UNIVERSIDADE FEDERAL DE SANTA CATARINA – UFSC

JANAÍNA OLIETE DE SIQUEIRA

DESCOBERTA DE CONHECIMENTO ESPAÇO-TEMPORAL ATRÁVES DA ANÁLISE DAS TRAJETÓRIAS DOS USUÁRIOS DA REDE SOCIAL *TWITTER*

Florianópolis - SC.

2012/2

UNIVERSIDADE FEDERAL DE SANTA CATARINA – UFSC

CENTRO TECNOLÓGICO – CTC

CURSO DE SISTEMAS DE INFORMAÇÃO

JANAÍNA OLIETE DE SIQUEIRA

DESCOBERTA DE CONHECIMENTO ESPAÇO-TEMPORAL ATRÁVES DA ANÁLISE DE TRAJETÓRIA DOS USUÁRIOS DA REDE SOCIAL *TWITTER*

Trabalho de Conclusão de Curso apresentado como parte dos requisitos para obtenção do grau de Bacharel em Sistemas de Informação da Universidade Federal de Santa Catarina.

Orientador(a): Luis Otávio Campos Alvares

Universidade Federal de Santa Catarina, 2012-2.

JANAÍNA OLIETE DE SIQUEIRA

DESCOBERTA DE CONHECIMENTO ESPAÇO-TEMPORAL ATRÁVES DA ANÁLISE DAS TRAJETÓRIAS DOS USUÁRIOS DA REDE SOCIAL *TWITTER*

Trabalho de Conclusão de Curso apresentado como parte dos requisitos para obtenção do grau de Bacharel em Sistemas de Informação.

Banca examinadora:

Orientador: _____

Prof. Luís Otávio Campos Álvares, Dr.

Membro: _____

Profª. Luciana de Oliveira Rech, Dra.

Membro: _____

Profª. Patricia Della Méa Plentz, Dra.

Florianópolis – SC, junho de 2012

AGRADECIMENTOS

Primeiramente agradeço a Deus, por ter se feito presente ao longo de toda a minha vida, por seu infinito amor e zelo para comigo e por permitir que eu tivesse a oportunidade de viver esta experiência por completo.

Agradeço a minha mãe Oliete Izaura de Siqueira, pelo carinho e amor que tem me dedicado durante toda uma vida e principalmente por ter me ensinado o verdadeiro valor do caráter e do conhecimento. Aos meus irmãos Luiz Claúdio de Siqueira, Maria Aparecida de Siqueira e Maria Eliane de Siqueira, por estarem sempre ao meu lado nos momentos de dificuldades. A toda minha família por cada momento de preocupação e cada palavra de apoio que me deram forças nos momentos desânimo.

Ao meu namorado Augusto Bonatelli, pela compreensão, amor e paciência que soube ter não somente durante este período de conclusão de curso, mas também durante todo o tempo que temos compartilhado nossas vidas.

Ao amigos que participaram direta ou indiretamente deste momento de minha vida. Aos amigos e colegas com quem dividi estes quatro anos e meio de curso, especialmente a Lara Cristina Wilpert, Marcelo Scheidt e Renata de Jesus Silva, com os quais pude compartilhar de momentos de alegrias e tristezas, provas e trabalhos e finalmente deste momento de vitória pela conclusão do curso.

Agradeço ao professor e orientador Luis Otavio Campos Alvares, por ter aceitado o desafio de ser responsável por este trabalho e me dar a oportunidade de realiza-lo. Por toda atenção que me dedicou ao longo deste um ano e meio de orientação. As professoras Patrícia Della Mea Plentz e Luciana de Oliveira Rech, por terem aceitado compor a banca deste trabalho.

RESUMO

A crescente popularização da internet fez com que houvesse um enorme aumento no número de pessoas que utilizam as redes sociais cotidianamente e também contribuiu para o aumento da quantidade de redes sociais disponíveis. Com isso as pessoas passaram a dedicar muitas horas de seu dia para acompanhar e atualizar as mesmas, disponibilizando dados de onde estão e o que estão fazendo, além de diversas outras informações.

Devido a esse fato estas redes sociais têm gerado um grande volume de dados, rico em informações sobre seus usuários, podendo conter, seus perfis, atualizações e atualmente muitas delas disponibilizam dados georreferenciais de seus usuários. Essas informações têm interessado cada vez mais empresas e pessoas, dos mais diversos setores da economia, a fim de que possam fazer uso dessas informações para beneficiarem seus negócios e manterem um contato mais pessoal com o cliente.

O Twitter é uma rede social de características instantâneas, baseada na pergunta "O que você está fazendo agora?", que ficou bastante popular entre as celebridades e empresas e por isso chamou atenção na conquista de muitos outros usuários. Este trabalho foca no estudo das trajetórias dos usuários desta rede social. Dois algoritmos são descritos e avaliados com dados reais. O primeiro mostra sequências das regiões de onde o usuário Tuitou e o segundo identifica o local de moradia e ocupação do usuário.

Palavras –chave: *Twitter*, dados espaço-temporais, mineração de dados, trajetórias.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS

CB-SMOT: Clustering-Based Stops and Moves of Trajectories
DB-SMOT: Direction-Based Stops and Moves of Trajectories
DCBD: Descoberta de Conhecimento em Banco de Dados
GIS: Geographic Information System
GPS: Global Positioning System
IB-SMOT: Intersection – Based Stops and Moves of Trajectories
KDD: Knowledge Discovery in Databases
PMO-RS: Predição de Moradia e Ocupação em Redes Sociais
RIP-RS: Regiões de Interesse de Postagem em Redes Sociais
SGBD: Sistema Gerenciador de Banco de Dados
SIG: Sistema de Informações Geográficas
SQL: Structured Query Language
STPM: Semantic Trajectory Preprocessing Module
Tweet: mensagem postada por um usuário na rede social Twitter
WEKA: Waikato Environment for Knowledge Analysis

SUMÁRIO

1 INTRODUÇÃO

A busca por automação e sistematização dos processos empresariais, assim como, a facilidade de informatização que vem acontecendo na última década, gerou um grande aumento do volume de dados armazenados em bancos de dados sejam eles empresariais, governamentais, pessoais ou comerciais. Com esse acúmulo de informações percebeu-se a oportunidade de utilizá-las de forma estratégica, de maneira a auxiliar os gestores nas tomadas de decisões em suas organizações.

Neste âmbito, a área de mineração de dados ganhou ênfase nos esforços de pesquisa e desenvolvimento no ramo da computação. "A mineração de dados consiste em encontrar tendências ou padrões interessantes em grandes conjuntos de dados para orientar decisões sobre atividades futuras" (RAMAKRISHNAN; GEHERKE, 2008, p.737). Portanto, nos últimos anos vem se desenvolvendo um grande número de algoritmos e técnicas para extração de dados em cima de grandes bases de informação. A mineração de dados vem evoluindo e tomando grandes proporções e sua crescente aplicação deu ênfase a uma área que atualmente vem sendo muito discutida e proporcionando grandes ganhos nas mais diversas áreas da economia, a qual recebeu o nome de Descoberta de Conhecimento de Banco de Dados (DCBD), que vem do inglês *Knowledge Discovery in Databases* (KDD). A descoberta de conhecimento é mais ampla que a mineração de dados. "O processo é composto por seis fases: seleção de dados, limpeza, enriquecimento, transformação ou codificação, mineração de dados e construção de relatórios e apresentação das informações descobertas" (ELMASRI; NAVATHE, 2005, p. 624). Portanto, a mineração dos dados é essencial para o processo de descoberta de conhecimento, mas além dessa, outras etapas são necessárias para que o processo ocorra corretamente. Conforme citado por Ramakrishnan e Gehrke (2008) essas etapas não ocorrem em sentido único e o resultado de qualquer uma das etapas pode levar a uma etapa anterior para que o processo possa ser refeito a partir do conhecimento obtido.

Atualmente os grandes esforços de pesquisa e desenvolvimento em KDD concentram-se em bases de dados tradicionais. No entanto com a crescente utilização de aparelhos móveis que utilizam o serviço de monitoramento por GPS ou triangulação, gerando grandes bases com dados temporais e de localização, tornou-se perceptível a importância da utilização dessas informações geográficas e espaciais nas aplicações atuais e o grande beneficio que sua exploração pode nos trazer no futuro, como melhor planejamento para o trânsito nas cidades,

estratégias de instalação de empresas e estudo de mercado através da análise de deslocamento de usuários, entre muitas outras. Dentro desta realidade enxerga-se cada vez mais a necessidade de intensificar o estudo de KDD dentro de bases geográficas, considerando ainda as informações temporais, que se tornam cada vez mais comuns nessas bases.

A descoberta de conhecimento considerando dados geográficos (ou espaciais) se diferencia da descoberta em dados tradicionais principalmente por considerar os relacionamentos espaciais entre os dados (topológicos, de distância e ordem) (GÜTING, 1994 apud ALVARES, 2011, p. 2). Esta área, portanto, é caracterizada como uma linha de pesquisa que caminha paralelamente ao KDD tradicional, já que os relacionamentos espaciais não ficam armazenados nos bancos de dados.

Os dados gerados por dispositivos móveis são normalmente representados no formato de trajetórias, as quais possuem registros com o formato (tid, x, y, t), onde *tid* é o identificador da trajetória, *x* e *y* são as coordenadas da localização geográfica do objeto no espaço e no tempo, representado por *t*. Os aparelhos de GPS geram dados de localização usualmente a cada um segundo. A Figura 1 apresenta um exemplo de armazenamento dos dados de trajetória em um banco de dados.

Gid	Tid	Time	The_Geom
126	55	"2009-02-01 15:30:05"	"0101000000D675160591ED2441128A7F1A0F7B5C41"
127	55	"2009-02-01 15:30:06"	"010100000068F4065D745B9C5D8965C78A0F7B5C41"
...	...	...	...
209	55	"2009-02-01 16:55:38"	"0101000000645289D65F42C5BA42986F5210F78C41"
210	55	"2009-02-01 16:55:09"	"0101000000256378 9F427C4298B64875A0078D6C41"

Figura 1 - Tabela de trajetórias em banco de dados espacial

A partir da Figura 1 é possível identificar cada atributo essencial em uma tabela de trajetórias em um banco de dados, sendo eles, o *Gid* que é o identificador único de cada registro dentro da base, o *Tid* que é o identificador único de cada trajetória, ou seja, todos os registros pertencentes a uma mesma trajetória recebem o mesmo identificador *Tid*, o campo *Time* que apresenta a data e a hora em que o registro foi obtido e o campo *The_Geom* que apresenta a localização do ponto no espaço.

A complexidade destes dados é o que torna a extração de conhecimento interessante destas bases, uma tarefa difícil. Apesar de ser uma área nova no campo de estudo já existem

muitas pesquisas relacionadas à construção de teorias e algoritmos para extrair informações com conteúdo útil. Conforme descrito em Loy (2011), os estudos estão focados na busca por padrões comportamentais dentro das trajetórias e estão divididos em dois grandes grupos, o de ponto de vista geométrico (LAUBE, 2005),(GUDMUNDSSON, 2006), (CAO, 2007), (GIANNOTTI, 2007), (ANDERSSON, 2008), que trabalha com as trajetórias em seu estado bruto e o de ponto de vista semântico(ALVARES, 2007), (BOGORNY, KUIJPERS, ALVARES, 2008) que adiciona informações semânticas a trajetória em suas análises.

Com a popularização da internet uma das aplicações web que mais ganhou popularidade na última década foram as redes sociais, atraindo pessoas de todo o mundo para suas redes e gerando bases com uma enorme quantidade de dados, que recebem diariamente milhões de novas atualizações.

O Twitter é uma rede social de mensagens instantâneas, lançada em 2006 que ganhou muita popularidade em um curto espaço de tempo, impulsionado pelo fato de muitas celebridades, empresas e marcas terem aderido ao uso da rede e a mesma permitir relacionamentos unilaterais, ou seja, um fã pode seguir um ídolo sem que o mesmo o aceite em sua rede, como é o caso de outras ferramentas de mesmo fim, como Orkut e Facebook. Assim como outras redes sociais, o Twitter permite que o usuário habilite a disponibilização de sua localização geográfica no momento de suas postagens, gerando informações espaço-temporais e consequentemente formando a trajetória de seus usuários.

O padrão de trajetórias gerado pelas redes sociais difere bastante daquelas mencionadas anteriormente, que podem ser geradas por GPS em um automóvel por exemplo, afinal no caso das redes sociais não existe um intervalo de tempo definido entre os pontos pertencentes a um trajeto específico, neste caso o que define cada ponto é o momento da postagem da mensagem na rede, o que varia muito de um usuário para outro, ou mesmo de um dia para o outro.

Como a disponibilização dos dados espaço-temporais dos usuários nas redes sociais é algo recente, os estudos de técnicas e a construção de algoritmos destinados especificamente a manipular e extrair conhecimento espaço-temporal das redes sociais ainda se encontra em estado emergente. Este trabalho pretende realizar a análise dos algoritmos de mineração de dados voltados para trajetórias comuns, a fim de verificar se os mesmos são aplicáveis as trajetórias, atípicas, geradas pelas redes sociais, identificando a necessidade de alterações ou eventualmente a criação de novos algoritmos para trabalhar com este tipo de trajetória. A

correta exploração deste tipo de dados proporcionará a extração de conhecimento real, novo e importante a respeito das redes e seus usuários, que poderão futuramente auxiliar os mais diversos setores econômicos, de pesquisa e governamentais, dentre muitas outras áreas.

1.1 OBJETIVOS

O objetivo geral deste trabalho é a exploração dos dados e o estudo de algoritmos que possibilitem a análise das trajetórias geradas pela rede social Twitter, na microrregião de Florianópolis, proporcionando a extração de conhecimento espaço-temporal útil, sobre os usuários desta rede.

Os principais objetivos específicos são:

- Fazer a utilização dos algoritmos já existentes e se necessário modificá-los ou definir novos métodos e algoritmos para a mineração dos dados;
- Identificar tipos de usuários a partir da análise de suas trajetórias, classificando-os com base na divergência ou não de seu comportamento durante a semana e no fim de semana;
- Identificar padrões de deslocamentos entre os usuários da rede;
- Identificação de pontos de interesse dos usuários, ou seja, pontos de onde muitos usuários costumam postar suas mensagens.
- Definir um algoritmo para predição de local de moradia e ocupação dos usuários, através da análise de suas trajetórias.

1.2 DELIMITAÇÃO DO TEMA

O presente trabalho tem como foco trabalhar apenas com os dados da rede social Twitter, não sendo analisada nenhuma informação de outras redes sociais existentes.

Apesar de ter acesso a dados de todo o Brasil, o escopo do projeto abrange somente a descoberta de conhecimento espaço-temporal na microrregião de Florianópolis, que segundo

o IBGE é formada pela capital do estado e mais os municípios de Antônio Carlos, Biguaçu, Governador Celso Ramos, Palhoça, Paulo Lopes, Santo Amaro da Imperatriz, São José e São Pedro de Alcântara. Esta delimitação foi feita para facilitar a análise e diminuir a quantidade de dados, pois a base de dados inteira compreendia uma quantidade em torno de dezoito milhões de registros.

1.3 ESTRUTURA DO TRABALHO

As seções restantes deste trabalhão estão organizadas da seguinte forma:

Capítulo 2 – Levantamento Bibliográfico – Neste capítulo serão levantados os principais conceitos referentes ao tema abordado, apresentando os principais trabalhos correlatos e a descrição de conceitos como descoberta de conhecimento em banco de dados, mineração em bancos de dados espaço-temporais e em redes sociais, apresentação da rede social Twitter.

Capítulo 3 – Apresentação dos dados e avaliação dos algoritmos – O objetivo é descrever os dados e realizar um levantamento sobre a aplicabilidade dos algoritmos IB-SMOT, CB-SMOT e DB-SMOT aos dados de trajetórias de redes sociais, além de apresentar modificações aos mesmos.

Capítulo 4 – Algoritmos propostos – Serão apresentados os algoritmos desenvolvidos durante a pesquisa realizada, além de eventuais extensões de métodos já existentes.

Capítulo 5– Análise e apresentação dos resultados – Neste capítulo serão apresentados os resultados obtidos após o processo de descoberta de conhecimento a ser aplicado aos dados, avaliando a efetividade de cada um deles.

Capítulo 6 – Conclusões e trabalhos futuros – Apresentará as considerações finais sobre o assunto estudado, bem como fará indicação de trabalhos futuros sobre o mesmo tema.

2 LEVANTAMENTO BIBLIOGRÁFICO

2.1 DESCOBERTA DE CONHECIMENTO E MINERAÇÃO DE DADOS

Os conceitos de descoberta de conhecimento e mineração de dados são frequentemente confundidos, ou compreendidos como sinônimos. No entanto o segundo refere-se a apenas uma etapa, embora um das mais importantes, dentro de um processo maior que é a descoberta de conhecimentos em banco de dados. De acordo com Fayyad (1996) a Descoberta de Conhecimento em Banco de Dados (DCBD) é definida como sendo um processo não trivial para identificação de padrões válidos, potencialmente útil e compreensíveis a partir dos dados.

A utilização de DCBC tem crescido em resposta ao grande aumento de dados que estão sendo armazenados em bancos de dados. Através da aplicação de seus conceitos é possível a extração de conhecimento de base de dados cientificas e comerciais, facilitando processos de tomada de decisão estratégica, diagnósticos médicos, respostas a pesquisas acadêmicas, entre outras. A Figura 2 apresenta as principais etapas do processo de descoberta de conhecimento.

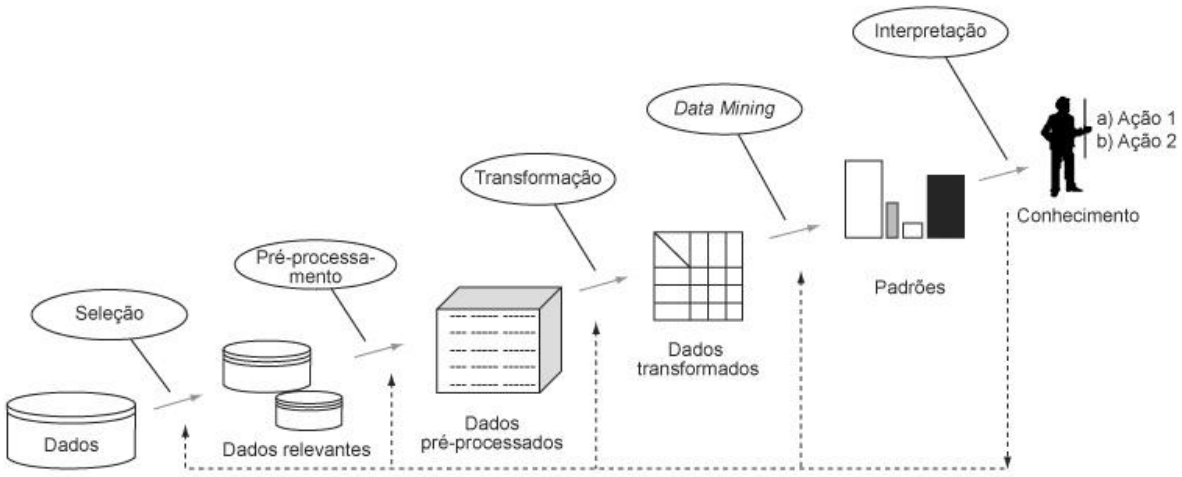

Figura 2 - Etapas do processo de KDD (Fayyad, 1996)

Como mostrado na Figura 2 a descoberta de conhecimento inicia na etapa de seleção dos dados, na qual a partir dos dados brutos existentes na base de dados são selecionados os dados necessários para o aplicação a que estão sendo submetidos. Uma base completa pode

não ser necessária para o objetivo da descoberta de conhecimento e nesta fase somente os dados de efetivo interesse são escolhidos.

A segunda etapa é chamada de pré-processamento e nesta fase são feitas operações de limpeza dos dados, correções de inconsistência, entre outras diversas operações necessárias para eliminar toda a "sujeira" dos dados e garantir a confiabilidade para o processo de mineração.

A terceira etapa trata da transformação necessária aos dados para que os algoritmos de mineração de dados possam executa-los, podem ser realizadas tarefas de alisamento, generalização, normalização, transformações de atributos numéricos para categóricos e ainda o inverso. Todas essas transformações devem ser realizadas com foco no objetivo da análise, identificando a necessidade de aplica-las para sucesso da mineração.

A quarta etapa é a mineração de dados propriamente dita. "*Data Mining* se refere à mineração ou a descoberta de novas informações em função de padrões ou regras em grandes quantidades de dados"(ELMASRI, NAVATHE,2005, p.620). Nesta etapa são selecionados os algoritmos para extração de padrões e aplicados sobre os dados.

A quinta etapa da descoberta de conhecimento é responsável pela interpretação dos resultados, avaliando padrões, regras e outros resultados obtidos pelo processo de mineração, com o objetivo de utilizar este conhecimento de maneira útil para o propósito para o qual foi aplicado.

O KDD é uma ferramenta de extrema importância para trabalhar com a exploração de domínios que são de difícil compreensão para as capacidades humanas, principalmente com bases de dados com grande número de informações. A descoberta de conhecimento suprime a utilização da estatística quando se trata de um busca por grandes volumes de informações, mas em se tratando de poder de confirmação a estatística ainda vence a descoberta de conhecimento, pois trabalha com padrões conhecidos (MILLER; HAN, 2009).

2.2 MINERAÇÃO DE DADOS ESPAÇO-TEMPORAL

Conforme apresentado em Fajardo (2008), a tarefa de minerar dados espaço-temporais se difere da mineração de dados em bancos tradicionais, como o que registra as compras dos clientes em uma loja, pois a primeira tem o objetivo de extrair relações sobre os dados envolvendo o tempo e o espaço. Um dos maiores problemas no trato com as informações

espaço-temporais é a complexidade dos dados envolvidos, o que dificulta o entendimento do usuário, tanto com relação aos dados, como também aos resultados.

O aumento na quantidade de dados espaços-temporais, gerados por dispositivos móveis, armazenados e apresentados no formato de trajetórias, geram uma grande quantidade de informação implícita que necessita ser explorada para gerar conhecimento. Uma das vertentes de estudo sobre o padrão de comportamento em trajetórias, apresentadas na introdução deste trabalho, é a geométrica que realiza a análise das trajetórias em seu estado bruto, pontos simples, e como resultado obtém padrões geométricos para suas análises. Os estudos de Patrick Laube destacaram-se neste assunto e posteriormente outros pesquisadores evoluíram algumas de suas teorias. Na seção 2.2.1 o assunto referente a trajetórias brutas e seus padrões será apresentado com maior nível de detalhes.

Conforme descrito em Alvares (2007), para muitas aplicações, a extração de padrões significativos não pode ser realizada a partir de trajetórias brutas, sem associação de alguma informação geográfica adicional. Neste sentido caminha a vertente semântica de análise de padrões comportamentais em trajetórias, adicionar informações importantes às trajetórias para facilitar sua análise e possibilitar maiores ganhos. Ainda em Alvares (2007), são apresentados os conceitos de *stops*, que representam pontos importantes dentro dos trajetos e *moves*. Na seção 2.2.2 deste trabalho será descrita com maiores detalhes a adição de semântica às trajetórias.

2.2.1 Trajetórias brutas

A todo o momento trajetórias estão sendo geradas. O caminho percorrido por uma pessoa que dirige seu automóvel de casa até o serviço representa uma trajetória e assim como as pessoas podem se movimentar de várias maneiras diferentes, diversos são os tipos de trajetórias que podem ser geradas a partir destes movimentos. Spaccapietra et al. (2008) define trajetória como sendo o segmento espaço-temporal do caminho percorrido por um objeto.

As trajetórias brutas são os dados puros, que são fornecidos diretamente pelos aparelhos eletrônicos, sem nenhum acréscimo de informação de contexto acrescido aos dados. Em suma, as trajetórias brutas são apenas sequências de pontos espaciais ordenados pelo

tempo de ocorrência de cada ponto (CHIECHELSKI, BOGORNY, 2008). A figura 3 ilustra a representação de uma trajetória em seu estado bruto.

