

SHEKHAR, S.; LU, C.; ZHANG, P. A unified approach to detecting spatial outliers. **GeoInformatica**, Boston, v.7, n.2, p.139-166, 2003.

_____; LU, C-T.; ZHANG, P. Detecting graph-based spatial outliers: algorithms and applications (a summary of results). In: ACM SIGKDD INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING, 2001, Washington. **Proceedings**… Washington: ACM, 2001. p.371-376.

_____; XIONG, H. (Ed.). **Encyclopedia of GIS**. Springer, 2008.

_____.; ZHANG, P.; HUANG, Y. Trends in spatial data mining. In: KARGUPTA, H. et al. (Ed.). **Next generation challenges and future**. Boca Raton: CRC Press, 2003.

SHENGWU, H. Method development about spatial data mining and its problem analysis. In: INTERNATIONAL CONFERENCE ON ELECTRONICS AND OPTOELECTRONICS, 2011, Shenyang. **Proceedings**… Shenyang, 2011. v.2, p. 144-147.

SONGMA, S. et al. Classification via k-means clustering and distance-based outlier detection. In: INTERNATIONAL CONFERENCE ON ICT AND KNOWLEDGE ENGINEERING,2012, Bangkok. **Proceedings**… Bangkok, 2012. p.125-128.

SURI, V. et al. Spatial outlier detection using improved z-score test. **International Journal of Engineering Science**, Texas, v. 5, p.1962-1966, 2013.

TORRES, J. M. et al. Detection of outliers in gas emissions from urban areas using functional data analysis. **Journal of Hazardous Materials**, Amsterdam, v.186, n.1, p.144-149, feb. 2011.

YEUNG, A. K. W.; HALL, G. B. **Spatial database systems**: design, implementation and project management. Amsterdam: Springer, 2007.

XI, J. Outlier detection algorithms in data mining. In: INTERNATIONAL SYMPOSIUM ON INTELLIGENT INFORMATION TECHNOLOGY APPLICATION, 2008, Shangai. **Proceedings**… Shangai: IEEE, 2008. v.1, n.4, p.94-97.

WONG, W-K. et al. Bayesian network anomaly pattern detection for disease outbreaks. In: INTERNATIONAL CONFERENCE ON MACHINE LEARNING, 2003, Atlanta. **Proceedings**… Atlanta, 2003. p.808-815.