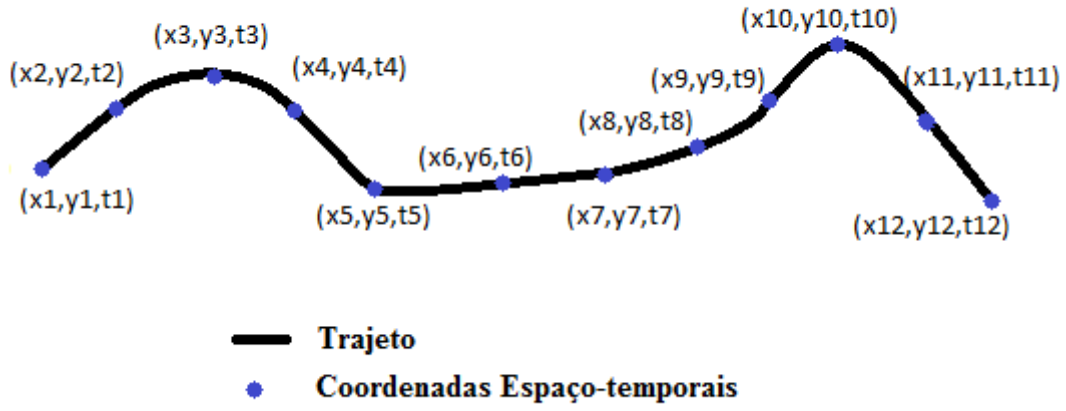

Figura 3 - Tajetória bruta

Recentemente, muitos algoritmos que objetivam realizar a tarefa de mineração de dados de trajetórias em seu estado bruto vêm sendo desenvolvidos. Seus focos, em geral, são a descoberta de padrões através da similaridade de trajetórias ou regiões densas (BOGORNY, KUIJPERS, ALVARES, 2008). Em 2002, Patrick Laube apresenta o primeiro estudo de maior notoriedade nesta área e que viria a servir de base para diversos outros trabalhos posteriormente desenvolvidos. Neste trabalho foi definida uma coleção de padrões espaço-temporais que se baseiam na direção do movimento e localização do objeto móvel (LAUBE, IMFELD, 2002). Para definir esta coleção de padrões o autor apresentou o conceito de Análise REMO (*RElative MOtion*), que define parâmetros básicos para a identificação de padrões comportamentais.

Continuando os estudos de 2002, outros grupos de padrões comportamentais foram descritos em Laube(2004), no qual é proposta uma abordagem de mineração espaço-temporal para detectar padrões genéricos de agregação, baseados na relação de grupos e objetos. Os padrões descoberto nas análises foram denominados *flock, leadership, convergence e encounter*. O padrão *flock* acontece no momento em que um grupo de trajetórias se move na mesma direção, dentro de um raio dado, por um determinado período de tempo. No mundo animal um bom exemplo deste comportamento é o movimento de um rebanho de ovelhas. O padrão *leadership*, consiste no deslocamento que apresenta o comportamento de um criador

de tendências, ou seja, um padrão de liderança onde os seguidores caminham atrás de seu líder respeitando uma restrição de tolerância referente a distancia entre eles. O padrão de deslocamento *convergence* realiza a identificação de trajetórias que convergem para um ponto, dentro de um intervalo de tempo. Por fim, o padrão *encounter* apresenta um deslocamento semelhante ao *convergence*, mas neste caso é dado um raio, dentro do qual as trajetórias devem permanecer juntas por um período de tempo. A Figura 4 mostra um exemplo de cada um dos padrões descritos.

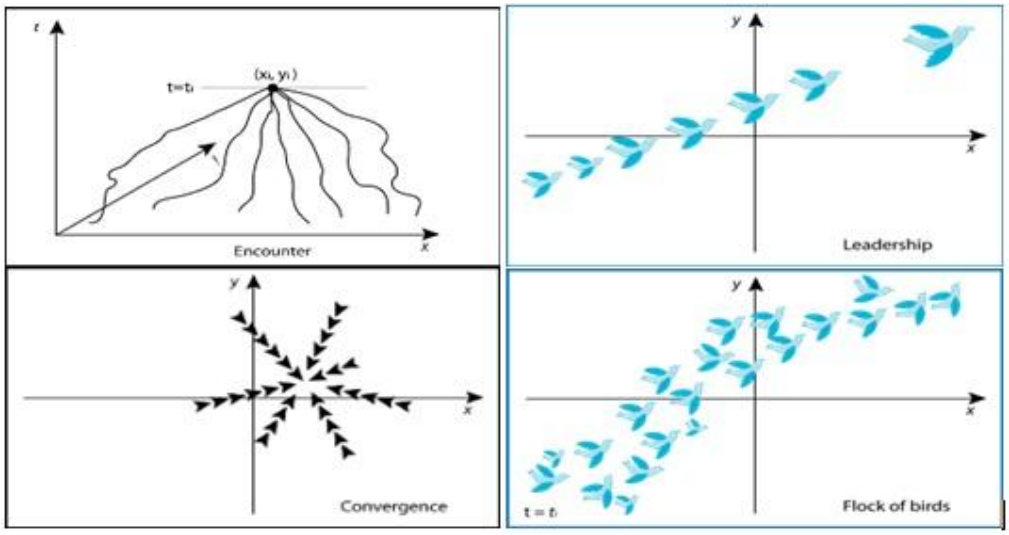

Figura 4 - Padrões geométricos, adaptado de http://movementpatterns.pbworks.com.

No ano seguinte, Laube(2005), apresenta mais uma contribuição para a descoberta de padrões geométricos em bases de dados geográficas, onde aplica o conceito de nível de granularidade temporal, possibilitando um nível diferente de detalhes dos dados e a descoberta de novos parâmetros de movimento.

No mesmo ano, Hwang et al.(2005) descreve um novo algoritmo, no qual considera somente o conjunto de trajetórias próximas umas das outras, sem levar em consideração a direção para a qual se destinam. Esse algoritmo produz os padrões frequentes da mesma forma que o algoritmo Apriori(AGRAWAL, SRIKANT, 1994) e utiliza como parâmetro uma distancia mínima que deve ser obedecida entre as trajetórias pertencentes ao conjunto e um tempo mínimo que as mesmas devem se encontrar próximas.

Dando continuidade aos estudos desenvolvidos inicialmente por Laube, em 2006 e 2007, outros autores apresentam extensões de suas teorias. Em Gudmundsson(2006), o

conceito de *flock* é estendido para trabalhar com trajetórias de longa duração e em Gudmundsson(2007), são apresentadas maneiras mais eficientes de calcular os padrões geométricos definidos por Laube em 2004.

O padrão denominado *trajectory pattern* representa um conjunto de trajetórias que passam pela mesma sequência de locais em tempos similares de deslocamento (GIANOTTI, 2007). No mesmo estudo é apresentado o algoritmo *T-Pattern Minner*, que busca regiões espaciais que normalmente são visitadas pelos objetos e o tempo de deslocamento entre duas regiões deste tipo, neste caso os padrões traduzem além da sequencia de pontos visitados o tempo de transição entre eles.

Existem ainda outros estudos referentes a estas buscas de padrões geométricos, como Cao(2006) e Elnekave(2007), mas estes padrões são mais específicos para determinadas situações e estão fora do escopo deste trabalho.

2.2.2 Trajetórias semânticas

Na teoria de compiladores o termo semântica está associado a outras expressões equivalentes a ele, como significado e sentido lógico, o que é muito semelhante quando associamos o mesmo termo ao estudo de análises de trajetórias, pois adicionar semântica a uma trajetória nada mais é do que dar significado aos pontos pertencentes a mesma. Em outras palavras, uma trajetória semântica é composta pelos dados brutos, associados a dados semânticos, que podem ser informações adicionais da própria trajetória ou da região onde ela está inserida, através do cruzamento das informações.

De posse das trajetórias adicionadas de semântica, é possível responder perguntas como, quais locais um turista visitou durante sua viajem ou em quais hotéis o mesmo dormiu. Se analisado um conjunto de trajetórias e não uma isoladamente pode-se tirar conclusões interessantes como quais pontos turísticos são visitados com maior frequência pelo viajante ou quais deles são visitados em sequencia. Enfim, através da adição de semântica é possível por meio da mineração de dados, alcançar vários resultados interessantes referentes ao contexto das trajetórias em análise. A Figura 5 apresenta a imagem de uma mesma trajetória no estado bruto e em seguida adicionada de semântica.

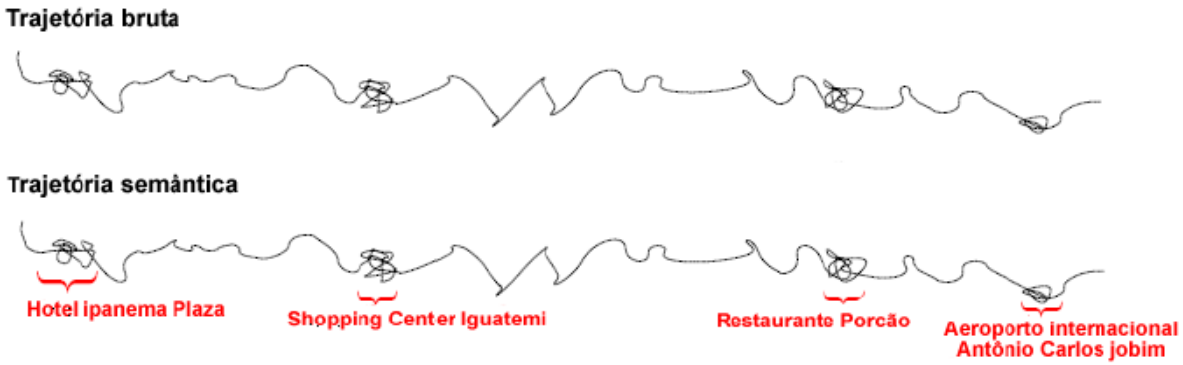

Figura 5 - Trajetória semântica (CHIECHELSKI, BOGORNY, 2008)

Uma das formas mais utilizadas para realizar a tarefa de adição de semântica as trajetórias é o modelo de *stops* e *moves*, conceitos definidos em Spaccapietra et al.(2008) que facilitam as funções de pré-processamento e identificação de padrões significativos. O modelo de enriquecimento semântico de trajetórias, construído com base nestes conceitos, serve para fazer a definição de pontos relevantes dentro das trajetórias e realizar o cruzamento das informações.

De acordo com a definição em Spaccapietra et al.(2008), *stops* são os pontos importantes dentro de uma trajetória, no qual o objeto permanece por um determinado intervalo de tempo. Os *stops* são definidos do ponto de vista da aplicação e dependendo do contexto ao qual ela está inserida a granularidade espacial dos *stops* pode variar. Para uma aplicação turística, por exemplo, podem ser considerados como pontos relevantes, hotéis onde o turista dormiu, pontos turísticos pelos quais ele pode ter passado como museus, praias ou shoppings, enquanto para outras aplicações pode ser importante demarcar somente os bairros ou municípios por onde o objeto transitou, assim como hospitais ou semáforos, enfim os pontos de interesse em uma trajetória dependem dos objetivo pelo qual ela esta sendo analisada.

Ainda segundo Spaccapietra et al.(2008) a duração de um *stop* tem que ser um valor maior que zero e neste tempo o objeto não deve se deslocar para fora de seus limites, traduzindo um comportamento em que o objeto esteve em um determinado lugar ou região, por determinado espaço de tempo. No mesmo estudo é definido o conceito de *moves*, como sendo partes da trajetória entre duas extremidades, que podem ser: o ponto inicial de uma trajetória e seu primeiro *stop*, dois *stops* consecutivos, o último *stop* encontrado e o ponto

final do trajeto ou a própria trajetória, se a mesma não possuir nenhum *stop*. Um *move* é composto somente por uma linha espaço-temporal com seus pontos de inicio e fim e o tempo de duração deste *move* não pode ser nulo.

A partir destes conceitos foram desenvolvidos alguns métodos para a descoberta de *stops* e agregação de semântica as trajetórias. Existem três principais algoritmos desenvolvidos utilizando conceitos como tempo (ALVARES et al., 2007), velocidade (PALMA et al., 2008) e direção (MANSO et al., 2010), que realizam a adição de semântica às trajetórias no estado bruto, em uma etapa de pré-processamento dos dados. O resultado destes algoritmos é a criação de uma tabela, onde posteriormente será aplicada a mineração de dados ou a execução de consultas, gerando um ganho de qualidade e performance nas tarefas executadas sobre esses resultados. Na seção seguinte serão apresentados maiores detalhes de cada um dos três algoritmos em particular.

2.2.3 Métodos para adição de semântica as trajetórias.

O primeiro modelo desenvolvido para adicionar semântica as trajetórias foi introduzido em Alvares et al. (2007) e recebeu o nome de IB-SMoT (*Intersection – Based Stops and Moves of Trajectories*). O método se baseia na intersecção de cada ponto de uma trajetória com a área geométrica de algum candidato a *stop,* que são todas as regiões definidas como de interesse para a aplicação.

O primeiro passo realizado pelo algoritmo é selecionar todos os pontos de uma determinada trajetória e em posse disto, realizar a análise de todos os pontos, verificando se eles fazem intersecção com algum candidato a *stop*. Se um ponto intercepta um candidato a *stop* o algoritmo realiza um *looping* até encontrar o último ponto em sequência que esteja dentro desta intersecção, para calcular o tempo que o objeto ficou naquela região. Após, é verificado se o tempo que o objeto esteve na região de interesse é maior ou igual ao mínimo pré–definido. Se for, este conjunto de pontos é definido como um *stop*. Para finalizar, o algoritmo cria duas tabelas após percorrer toda a trajetória, uma de *stops* e outra de *moves,* e insere nas mesmas todos os registros encontrados.

Este método pode ser utilizado com muita qualidade para encontrar pontos turísticos que foram visitados por alguém em sua passagem por determinada cidade, e outras aplicações semelhantes.

O segundo método foi apresentado em Palma et al. (2008) e foi denominado CB-SMoT(*Clustering-Based Stops and Moves of Trajetories*). De uma maneira geral, funciona para encontrar trechos dentro de uma trajetória onde o objeto esteve em uma velocidade abaixo de um valor mínimo, por um determinado período de tempo. Portanto, além de considerar o tempo que um objeto esteve parado em um candidato a *stop*, como é o caso do IB-SMOT, ainda considera o valor da velocidade em suas análises.

O algoritmo CB-SMOT funciona dividido em três etapas, na primeira é realizada a seleção dos pontos que cumprem a regra de estarem em uma velocidade abaixo da mínima definida, que é calculada pela multiplicação do valor percentual passado como parâmetro pelo usuário e a velocidade média da trajetória. A velocidade deve se manter mínima por um intervalo de tempo pré-determinado pelo usuário. Após esta etapa, é realizado o cruzamento dos *clusters* de baixa velocidade com os candidatos a *stop*, que se tornam *stops* respeitando a mesma regra do algoritmo anterior, ou seja, permanecendo um tempo mínimo naquele local.

Uma particularidade deste algoritmo é que os *clusters* que não cruzam com nenhum candidato a *stop* são guardados como *stops* desconhecidos e na terceira e última etapa do algoritmo estes clusters são comparados entre si, se dois ou mais acontecem no mesmo lugar, eles recebem o mesmo nome e um novo *stop* é criado.

A principal aplicação deste algoritmo é a descoberta de regiões ao longo da trajetória em que um automóvel tenha passado por congestionamentos.

O terceiro modelo foi desenvolvido em Manso et al. (2010) e apresenta uma metodologia que tem como principal característica a avaliação da trajetória considerando a variação da direção que ocorre ao longo da mesma. Este método recebeu o nome de DB-SMoT (*Direction-Based Stops and Moves*).

Inicialmente o algoritmo faz a seleção de todos os pontos da trajetória e começa a compará-los de dois em dois, verificando a variação da direção entre o primeiro e o segundo. Se esta variação for maior que uma variação mínima pré-estabelecida, os próximos dois entram na comparação e assim sucessivamente até que o primeiro ponto que não atende a variação mínima seja encontrado. Neste momento é feita a conferencia do tempo mínimo que

o objeto permaneceu variando, e assim como nos outros algoritmos, se obedecerem a um mínimo definido, estes pontos são inseridos em uma tabela de *clusters*.

Este algoritmo foi desenvolvido para uma aplicação que serve para definir o local em que um barco de pesca se manteve pescando, pois do ponto inicial até o local onde é realizada a pesca, o barco segue praticamente em linha reta e no local da pesca o motor do barco é desligado e o mesmo permanece parado, mas com variações de direção, devido ao movimento do próprio mar. Para esta aplicação o algoritmo apresentou resultados de ótima qualidade.

2.3 MINERAÇÃO DE DADOS ESPAÇO-TEMPORAIS EM REDES SOCIAIS

As redes sociais existem desde a antiguidade e tem acompanhado a evolução das tecnologias comunicativas, indo desde a escrita até a internet, gerando aquilo que atende hoje por rede social online, comumente chamada apenas de redes sociais, com exemplos consolidados como o caso do Facebook, do Orkut e do Twitter. Uma rede social é estruturada através das pessoas e seus relacionamentos e são espaços que servem para manter relações pessoais e profissionais, bem como criar novas relações.

Em entrevista concedida a revista Época em novembro de 2010, o especialista em ciência da computação Gregory Piatesky-Shapiro, afirmava que “as áreas mais quentes de novas pesquisas em *data mining* são as que envolvem redes sociais. A empresa poderá monitorar o comportamento dos consumidores nas redes sociais e de se comunicar com eles pelas redes e também pelas plataformas móveis, como celulares” (ÉPOCA NEGÓCIOS, 2010).

Dois anos após esta declaração, percebe-se que realmente o autor estava certo e já é possível encontrar inúmeros métodos e modelos de mineração de dados tradicionais capazes de extrair informações importantes dos dados gerados pelas redes sociais, como algoritmos para exploração de publicidade segmentada, ou seja, direcionar o conteúdo da propaganda para determinado perfil de usuário desejado, sistemas de recomendação, identificação de usuários influenciadores dentro das redes sociais, entre muitas outras aplicações. Um fator importante para que este tipo de mineração fosse realizado, foram as API’s para obtenção dos dados, disponibilizadas pelas próprias redes.

O formato de dados geográficos, coletados no momento das postagens dos usuários, por exemplo, passou a ser disponibilizado recentemente pelas principais redes sociais e já vêm atraindo a atenção de alguns pesquisadores, interessados em utilizar estes dados para gerar um novo tipo de conhecimento, denominado espaço-temporal.

Em Santos (2011) o estudo dos algoritmos de mineração de dados geográficos em redes sociais foi dividido em dois tipos de análises: estática e semântica. No caso da análise estática trabalha-se com dados geográficos que não se alteram ao longo do tempo, como localização de moradia de um usuário, que na maioria dos casos não se alteram ao longo do tempo. A análise dinâmica, apesar de ser um tipo de análise mais complexa pode gerar resultados mais ricos quanto ao ganho de conhecimento, pois neste caso as informações geográficas estão associadas às informações temporais. São exemplos deste tipo de dados as trajetórias percorridas pelos usuários, que serão mais exploradas no desenvolvimento deste trabalho.

Atualmente já existem alguns métodos para mineração de dados espaciais em redes sociais que utilizam a análise estática. Um dele foi desenvolvido pela equipe do Facebook e denomina-se *Find Me If You Can* (BACKSTROM et al., 2010) e realiza a predição da cidade de moradia do usuário, através de seu relacionamento dentro da rede utilizando algum conhecimento da geografia do local. Uma das premissas para este desenvolvimento foi a maior atenção a densidade e aos relacionamentos dos usuários em função da distância entre eles. Para realizar o estudo foram capturados os dados de aproximadamente três milhões de usuários que tinham todas as informações de endereço devidamente preenchidas em seus cadastros, para que fosse possível encontrar os dados de latitude e longitude do ponto e ainda era necessário que este usuário tivesse pelo menos um relacionamento com outro endereço também devidamente preenchido. O algoritmo utilizou as medidas de distância e densidade como métrica, pois, em geral, conforme aumenta a distância entre os usuários diminui a probabilidade de se conhecerem e da mesma forma quanto maior a densidade demográfica do local onde você mora menor a probabilidade de conhecer algum de seus vizinhos.

Foram utilizadas três metodologias para testar o algoritmo, uma deixando o usuário que deveria ser previsto sem a localização e usando todos os seus amigos para prever sua geolocalização, a segunda deixando vários usuários sem a localização e a terceira fazendo a combinação com a geolocalização por IP, que faz obtenção da localização de um usuário através do seu IP. Este algoritmo apresentou resultados interessantes, mas principalmente

reforçou a importância dos relacionamentos dentro da rede para a descoberta de conhecimento em redes sociais.

A segunda pesquisa de destaque foi realizada em (MISLOVE et al. 2010) e reproduzida em (BIEVER, 2010), utilizando os dados de localização dos usuários do Twitter. Neste caso a informação geográfica foi associada a palavras dentro dos *posts* do usuário que identificassem seu estado de humor. Foram utilizados cerca de 300 milhões de tweets dos Estados Unidos, em um período de aproximadamente três anos.

Foi realizado um trabalho de identificação do estado de humor da mensagem baseado na quantidade de palavras positivas e negativas dentro do texto. O dicionário *Affective Norms* for *English Words*, foi utilizado como base para estas classificações. Para este escopo não foram feitas análises de contexto da frase. O resultado deste algoritmo foi a caracterização do estado emocional de cada região dos Estados Unidos durante as horas do dia e nos dias da semana. A figura 6 apresenta um exemplo de resultado do algoritmo e descreve o grau de felicidade nas costas Leste e Oeste do país ao longo de 24 horas.

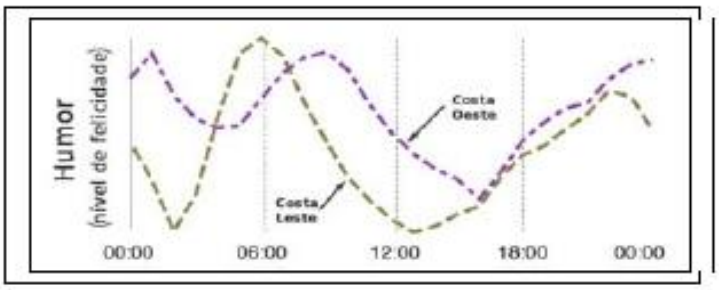

Figura 6 - Resultado de um dos testes do Mapa de humor (BIEVER, 2010)

As análises dinâmicas existentes, em geral, estudam o comportamento dos usuários e possibilitam a descoberta de conhecimento mais elaborado e padrões mais significativos. Em 2010 (YE et al., 2010) foi realizada uma pesquisa que trabalha com a idéia de recomendação de lugares, que utiliza os conceitos de confiança e similaridade de interesses, extraído dos relacionamentos. Neste caso os dados utilizados foram do Foursquare e seu objetivo foi desenvolver técnicas mais eficientes de recomendação.

A abordagem utilizada neste estudo foi a *friend-based collaborative* (FCF), baseado na classificação dos locais mais visitados, sua variante foi o modelo GM-FCF que utiliza as características geoespaciais dos dados. Para ambos, as seguintes premissas foram

estabelecidas: amigos tem maior interesse em lugares em comum do que pessoas que não se conhecem, amigos vizinhos tem maior possibilidade de visitarem os mesmos locais e atividades sociais são afetadas pela proximidade geográfica. Baseado nisto os algoritmos utilizam somente os amigos, sendo que no GM-FCF o peso é inversamente proporcional a distancia entre os mesmos.

Os algoritmos foram comparados a outros algoritmos de recomendação e foi concluído que os mesmos mostraram-se bastante competitivos. O FCF é bastante competitivo com relação a precisão e tem maior eficiência. O GM-FCF perde em precisão se comparado ao FCF, mas tem vantagens de desempenho.

Em (DENG et al. , 2009) foi realizado um estudo que analisa *tags* e *geotags*, para explorar o significado que os usuários dão para localidades. A base de dados utilizada foi a do Flickr, uma rede social para compartilhamento de fotos, e as informações geográficas foram extraídas dos metadados da imagem ou da marcação do local no foto, feita manualmente pelo usuário. A base contava com dados de fotos do período de um ano, tiradas em Amsterdan e arredores.

O algoritmo de mineração de dados utilizado foi o DBSCAN (Ester et al., 1996) para agrupamento espacial das imagens e faz a transformação de fotos e *tags* em uma matriz**.** Como resultado os autores observaram que quanto menor a distancia geográfica, maior a relação entre *tags*. Portanto a utilização de informações georeferenciadas pode auxiliar aplicações para sugestão de *tags*.

Utilizando as características de tempo real do Twitter, em Sakaki et al. (2010) é feita a utilização das mensagens e usuários do Twitter, como sensores de eventos em tempo real. Para realização de testes desse algoritmo foram utilizados apenas dados de usuários do Japão e busca-se fazer estudar o evento natural dos terremotos.

As diretrizes básicas são a análise semântica das mensagens, associando-as a uma posição no tempo e no espaço, para verificar se o evento que está sendo comentado não é algo passado ou um acontecimento presente. O artigo mostra que é possível descobrir o percurso de um evento probabilisticamente. Na Figura 7 é apresentada uma comparação entre a posição e trajetória do evento detectado em relação ao fenômeno natural.

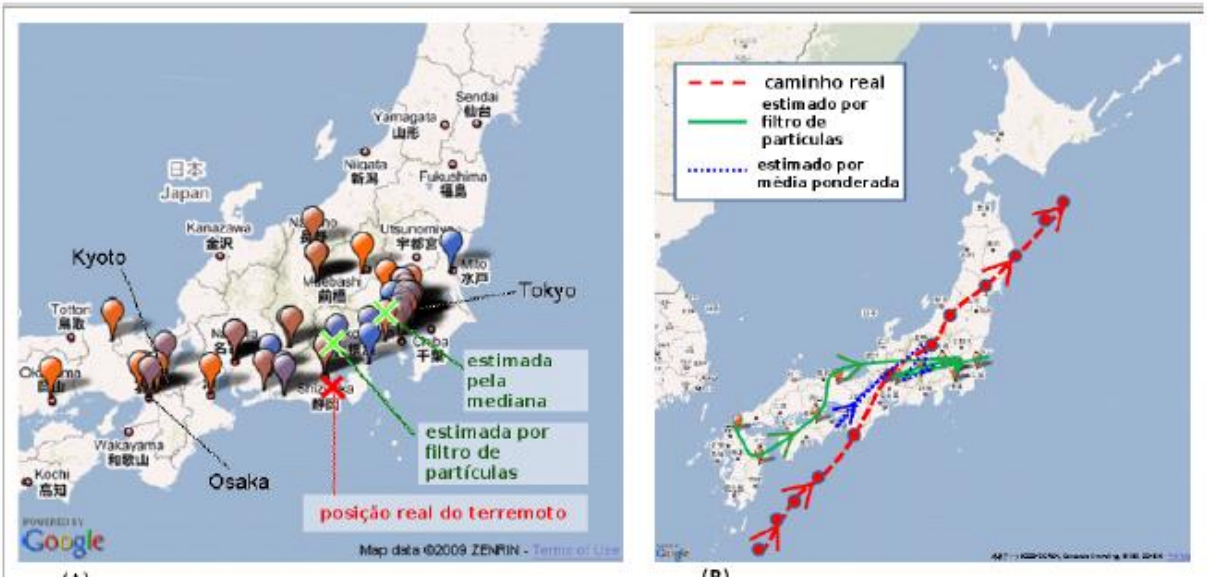

Figura 7 - Detecção de evento através do Twitter (SAKAKI et al., 2010)

Um dos estudos mais recente foi realizado sobre dados do Twitter e é descrito em (LEE et al. 2011), sendo que este trabalho foi apresentado em quatro outros artigos, todos apresentando uma evolução em relação ao anterior. Fujikasa et al. (2010a) apresenta os principais conceitos sobre o que seria a proposta final do método para análise e extração de regiões com atividades fora do padrão esperado, Fujikasa et al.(2010b) constata a importância na análise do conteúdo das mensagens para traduzir emoções e pensamentos dos usuários e facilitar a detecção do tipo de evento que disparou a atividade e em Fujikasa et al. (2010c) define-se os argumentos de pontos como regiões de interesse, culminado no trabalho final de Lee.

No quarto trabalho, Lee e Sumiya (2010), é feita a evolução dos conceitos tratados nos trabalhos anteriores e é realizada a definição de três métricas que devem ser utilizada para identificar determinado tipo de evento, sendo elas: número de tweets, número de pessoas e movimentação da multidão, respectivamente #tweets, #Crowd e #movCrowd. Sendo que para um evento do tipo festival existe um aumento das três variáveis, se o tipo do evento for festival local o aumento ocorre em tweets e movCrowd e para feriados o aumento ocorre em crowd e movCroud. Para utilizar o parâmetro de aumento das variáveis é necessária que previamente tenha sido calculada a média de cada uma para cada região.

No trabalho final apresentado em (LEE et al., 2011) são utilizados cinquenta eventos como testes para validação do algoritmo e o nome das métricas é alterado.

2.4 O TWITTER

Conforme foi possível observar na seção anterior os algoritmos já desenvolvidos para mineração de dados espaço-temporais em redes sociais normalmente são aplicados inicialmente a uma das redes existentes em específico. Uma dessas redes sociais para os quais alguns algoritmos já foram desenvolvidos é o Twitter.

Esta rede social foi criada em 2006, por Jack Dorsey e é um serviço gratuito de microblog, que se apresenta na forma de publicação de blog que permite aos usuários atualizações breves de texto e sua divulgação para que sejam vistas publicamente ou apenas por um grupo restrito.

O Twitter é o mais popular dentre os microblogs, permite o envio de mensagens com no máximo 140 caracteres, denominadas tweets, que podem ser visualizadas no perfil de quem postou e também na página de todos os seus seguidores. Basicamente o Twitter funciona como um diário, onde o usuário pode publicar desde o que está fazendo no momento até links interessantes ou notícias importantes. Atualmente muitas empresas e programas de rádio e televisão vêm utilizando como meio de divulgação da marca, realização de promoções e interação mais direta com os seus seguidores.

Os relacionamentos no Twitter apresentam características diferentes das demais redes sociais, pois atuam sobre uma infraestrutura assimétrica que permite conexões unilaterais, ou seja, para seguir alguém aquela pessoa não precisa aceitar seguir você também. As pessoas que um usuário X resolve acompanhar são os seus amigos e todos os usuários que desejam acompanhar a conta do usuário X são os seus seguidores.

Uma das análises feitas em Kwak et al. (2010) que estudou a topologia da rede do Twitter, foi a capacidade influenciadora de determinados usuários baseadas no conceito de re-tweets, que são os encaminhamentos de mensagens de outras pessoas e só podem ser feitos se o encaminhamento contiver exatamente os mesmo dados postados originalmente. Essa característica mostra a capacidade de difusão de informações que o Twitter oferece, sendo que em média as mensagens (*re-tweets*) chegam a alcançar cerca de 1000 usuários, ultrapassando o número de seguidores dos usuários, em geral.

O Twitter é atualmente muito importante, tendo em vista o número alto e crescente de usuários, bem como, por sua larga utilização como ferramenta de marketing e como

fornecedor de serviço para troca de mensagens. Segundo o levantamento apresentado em 2011, por uma empresa francesa que estuda redes sociais, chamada Semiocast, o Twitter começou o ano de 2012 com 383 milhões de contas registradas, sendo que deste total 107,7 milhões são dos Estados Unidos, ocupando este o primeiro lugar mundial em número de usuários na rede. Na segunda colocação, mesmo que com um número muito menor de contas, aparece o Brasil com 33,3 milhões de usuário.

A agência brasileira Monkey Business, divulgou um material com estatísticas e curiosidades a respeito do Twitter em 2011, esta pesquisa ocorreu na esfera mundial e consegue representar bem a dimensão do Twitter, mostrando que um bilhão de mensagens são postadas por semana nesta rede e 40% deste total é gerado via dispositivo móvel, número que vem crescendo a cada trimestre. A mesma pesquisa apresenta ainda que 43% das pessoas seguem alguma marca para ter acesso a ofertas e promoções e apenas 5% as seguem para fazer reclamações e xingamentos, além disto, 75% são propensos a comprar produtos da marca que seguem e outros 67% tem propensão a recomendar a marca.

3 APRESENTAÇÃO DOS DADOS E AVALIAÇÃO DOS ALGORITMOS

O desenvolvimento deste trabalho será realizado sobre uma base de dados coletada da rede social Twitter. Na secção 3.1 será feita uma apresentação breve dos dados a que se teve acesso para aplicação do processo de descoberta de conhecimento, bem como suas peculiaridades e a forma como foi realizada a captura destes dados. Como as trajetórias geradas pelas redes sociais se comportam de maneira um pouco diferente das trajetórias comuns, como as geradas por automóveis em movimento, a secção 3.2 detalha alguns dos algoritmos existentes para adição de semântica as trajetórias, IB-SMOT, CB-SMOT e DB-SMOT, mencionados nesta pesquisa e avalia sua aplicabilidade a trajetórias nas redes sociais.

3.1 APRESENTAÇÃO E PREPARAÇÃO DOS DADOS

Toda a tarefa de coleta dos dados da base utilizada para estudo foi realizada pelo mestrando em Ciência da Computação da Universidade Federal do Rio Grande do Sul (UFRGS), Augusto dos Santos, que também está utilizando os dados no desenvolvimento de sua tese. Em função do foco do trabalho ser a análise de trajetórias, foram coletados apenas os *tweets* que possuíam informações geográficas associadas à posição do usuário no momento das postagens, o que corresponde a aproximadamente 15% do total de mensagens.

Os dados foram disponibilizados por Augusto dos Santos através de um link na web, para que pudessem ser utilizados neste desenvolvimento. Os dados iniciais continham uma tabela com informações coletadas entre os meses de abril e novembro do ano de 2011, em qualquer localidade dentro do Brasil e outra tabela com informações cadastrais dos usuários.

A tabela de cadastros contem 208.310 usuários diferentes e armazena as informações: *id*, referente ao identificador único do usuário, *screen_name* que apresenta o nome pelo qual o usuário é reconhecido na rede, *name* que representa o nome completo do usuário, *fullprofile* com todas as configurações da página e *datetime*. Já a tabela de *tweets* conta com um número de 18.949.900 registros e apresenta os seguintes atributos: id, text, lat, long, datetime, created_at, user_id, country, full_place, place_type, state e city.

Inicialmente observou-se que o volume de dados era muito grande e em virtude do processamento de trajetórias ainda ser algo custoso computacionalmente, a demanda de tempo

requisitada para cada teste era bastante alta. Por ser um trabalho em caráter de pesquisa, onde muitos testes seriam necessários, optou-se por diminuir a quantidade de dados na base e por esta razão uma seleção apenas com os dados de Santa Catarina foi realizada para início das análises. Foi criada uma nova tabela que continha somente os dados do Estado citado, que dispunha de um número de 664.857 registros, que representam 3,5% do total de dados do Brasil. A Figura 8 apresenta visualmente o mapa político do estado de Santa Catarina, disponível no site do IBGE, acrescido dos pontos de *tweets* postados dentro de seus limites geográficos.

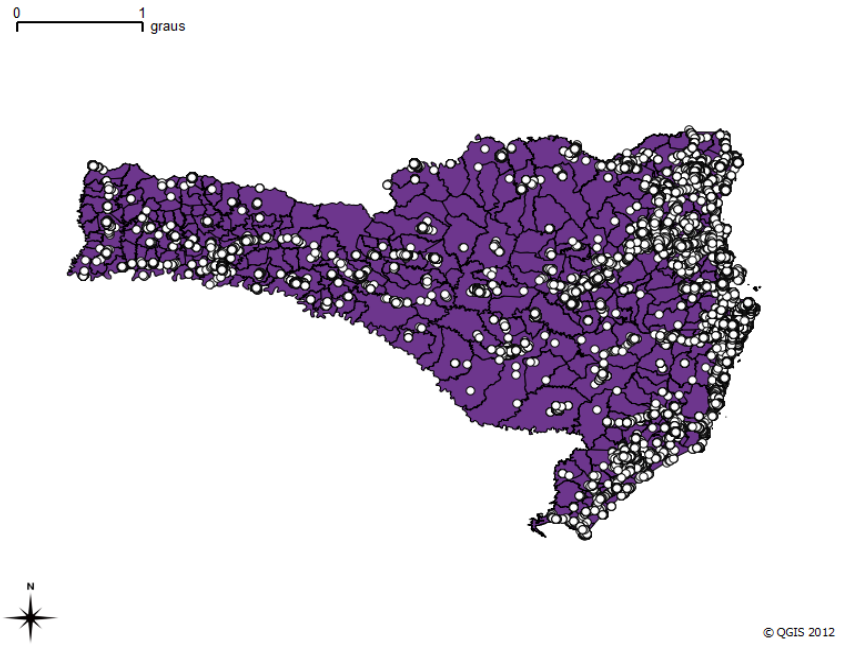

Figura 8 - Mapa de *tweets* em Santa Catarina

Através de uma simples análise visual da Figura 8 é possível diagnosticar que a maior concentração de *tweets* se encontra próximo a região litorânea do estado, coincidindo com a maioria das cidades mais populosas, que segundo dados do IBGE(2005), são: Joinville, Florianópolis, Blumenau, São José, Criciúma, Itajaí, Chapecó, Lages, Jaraguá do Sul, Palhoça e Tubarão. Analisando um pouco melhor os dados dentro da base, observa-se que o número de *tweets* nessas cidades corresponde a aproximadamente 75% em relação ao total do Estado, sendo que as cidades que mais tuitam são: Florianópolis, Joinville e Blumenau, representando aproximadamente 25%, 15% e 11% respectivamente, do total encontrado para o Estado. A Tabela 1 mostra que dentre as 10 cidades que mais tuitam, apenas uma delas não está entre as mais populosas do estado, sendo ela, Balneário Camboriú, que apesar de não ser uma das

mais populosas é a que possui maior densidade demográfica, além de ser extremamente turística.

Tabela 1 - Número de *tweets* por cidade

Número	Cidade	Número de *tweets*
1	Florianópolis	172.838
2	Joinville	104.417
3	Blumenau	74.281
4	Balneário Camboriú	36.576
5	São José	29.076
6	Chapecó	26.564
7	Itajaí	22.195
8	Jaraguá do Sul	17.571
9	Criciuma	16.664
10	Lages	16.618

Nesta análise inicial é possível perceber que os dados do Twitter, de maneira geral, podem retratar o que ocorre no mundo real como o fato de as cidades mais populosas terem maior número de *tweets*, assim como a região litorânea apresentar uma densidade maior de *tweets*, enquanto que no interior isso aparece de uma maneira mais dispersa, entre outras constatações.

Na fase de preparação dos dados foi necessária a criação de uma coluna no banco de dados, que recebeu o nome de *the_geom*, sendo esta efetivamente a coluna geométrica da base de dados. Inicialmente a base continha somente os campos latitude e longitude armazenados em colunas separadas e através da execução do código SQL mostrado na Figura 9a, foi feita a atualização de cada linha da tabela, para que todas tivessem a coluna geométrica preenchida.

Durante todo o desenvolvimento deste trabalho foi utilizado o banco de dados PostgreSQL e sua extensão para trabalhar com dados espaciais, denominada PostGIS. A grande maioria das funções já existentes nesta extensão e que facilitam a manipulação dos dados geográficos, fazem uso da coluna geométrica e por isso optou-se pela sua criação desde o início da pesquisa.

Ainda nesta fase, foram criadas na tabela de *tweets* duas colunas adicionais no banco, uma para armazenar o dia da semana em que o usuário tuitou, denominada “dia_da_semana” e a outra para guardar o período de tempo em que ocorreu o *tweet*, chamada “intervalo_tempo”, que poderia receber quatro valores, sendo eles: manhã, tarde, noite e

madrugada. Estes campos foram inseridos, porque após algumas análises observou-se que poderiam ser informações interessantes a serem utilizadas para extrair algum conhecimento dos dados, além de facilmente obtida através dos próprios dados já existentes na tabela. A Figura 9b apresenta também os códigos utilizados para realizar estas conversões.

```
(A)
-- ATUALIZAÇÃO DA COLUNA THE_GEOM

Update tweetssc SET the_geom = ST_SetSRID(ST_MakePoint(j.long, j.lat),4326)
_____________________________________________________________
(B)
-- ATUALIZA COLUNA DIA_DA_SEMANA, DEPOIS DE SUA CRIAÇÃO

UPDATE  trajectoriamicro_todos SET dia_da_semana = CASE Extract(dow from time)
when 0 then 'Domingo' when 1 then 'Segunda' when 2 then 'Terça' when 3 then 'Quarta'
when 4 then 'Quinta' when 5 then 'Sexta' when 6 then 'Sábado'
end

--ATUALIZA COLUNA INTERVALO_TEMPO, DEPOIS DE SUA CRIAÇÃO
--Manhã
UPDATE trajetoriamicro_todos SET intervalo_tempo = 'Manhã'
Where to_char (time, 'HH24:MI:SS')>='06:00:00' AND to_char (time, 'HH24:MI:SS')<
'12:00:00'

--Tarde
UPDATE trajetoriamicro_todos SET intervalo_tempo = 'Tarde'
Where to_char (time, 'HH24:MI:SS')>='12:00:00' AND to_char (time, 'HH24:MI:SS')<
'18:00:00'

--Noite
UPDATE trajetoriamicro_todos SET intervalo_tempo = 'Noite'
Where to_char (time, 'HH24:MI:SS')>='18:00:00' AND to_char (time, 'HH24:MI:SS')<=
'23:59:59'

--Madrugada
UPDATE trajetoriamicro_todos SET intervalo_tempo = 'Madrugada'
Where to_char (time, 'HH24:MI:SS')>='00:00:00' AND to_char (time, 'HH24:MI:SS')<
'06:00:00'
```

Figura 9 - (a) script para aualizar the_geom,
(b) script para atualizar colunas dia_da_semana e intervalo_tempo

Para finalizar a etapa de preparação dos dados e adequá-los ao escopo desta pesquisa foi necessária a construção da tabela de trajetórias de objetos móveis a partir da tabela espaço-temporal original. Para isto foi feita uma reorganização dos dados e uma seleção apenas dos atributos considerados importantes ou com possibilidades de interesse para etapas posteriores de análise. Todos os dados foram ordenados primeiramente pelo identificador do usuário seguido do tempo dos *tweets*, ou seja, todas as postagens de um determinado usuário aparecem ordenadas pela data e horário de sua geração, sendo que o identificador de usuário passa a ser o identificador da trajetória. Portanto, esta tabela foi formada pelos seguintes atributos: *gid* - um número sequencial que identifica cada registro da tabela, *tid* – identificador da trajetória de cada usuário, neste caso, como já mencionado, foi selecionado o identificador

do usuário já existente na tabela anterior para preencher o campo, *time* – formado pela data e a hora em que o *tweet* foi postado na rede social e o *the_geom* – que é a coluna geométrica.

Estes quatro campos são exigidos pelos métodos já implementados na ferramenta de análise WEKA – STPM (Álvares, 2007), para que o mesmo possa trabalhar com os dados e reconhecê-los como trajetórias. Além dos quatro primeiros, mantiveram-se na tabela os seguintes campos: estado, cidade, *tweet*, dia_da_semana e intervalo _tempo. Foram descartados na nova tabela os atributos: *lat* e *long* – sem utilidade devido à inserção da coluna *the_geom* e sua mais fácil manipulação, *country* – pois todos os dados da tabela são do Brasil o que tornava o campo redundante, *full_place* – que apresentava a Cidade e o Estado em um campo único, mas optou-se por manter Cidade e Estado em campos separados e place_type – este campo era preenchido com o valor "*city*" para todos os registros e não era usado para nada. A Figura 10a apresenta o código SQL utilizado para selecionar os dados que foram inseridos na tabela de trajetórias e a 10b apresenta parte da tabela de trajetórias.

(A)

```
--Insere dados na tabela de trajetórias
INSERT INTO trajetoriamicro_todos (tid, "time", estado, cidade, tweet, the_geom, dia_da_se-
mana, intervalo_tempo) SELECT user_id, created_at, state, city, "text", the_geom, dia_da_se-
mana, intervalo_tempo FROM tweets_micro ORDER BY user_id, created_at
```

(B)

-- Parte da tabela de trajetórias

tid bigint	time timestamp w	estado character var	cidade character var	tweet character var	the_geom geometry	dia_da_sema character var	gid integer	intervalo_ter character var
4136	2011-05-15	Santa Catar	Florianópol	Dia inútil,	01010000008	Domingo	1	Manha
4136	2011-10-24	Santa Catar	Florianópol	assim que e	01010000009	Segunda	2	Noite
4341	2011-04-16	Santa Catar	Florianópol	Indo pro Ch	01010000006	Sábado	3	Noite
4341	2011-04-17	Santa Catar	Florianópol	At home...	01010000005	Domingo	4	Noite
4341	2011-04-18	Santa Catar	Florianópol	Pastelzinho	01010000000	Segunda	5	Madrugada
4341	2011-04-18	Santa Catar	Florianópol	@debystrega	01010000005	Segunda	6	Madrugada

Figura 10 - (a) script que insere dados na tabela trajetoria, (b) Parte da tabela de trajetória

Esta etapa foi realizada de forma cíclica e não linear, ou seja, as modificações não ocorreram somente uma vez e nem exatamente na ordem em que foram descritas. Foram sendo realizados vários testes com a base, a fim de descobrir novos atributos que poderiam ser importantes na tabela e também para perceber que tipo de informação que a tabela não continha e que poderia ser útil futuramente. Nesta primeira parte os testes foram realizados através de consultas SQL no banco de dados e também a visualização dos dados na ferramenta Quantum GIS, um sistema de informações geográficas de código aberto, cujo objetivo inicial era ser um visualizador de dados SIG, conforme descrito no manual do

usuário do próprio software. A ferramenta Weka foi utilizada nesta fase também, mas apenas para verificar se a tabela de trajetória estava de acordo com as exigências dos métodos implementados.

Após as análises e adequações das tabelas, foi definido como objetivo de estudo somente os dados gerados na microrregião de Florianópolis, que segundo o IBGE além da capital Catarinense é formada também pelos municípios de São José, Biguaçu, Governador Celso Ramos, Palhoça, Antonio Carlos, Paulo Lopes, Santo Amaro da Imperatriz e São Pedro de Alcântara. Foram selecionadas dentro desta região somente as trajetórias dos usuários que tinham um número mínimo de dois *tweets*. Esta seleção final e com a qual se trabalhou no restante do estudo, totalizou um número de 212.085 registros, distribuídos em 3.489 usuários diferentes, o que representa uma média de aproximadamente 60,75 *tweets* por pessoa no período e podendo este valor variar entre 2 para o usuário com o menor número de *tweets* e 6.357 para aquele que tem o maior número.

3.2 ANÁLISE DOS ALGORITMOS EXISTENTES

A mineração de dados espaço-temporais especialmente em se tratando de trajetórias de objetos móveis, ainda é uma área bastante nova no universo de estudo da computação e por isso ainda existem poucos algoritmos criados e disponíveis para uso em programas comerciais, a grande maioria daquilo que é produzido é fruto de trabalhos e pesquisas acadêmicas e normalmente desenvolvidos para algum objetivo especifico.

A ferramenta Weka é um software de código aberto, desenvolvido em Java, por um grupo de pesquisadores da Universidade de Waikato, em 1993 e que se destina a tarefa de mineração de dados não espaciais, fazendo a integração de vários algoritmos existentes para este fim. O Weka é atualmente uma das ferramentas de mineração mais utilizadas no meio acadêmico. Em (Bogorny et al., 2006) foi feita uma extensão do software para que pudesse trabalhar com dados geográficos e em (Alvares, 2007) o mesmo foi estendido novamente para suportar, a partir de então, o pré-processamento e a mineração em trajetórias de objetos móveis.

O módulo desenvolvido em 2007 recebeu o nome de STPM e foi totalmente integrado a ferramenta, dando origem ao Weka-STPM. Esta versão é capaz de trabalhar tanto com dados não espaciais quanto com dados espaço-temporais de trajetórias, estes últimos obtidos

diretamente do acesso a uma base de dados, sendo ainda capaz de se conectar com qualquer banco devido a suas especificações Open GIS Consortium. Os três métodos para mineração de dados já citados no início deste capítulo foram implementados nesta versão do software, e foi através dela que foram feitas as análises sobre cada um dos algoritmos. A Figura 11 mostra a tela do Weka que faz conexão com o banco de dados e por onde posteriormente pode ser feito o acesso ao módulo de mineração de trajetórias, através do botão "Trajectory Data".

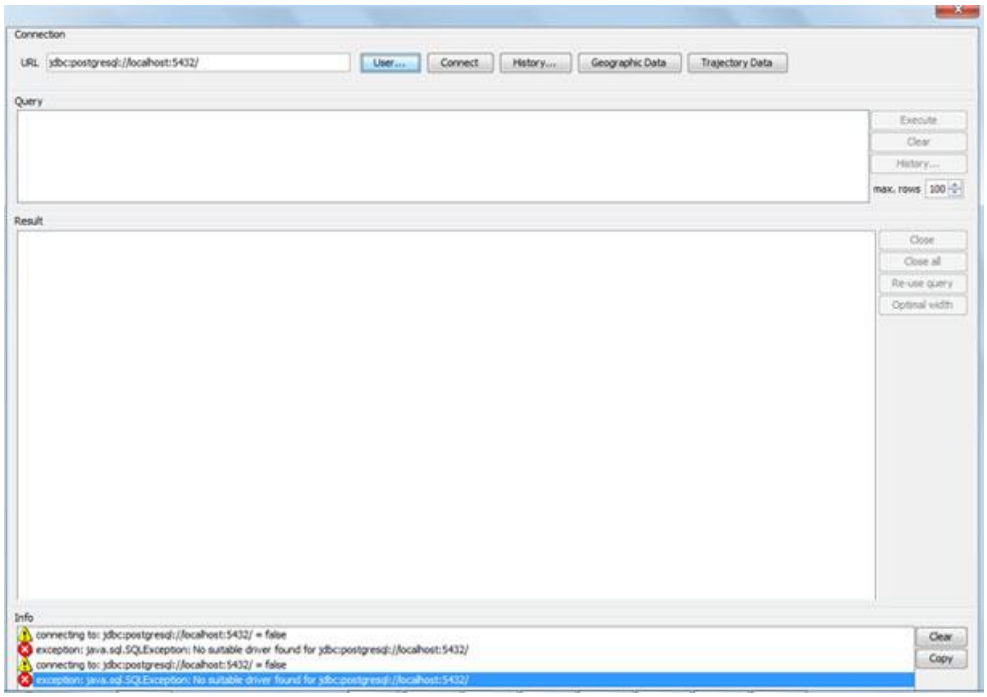

Figura 11 - Weka - Tela de conexão com o banco de dados

O principal objetivo desta análise foi verificar a aplicabilidade dos métodos implementados no Weka ao tipo de trajetória gerada por usuários de uma rede social, sendo que todos eles foram desenvolvidos baseados em dados gerados por GPS, normalmente considerando um carro ou uma pessoa em posse do aparelho.

A principal característica que diferencia estes dois tipos de trajetórias é a forma e a temporalidade em que os dados são gerados. Nos equipamentos de GPS, instalados em um carro ou em posse de um ser humano, os dados obedecem a uma temporalidade constante de geração, ou seja, a cada determinado instante de tempo, pré-definido, um ponto de localização é gerado e armazenado na base de dados, portanto toda a trajetória obedece a um intervalo de tempo constante entre um ponto e outro ao longo do trajeto. Já no caso específico das redes sociais, os pontos de localização são emitidos no momento em que o usuário posta uma

mensagem na rede, não obedecendo assim nenhuma regra de temporalidade entre os pontos gerados. Cada usuário pode enviar mensagens no momento que desejar e na quantidade que achar necessário, variando de um dia para o outro e dependendo das características de utilização da rede de cada usuário.

Outra característica que as diferencia é o fato de que a trajetória que um automóvel gera, por exemplo, visualmente forma o trajeto completo descrito por ele. Os pontos são gerados normalmente em um intervalo de tempo pequeno e por isso extremamente próximos uns aos outros e dessa maneira pode evidenciar algumas características da trajetória como, curvas, variação de direção, inicio e fim, entre outros. Um usuário de rede social, por mais que tenha um número grande de mensagens, o intervalo de tempo entre elas não costuma ser tão pequeno e também não ocorre durante toda a trajetória a ponto de descrever o trajeto percorrido, normalmente é possível visualizar alguns pontos nas regiões por onde o usuário passou, mas isso não significa que ele esteve somente naqueles locais e não é possível saber exatamente qual caminho percorreu entre um ponto e outro. De forma geral se traça uma linha reta entre os pontos indicando que o usuário transitou entre aqueles dois locais. Esta diferença pode ser mais bem compreendida quando analisadas visualmente conforme mostrado na Figura 12.

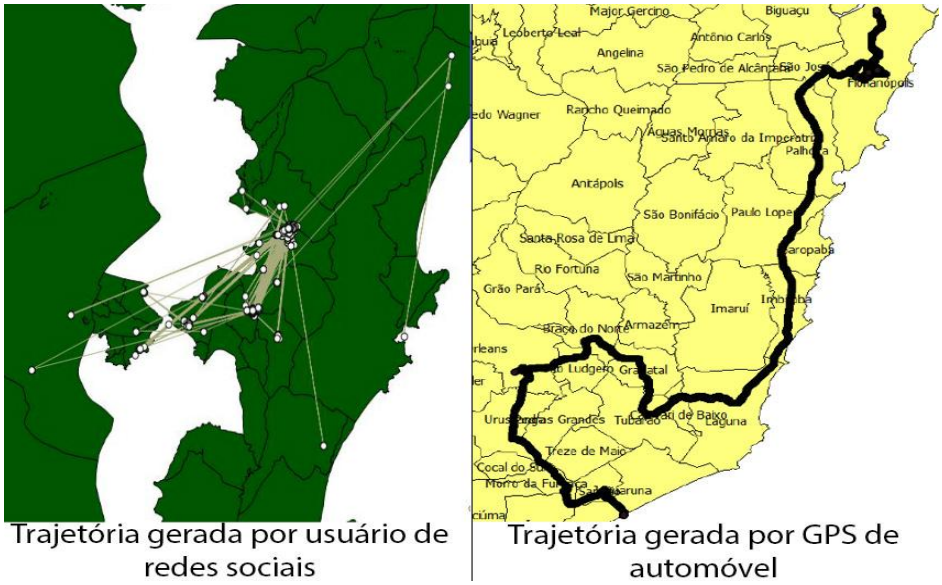

Figura 12 - Exemplos de trajetórias

As características descritas acima foram concluídas durante os testes de análise dos algoritmos. Os três algoritmos analisados trabalham com o conceito de *stops* e *moves*,

portanto buscam regiões de interesse no trajeto baseada em determinadas características. O método CB-SMoT e o método DB-SMoT, levam em consideração, respectivamente a variação da velocidade e da direção em uma trajetória para determinar os pontos de *stop* dentro do trajeto.

O algoritmo CB-SMoT se mostrou não aplicável às trajetórias de redes sociais, pois devido às características já apresentadas para esta, não é possível considerar a variação de velocidade entre dois pontos dentro do trajeto já que como eles não são gerados em um intervalo de tempo pequeno, não é possível delimitar a velocidade média em que o objeto transita e muito menos se houve uma queda nesta variável em determinado instante de tempo. O mesmo ocorre em relação ao método DB-SMoT, pois assim como a velocidade a mudança de direção também não pode ser calculada neste caso, já que os pontos são dispersos e normalmente conectados por uma linha reta que os interliga, sem retratar o verdadeiro caminho percorrido pelo usuário e as nuances de sua trajetória.

Por fim, foram realizadas as análises com o método IB-SMoT e este por considerar apenas o tempo em que a trajetória esteve interceptando aquela região para fazer a descoberta de um *stop* foi o único método que conseguiu executar com um pouco mais de êxito neste novo formato de trajetória. No entanto após algumas análises mais detalhadas percebeu-se que o conceito original do método, que era a descoberta de *stops* e *moves*, se perdeu ao ser aplicado em redes sociais, pois quando o usuário faz muitas postagens de um mesmo ponto, não quer dizer necessariamente que o usuário esteve parado naquele local, por exemplo, existem pessoas que utilizam a rede apenas de um mesmo lugar, como suas casas ou escolas, por exemplo, apesar de transitarem em outros locais. Além disto, nas redes sociais as pessoas costumam fazer postagens apenas uma vez de uma mesma região, como um ponto turístico, um lugar onde esteve ou mesmo um único *tweet* diário de sua casa ou trabalho, mas na execução deste método todos estes pontos eram excluídos da tabela de resultados gerada pelo algoritmo, pois se existe apenas um ponto e o próximo gerado ocorre em outro local, o método não pode realizar o calculo do tempo, porque teoricamente o usuário não permaneceu naquela região sendo que só existe um ponto que a intercepta. Para uma trajetória comum, realmente este ponto não interessa, no entanto para uma rede social ele não só é importante como é um dos mais comuns de ocorrer ao longo de um trajeto.

A partir desta análise foi possível perceber que o tipo de trajetória encontrada nas redes sociais apresenta algumas características particulares, que não são comuns àquelas que

vinham sendo estudadas e utilizadas como entrada para os métodos já implementados no Weka. Dentre os três métodos estudados o que obteve melhor resultado e um maior indício de aplicabilidade foi o IB-SMoT e no decorrer da análise ficou perceptível que muitas das manipulações que o método faz com as trajetórias são interessantes para serem realizadas com este novo modelo de trajeto, portanto na seção 4.1 deste trabalho será mostrada uma extensão realizada no método IB-SMoT, para que o mesmo possa trabalhar tanto com redes sociais quanto com trajetórias comuns.

4 ALGORITMOS PROPOSTOS

A pesquisa exploratória realizada nos dados evidenciou que o comportamento dos usuários nas redes sociais dá origem a um tipo particular de trajetória, com importantes características distintas e para as quais os métodos já desenvolvidos não estão preparados. Portanto, ao longo da pesquisa o foco esteve em procurar formas de tornar estes dados mais simples de serem minerados, devido ao grande volume em que são gerados e ainda descobrir que tipo de informações interessantes poderiam ser extraídas dos mesmos.

Com base nisto, foram desenvolvidos dois métodos capazes de trabalhar com este novo modelo de trajetória. O primeiro deles, detalhado na sessão 4.1, recebeu o nome de Regiões de Interesse de Postagem em Redes Sociais e é uma adaptação de um método já existente, que tem o foco na fase de pré-processamento dos dados. Já o segundo é um algoritmo novo que objetiva efetivamente a extração de conhecimento através do processamento das trajetórias, que será detalhado na sessão 4.2.

4.1 REGIÕES DE INTERESSE DE POSTAGEM EM REDES SOCIAIS (RIP-RS)

O grande número de dados gerados pelas redes sociais dificulta o processamento destas informações. Para análises de cunho mais generalizado e estatístico, na grande maioria das vezes pouco importa o conteúdo da mensagem dos usuários. Por exemplo, para descobrir a média de postagens de determinada cidade em relação a um país, não importa quem são os usuários ou o que estava escrito em suas mensagens, mas apenas a quantidade de vezes que as pessoas utilizaram a rede de dentro dos limites geográficos da região.

Durante a análise realizada sobre o método IB-SMOT, percebeu-se que apesar de o método não ter gerado bons resultados para os dados das redes sociais, o algoritmo realizava muitas manipulações intermediarias importantes com os dados e que se adaptaram bem a este novo tipo de trajetória. Assim, foi proposto um novo método denominado Regiões de interesse de postagem em redes sociais (RIP - RS), que na verdade é uma adaptação do IB-SMoT para trabalhar com o modelo de trajetórias geradas pelas redes sociais, neste caso específico o Twitter.

O RIP – RS tem um objetivo diferente do IB-SMoT, pois seu foco principal é o pré-processamento da trajetória para fazer a redução de pontos ao longo da mesma e assim facilitar a mineração dos dados. Para que um local seja considerado uma região de interesse neste novo método basta que o usuário tenha feito uma simples postagem deste local, enquanto o primeiro algoritmo tinha como meta encontrar estas regiões baseado no tempo em que o usuário permaneceu em determinada região. Apesar de ser um método que trabalha com um conceito novo e não mais com a definição original de *stops* e *moves* ele faz uso dos métodos já existentes no algoritmo anterior, apenas adaptando-os. Baseado nisto pode-se definir formalmente o RIP-RS, como um método que a partir de um conjunto de *tweets* de um usuário, gera uma sequência de regiões nas quais as mensagens foram emitidas. A granularidade da região é definida pelo usuário e pode ser, por exemplo, País, Estado, Município, Bairro, etc. Além disto, o método é responsável por fazer uma redução de pontos ao longo da trajetória, baseado nas regiões de postagens e que objetiva preparar os dados de forma a facilitar a etapa de mineração.

A redução de pontos é feita de maneira que a trajetória do usuário passe a ser formada por uma sequência de regiões da qual o usuário postou alguma mensagem, agrupando todas as postagens que o usuário fez de uma mesma região em um único ponto até que a sequência seja desfeita por um registro que faça interseção com outro local. A mesma localização pode aparecer diversas vezes em um trajeto desde que estejam intercaladas por uma região diferente. Por exemplo, considerando Florianópolis em uma situação hipotética com uma única trajetória, se o usuário fez vinte postagens da região Centro, seguidas de uma do bairro Estreito e mais duas novamente do bairro Centro, totalizando vinte e três registros na base de dados, após a execução do método, a tabela resultante contará com apenas três registros e uma trajetória com a seguinte sequência: Centro, Estreito, Centro. Se fosse utilizado o método IB-SMoT com essa finalidade o resultado seria diferente, pois o método desconsideraria o bairro Estreito no resultado, em virtude de existir apenas uma postagem em sequencia desta região, e apresentaria como resultado: Centro, Centro. O que ocasionaria em erro porque sinalizaria que o usuário só utiliza a rede social quando está na região Central da cidade, já o RIP-RS, nunca apresenta resultados com duas regiões iguais seguidas uma da outra, pois se o usuário realmente só utiliza a rede a partir de um local, o resultado é apenas um registro.

No desenvolvimento deste método de redução o conteúdo das mensagens postadas foi considerado não relevante para esta fase da análise, mas após a execução do método e testes realizados a partir dos resultados gerados pelo algoritmo, percebeu-se que o número de

postagens agrupadas em um mesmo ponto é uma informação importante a ser guardada, pois dependendo da situação pode-se extrair conhecimento útil como descobrir qual a região dentro de uma trajetória é a mais utilizada pelo usuário para fazer postagens na rede social, ou quais as regiões de onde os usuários mais utilizam a rede, entre outras. Portanto além de agrupar os pontos, o método passou a fazer a contagem de quantas postagens tinham sido agrupadas naquele ponto. Para armazenar essa informação foi criada mais uma coluna na tabela de resultados, denominada *numpostagem*.

A tabela resultante gerada após a execução do método sofreu algumas alterações em relação a que é gerada pelo IB-SMoT, principalmente por uma questão de nomenclatura das colunas, já que o conceito de *stop* não é mais utilizado. A tabela resultante sempre é nomeada contendo o prefixo "ri" mais o nome da tabela que foi processada e o nome da tabela de *Relevant Features*, e contém as seguintes colunas: *gid*, *tid*, *ri_id* e *ri_name* – identificador e nome da região de interesse, *start_time* e *end_time* – data e hora da primeira e da última postagem em sequencia dentro da região e *numpostagem*. Para casos de regiões que são interceptadas por somente um ponto a coluna *end_time* recebe o mesmo valor de *start_time*.

Na primeira execução do algoritmo é gerada uma tabela que contém o nome final da tabela gerada pelo método RIP-RS, na coluna seguinte o nome da tabela de trajetórias e na última coluna o nome da tabela de *Relevant Features* utilizada no processamento realizado pelo método. Caso a tabela já tenha sido criada então o método apenas adiciona o registro a mesma. Esta funcionalidade foi adicionada ao método para que possam ser realizados processamentos futuros sobre a tabela resultante em que seja necessário acessar as tabelas que serviram de entrada ao algoritmo.

4.1.1 Implementação

O algoritmo IB-SMoT encontra-se em funcionamento na versão 3.5.8 do Weka_STPM. Por este motivo esta foi a versão escolhida para implementar o RIP-RS. Como o módulo de trajetórias foi desenvolvido em 2007 a primeira alteração feita no sistema foi adaptá-lo ao PostGIS 2.0, que exige que todas as funções geográficas sejam padronizadas com o prefixo *"st_"* seguido do nome da função, enquanto nas versões anteriores o prefixo era opcional, podendo ser ou não utilizado.

Para ter acesso a parte de processamento de trajetórias do weka, o usuário deve executar o programa, em seguida clicar no menu *Applications* e em seguida *Explorer* que abrirá uma nova tela. Nesta tela deve escolher a opção *Open DB* que abrirá uma tela de conexão com o banco de dados, então deve ser pressionado o botão *User* em que é necessário informar os dados de conexão: *database*, *username* e *password*, e por fim o usuário clica no botão *Trajectory Data* e se todas as informações de conexão estiverem corretas é aberta a janela de processamento de trajetórias, como apresenta a Figura 13a.

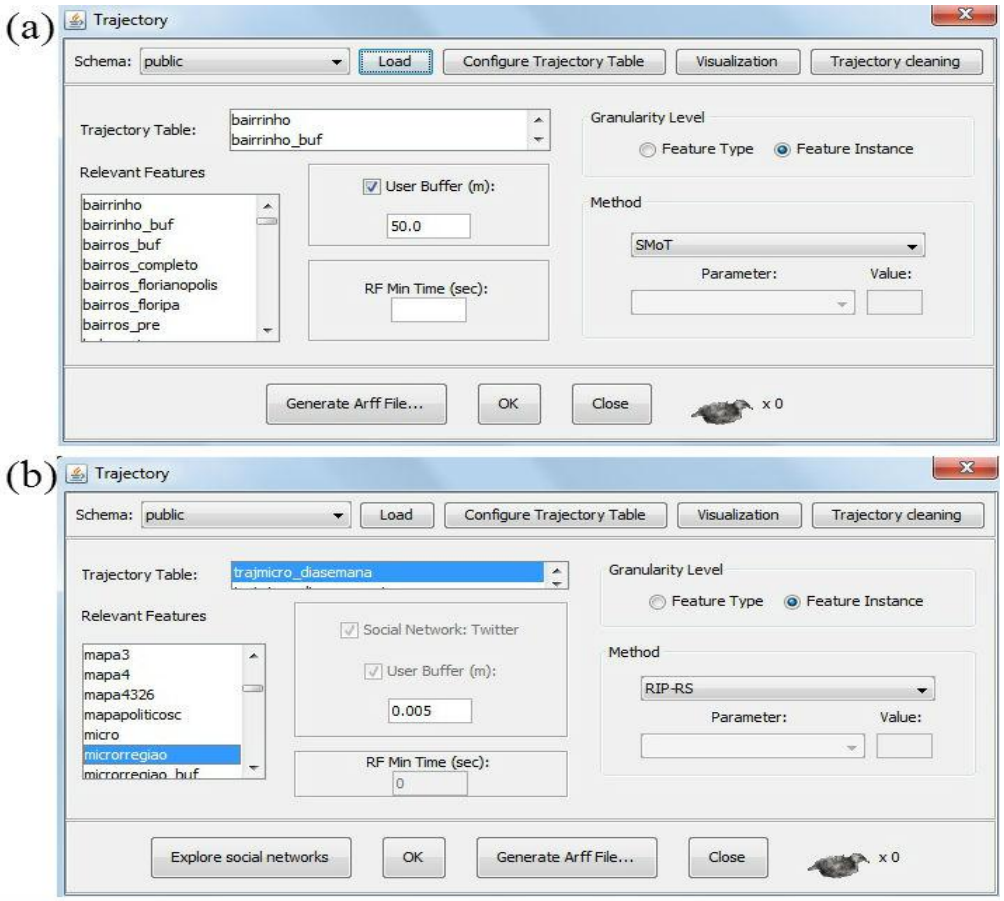

Figura 13 - (a) Tela original do Weka-STPM (b) com alterações do RIP-RS

Conforme pode ser observado na figura 13b, foram feitas pequenas alterações na tela do sistema no desenvolvimento do método RIP-RS. Como o novo algoritmo é feito como uma extensão do outro adaptado a redes sociais, eles utilizam muitos métodos em comum que praticam ações diferentes dependendo do tipo de trajetória dada como entrada. Para utilizar os algoritmos o usuário deve primeiramente escolher um esquema e clicar em *Load*, isto irá carregar na tela o nome de todas as tabelas com conteúdo geográfico que podem servir como tabelas de trajetórias, no campo de texto *Trajectory Table* e também como candidatos a região de interesse no campo *Relevant Features*.

Para executar o método desenvolvido para redes sociais o usuário deve escolher a opção RIP-RS no campo *Method*. Foi adicionada à tela uma *flag* chamada *Social Network: Twitter* que define o tipo de trajetória com a qual o método estará trabalhando, esta flag não pode ser editada pelo usuário, é preenchida pelo próprio sistema de acordo com o método escolhido. Ao selecionar o RIP-RS é inserido no campo *buffer* o valor pré-definido 0.005 e o valor zero para o campo *RF Min Time (sec)*, sendo que este último passa a não poder ser mais editado pelo usuário.

Os valores pré-definidos foram estabelecidos conforme análises anteriores de execução do método. O valor zero foi utilizado no campo *Min Time* por não ser mais utilizado o tempo para definição de um ponto de interesse, portanto não tem utilidade no método RIP-RS, no entanto o atributo não foi eliminado, pois caso o usuário esteja trabalhando com trajetórias comuns através do método IB-SMOT o campo é extremamente necessário. No método RIP-RS o valor do *buffer* está sendo utilizado somente no resultado final do algoritmo, ou seja, quando o algoritmo encontra os pontos de interesse ele gera sobre a junção dos pontos um buffer de 0.005 sobre a geometria deste ponto e este foi o valor que se mostrou mais adequando quando foi feita a visualização dos dados no QuantumGIS. Foram feitos testes com diversos valores, entre eles 0.05 que se mostrou muito alto e 0.003 que apesar da pequena diferença mostrou-se um pouco baixo, tornando mais difícil a visualização dos dados.

Após selecionar as tabelas, o método e manter ou modificar o valor pré-definido para buffer, o usuário deve clicar no botão *OK* para executar o pré-processamento da trajetória. Inicialmente o método verifica se a tabela é composta por trajetórias de rede social. Se for, seleciona a primeira das trajetórias e faz o teste de intersecção de cada um dos pontos do trajeto com os candidatos a regiões de interesse, mas neste caso, como o atributo *Min Time* foi definido como zero, todas as regiões que possuem somente um ponto também são consideradas como região de interesse. O algoritmo faz isso ciclicamente até a última trajetória contida na tabela. Todo o resultado da execução do método é armazenado em uma tabela no mesmo banco de dados da tabela original, conforme já descrito anteriormente.

Como o programa Weka foi desenvolvido na linguagem Java, também foi nesta linguagem de programação que foram feitas as alterações e adaptações necessárias para o funcionamento do RIP–RS. Durante o desenvolvimento não foram criadas novas classes no

código do sistema, somente foram realizadas adaptações às classes já existentes: *stop*, *trajecoryMethods* e *trajectoryFrame*.

4.2 PREDIÇÃO DE MORADIA E OCUPAÇÃO EM REDES SOCIAIS (PMO-RS)

Assim como em outras redes sociais, para ter acesso ao Twitter é necessário que o usuário faça um cadastro, neste caso, considerado bastante simples e que contém apenas as informações, nome, e-mail, senha e username do usuário na rede. Nenhuma informação adicional é solicitada, portanto com base nisto foi realizada uma pesquisa a fim de descobrir como extrair alguma informação pessoal do usuário a partir dos dados obtidos e que pudesse ser útil futuramente.

Apesar de contar com os mais diversos tipos de perfis, pode-se perceber que os usuários, de uma maneira geral, tendem a utilizar a rede como uma forma de diário pessoal, em que dizem o que estão fazendo em determinados momentos, portanto fazendo uso da rede diariamente nos locais por onde transitam. Em uma análise exploratória no conteúdo das mensagens postadas mostrou-se bastante claro que muitos dos usuários costumam postar mensagens de sua casa, trabalho, local de estudo, além de eventuais lugares visitados durante passeios, e estas informações são capazes de descrever o fluxo diário das pessoas.

O conhecimento sobre entre quais locais um grande número de pessoas que utilizam as redes sociais transitam diariamente pode ser útil para diversos setores, como por exemplo, campanhas de marketing focadas neste público alvo, podendo utilizar recursos das redes sociais como *hashtags* ou endereço da página social da empresa em banners expostos entre estes locais. Hoje em dia outro problema comum enfrentado pela sociedade são crimes virtuais dos mais variados tipos muitas vezes iniciados através das redes sociais como: sequestros, pedofilia, entre outros. A falta ou mesmo a falsidade das informações sobre os usuários dificulta a identificação do criminoso, no entanto o conhecimento do fluxo descrito acima poderia auxiliar os profissionais da polícia a identificar algumas informações como onde potencialmente poderia ser a casa ou mesmo o local de onde o usuário mais faz acesso à rede, abrindo novos horizontes que poderiam auxiliar o trabalho profissional da polícia. Atualmente a principal forma de obter a localidade de um usuário qualquer da internet, é através do endereço IP, por ser uma identificação obrigatória em qualquer conexão.

Baseado na utilidade que pode ter o conhecimento adquirido, aliado as informações que são possíveis extrair da base de dados, foi proposto o desenvolvimento de um algoritmo capaz de identificar de maneira probabilística os locais de moradia e ocupação dos usuários da rede, baseado na análise de suas trajetórias e ainda no conteúdo de suas postagens, o algoritmo foi denominado método de Predição de Moradia e Ocupação em Redes Sociais (PMO-RS).

Inicialmente foi feita a tentativa de utilizar todos os dados como entrada do método, mas como um de seus objetivos é encontrar o local de ocupação do usuário, chegou-se a conclusão de que o ideal seria desenvolvê-lo para trabalhar apenas com as postagens ocorridas durante os dias da semana, de segunda a sexta-feira. Dado que em geral as pessoas não trabalham ou estudam nos finais de semana e costumam transitar por locais diferentes, como casas de praia, locais turísticos e outros, descrevendo uma trajetória de lazer e não de ocupação e por isso induziam o algoritmo ao erro. Além disto, para servir de entrada para o PMO-RS a tabela precisa ter sido pré-processada pelo método RIP-RS já descrito anteriormente. Esta opção foi feita, pois em virtude do processamento de trajetória ser lento, com a tabela pré-processada o novo método pode ser executado mais rapidamente e algumas informações que precisam ser utilizadas pelo novo método podem ser mais facilmente obtidas diretamente do resultado da execução do RIP-RS.

Para predizer tanto o local de moradia quanto o de ocupação, o método faz a utilização de três métricas diferentes, sendo elas frequência de postagem das mensagens, horário das mensagens e semântica das mensagens, para simplificar denominadas respectivamente: #freqMsg, #horarioMsg e #semanticaMsg.

A métrica #freqMsg faz a seleção das regiões que serão candidatas a local de ocupação e moradia que o método busca encontrar. Após diversas análises dos dados foi possível perceber que a grande maioria das mensagens dos usuários são postadas a partir dos locais pelos quais ele costuma circular. Portanto o algoritmo seleciona um determinado número de regiões, que pode variar entre duas e cinco, que tenham o maior número de postagens dentre todas as regiões das quais um usuário já tenha postado pelo menos uma vez. Esta seleção torna-se então a relação de candidatos à ocupação e moradia daquele usuário e esta tarefa é repetida em particular para cada um dos usuários.

Após ter sido feita a escolha das regiões candidatas são utilizadas outras duas métricas para comparar dentre as regiões selecionadas qual tem a maior probabilidade de ser moradia

ou ocupação. A métrica #horarioMsg serve para realizar a contagem do número de mensagens enviadas entre dois intervalos de tempo pré-determinados. Inicialmente foi pensado em dividir o tempo em quatro grupos diferentes, sendo eles: manhã, tarde, noite e madrugada, mas acabou sendo feita a opção de agrupar estes quatro intervalos em grupos maiores e trabalhar apenas com dois intervalos de doze horas, sendo que fosse utilizado para ocupação um horário próximo ao padrão comercial, já que a maioria das pessoas encontra-se em seus locais de trabalho ou estudo neste horário, enquanto durante a noite e a madrugada as pessoas tendem a estar em suas casas. Foram feitos testes com vários intervalos diferentes e comparados os resultados finais de cada um deles e dentre todos os intervalos testados os que obtiveram os melhores resultados e foram escolhidos para ser utilizados no algoritmo, foram os seguintes: das 08:00h às 19:59h para ocupação e das 20:00 às 07:59h para moradia, embora nem mesmo este tenha apresentado 100% de acertos.

Para evitar erros advindos de usuários que não pertençam ao perfil geral utilizado como base para a definição dos intervalos de horário que foram utilizados para apontar prováveis locais de moradia e ocupação baseados nesta variável, como por exemplo, pessoas que estudam durante a noite ou trabalham durante a madrugada ou mesmo que estudam somente no período vespertino ou matutino e no outro se encontram em casa, buscou-se encontrar uma nova métrica que pudesse analisar o usuário independente do horário, para isto foi feita a opção de utilizar a avaliação semântica da própria mensagem através da métrica #semanticaMsg, tendo em vista que muitas postagens são capazes de retratar perfeitamente ou indicar onde o usuário está.

Inicialmente foi pensado em palavras bastante genéricas para fazer a avaliação semântica, como *casa* e *home* para indicar que o usuário esta em sua localidade de moradia e *trabalho* ou *escola* para indicar que o mesmo encontra-se no local de ocupação. No entanto apesar destas palavras serem bastante eficazes e apresentarem um grande número de acertos, também são capazes de induzir o método a um grande número de erros, em frases como: “Chegando do trabalho” ou “Agora vou para casa”. Na primeira seleção de palavras a porcentagem entre acertos e erros para algumas delas chegou a uma relação de 50% para cada uma e a partir de então se decidiu por fazer uma análise inicial das palavras separadamente somente no banco de dados para encontrar termos que pudessem apresentar uma proporção maior de acertos nos resultados. Cada palavra encontrada que pudesse representar moradia ou ocupação era testada no banco de dados e feita a proporção de acertos em relação ao número de registros nas quais apareciam.

Com esta lista em mãos as palavras foram adicionadas ao código do algoritmo, e passaram por uma nova fase de modificações que foi a análise dos resultados finais do método, para descobrir se existiam palavras que não haviam sido selecionadas e que poderiam ter encontrado melhores resultados. Nesta etapa foram encontradas muitas palavras com sintaxe da rede social como as postagens que utilizam o prefixo *"I'm at"* ou *"@ "* e que indicam uma certeza ainda maior de onde o usuário pode estar e que não tinham sido pensadas originalmente. A Tabela 2 apresenta a relação final de todas as palavras que foram utilizadas no algoritmo, sendo que esta última listagem apresenta um maior número de palavras, que indicam com um maior nível de certeza onde o usuário está.

Tabela 2 - Listagem de palavras - métrica #semanticaMsg

Palavras Moradia	Palavras Ocupação
"em casa"	" trabalhar" && !(ir trabalhar)
"home sweet home"	"work"
"lar doce lar"	"essa aula"
"dormi"	"m at unisul"
" at minha casa"	"m at ufsc"
" tv" && !(", tv")	"m at univali"
"@ my home"	"facul"
"banho"	"m at colégio"
"morando aqui"	"@ colégio"
"m at condomínio"	"campus"
"@ condomínio"	"instituto"
"m at residencial"	"m at escola"
"@ residencial"	"empresarial"
	"aula % agora"

Em geral a métrica #horarioMsg apresenta valores de contagem bem maiores do que #semanticaMsg. Em função disso, um método que apenas somasse os resultados das duas métricas não seria eficaz. Por exemplo, avaliando dois usuários hipotéticos, um com 200 postagens no horário de ocupação, mas sem nenhuma palavra que indique esta característica e outro com 100 postagens no mesmo horário, mas com 5 *tweets* com palavras que também indicam ocupação, se fosse considerado o número real de *tweets* percebe-se que as cinco palavras não teriam valor algum na análise, pois realizando a soma seriam encontrados os valores 200 contra 105 e mesmo que o segundo tenha representatividade nas duas métricas não seria o escolhido pelo método. O mesmo comportamento é encontrado para um grande

número de usuários, sendo que em grande parte deles a métrica #semanticaMsg não teria valor algum para representar mudança no resultado final do método. Como foi concluído que nenhuma das duas métricas tinham tanta superioridade em relação à outra foi feita a opção de normalizar estes dados para valores entre 0 e 1, para fins de igualar as duas métricas na análise.

Esta normalização foi realizada calculando a proporção dos dados em relação ao total daquela variável. A Tabela 3 retrata um exemplo hipotético de um usuário que publicou um total de 40 mensagens no intervalo de horário que indica Ocupação, sendo que deste total dez ocorreram na região 1, vinte aconteceram na região 2 e outras dez na região 3 e na terceira coluna é apresentada a proporção desses números em relação ao total. Na quarta coluna da tabela é possível verificar que foram encontradas quatro palavras que indicam ocupação na região 3 e apenas uma na região 1, sendo apresentada a proporção de cada uma delas na coluna ao lado desta.

Tabela 3 - Exemplo de predição de ocupação

Região	#horarioMsg	Prop #horarioMsg	#semanticaMsg	Prop #semanticaMsg	Somatório
1	10	0,25	1	0,2	0,45
2	20	0,50	0	0	0,50
3	10	0,25	4	0,8	1,05

Como é possível observar a última coluna da Tabela 3, apresenta o somatório da proporção de cada um dos candidatos para ser o local de ocupação daquele usuário, sendo que neste caso a Região 3 seria a indicada como o local onde o usuário estuda ou trabalha. Toda esta tarefa é repetida para a predição do lugar de moradia do usuário e é igualmente realizada para todos os usuários da base de dados em análise. O somatório de proporções pode atingir um valor máximo de 2 para uma região, sendo assim, o maior grau de certeza de que uma região é ocupação ou moradia, acontece se: #freqMsg + #semanticaMsg = 2. Em casos de empate entre regiões com maior somatório, o algoritmo escolhe primeiramente o de maior #freqMsg, permanecendo a situação empatada a escolha se direciona para aquele que tem

maior #semanticaMsg, mas se mesmo assim persistir o empate o método conclui que não pode definir qual região escolher.

Para armazenar todo o resultado encontrado após o processamento do método PMO-RS é criada uma tabela no banco de dados, que contém o identificador do usuário e os dados referentes à moradia e à ocupação, sendo eles nome do local, identificador do local, proporção e número de palavras encontradas e de mensagens no horário da função para a qual foram escolhidos (moradia ou ocupação), além do *the_geom* que é a geometria do polígono resultante da junção dos pontos (coordenadas dos *tweets*) que possuem as características de moradia ou ocupação.

4.2.1 Implementação

O desenvolvimento do PMO-RS também foi integrado a versão 3.5.8 do Weka_STPM, já com o algoritmo RIP-RS implementado. Portanto, devido às características já descritas para o software foi mantida a linguagem de programação orientada a objetos Java para sua codificação.

Para acessar a parte de mineração de dados de redes sociais o usuário deve primeiramente seguir o mesmo caminho que o leva até a tela para processamento de trajetórias. Nesta tela, se o usuário quiser apenas manipular os dados de redes sociais, deve clicar no botão *"Explore Social Networks"* e a partir deste clique abrirá uma janela, conforme mostrado na Figura 14, que atualmente dispõe somente do algoritmo PMO-RS implementado para execução, mas que foi pensada para ser um local onde futuramente possam se agrupar todos os métodos que possam vir a ser desenvolvidos para a mineração de dados de redes sociais.

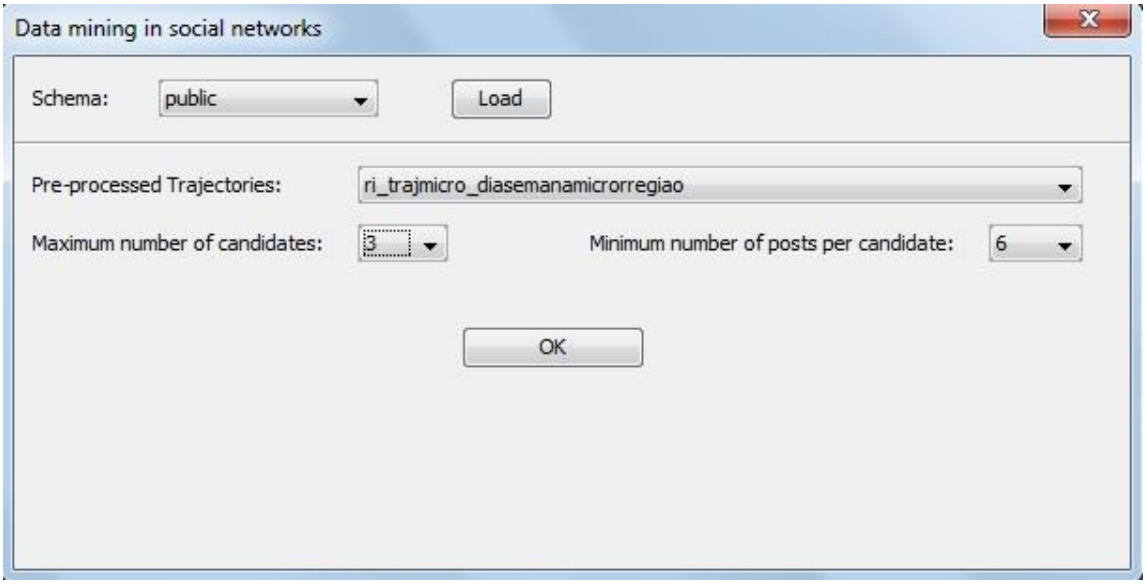

Figura 14 - Tela de mineração de redes sociais

Ao abrir a tela, apenas o combobox *Schema* e o botão *Load* estarão ativados, por isso inicialmente o usuário precisa escolher o *schema* no qual deseja trabalhar, clicar em *Load* e a partir deste momento todos os outros campos serão carregados e se tornarão ativos para utilização. Como já mencionado o método recebe como entrada somente as tabelas já pré-processadas pelo método RIP-RS, por isso no combobox *Pre-processed Trajectories* são carregadas somente as tabelas de redes sociais que já foram manipuladas por este método.

Existe ainda na tela um campo para escolher o número máximo de candidatos por usuário (*maximum number of candidates*), ou seja, este número determina quantas regiões, dentre as que o usuário postou alguma mensagem, serão selecionadas para ser candidatos a local de ocupação e moradia deste usuário. Esta seleção é feita baseada na ordem decrescente de número de postagens de cada região. Portanto, se escolhemos o número três, o método selecionará no máximo as três regiões de onde o usuário mais utilizou a rede social para fazer a análise de predição. O último parâmetro que pode ser definido na tela é o número total mínimo de postagens que devem ter sido feitas de uma região candidata para que esta seja considerada nas análises.

Após ser selecionado cada campo o usuário deve clicar no botão *"OK"* para que o algoritmo inicie sua execução. No desenvolvimento do algoritmo foram criadas quatro novas classes, conforme pode ser observado no diagrama apresentado na Figura 15, sendo elas: Usuários, Candidato, ExplorarRedesSociaisMetodos onde estão armazenados os principais

métodos do algoritmo e ExplorarRedesSociais que representa a tela já apresentada na Figura 15.

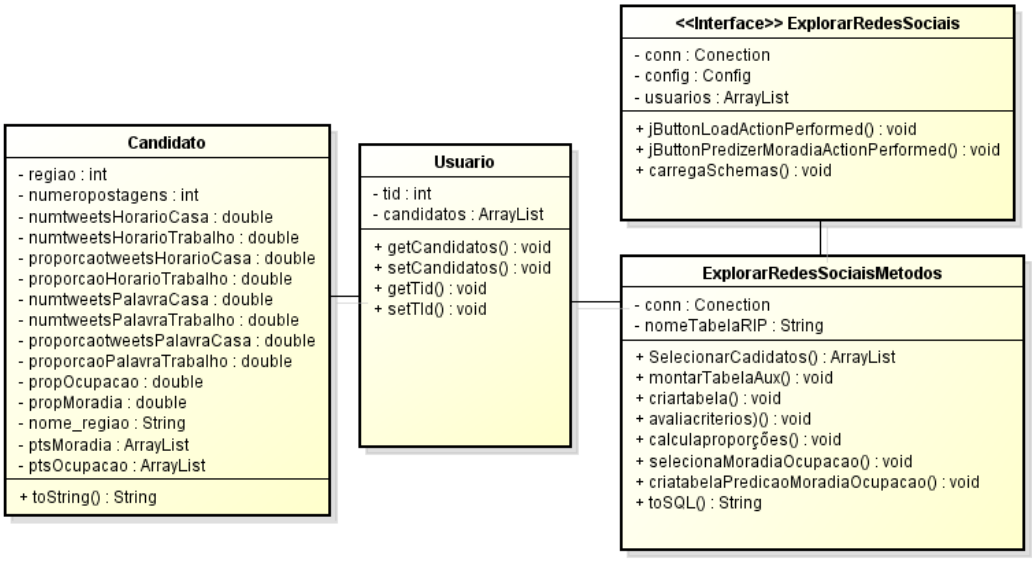

Figura 15 - Diagrama de classes

Inicialmente o PMO-RS faz busca na tabela gerada pelo método RIP-RS que contém o nome da tabela de trajetórias e o nome da tabela de *Relevant Features*, que foram utilizadas para dar origem a tabela de trajetos pré-processados selecionada. De posse disto, é feita a seleção de todos os usuários que aparecem na tabela de trajetórias e que respeitam o número mínimo de postagens escolhido e para cada um deles seleciona o número máximo de candidatos definido na tela do algoritmo, através do método SelecionaCandidatos() que faz a contagem de quantas postagens cada um tem em cada região e posteriormente seleciona de acordo com o número de candidatos definido. Com estes dados, o algoritmo faz a execução do método montaTabelaAux(), onde é criada uma tabela auxiliar com todos os *tweets* de cada usuário que fazem intersecção com os locais que são candidatos a ocupar a posição de moradia e ocupação para cada um deles.

Tendo a tabela auxiliar montada, o algoritmo cria a tabela que armazenará o resultado final do algoritmo, através do método criaTabelaPredicaoMoradiaOcupacao(). Após criar a tabela o algoritmo inicia a análise das métricas #horarioMsg e #semanticaMsg para cada candidato de cada um dos usuários, armazenando o número real encontrado para cada métrica em seus respectivos candidatos, através da execução do método avaliaCriterios(). O próximo

passo é a execução do método calculaProporcoes(), que é responsável por realizar a normalização dos valores das métricas no intervalo entre zero e um.

Para finalizar é feita a execução do método selecionaMoradiaOcupacao(), que é responsável por fazer o somatório e as comparações entre as proporções finais de cada candidato e selecionar dentre eles qual local é moradia e qual é o local de ocupação. Neste método são tratadas as situações de empate entre os candidatos e ainda é feita a inserção dos registros na tabela de resultados que foi criada durante a execução do algoritmo. Este método faz as análises por usuário, ou seja, para cada usuário ele faz a determinação de moradia e ocupação.

5 EXPERIMENTOS E RESULTADOS

Para verificar a efetividade e os resultados dos algoritmos desenvolvidos neste estudo, tanto RIP-RS quanto PMO-RS, foram realizados alguns experimentos com a base de dados de postagens da rede social Twitter, utilizando os dados gerados no período de oito meses, de abril à novembro de 2011, que tenham sido postados dentro da microrregião de Florianópolis.

Foi criada uma tabela contendo o mapa da microrregião de Florianópolis, sendo que a capital foi dividida em bairros e as outras cidades permaneceram sem divisão, dado que o maior objetivo era analisar o fluxo de pessoas dentro da capital, e desta com as cidades adjacentes. Essa tabela foi utilizada como candidato a pontos de interesse ou *Relevante Features*, em todos os testes.

Os testes foram feitos através do Weka-STPM já com os algoritmos implementados e algumas análises extras foram feitas nos dados resultantes. O SGBD escolhido para ser utilizado na realização dos testes foi o Postgres extensão PostGIS. Algumas análises foram realizadas através do Microsoft Excel e a visualização dos dados foi feita no Quantum – GIS, através de sua conexão com o PostGIS.

5.1 EXPERIMENTOS COM O MÉTODO RIP-RS

As primeiras experiências realizadas com o método RIP-RS, tem o objetivo de avaliar se realmente houve redução de pontos e se essa redução não prejudicou a trajetória. Inicialmente foi feita a execução do método, buscando a intersecção dos locais de postagem com a tabela de pontos de interesse gerada para esta região.

A execução do método RIP-RS não teve alterações de tempo em relação ao que já acontece com o IB-SMOT, depende do tamanho da tabela que se leva para análise, para tabelas muito grandes pode ser lento em virtude do trabalho com dados geométricos, mas para tabela com um número razoável de dados pode processar em questão de minutos. A tabela de postagens da microrregião utilizada como entrada para o algoritmo no primeiro teste realizado continha um total de 212.085 registros e a tabela de candidatos a *Relevant Features* possui um total fixo de 53 registros. Neste caso o tempo total de processamento foi de aproximadamente 20 minutos, incluindo seleção das trajetórias, intersecção com os candidatos a ponto de

interesse e manipulação destes dados até sua inserção na tabela de resultados no banco de dados. Este teste foi realizado sem a utilização de índices geométricos que podem diminuir consideravelmente o processamento de trajetórias.

Após a execução do RIP-RS a tabela de trajetórias resultante totalizou 45.270 registros, demonstrando que o método foi capaz de fazer uma boa diminuição no número de registros da tabela, apresentando como resultado uma tabela com aproximadamente 21,34% do número de registros da original, além do que consultas como: soma do número de postagens por bairros ou número de vezes que uma região de interesse de postagem aparece dentre todas as trajetórias, tornam-se extremamente simples com a nova tabela, sendo a primeira capaz de ser executada em 157ms e a segunda em 151ms. A Tabela 4 apresenta a diferença em tempo de execução destas duas consultas feitas através de um comando SQL diretamente no banco de dados, sobre a tabela resultante do RIP-RS e a tabela original.

Tabela 4 - Comparação de tempo de execução de consultas

	Tabela resultante do RIP-RS	Tabela de trajetórias Original
Consulta 1 – postagens/bairro	157 ms	256440 ms
Consulta 2 – número de aparições RIP	151 ms	+ 256440 ms

A diferença de tempo de execução das consultas 1 e 2 em grande parte se dá pelo fato de que a tabela pré-processada já apresenta as informações pesquisadas diretamente para serem utilizadas, além do que os dados já passaram pelo processo de intersecção geométrica com os candidatos a ponto de interesse, enquanto na tabela original isso precisa ser feito para chegar a esta informação. A segunda consulta não foi realizada até o final, sendo suspensa quando passou o tempo da anterior, pois somente assim já válida a questão da melhora de tempo para chegar a um resultado utilizando-se os dados resultantes do RIP-RS para este tipo de análises estatísticas com cruzamento de informações entre duas tabelas. O RIP-RS apesar de não realizar todo o seu processamento em um tempo considerado muito baixo, faz a intersecção entre as duas tabelas apenas uma vez e disponibiliza os dados para facilitar a análise, enquanto com as tabelas originais a cada consulta é necessário que se faça o

cruzamento dos dados, dificultando o processo caso a consulta precise ser executada mais de uma vez.

Depois de verificar que os resultados obtidos com o RIP-RS são satisfatórios no quesito tempo de execução e benefícios que podem trazer para análises futuras que dependam dos dados, houve a preocupação de se fazer uma análise do resultado final da trajetória para confirmar que não havia distorções e alterações de trajeto em relação à trajetória apresentada originalmente, seja omitindo alguns pontos de visitação ou fazendo algum tipo de deslocamento da trajetória em virtude do agrupamento de pontos.

Após várias análises verificou-se que não existem alterações no trajeto após a execução do método. Para exemplificar foi escolhido aleatoriamente um usuário que possui uma trajetória original composta por 19 postagens e percorreu 4 locais diferentes. O resultado do RIP-RS encontrou os mesmos quatro locais, mas composto por apenas 8 registros que formam o seguinte trajeto <Itacorubi, Lagoa, Itacorubi, Lagoa, Itacorubi, Biguaçu, Itacorubi e São José>. Portanto, como esperado, o método não modifica o trajeto percorrido pelo usuário, apenas agrupa os pontos emitidos na sequencia de um mesmo local, evitando que exista uma trajetória em que a mesma região apareça mais de uma vez em sequencia, por exemplo: <Itacorubi, Itacorubi>, mas mantendo na trajetória a ocorrência do mesmo local desde que o usuário tenha postado de outra localidade entre eles. Dessa forma o algoritmo mantém as características da trajetória e elimina o excesso de pontos emitidos do mesmo lugar.

Dado que os resultados apresentados pelo RIP-RS podem ser úteis e confiáveis, foram feitas algumas análises de cunho estatístico em cima desses dados encontrados na tabela final do método. Esses estudos serão apresentados nas seções 5.1.1 na qual foi feito um comparativo entre os dados durante a semana e no final de semana e 5.1.2 na qual foi aplicado um método de descoberta de padrões em trajetórias já existente no Weka-STPM.

5.1.1 Análise Comparativa – Dia de semana versus Final de semana.

O estudo dos dados das redes sociais vem apresentando um perfil bastante parecido com o movimento real das pessoas em seu dia-a-dia. Por conta disso foi realizada uma análise das trajetórias do Twitter, afim de comparar as semelhanças e diferenças do comportamento dos usuários durante a semana e no final de semana e ainda fazer uma analogia entre o

comportamento virtual em relação ao comportamento real relativos ao fluxo de movimento estabelecido pelas pessoas.

Para iniciar estas análises foram criadas duas tabelas separadas no banco de dados, baseadas na tabela formada pelos *tweets* postados na microrregião de Florianópolis, sendo que uma apresentava as trajetórias com postagens feitas durante a semana e a outra as postagens do final de semana e ambas foram levadas ao pré-processamento através do método RIP-RS. Sobre o resultado final foram realizados alguns testes e três planilhas foram montadas na ferramenta Microsoft Excel, a primeira contendo as porcentagens encontradas para cada região de interesse (RIP), a segunda composta pelo número total de *tweets*, a média diária e a porcentagem para cada local e uma terceira trazendo um comparativo do número de *tweets* por período do dia. Todas elas formadas pelos dados emitidos durante a semana e no final de semana, organizados separadamente para comparação. Nesta seção serão apresentados alguns trechos das planilhas criadas para melhor compreensão dos resultados do método.

Inicialmente foi feita a análise dos dados, através do número total de vezes que uma região de interesse de postagens (RIP's) aparece ao longo das trajetórias. Neste caso não é possível obter média diária ou por usuário, dado que é feito o agrupamento de pontos conforme já explicado. A Tabela 5 apresenta apenas uma parte dos resultados obtidos, mostrando as regiões que ficaram entre as dez que mais apareceram na base de dados.

Tabela 5 - Análise de RIP's – 10 primeiros

DIAS DE SEMANA				FINAL DE SEMANA		
RIP	**Aparições**	**%**		**RIP**	**Aparições**	**%**
Centro	6558	20,80%	1	Centro	2551	17,40%
São José	2990	9,48%	2	São José	1556	10,61%
Trindade	2864	9,08%	3	Itacorubi	975	6,65%
Itacorubi	2736	8,68%	4	Trindade	757	5,16%
Estreito	1326	4,21%	5	Lagoa da Conceição	713	4,86%
Agronômica	1094	3,47%	6	Estreito	625	4,26%
Tapera	1026	3,25%	7	Tapera	532	3,63%
Palhoça	955	3,03%	8	Palhoça	518	3,53%
Capoeiras	946	3,00%	9	Agronômica	466	3,18%
Córrego Grande	929	2,95%	10	Jurere	438	2,99%

Através da Tabela 5, percebe-se que as duas primeiras posições se mantêm as mesmas, mas demonstra uma diferença importante e que retrata a primeira diferença entre os perfis analisados, pois a região Centro apesar de se manter na primeira colocação apresenta uma queda percentual no final de semana ao contrário da cidade de São José que apresenta aumento em sua porcentagem, retratando o fluxo de trabalho São José – Florianópolis. Outra alteração importante de uma tabela em relação a outra é a troca de posições entre Trindade e Itacorubi, sendo que apesar de perder apenas uma posição a primeira região tem uma queda percentual notável de 9,08 para 5,16 enquanto Itacorubi apesar de ganhar uma posição também apresenta queda percentual, mas em uma escala menor se comparada a Trindade. Neste caso pode-se evidenciar a queda nestas regiões de interesse ao fato de que as duas regiões possuem Universidades instaladas dentro de seus limites e isso faz com que o número de *tweets* seja maior durante a semana, mas também são bastante residenciais e por isso apesar da redução ainda apresentam um bom número no final de semana.

Ao realizarmos a análise do restante das dez primeiras posições, percebe-se um aumento alto, quase o dobro, da porcentagem de vezes que a região de interesse Lagoa da Conceição apareceu no final de semana em relação aos dias de semana, passando de décimo segundo local de onde mais se esteve utilizando a rede social para a quinta colocação. Outra observação interessante está na décima posição da parte que representa os dados do final de semana, onde aparece o bairro de Jurere, o qual nem ao menos aparece entre os dez primeiros nos dados que retratam os dias de semana, apresentando um aumento de aproximadamente 51% no valor absoluto de porcentagem, saindo de 1,61% para 2,99%. Ambos os casos apresentam bairros da região mais litorânea de Florianópolis, que realmente apresentam maior movimento nos finais de semana e em virtude desta semelhança foram feitas análises de outros bairros que possuem características semelhantes a estes dois, e observou-se que a grande maioria tem o mesmo perfil na rede social, como é o caso, por exemplo, dos bairros Campeche, Ribeirão da Ilha e Ingleses, todos apresentando uma percentagem maior de *tweets* no final de semana. Essa análise mostra mais uma situação em que a análise baseada na rede social é capaz de traduzir a realidade de um local e assim dá maior confiabilidade aos resultados de estudos que possam vir a ser feitos sobre estes dados.

Em última análise observou-se que muitos bairros dentro da capital apresentam um número de *tweets* maior do que o de algumas cidades, destacando-se aqueles que ficam a frente de Palhoça que é a décima quinta cidade do Estado que mais apresenta postagens

dentro da base em estudo. Já a cidade de São José que é a quinta colocada dentre as cidades do estado que mais tuitam perde apenas para o bairro Centro na análise.

Na análise feita considerando o número exato de *tweets* por região de interesse, os resultados foram bastante parecidos com os da análise apresentada anteriormente, mas deixa em evidência o fato de que muitos usuários fazem suas postagens muitas vezes de um mesmo local, como é o caso da região Centro, que apareceu 6.558 vezes como RIP, mas quando somada a quantidade de *tweets* que existiam dentro dessas 6.558 aparições, chegou-se ao número de 34.330 postagens somente durante os dias de semana. Além disto algumas regiões aumentam sua representatividade quando a análise é focada no número exato de tweets e não na quantidade de vezes que aparece como RIP, como o caso do bairro Carianos.

As análises de aumento e diminuição do número de postagens apresentadas anteriormente são reafirmadas nesta segunda análise, como é evidenciada a questão já abordada de redução de postagens na Trindade durante o fim de semana, sendo que nos dias de semana a média diária é de 78,56 caindo para 44,09 nos finais de semana. Analisando a média diária de *tweets* neste período, considerando todos os locais, ocorre aumento no final de semana apresentando um número médio de 900,13 postagens por dia contra 842,59.

Para finalizar esta análise, foi feito o estudo do número de postagens por horário, apresentado na Tabela 6 e que traz o período noturno como sendo o portador do maior número de postagem nos dois perfis (dias de semana e fim de semana), assim como o período da manhã fica com os menores números também nos dois perfis, com uma média muito abaixo dos outros três períodos. Os períodos tarde e madrugada trocam de posição, ficando a primeira na segunda colocação durante a semana e em terceiro nos finais de semana, enquanto com o período madrugada ocorre o inverso, evidenciando mais uma vez o comportamento real do usuário já que no intervalo de segunda a sexta as pessoas trabalham e tendem a dormir um pouco mais cedo, já durante o sábado e o domingo, as pessoas, principalmente os mais jovens, costumam sair para se divertir durante a madrugada, permanecendo acordados por mais tempo e tuitando proporcionalmente mais.

Tabela 6 - Tabela de média de postagem por período

	Final de Semana		Dia Semana		Todos	
	Número	%	Número	%	Número	%
Noite	22912	36%	48033	32%	70945	33%
Madrugada	18645	29%	40583	27%	59228	28%
Tarde	17934	28%	45079	30%	63013	30%
Manhã	4024	6%	14875	10%	18899	9%
Total	63515	100%	148570	100%	212085	100%

5.1.2 Utilização do método *TrajectorySequentialPattern*

O método *TrajectorySequentialPattern* foi elaborado para trabalhar com a busca por padrões sequenciais especificamente dentro de trajetórias. Um padrão sequencial busca encontrar uma lista de acontecimentos que ocorram em sequencia, por exemplo, quais produtos costumam ser comprados sequencialmente em um supermercado. Este algoritmo propõe a procura por padrões deste tipo nas trajetórias de objetos móveis, e pode encontrar resultados como locais que normalmente são visitados em sequencia pelos usuários. Quando encontra um padrão de locais visitados como <Florianópolis, Joinville, Criciúma>, não significa que o usuário tenha visitado somente estes locais, mas que dentro de seu trajeto ele tenha visitado estes locais nesta sequencia, sendo uma trajetória que possui este padrão <Florianópolis, São José, Joinville, Palhoça, Criciúma>.

A implementação original do método no Weka-STPM, recebe como entrada um arquivo na extensão .arff de trajetórias verticais, geradas pelo próprio programa, mas somente as pré-processadas pelo algoritmo IB-SMoT, portanto com as alterações na tabela final gerada pelo RIP-RS em relação ao primeiro, foi necessária também algumas alterações no método *TrajectorySequentialPattern* para que pudesse trabalhar com trajetórias de redes sociais, já processadas pelo método RIP-RS. Apenas algumas alterações de código foram realizadas, mas o método continua funcionando da mesma maneira como o original e gerando os mesmos resultados, sendo estes apresentados na própria tela do weka e se necessário pode ser salvo como um documento de texto simples.

Foram levados para experimentos neste algoritmo, os dois conjuntos de dados já analisados anteriormente na busca por diferenças e semelhanças entre os padrões sequenciais encontrados para os dias de semana e os finais de semana. Todos os testes foram efetuados

com suporte mínimo de 0.03. No primeiro teste considerando apenas os dias da semana foi encontrado um total de 729 padrões frequentes diferentes, sendo que os principais envolvem em conjunto os locais Centro e São José, bem como Centro e Palhoça; Itacorubi, Trindade e Agronômica; Lagoa da Conceição e Itacorubi. Existem ainda outros locais que aparecem em uma grande quantidade de padrões sendo junto com os lugares já citados ou entre si, são eles: Tapera, Córrego Grande, Agronômica e Capoeiras. A Tabela 7 mostra alguns exemplos de padrões envolvendo estes locais.

Tabela 7 - Exemplos de padrões seqênciais - durante a semana

Trajetórias	Suporte
Centro, Capoeiras, Centro, São José	99
Centro, Córrego Grande, Itacorubi	106
Itacorubi, Agronômica, Trindade	92
Lagoa da Conceição, Itacorubi	125
São José, Palhoça, Centro	105

Ao serem levados à análise os dados de final de semana, foram encontrados 89 padrões sequenciais diferentes, um número bastante inferior se comparado aos dias de semana, consequência não só do menor número de dados, mas também da falta de convergência para um local característico no final de semana, como é caso no período de segunda a sexta em que muitas pessoas tendem a se deslocar para os bairros citados acima seja para trabalho ou estudo, tendendo a realizar um deslocamento entre os mesmos locais durante todos os dias da semana. Portanto durante o final de semana não apareceram padrões tão característicos deste tipo de trajetória, aparecendo em muitos padrões os locais já citados apesar de ser em menor escala e alguns padrões novos que fazem referencia a regiões litorâneas.

Concluiu-se ao final destes testes que os trajetos que ocorrem durante a semana, quando processados pelo método *trajectorySequentialPattern*, são capazes de reproduzir as trajetórias de ida vinda do usuário para locais de trabalho ou estudo. Já para os dados de final de semana acaba não sendo tão eficaz devido a própria característica dos dados que conta com muitas variações e pouca previsibilidade em relação aos locais pelos quais um usuário se

desloca nestes dias. O fato de a base de dados ter sido coletada entre os meses de abril e novembro, faz com que não apresentem nenhum dado referente ao período de verão e dificultam análises referentes a trajetórias turísticas, o que seria bastante interessante já que Florianópolis é uma cidade muito popular por atrair turistas no verão.

5.2 EXPERIMENTOS COM O MÉTODO PMO-RS

Tendo sido finalizado o desenvolvimento do método PMO-RS, foram realizados testes e experiências afim de validar os resultados aos quais sua execução havia chegado. Como os cadastros dos usuários no Twitter não apresentam nenhuma informação referente ao endereço da casa do usuário e muito menos a respeito de seu local de ocupação, foi feita avaliação manual dos *tweets* contidos nas tabelas para verificar se era possível identificar através deles se o método havia acertado, errado, ou se realmente não era possível identificar estes locais. Em casos de erros foi feita a tentativa de encontrar uma razão para que o método tenha errado, ficando como sugestões para que em trabalhos futuros os erros possam ser corrigidos.

O processamento das trajetórias, a criação da tabela auxiliar e a finalização do método através da inserção dos registros finais em nova tabela criada no banco de dados durante a execução do método, transcorreram em um tempo total de oito minutos tendo como entrada uma tabela inicial pré-processada pelo método RIP-RS contendo 31.527 registros e considerando para as análises apenas os trajetos formados por um número mínimo de seis postagens e um máximo de três regiões selecionadas como candidatos a serem escolhidos como ocupação e moradia para cada usuário. A Figura 16 mostra um exemplo da visualização do resultado da execução do método PMO-RS para cinco diferentes trajetórias, sendo que os resultados de moradia para cada usuário aparecem na cor rosa e seus respectivos locais de ocupação aparecem na cor branca, sendo identificados pelos mesmos números a dupla que pertence a um mesmo usuário.

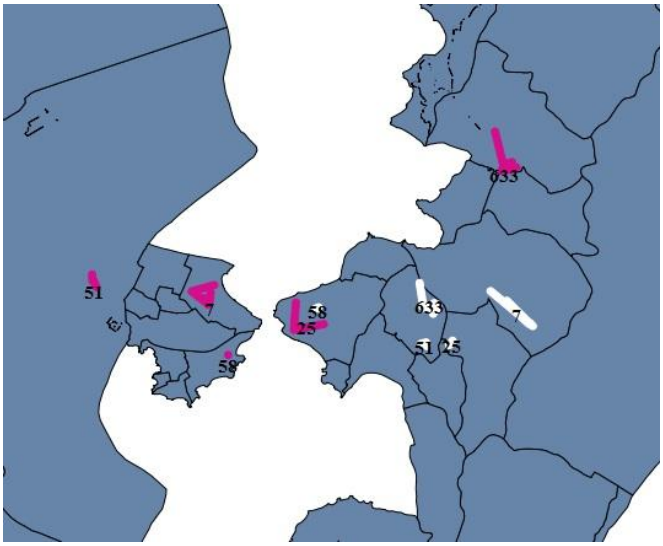

Figura 16 - Exemplo de visualização dos dados resultantes do método PMO-RS

A tarefa de verificação manual das mensagens mostrou-se bastante onerosa e lenta, pois muitos usuários possuem uma média alta de postagens o que dificulta a análise dos textos para identificação do local de moradia e ocupação. Em virtude disto, comprovou-se que seria inviável realizar a busca em todos os registros, dado que a tabela auxiliar criada pelo método, quando executado para um número mínimo de postagens por usuário igual a seis, possui 131.153 registros distribuídos entre 1.675 usuários distintos. Para facilitar os testes foram feitas três seleções de dados sobre os resultados encontrados, a primeira contendo os vinte usuários com maior número de postagens na base dados, a segunda composta por outros vinte usuários com total de postagens entre 160 e 480 e por último foram realizados os testes sobre vinte candidatos com postagens abaixo de 160 mensagens totais.

Para os usuários que possuem o maior número de postagens dentre todas as trajetórias considerando os três locais considerados candidatos a uma das posições, foram selecionados vinte usuário com um número de postagens que variou de 4.734 para o usuário que apresentou o maior número delas e 794 para o que ocupou a vigésima posição. Para estas trajetórias o método se apresentou bastante eficaz, tendo efetivamente acertado o local de moradia para todos os vinte candidatos e para quinze deles acertou também o local de ocupação, a Tabela 8 apresenta o exemplo de dois usuários para os quais o método acertou tanto o local de moradia, quanto o de ocupação.

Tabela 8 - Exemplos de acerto para moradia e ocupação

Usuário	Postagens	Moradia predita	Mensagens
187729514	4734	12 - Saco Grande	"É, agora vou ficar em casa, que saco.";12
			"Odeio quando tem gente aqui em casa, não da pra ficar a vontade..." ;12
		Ocupação predita	
		10 - Trindade	"Hdjxgzjsbsuejdhejhsidhe tedio essa aula.....";10
			"Aula de inglês bem lazy agora.";10
186930878	2209	44- Estreito	"Acho que vou tomar um banho .";44
			"Vou ver se tem alguém aqui em casa";44
		Ocupação predita	
		26 - Jardim Atlântico	"Bom dia , aula de química agora --'";26
			"Aula de religião agora";26

Conforme mostrado na Tabela 8, é possível identificar através das mensagens selecionadas que todas as predições feitas pelo método para estes usuários são verdadeiras e, portanto diagnosticam êxito por parte do algoritmo desenvolvido. Ambos os usuários tem frases bastantes características e que por sua vez não deixam margens para dúvidas em relação ao êxito do método. A Tabela 9 por sua vez apresenta duas situações para as quais não foi possível afirmar pela análise visual das mensagens se o método errou ou acertou a predição do local de ocupação do usuário.

Tabela 9 - Exemplos em que não é possível afirmar se houve êxito na predição de ocupação

Usuário	Postagens	Ocupação predita	Diagnóstico
40349380	794	**13 - são José**	Não é possível ter certeza. Usuário posta mensagens somente do mesmo local. Conteúdo das mensagens não esclarece.
57459615	1859	**13 - são José**	Aparentemente não possui ocupação. Mensagens quase sempre do mesmo local em qualquer horário do dia. Conteúdo das mensagens não esclarece.

Para os usuários 40349380 e 57459615 não é possível afirmar se houve erro ou acerto do método, pois analisando todas as mensagens do primeiro percebe-se que o mesmo utiliza a rede social de apenas um local e neste torna evidente apenas seu local de moradia, pois apesar de ter muitas postagens no horário definido para ocupação não apresenta nenhum texto nas mensagens que seja capaz de caracterizar que o mesmo esteja em seu local de ocupação. Já o segundo usuário aparentemente não tem uma ocupação como trabalho ou escola, pois apresenta postagens em todos os períodos do dia no ponto onde mora e esporadicamente faz

uso da rede em outros locais, mas que também não apresentam características para serem seu lugar de ocupação.

Tabela 10 - Exemplos de erro para predição de ocupação

Usuário	Postagens	Ocupação predita	Diagnóstico
202300068	1558	38 - Itacorubi	**Correto seria 10 - Trindade**
			"Estamos ganhando estrelinhas na aula de matemática, HAIAHAUHAHA";10
112795696	1527	50 - Carianos	**Correto seria 6 - Coqueiros**
			"Primeiro dia de trabalho da sendo legal, to aprendendo muita coisa legaaal";6
60674940	887	45 - Rio Tavares	**Correto seria 53 - Centro**
			Pelas mensagens postadas na região 45 é possível identificar que estuda no Colégio Energia Centro

A Tabela 10 mostra os três casos em que o método errou efetivamente na predição do local de ocupação. Nestes três casos é possível verificar através da análise das mensagens que houve realmente um equívoco como no caso do usuário 202300068. Para o usuário 112795696 é possível que o algoritmo não tenha obtido êxito em virtude de o usuário estar começando a trabalhar durante o período de coleta das mensagens como se pode observar na postagem. Já para o usuário 60674940 o erro ocorre devido ao fato de que todas as postagens do usuário relativas a suas atividades escolares ou ao colégio onde estuda são postadas do bairro Rio Tavares onde é a moradia do usuário e poucas postagens são feitas da região Centro, sendo que nenhuma destas é relacionada a ocupação do usuário.

Conclui-se, portanto que o método é eficiente na predição de locais para trajetórias que são compostas por um número alto de pontos apresentando um resultado final de 100% de acerto para moradia e 75% de acerto para ocupação na amostra selecionada. Além do que é para estes casos que o método é mais útil, pois devido ao alto número de mensagens a análise destes usuários mostra-se extremamente custosa para tentativas de realiza-la manualmente.

A análise seguinte foi realizada com uma amostra aleatória de vinte usuários que tivessem um número total de postagens entre 160 e 480, valores escolhidos por ser o resultado do cálculo feito para selecionar usuários que tivessem uma média diária mínima de uma postagem diária e no máximo três. Para esta amostra o algoritmo permaneceu com média de 100% de acerto para predição de moradia, mas para seis usuários o método fez a predição equivocada do local de ocupação, resultando em 70% de acerto contra 20% de erro, sendo que, dois casos foram caracterizados como não sendo possível afirmar, pois analisando todas

as mensagens dos usuários, ainda assim não é possível identificar o local de ocupação do mesmo.

Na análise final considerando apenas os usuários com uma média diária de mensagens inferiores a um, percebeu-se que o método dificilmente consegue prever a moradia e a ocupação destes usuários, mas notou-se também que isso não representa necessariamente um erro, pois muitos destes usuários não residem e também não trabalham na localidade, muitos são turistas ou estão em viajem de trabalho pela cidade. Portanto, conclui-se que o PMO-RS não é recomendado para fazer predições sobre trajetórias de usuários que utilizam pouco a rede social, pois primeiramente não são capazes de serem representativos para aplicações direcionadas a este meio, afinal se não utilizam diariamente a rede não serão alvos de uma aplicação voltada para este tipo de público, além de como já foi mencionado dependendo do limite geográfico da aplicação, como neste caso a microrregião de Florianópolis, pode haver trajetórias que não são de moradores da região e como o algoritmo foca na predição deste dado especificamente acaba falhando nestas trajetórias.

Por fim foi realizada uma análise a partir das informações que podem ser extraídas a partir dos resultados da execução do método, não buscando testar os erros e acertos do algoritmo, mas sim a utilização das informações geradas na busca de que algum conhecimento possa ser extraído do conjunto resultante, analisando-os de forma geral nesta fase e não individualmente como é o caso da predição do local de moradia e ocupação. Foi definido como escopo nestes testes finais, analisar os trajetos mais comuns dentro da microrregião de Florianópolis e que podem ser descritos através das redes sociais, para isso foram utilizadas as divisões regionais da cidade: Norte, Sul e Leste, além da região Central, que foi definida para estas análises como sendo composta pelos bairros: Centro, Itacorubi, Trindade, Agronômica, José Mendes, Córrego Grande e Pantanal.

Foi feita a seleção de uma amostra com 511 usuários que residem e têm ocupação em locais diferentes, segundo predição do método PMO-RS. Através desta análise observou-se que o método pode retratar a característica da cidade de que a grande maioria das pessoas tem seus locais de ocupação na região Central, afinal é nesta que se encontram a grande maioria das empresas, além de Universidades e também um grande número de colégios. A Tabela 11 mostra os cinco principais fluxos entre moradia e ocupação que ocorrem dentro da microrregião da grande Florianópolis e através dele é possível perceber que todos estes fluxos resultam em local de ocupação na região Central.

Tabela 11 - Cinco princípais fluxos de moradia e ocupação

Moradia	Ocupação	%
Região Central	Região Central	18,20%
Outras cidades	Região Central	11,94%
Região Continental	Região Central	8,41%
Região Norte	Região Central	8,22%
Região Sul	Região Central	7,63%

O principal destaque apresentado na Tabela 11 está no grande número de pessoas que além de manterem seus locais de ocupação na região Central também optam por residir mais próximo de seus lugares de ocupação, em virtude disto aparecendo como percentual mais alto da tabela, o daqueles que moram e trabalham nesta região. Este fato evidencia também a dependência que os demais locais ainda apresentam da região Central de Florianópolis, afinal tanto a região Norte, quanto a Sul e a Leste da cidade, apresentam um pequeno percentual de pessoas que residem e mantém seu local de ocupação dentro da própria região de moradia, evidenciando o fato de que a maior parte da população destas regiões precisa transitar diariamente até um dos bairros da região Central da cidade. Vale resaltar aqui, que a região Norte apresentou a maior percentagem dentre as três regiões para pessoas que trabalham e residem dentro de seus limites.

Outra característica interessante que pode ser diagnosticada na análise dos dados foi o trajeto entre o continente e a região central da cidade, onde se somado o percentual da região continental ao das outras cidades pertencentes à microrregião, tem-se um valor de 20,35% do total de fluxos no sentido <Continente - moradia, Centro - ocupação> e mais 7,63 % no sentido inverso. Este fato é capaz de retratar o quanto à população das cidades próximas a Capital ainda dependem desta, além de realçar o fato, perceptível na realidade, que um grande número de moradores da Ilha vem fazendo um fluxo diário no sentido contrário, devido ao crescimento que cidades como São José e Palhoça vêm apresentando nos últimos anos, com grandes empresas e Universidades sendo instaladas nestas regiões.

Além de descrever os fluxos mais comuns entre o local de moradia e ocupação dos usuários dentro da região, é possível através do resultado da execução deste algoritmo calcular a distancia que cada usuário percorre entre estes dois locais através de uma simples consulta em SQL. É possível ainda apresentar informações como as distancias médias que os usuários que possuem ocupação na região central costumam percorrer para chegar de suas casas até este local, entre outras informações úteis que podem ser extraídas destes dados.

Como foi possível observar o método PMO-RS além de predizer o local de moradia e o de ocupação, é capaz de fornecer dados úteis para que informações adicionais, relacionadas ao fluxo diário de uma cidade ou região, possam ser extraídas de forma mais simples com base em seus resultados. Conforme mostrado na análise anterior essas informações apresentam um alto grau de confiabilidade sendo capazes de traduzir com êxito a realidade apresentada para a região em análise.

6 CONCLUSÃO E TRABALHOS FUTUROS

A crescente utilização de dispositivos móveis capazes de gerar dados espaço-temporais e principalmente a maior acessibilidade da população a equipamentos como celulares e computadores que são capazes de gerar este tipo de dados, bem como a grande utilização das redes sociais a partir destes dispositivos, ocasionou a geração de uma grande quantidade de dados espaço-temporais advindos de redes sociais. Esses dados agrupados podem ser uma fonte de informação, mas sua análise ainda é algo bastante novo e este trabalho vem como uma proposta de pesquisa exploratória dos dados de uma rede social específica chamada Twitter, visando além de estudar os dados, propor novos algoritmos capazes de extrair algum tipo de conhecimento de bases de dados espaço-temporais oriundas de redes sociais.

Os estudos sobre a base de dados puderam diagnosticar e reafirmar a tendência de que os dados das redes sociais associados às informações espaço-temporais retratam o movimento real das pessoas, tendo esta característica o poder de transformar uma rede social em algo extremamente útil do ponto de vista mercadológico e científico. Conforme já mencionado no decorrer deste trabalho, esta realidade retratada no mundo virtual é capaz de auxiliar diversos setores, como: facilitar a publicidade e a propaganda focada neste tipo de público, a utilização destes dados para auxiliar algum órgão ou setor da sociedade a encontrar um individuo, servir como insumos para pesquisas referentes ao fluxo de pessoas dentro de uma cidade ou país, entre outras várias formas de aplicações para as quais estes dados podem servir.

O primeiro método desenvolvido ao longo deste estudo, denominado RIP-RS, foi capaz de realizar com sucesso seu objetivo de fazer a redução de pontos da trajetória afim de facilitar a manipulação dos dados, através da adição de semântica às trajetória. O RIP-RS funciona com muita semelhança ao IB-SMoT, sendo este focado na análise de trajetórias de objetos móveis comuns enquanto a extensão implementada neste estudo é focada nas redes sociais, apresentando o mesmo nível de efetividade e sucesso encontrado no primeiro. Com a inserção do RIP-RS no Weka-STPM, esta versão do software passa a disponibilizar o processamento de dados originários de redes sociais, uma função ainda não existente na ferramenta.

O método PMO-RS, surgiu a partir da análise dos dados e foi colocado em prática a partir da efetividade de que os dados podem retratar com certo nível de confiança a realidade.

Este método se mostrou extremamente eficaz nas predições de local de moradia e de ocupação feitas para trajetórias com um número grande de pontos, encontrando certo nível de dificuldade em trajetórias com pequeno número de pontos. O algoritmo também se mostrou mais efetivo quando tenta predizer o local de moradia do que o de ocupação, isso porque as pessoas tendem a apresentar um comportamento mais parecido na primeira situação do que na segunda, além do que, a grande maioria dos usuários faz postagens de sua residência, já no local de trabalho ou estudo não são todos que utilizam a rede social, portanto muitas vezes não é mesmo possível identificar esta segunda situação devido ao fato de que a pessoa não faz postagens na rede a partir deste tipo de localidade.

Para trabalhos futuros sugere-se que a base de dados possa ser ainda mais explorada, para que se possam ser encontradas possibilidades de mais algoritmos para extração de conhecimento destes dados. Para o algoritmo já implementado PMO-RS fica a sugestão de aprimorar a questão da predição de local de ocupação, buscando novas variáveis que possam aumentar a percentagem de acerto do método para esta variável.

7 REFERÊNCIAS

AGRAWAL, R; SRIKANT, R, 1994. **Fast algorithms for mining association rules in large databases**. In VLDB, pages 487 – 499.

ALVARES, Luis Otávio. **Descoberta de Conhecimento espaço-temporal em Redes Sociais na Internet.** Florianópolis: 2011.

ALVARES, L.O., BOGORNY, V., Kuijpers, B., MACEDO, J.A.F., MOELANS, B. **A model for enriching trajectories with semantic geographical information**. In GIS'07: Proceedings of the 15th Annual ACM International Symposium on Advances in Geographic Information Systems, New York, NY, USA, 2007. ACM Press.

ANDERSON, Mattias; GUDMUNSSON, Joaquim; LAUBE, Patrick and WOLLE, Thomas. **Reporting Leaders and Followers Among Trajectories of Moving Point Objects, Geoinformática (2008),** Volume 12, Number 4, 497-528, DOI: 10.2007/s10707-007-0037-9

BACKSTROM, L., SUN, E., and MARLOW, C. 2010. **Find me if you can: improving geographical prediction with social and spatial proximity.** In Proceedings of the 19th international Conference on World Wide Web

BOGORNY, V; PALMA, A; ENGEL, P; ALVARES, L.O. **Weka-GDPM: Integrating Classical Data MInig Tooltik to Geographic Information Systems**. In: SBBD Workshop on Data Mining Algorithms and Aplications (WAAMD'06), Florianópolis, Brazil, October 16-20, (2006), pages 9 – 16

CHIECHELSKI, G. O; BOGORNY, V., **Uma Extensão do PostGIS para a Geração Automática de Trajetórias Semânticas.** 61 f. Trabalho de Conclusão (Graduação) - Instituto de Informática e estatística da UFRGS, 2008.

DENG, Dong-Po., CHUANG, Tyng-Ruey. and LEMMENS, Rob. 2009. **Conceptualization of place via spatial clustering and co-occurrence analysis**. In Proceedings of the 2009 International Workshop on Location Based Social Networks (LBSN '09). ACM, New York, NY, USA, 49-56.

ELMASRI, R.; NAVATHE, S. B. **Sistemas de Banco de Dados.** 4 ed. São Paulo: Pearson Addison Wesley, 2005.

FAJARDO, B. S. **Uma extensão nativa do SQL para mineração de trajetórias semânticas.** 60 f. Trabalho de Conclusão (Graduação) - Curso de Ciência da Computação, Instituto de Informática e Estatística da UFRGS, Porto Alegre, 2008.

FAYYAD, U.M., PIATETSKY-SHAPIRO, G. and SMYTH, P. (1996) **"From data mining to knowledge discovery: An overview"** in U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth

and R. Ulthurusamy (Eds.) **Advances in Knowledge Discovery and Data Mining**, Cambridge, MA: MIT Press, 1–34.

FUJIKASA, T.; LEE, R. and SUMIYA, K. **Exploring Urban Characteristics Using the Movement History of Mass Mobile Microbloggers.** The Eleventh Workshop on Mobile Computing Systems and Applications (HotMobile2010), February 2010

_____. **Monitoring Geo-Social Activities through Micro-Blogging Sites.** Workshop on Social Networks and Social Media Mining on the Web (SNSMW), April 2010.

_____. **Detection of Unusually Crowded Places through Micro-Blogging Sites**. The 6th International Symposium on Web and Mobile Information Services (WAMIS 2010), April 2010

GIANNOTTI, F.; NANNI, M. AND PEDRESCHI, D. **Efficient mining of sequences with temporal annotations**. In Proc. SIAM Conference on Data Mining, pages 346 – 357. SIAM, 2006.

HAN, J.; MILLER, Harvey J. **Geographic Data Mining and Knowledge Discovery.** 2 ed. CRC Press, 2009.

KWAK, H., LEE, C., PARK, H., and MOON, S. 2010. **What is Twitter, a social network or a news media?**. In Proceedings of the 19th international Conference on World Wide Web (Raleigh, North Carolina, USA, April 26 - 30, 2010). WWW '10. ACM, New York, NY, 591-600.

LAUBE, P.; IMFELD, S., 2002,**Analyzing relative motion within groups of trackable moving points objects**. In Egenhofer, M. J. and Marks, D. M., editors, GIScience, volume 2478 of Lecture Notes in Computer Science, pages 132 -144.

LAUBE, P.; VAN K.; M., IMFELD, S.; **Finding remo - detecting relative motion patterns in geospatial lifelines**. In Fisher, P.F., ed.: Developments in Spatial Data Handling. Proceedings of the 11th International Symposium on Spatial Data Handling. Springer, Berlin Heidelberg, DE (2004) 201–214.

LAUBE, P. ; IMFELD, S.; WEIBEL, R. **Discovering relative motion patterns in groups of moving point objects**, in: International Journal of Geographic Information Science, vol.19, Taylor & Francis Goup, pp. 639-668, 2005.

LEE, R.; WAKAMIYA, S. and SUMIYA, K. **Discovery of unusual regional social activities using geo-tagged microblogs.** World Wide Web Journal. 2011.

LEE, R. and SUMIYA, K. **Measuring geographical regularities of crowd behaviors for Twitter-based geo-social event detection.** In Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks (LBSN '10). ACM, New York, NY, USA, 1-10

MANSO, J. A; TIMES, V. C.; OLIVEIRA, G.; ALVARES, L.O.; BOGORNY, V. **DB-SMoT: a Direction-based spatio-temporal clustering method.** Fifth IEEE International Conference on Intelligent Systems (IEEE IS 2010), 2010.

PALMA, A.T., BOGORNY, V. and ALVARES, L.O., 2008, **A Clustering-based approach for discovering interesting places in trajectories.** In Proceedings of the ACMSAC, Fortaleza, Brazil (New York, NY, USA: ACM Press).

RAMAKRISHNAN, Raghu; GEHRKE, Johannes. **Sistemas de Gerenciamento de Banco de Dados.** 3 ed. São Paulo: McGraw-Hill, 2008.

SPACCAPIETRA, S.; PARENT, C.; DAMIANI, M.L.; MACEDO, J. A. F. de ; Porto, F.; Vangenot, C.; **A conceptual view on trajectories, Data & Kownledge Egineering**, 65 n.1, p126-146, April 2008.

Referências na web

Imagens de padrões geométricos. Disponível em URL: <**http://movementpatterns.pbworks.com/w/page/21692527/Patterns%20of%20Movement**>. Acessado em abril de 2012.

Documentação sobre o aplicativo Quantum GIS, versão 1.4.0. Disponível em URL: <**http://www.qgis.org/pt/documentation/manuals.html**>. Acessado em março de 2012

Estatísticas do Twitter da agência Monkey Business. Disponível em URL: **<http://www.slideshare.net/mkbusiness/twitter-stats-2011> .** Acessado em outubro de 2012

Estatísticas do Twitter da empresa Semiocast. Disponível em URL: <**http://semiocast.com/publications/2012_07_30_Twitter_reaches_half_a_billion_accounts_140m_in_the_US**>. Acessado em outubro de 2012.

Site do postGIS. Disponível em URL: **<http://postgis.refractions.net/**> . Acessado em março de 2012

APÊNDICE 01 – ARTIGO

Descoberta de Conhecimento Espaço-Temporal Através da Análise das Trajetórias dos Usuários da Rede Social *Twitter.*

Janaína Oliete de Siqueira[1], Luis Otavio Alvares[1]

[1]Instituto de Informática e Estatística – Universidade Federal de Santa Catarina (UFSC)
Florianópolis – SC – Brasil

{janainasiqueira, alvares}@inf.ufsc.br

Abstract. *The growing popularity of the internet has meant that there was a huge increase in the number of people who use social networks every day. This work focuses on the study of the trajectories of social network Twitter users and two algorithms are described and evaluated using real data. The first sequence shows the regions where the user tweeted and the second identifies the place of residence and occupation of the user.*

Resumo. *A crescente popularização da internet fez com que houvesse um enorme aumento no número de pessoas que utilizam as redes sociais cotidianamente. Este trabalho foca no estudo das trajetórias dos usuários da rede social Twitter e dois algoritmos são descritos e avaliados com dados reais. O primeiro mostra sequências das regiões de onde o usuário Tuitou e o segundo identifica o local de moradia e ocupação do usuário.*

1. INTRODUÇÃO

A busca por automação e sistematização dos processos empresariais, assim como, a facilidade de informatização que vem acontecendo na última década, gerou um grande aumento do volume de dados armazenados em bancos de dados sejam eles empresariais, governamentais, pessoais ou comerciais. Com esse acúmulo de informações percebeu-se a oportunidade de utilizá-las de forma estratégica, de maneira a auxiliar os gestores nas tomadas de decisões em suas organizações.

Neste âmbito, a área de mineração de dados ganhou ênfase nos esforços de pesquisa e desenvolvimento no ramo da computação. "A mineração de dados consiste em encontrar tendências ou padrões interessantes em grandes conjuntos de dados para orientar decisões sobre atividades futuras" (RAMAKRISHNAN; GEHERKE, 2008, p.737). A mineração de dados vem evoluindo e tomando grandes proporções e sua crescente aplicação deu ênfase a uma área denominada descoberta de conhecimento que é mais ampla que a mineração de dados. "O processo é composto por seis fases: seleção de dados, limpeza, enriquecimento, transformação ou codificação, mineração de dados e construção de relatórios e apresentação das informações descobertas" (ELMASRI; NAVATHE, 2005, p. 624).

A descoberta de conhecimento considerando dados geográficos (ou espaciais) se diferencia da descoberta em dados tradicionais principalmente por considerar os relacionamentos espaciais entre os dados (topológicos, de distância e ordem) (GÜTING, 1994 apud ALVARES, 2011, p. 2). Esta área, portanto é caracterizada como uma linha de pesquisa

que caminha paralelamente ao KDD tradicional, já que os relacionamentos espaciais não ficam armazenados nos bancos de dados.

Como a disponibilização dos dados espaço-temporais de usuários nas redes sociais é algo recente, os estudos de técnicas e a construção de algoritmos destinados especificamente a manipular e extrair conhecimento espaço-temporal das redes sociais ainda se encontra em estado emergente. Este trabalho pretende realizar a análise dos algoritmos de mineração de dados voltados para trajetórias comuns, a fim de verificar se os mesmos são aplicáveis às trajetórias, atípicas, geradas pelas redes sociais, identificando a necessidade de alterações ou eventualmente a criação de novos algoritmos para trabalhar com este tipo de trajetória. A correta exploração deste tipo de dados proporcionará a extração de conhecimento real, novo e importante a respeito das redes e seus usuários, que poderão futuramente auxiliar os mais diversos setores econômicos, de pesquisa e governamentais, dentre muitas outras áreas.

2. LEVANTAMENTO BIBLIOGRÁFICO

As redes sociais existem desde a antiguidade e tem acompanhado a evolução das tecnologias comunicativas, indo desde a escrita até a internet, gerando aquilo que atende hoje por rede social online, comumente chamada apenas de redes sociais, com exemplos consolidados como o caso do Facebook, do Orkut e do Twitter. Uma rede social é estruturada através das pessoas e seus relacionamentos e são espaços que servem para manter relações pessoais e profissionais, bem como criar novas relações.

Em Santos (2011) o estudo dos algoritmos de mineração de dados geográficos em redes sociais foi dividido em dois tipos de análises: estática e semântica. No caso da análise estática trabalha-se com dados geográficos que não se alteram ao longo do tempo, como localização de moradia de um usuário, que na maioria dos casos não se alteram ao longo do tempo. A análise dinâmica, apesar de ser um tipo de análise mais complexa pode gerar resultados mais ricos quanto ao ganho de conhecimento, pois neste caso as informações geográficas estão associadas às informações temporais. São exemplos deste tipo de dados as trajetórias percorridas pelos usuários, que serão mais exploradas no desenvolvimento deste trabalho.

Atualmente já existem alguns métodos para mineração de dados espaciais em redes sociais que utilizam a análise estática. Um dele foi desenvolvido pela equipe do Facebook e denomina-se *Find Me If You Can* (BACKSTROM et al., 2010) e realiza a predição da cidade de moradia do usuário, através de seu relacionamento dentro da rede utilizando algum conhecimento da geografia do local. O algoritmo utilizou as medidas de distância e densidade como métrica.

A segunda pesquisa de destaque foi realizada em (MISLOVE et al. 2010) e reproduzida em (BIEVER, 2010), utilizando os dados de localização dos usuários do Twitter. Neste caso a informação geográfica foi associada a palavras dentro dos *posts* do usuário que identificassem seu estado de humor. Foram utilizados cerca de 300 milhões de *tweets* dos Estados Unidos, em um período de aproximadamente três anos.

As análises dinâmicas existentes, em geral, estudam o comportamento dos usuários e possibilitam a descoberta de conhecimento mais elaborado e padrões mais significativos. Em 2010 (YE et al., 2010) foi realizada uma pesquisa que trabalha com a idéia de recomendação de lugares, que utiliza os conceitos de confiança e similaridade de interesses, extraído dos relacionamentos. Em (DENG et al. , 2009) foi realizado um estudo que analisa *tags* e *geotags*, para explorar o significado que os usuários dão para localidades.

Utilizando as características de tempo real do Twitter, em Sakaki et al. (2010) é feita a utilização das mensagens e usuários do Twitter, como sensores de eventos em tempo real. Para realização de testes desse algoritmo foram utilizados apenas dados de usuários do Japão e busca-se estudar o evento natural dos terremotos.

Um dos estudos mais recente foi realizado sobre dados do Twitter e é descrito em (LEE et al. 2011), sendo que este trabalho foi apresentado em quatro outros artigos, todos apresentando uma evolução em relação ao anterior. Fujikasa et al. (2010a) apresenta os principais conceitos sobre o que seria a proposta final do método para análise e extração de regiões com atividades fora do padrão esperado, Fujikasa et al.(2010b) constata a importância na análise do conteúdo das mensagens para traduzir emoções e pensamentos dos usuários e facilitar a detecção do tipo de evento que disparou a atividade e em Fujikasa et al. (2010c) define-se os argumentos de pontos como regiões de interesse, culminado no trabalho final de Lee, onde é feita a evolução dos conceitos tratados nos trabalhos anteriores e é realizada a definição de três métricas que devem ser utilizada para identificar determinado tipo de evento.

3. ALGORITMOS PROPOSTOS

A pesquisa exploratória realizada nos dados evidenciou que o comportamento dos usuários nas redes sociais dá origem a um tipo particular de trajetória, com importantes características distintas e para as quais os métodos já desenvolvidos não estão preparados. Portanto, ao longo da pesquisa o foco esteve em procurar formas de tornar estes dados mais simples de serem minerados, devido ao grande volume em que são gerados e ainda descobrir que tipo de informações interessantes poderiam ser extraídas dos mesmos.

Com base nisto, foram desenvolvidos dois métodos capazes de trabalhar com este novo modelo de trajetória. O primeiro deles, detalhado na sessão 3.1, recebeu o nome de Regiões de Interesse de Postagem em Redes Sociais e é uma adaptação de um método já existente, que tem o foco na fase de pré-processamento dos dados. Já o segundo é um algoritmo novo que objetiva efetivamente a extração de conhecimento através do processamento das trajetórias, que será detalhado na sessão 3.2.

3.1 Regiões de Interesse de Postagem em Redes Sociais (RIP-RS)

O RIP-RS é um método que a partir de um conjunto de *tweets* de um usuário, gera uma sequência de regiões nas quais as mensagens foram emitidas. A granularidade da região é definida pelo usuário e pode ser, por exemplo, País, Estado, Município, Bairro, etc. Além disto, o método é responsável por fazer uma redução de pontos ao longo da trajetória, baseado nas regiões de postagens e que objetiva preparar os dados de forma a facilitar a etapa de mineração.

O método recebe como entrada uma tabela com o conjunto de *tweets*, outra com os candidatos a regiões de interesse e um atributo chamado *buffer*. Inicialmente o método verifica se a tabela é composta por trajetórias de rede social. Se for, seleciona a primeira das trajetórias e faz o teste de intersecção de cada um dos pontos do trajeto com os candidatos a regiões de interesse, todas as regiões que possuem pelo menos uma postagem também são consideradas como região de interesse. O algoritmo faz isso ciclicamente até a última trajetória contida na tabela. Todo o resultado da execução do método é armazenado em uma tabela no mesmo banco de dados da tabela original, conforme já descrito anteriormente.

3.2 Predição de Moradia e Ocupação em Redes Sociais (PMO-RS)

Em uma análise exploratória no conteúdo das mensagens postadas mostrou-se bastante claro que muitos dos usuários costumam postar mensagens de sua casa, trabalho, local de estudo, além de eventuais lugares visitados durante passeios, e estas informações são capazes de descrever o fluxo diário das pessoas.

Baseado na utilidade que pode ter o conhecimento adquirido, aliado as informações que são possíveis extrair da base de dados, foi proposto o desenvolvimento de um algoritmo capaz de identificar de maneira probabilística os locais de moradia e ocupação dos usuários da rede, baseado na análise de suas trajetórias e ainda no conteúdo de suas postagens, o algoritmo foi denominado método de Predição de Moradia e Ocupação em Redes Sociais (PMO-RS).

O método trabalha com postagens feitas durante a semana, excluindo as mensagens geradas nos finais de semana, dado que em geral as pessoas não trabalham ou estudam nos finais de semana e costumam transitar por locais diferentes. Além disto, para servir de entrada para o PMO-RS a tabela precisa ter sido pré-processada pelo método RIP-RS já descrito anteriormente.

Para predizer tanto o local de moradia quanto o de ocupação, o método faz a utilização de três métricas diferentes, sendo elas frequência de postagem das mensagens, horário das mensagens e semântica das mensagens, para simplificar denominadas respectivamente: #freqMsg, #horarioMsg e #semanticaMsg. A métrica #freqMsg faz a seleção das regiões que serão candidatas a local de ocupação e moradia que o método busca encontrar, seleciona um determinado número de regiões, que pode variar entre duas e cinco, que tenham o maior número de postagens dentre todas as regiões das quais um usuário já tenha postado pelo menos uma vez. A métrica #horarioMsg serve para realizar a contagem do número de mensagens enviadas entre dois intervalos de tempo pré-determinados, sendo eles: das 08:00h às 19:59h para ocupação e das 20:00 às 07:59h para moradia.

Para evitar erros advindos de usuários que não pertençam ao perfil geral utilizado como base para a definição dos intervalos de horário que foram definidos para apontar prováveis locais de moradia e ocupação baseados nesta variável, foi feita a opção de utilizar a avaliação semântica da própria mensagem através da métrica #semanticaMsg, tendo em vista que muitas postagens são capazes de retratar perfeitamente ou indicar onde o usuário está. Após várias tentativas, chegou-se a uma lista final de palavras, apresentadas na Tabela 1, que foram efetivamente utilizadas no algoritmo e que indicam com maior grau de certeza onde o usuário está.

Tabela 1 - Listagem de palavras - métrica #semanticaMsg

Palavras Moradia	Palavras Ocupação
"em casa"	" trabalhar" && !(ir trabalhar)
"home sweet home"	"work"
"lar doce lar"	"essa aula"
"dormi"	"m at unisul"
" at minha casa"	"m at ufsc"
" tv" && !(", tv")	"m at univali"
"@ my home"	"facul"
"banho"	"m at colégio"
"morando aqui"	"@ colégio"
"m at condomínio"	"campus"
"@ condomínio"	"instituto"
"m at residencial"	"m at escola"
"@ residencial"	"empresarial"
	"aula % agora"

Em geral a métrica #horarioMsg apresenta valores de contagem bem maiores do que #semanticaMsg. Em função disso, um método que apenas somasse os resultados das duas métricas não seria eficaz. Como foi concluído que nenhuma das duas métricas tinham tanta superioridade em relação à outra foi feita a opção de normalizar estes dados para valores entre 0 e 1, para fins de igualar as duas métricas na análise. Esta normalização foi realizada calculando a proporção dos dados em relação ao total daquela variável. O método seleciona como local de ocupação daquele usuário, a região candidata que obtiver o maior valor no somatório de #freqMsg e #semanticaMsg e o mesmo ocorre para o local de moradia.

Para armazenar todo o resultado encontrado após o processamento do método PMO-RS é criada uma tabela no banco de dados, que contém o identificador do usuário e os dados referentes à moradia e à ocupação, sendo eles nome do local, identificador do local, proporção e número de palavras encontradas e de mensagens no horário da função para a qual foram escolhidos (moradia ou ocupação), além do *the_geom* que é a geometria do polígono resultante da junção dos pontos (coordenadas dos *tweets*) que possuem as características de moradia ou ocupação. A Figura 1 apresenta a tela desenvolvida para trabalhar com a exploração de dados de redes sociais, inicialmente possuindo apenas o método PMO-RS.

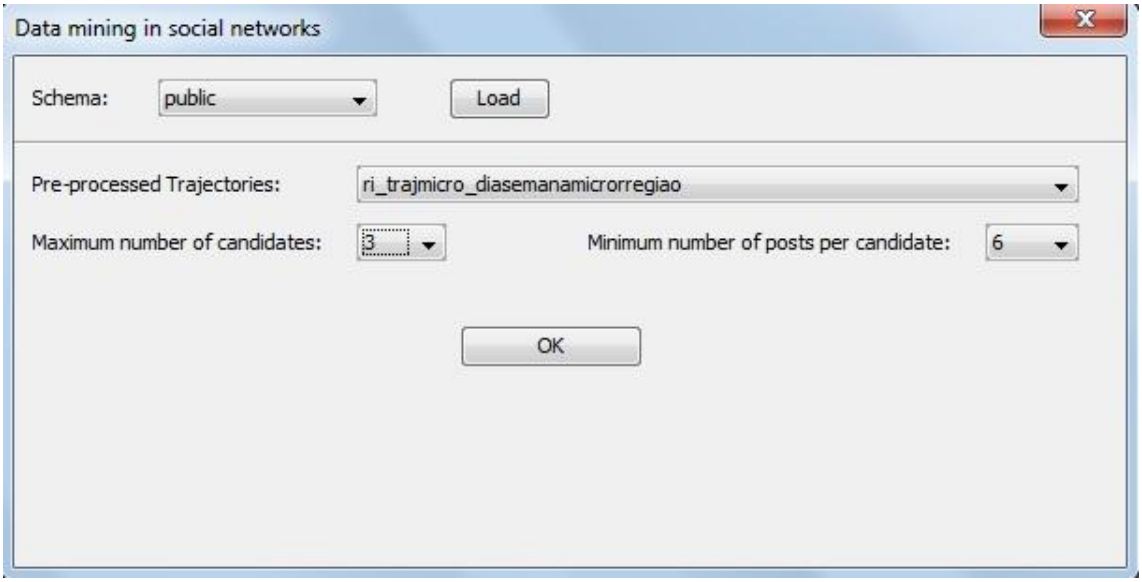

Figura 17 - Tela de Mineração de Redes Sociais

4. EXPERIMENTOS E RESULTADOS

Para verificar a efetividade e os resultados dos algoritmos desenvolvidos neste estudo, tanto RIP-RS quanto PMO-RS, foram realizados alguns experimentos com a base de dados de postagens da rede social Twitter, utilizando os dados gerados no período de oito meses, de abril à novembro de 2011, que tenham sido postados dentro da microrregião de Florianópolis.

As primeiras experiências realizadas com o método RIP-RS, tem o objetivo de avaliar se realmente houve redução de pontos e se essa redução não prejudicou a trajetória. Inicialmente foi feita a execução do método, buscando a intersecção dos locais de postagem com a tabela de pontos de interesse gerada para esta região. A tabela de postagens da microrregião utilizada como entrada para o algoritmo no primeiro teste realizado continha um total de 212.085 registros e a tabela de candidatos a *Relevant Features* possui um total fixo de 53 registros.

Após a execução do RIP-RS a tabela de trajetórias resultante totalizou 45.270 registros, demonstrando que o método foi capaz de fazer uma boa diminuição no número de registros da tabela, apresentando como resultado uma tabela com aproximadamente 21,34% do número de registros da original, além do que consultas como: soma do número de postagens por bairros ou número de vezes que uma região de interesse de postagem aparece dentre todas as trajetórias, tornam-se extremamente simples com a nova tabela, sendo a primeira capaz de ser executada em 157ms e a segunda em 151ms.

Dado que os resultados apresentados pelo RIP-RS podem ser úteis e confiáveis, foram feitas algumas análises de cunho estatístico em cima desses dados encontrados na tabela final do método. Primeiramente foi realizada uma análise comparativa entre as postagens durante a semana e nos finais de semana, na qual foi possível concluir que as redes sociais são capazes de retratar o fluxo real de pessoas, apresentando resultados como: aumento do percentual que as regiões de interesse Trindade e Itacorubi apresentam nos dias de semana e o declínio apresentado nos finais de semana. Enquanto nas regiões litorâneas da Ilha como: Jurere, Ingleses, Campeche e Ribeirão da Ilha ocorre o contrário, aumento percentual no final de semana. Outros testes foram realizados com base no resultado do RIP-RS, como a busca pelos padrões sequencias mais comuns nestes dados, sendo estes testes capazes também de retratar o fluxo real de movimentação dentro da cidade.

Para o método PMO-RS foi feita avaliação manual dos *tweets* contidos nas tabelas para verificar se era possível identificar através deles se o método havia acertado, errado, ou se realmente não era possível identificar os locais de moradia e ocupação dos usuários. Os testes foram divididos em três etapas.

O primeiro teste ocorreu para os usuários que possuem o maior número de postagens dentre todas as trajetórias, sendo selecionados vinte usuários com um número de postagens que variou de 4.734 para o usuário que apresentou o maior número delas e 794 para o que ocupou a vigésima posição. Concluiu-se que o método é eficiente na predição de locais para trajetórias que são compostas por um número alto de pontos apresentando um resultado final de 100% de acerto para moradia e 75% de acerto para ocupação na amostra selecionada. Além do que é para estes casos que o método é mais útil, pois devido ao alto número de mensagens a análise destes usuários mostra-se extremamente custosa para tentativas de realiza-la manualmente.

A análise seguinte foi realizada com uma amostra aleatória de vinte usuários que tivessem um número total de postagens entre 160 e 480, valores escolhidos por ser o resultado do cálculo feito para selecionar usuários que tivessem uma média diária mínima de uma postagem diária e no máximo três. Para esta amostra o algoritmo permaneceu com média de 100% de acerto para predição de moradia, mas para seis usuários o método fez a predição equivocada do local de ocupação, resultando em 70% de acerto contra 20% de erro, sendo que, dois casos foram caracterizados como não sendo possível afirmar, pois analisando todas as mensagens dos usuários, ainda assim não é possível identificar o local de ocupação do mesmo.

Na análise final considerando apenas os usuários com uma média diária de mensagens inferiores a um, percebeu-se que o método dificilmente consegue prever a moradia e a ocupação destes usuários, mas notou-se também que isso não representa necessariamente um erro, pois muitos destes usuários não residem e também não trabalham na localidade, muitos são turistas ou estão em viajem de trabalho pela cidade. Portanto, conclui-se que o PMO-RS não é recomendado para fazer predições sobre trajetórias de usuários que utilizam pouco a rede social, pois primeiramente não são capazes de serem representativos para aplicações direcionadas a este meio, afinal se não utilizam diariamente a rede não serão alvos de uma aplicação voltada para este tipo de público, além de como já foi mencionado dependendo do limite geográfico da aplicação, como neste caso a microrregião de Florianópolis, pode haver trajetórias que não são de moradores da região e como o algoritmo foca na predição deste dado especificamente acaba falhando nestas trajetórias.

Por fim foi realizada uma análise a partir das informações que podem ser extraídas a partir dos resultados da execução do método, não buscando testar os erros e acertos do algoritmo, mas sim a utilização das informações geradas na busca de que algum conhecimento possa ser extraído do conjunto resultante, analisando-os de forma geral nesta fase e não individualmente como é o caso da predição do local de moradia e ocupação. Foi definido como escopo nestes testes finais, analisar os trajetos mais comuns dentro da microrregião de Florianópolis e que podem ser descritos através das redes sociais, para isso foram utilizadas as divisões regionais da cidade: Norte, Sul e Leste, além da região Central, que foi definida para estas análises como sendo composta pelos bairros: Centro, Itacorubi, Trindade, Agronômica, José Mendes, Córrego Grande e Pantanal.

Foi feita a seleção de uma amostra com 511 usuários que residem e têm ocupação em locais diferentes, segundo predição do método PMO-RS. Através desta análise observou-se que o método pode retratar a característica da cidade de que a grande maioria das pessoas tem

seus locais de ocupação na região Central, afinal é nesta que se encontram a grande maioria das empresas, além de Universidades e também um grande número de colégios.

Como foi possível observar o método PMO-RS além de predizer o local de moradia e o de ocupação, é capaz de fornecer dados úteis para que informações adicionais, relacionadas ao fluxo diário de uma cidade ou região, possam ser extraídas de forma mais simples com base em seus resultados. Conforme mostrado na análise anterior essas informações apresentam um alto grau de confiabilidade sendo capazes de traduzir com êxito a realidade apresentada para a região em análise.

5. CONCLUSÃO E TRABALHOS FUTUROS

Os estudos sobre a base de dados puderam diagnosticar e reafirmar a tendência de que os dados das redes sociais associados às informações espaço-temporais retratam o movimento real das pessoas, tendo esta característica o poder de transformar uma rede social em algo extremamente útil do ponto de vista mercadológico e científico. Esta realidade retratada no mundo virtual é capaz de auxiliar diversos setores, como: facilitar a publicidade e a propaganda focada neste tipo de público, a utilização destes dados para auxiliar algum órgão ou setor da sociedade a encontrar um individuo, servir como insumos para pesquisas referentes ao fluxo de pessoas dentro de uma cidade ou país, entre outras várias formas de aplicações para as quais estes dados podem servir .

O primeiro método desenvolvido ao longo deste estudo, denominado RIP-RS, foi capaz de realizar com sucesso seu objetivo de fazer a redução de pontos da trajetória afim de facilitar a manipulação dos dados, através da adição de semântica às trajetória. O RIP-RS funciona com muita semelhança ao IB-SMoT, sendo este focado na análise de trajetórias de objetos móveis comuns enquanto a extensão implementada neste estudo é focada nas redes sociais, apresentando o mesmo nível de efetividade e sucesso encontrado no primeiro. Com a inserção do RIP-RS no Weka-STPM, esta versão do software passa a disponibilizar o processamento de dados originários de redes sociais, uma função ainda não existente na ferramenta.

O método PMO-RS, surgiu a partir da análise dos dados e foi colocado em prática a partir da efetividade de que os dados podem retratar com certo nível de confiança a realidade. Este método se mostrou extremamente eficaz nas predições de local de moradia e de ocupação feitas para trajetórias com um número grande de pontos, encontrando certo nível de dificuldade em trajetórias com pequeno número de pontos. O algoritmo também se mostrou mais efetivo quando tenta predizer o local de moradia do que o de ocupação, isso porque as pessoas tendem a apresentar um comportamento mais parecido na primeira situação do que na segunda, além do que, a grande maioria dos usuários faz postagens de sua residência, já no local de trabalho ou estudo não são todos que utilizam a rede social, portanto muitas vezes não é mesmo possível identificar esta segunda situação devido ao fato de que a pessoa não faz postagens na rede a partir deste tipo de localidade.

Para trabalhos futuros sugere-se que a base de dados possa ser ainda mais explorada, para que se possam ser encontradas possibilidades de mais algoritmos para extração de conhecimento destes dados. Para o algoritmo já implementado PMO-RS fica a sugestão de aprimorar a questão da predição de local de ocupação, buscando novas variáveis que possam aumentar a percentagem de acerto do método para esta variável.

REFERÊNCIAS

ALVARES, Luis Otávio. **Descoberta de Conhecimento espaço-temporal em Redes Sociais na Internet.** Florianópolis: 2011.

BACKSTROM, L., SUN, E., and MARLOW, C. 2010. **Find me if you can: improving geographical prediction with social and spatial proximity.** In Proceedings of the 19th international Conference on World Wide Web

DENG, Dong-Po., CHUANG, Tyng-Ruey. and LEMMENS, Rob. 2009. **Conceptualization of place via spatial clustering and co-occurrence analysis**. In Proceedings of the 2009 International Workshop on Location Based Social Networks (LBSN '09). ACM, New York, NY, USA, 49-56.

ELMASRI, R.; NAVATHE, S. B. **Sistemas de Banco de Dados.** 4 ed. São Paulo: Pearson Addison Wesley, 2005.

FUJIKASA, T.; LEE, R. and SUMIYA, K. **Exploring Urban Characteristics Using the Movement History of Mass Mobile Microbloggers.** The Eleventh Workshop on Mobile Computing Systems and Applications (HotMobile2010), February 2010

LEE, R.; WAKAMIYA, S. and SUMIYA, K. **Discovery of unusual regional social activities using geo-tagged microblogs.** World Wide Web Journal. 2011.

LEE, R. and SUMIYA, K. **Measuring geographical regularities of crowd behaviors for Twitter-based geo-social event detection.** In Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks (LBSN '10). ACM, New York, NY, USA, 1-10

RAMAKRISHNAN, Raghu; GEHRKE, Johannes. **Sistemas de Gerenciamento de Banco de Dados.** 3 ed. São Paulo: McGraw-Hill, 2008.

APENDICE 02 – CÓDIGO FONTE

```
/**
 *
 * @author Jana
 */

public class ExplorarRedesSociaisMetodos {
    Connection conn;
    String nomeTabelaRIP;
    int numeroCandidato;


    /*seleciona todos os usuários na existentes na tabela de ri's
    faz intersecção com bairros e seleciona os bairros candidatos de acordo com número máximo de candidatos escolhido pelo usuário*/

    public ArrayList<Usuario> SelecionarCadidatos(Connection conn, String nomeTabelaRIP ,int numeroCandidatos, int numeroPostagens) throws SQLException {
     ArrayList<Usuario> usuarios = new ArrayList<Usuario>();

     Statement s = conn.createStatement();
     String sql = "SELECT DISTINCT tid FROM " +nomeTabelaRIP+ " group by tid having sum(numpostagem) >= "+numeroPostagens;
     ResultSet rs = s.executeQuery(sql);
     while(rs.next()){

         Usuario usu = new Usuario();
        usu.setTid(rs.getInt("tid"));

         Statement s2 = conn.createStatement();

         String sql2 = "SELECT sum(numpostagem) as numpostagem, ri_gid FROM "+ nomeTabelaRIP+" WHERE tid= "+
           rs.getInt("tid")+" GROUP BY ri_gid ORDER BY sum(numpostagem) DESC LIMIT "+numeroCandidatos;

         ResultSet rs2 = s2.executeQuery(sql2);
         ArrayList<Candidato> cand = new ArrayList<>();

     while(rs2.next()){

     Candidato candidato = new Candidato();
     candidato.setRegiao(rs2.getInt("ri_gid"));
     candidato.setNumeropostagens(rs2.getInt("numpostagem"));
     cand.add(candidato);
     }
    usu.setCandidatos(cand);
    usuarios.add(usu);
```

```
    }

  return usuarios;
   }


   void montarTabelaAux(Connection conn, String nomeTabelaTrajetoria, String
nomeTabelaRF, ArrayList<Usuario> usuarios) throws SQLException {

    Statement s = conn.createStatement();
    String nomeTabelaTemp = nomeTabelaTrajetoria+"_temp";
    criartabela(nomeTabelaTemp, s);

    int regiaoInteresse = 0;
      for(Usuario usu : usuarios){
        for(Candidato cand : usu.getCandidatos()){
            regiaoInteresse=cand.getRegiao();
            String sql = "INSERT INTO "+nomeTabelaTemp+"(gid, tid, time, tweet,the_geom,
dia_da_semana, regiao_interesse)"+
              " SELECT t.gid, t.tid, t.time, t.tweet, t.the_geom, t.dia_da_semana, b.gid FROM "+
nomeTabelaTrajetoria +" t JOIN "+
               nomeTabelaRF +" b ON b.gid= "+regiaoInteresse+" AND t.tid="+usu.getTid()+"
AND st_intersects(t.the_geom,b.the_geom);";
             s.execute(sql);

        }


   }
     String nomeTabelaPMORS = "pmo_"+nomeTabelaTrajetoria+nomeTabelaRF;
     this.criatabelaPredicaoMoradiaOcupacao(nomeTabelaPMORS, s);
     this.avaliacriterios(conn, usuarios, nomeTabelaTemp, nomeTabelaRF,
nomeTabelaPMORS);

   }
   //cria tabela aux com todos os tweets de cada usuário, apenas nos candidatos a casa ou
escola/moradia
   private void criartabela( String nomeTabelaTemp, Statement s) throws SQLException {


       // tabela temporária
       System.out.println("\t\t criando tabela temporaria...");//testes...
       try {
          s.execute("DROP TABLE "+nomeTabelaTemp);
          s.execute("DELETE FROM geometry_columns WHERE f_table_name =
'"+nomeTabelaTemp+"'");
          //System.out.println("\t\tstops drop...");
       }catch (SQLException ex) {
       }finally {
```

```
        s.execute(
            "CREATE TABLE "+nomeTabelaTemp+"("+
            "    gid serial NOT NULL,"+
            "    tid integer NOT NULL,"+
            "    time timestamp without time zone,"+
            "    tweet character varying(150),"+
            "    the_geom geometry,"+
            "    dia_da_semana character varying(15),"+
            "    regiao_interesse integer,"+
            "    intervalo_tempo character varying(25),"+
            "    CONSTRAINT "+nomeTabelaTemp+"_gidkey PRIMARY KEY (gid)"+
            ") WITHOUT OIDS;"
          );
        s.execute("CREATE INDEX indice"+nomeTabelaTemp+" ON "+nomeTabelaTemp+
           " USING GIST ( the_geom );");

      }
    }
    // Método que avalia horário dos tweets, nas faixas casa ou trabalho/escola

  private void avaliacriterios(Connection conn, ArrayList<Usuario> usuarios, String
nomeTabelaTemp, String nomeTabelaBairros, String nomeTabelaPMORS) throws
SQLException{

     Statement s = conn.createStatement();
     for(Usuario usu : usuarios){
        int tid = usu.getTid();
        System.out.println("--- Avaliando Horário---");
        double somaTabalhoHorario=0;
        double somaCasaHorario=0;
        double somaCasaPalavras=0;
        double somaTrabalhoPalavras=0;
        for(Candidato cand : usu.getCandidatos()){
           String complemento ="";
           String sql = "SELECT * FROM "+nomeTabelaTemp+
            " WHERE tid="+tid+" AND regiao_interesse="+cand.getRegiao();
           // Tweets no horário Trabalho ou estudo
           complemento = " AND to_char(time, 'HH24:MI:SS')>= '08:00:00' AND to_char "
            + "(time, 'HH24:MI:SS')< '20:00:00'";
           ResultSet tweetsTrabEscola = s.executeQuery(sql+complemento);
           double numTweetsHorarioTrabEstudo = 0;
           double numTweetsHorarioCasa = 0;
           double numTweetsPalavraTrabEstudo = 0;
           double numTweetsPalavraCasa = 0;

           while(tweetsTrabEscola.next()){
              GPSPoint pt = new GPSPoint();
              org.postgis.PGgeometry geom = (org.postgis.PGgeometry)
tweetsTrabEscola.getObject("the_geom");
              pt.point = (org.postgis.Point) geom.getGeometry();
```

```
            cand.getPtsOcupacao().add(pt);
            numTweetsHorarioTrabEstudo=numTweetsHorarioTrabEstudo+1;

            String tweet = tweetsTrabEscola.getString("tweet").toLowerCase();
            if(tweet.contains("em casa") || tweet.contains("home sweet home") ||
tweet.contains("lar doce lar")
              || tweet.contains("dormi") || tweet.contains(" at minha casa ") || (tweet.contains("
tv")&& !(tweet.contains(", tv")))
              || tweet.contains("@ my home") || tweet.contains("banho") ||
tweet.contains("morando aqui")
              || tweet.contains("m at condomínio") || tweet.contains("@ condomínio")||
tweet.contains("m at residencial")
              || tweet.contains("@ residencial")) {
              numTweetsPalavraCasa = numTweetsPalavraCasa+1;
            pt.point = (org.postgis.Point) geom.getGeometry();
            cand.getPtsMoradia().add(pt);

            }
            if((tweet.contains(" trabalhar") && !(tweet.contains(" ir trabalhar"))) ||
tweet.contains("work ")
              || tweet.contains("essa aula") || tweet.contains("m at unisul")
              || tweet.contains("m at ufsc") || tweet.contains("m at univali") ||
tweet.contains("facul")
              || tweet.contains("m at colégio") || tweet.contains("@ colégio") ||
tweet.contains("campus")
              || tweet.contains("instituto")|| tweet.contains("m at escola") ||
tweet.contains("empresarial")
              || (tweet.contains("aula") && (tweet+".").split("aula")[1].contains("agora"))){
                numTweetsPalavraTrabEstudo = numTweetsPalavraTrabEstudo+1;
            }

          }
          complemento = " AND (to_char(time, 'HH24:MI:SS')>= '20:00:00' OR to_char "
          + "(time, 'HH24:MI:SS')< '08:00:00')";
          s=conn.createStatement();
          ResultSet tweetsCasa = s.executeQuery(sql+complemento);
          while(tweetsCasa.next()){
             GPSPoint pt = new GPSPoint();
             org.postgis.PGgeometry geom = (org.postgis.PGgeometry)
tweetsCasa.getObject("the_geom");
             pt.point = (org.postgis.Point) geom.getGeometry();
             cand.getPtsMoradia().add(pt);

             numTweetsHorarioCasa=numTweetsHorarioCasa+1;
             String tweet = tweetsCasa.getString("tweet").toLowerCase();
            if(tweet.contains("em casa") || tweet.contains("home sweet home") ||
tweet.contains("lar doce lar")
              || tweet.contains("dormi") || tweet.contains(" at minha casa ") || (tweet.contains("
tv")&& !(tweet.contains(", tv")))
```

```
                || tweet.contains("@ my home") || tweet.contains("banho") ||
tweet.contains("morando aqui")
                || tweet.contains("m at condomínio") || tweet.contains("@ condomínio")||
tweet.contains("m at residencial")
                || tweet.contains("@ residencial") ) {
                numTweetsPalavraCasa = numTweetsPalavraCasa+1;

              }
              if((tweet.contains(" trabalhar") && !(tweet.contains(" ir trabalhar"))) ||
tweet.contains("work ")
                || tweet.contains("essa aula") || tweet.contains("m at unisul")
                || tweet.contains("m at ufsc") || tweet.contains("m at univali") ||
tweet.contains("facul")
                || tweet.contains("m at colégio") || tweet.contains("@ colégio") ||
tweet.contains("campus")
                || tweet.contains("instituto")|| tweet.contains("m at escola") ||
tweet.contains("empresarial")
                || (tweet.contains("aula") && (tweet+".").split("aula")[1].contains("agora"))){
                numTweetsPalavraTrabEstudo = numTweetsPalavraTrabEstudo+1;
              pt.point = (org.postgis.Point) geom.getGeometry();
              cand.getPtsOcupacao().add(pt);

              }

          }
          somaCasaHorario=somaCasaHorario+numTweetsHorarioCasa;
          somaCasaPalavras=somaCasaPalavras+numTweetsPalavraCasa;
          somaTabalhoHorario=somaTabalhoHorario+numTweetsHorarioTrabEstudo;
          somaTrabalhoPalavras=somaTrabalhoPalavras+numTweetsPalavraTrabEstudo;

          cand.setNumtweetsHorarioCasa(numTweetsHorarioCasa);
          cand.setNumtweetsHorarioTrabalho(numTweetsHorarioTrabEstudo);
          cand.setNumtweetsPalavraCasa(numTweetsPalavraCasa);
          cand.setNumtweetsPalavraTrabalho(numTweetsPalavraTrabEstudo);

          sql= "SELECT nm_bairro FROM "+nomeTabelaBairros+" where gid=
"+cand.getRegiao();
          s = conn.createStatement();
          ResultSet nomebairro = s.executeQuery(sql);
          while(nomebairro.next()){
          cand.setNome_regiao(nomebairro.getString("nm_bairro"));
        }
        }
    System.out.println("---"+usu.getTid()+"---");

    this.calculaproporções(usu, somaCasaHorario, somaCasaPalavras, somaTabalhoHorario,
somaTrabalhoPalavras);
    this.selecionaMoradiaOcupacao(usu, nomeTabelaPMORS,conn);

    }
```

```
    }

    private void calculaproporções(Usuario usu, double somaCasaHorario, double
somaCasaPalavras, double somaTrabalhoHorario, double somaTrabalhoPalavras) {


   for(Candidato cand : usu.getCandidatos()){
       if(somaCasaHorario!=0){

cand.setProporcaotweetsHorarioCasa((Math.round((cand.getNumtweetsHorarioCasa()/somaC
asaHorario*100.0)))/100.0); //Horacasa/soma;
       }else{
       cand.setProporcaotweetsHorarioCasa(0);
       }
       if(somaCasaPalavras!=0){

cand.setProporcaotweetsPalavraCasa((Math.round((cand.getNumtweetsPalavraCasa()/somaC
asaPalavras*100.0)))/100.0);
       }else{
       cand.setProporcaotweetsPalavraCasa(0);
       }
       if(somaTrabalhoHorario!=0){

cand.setProporcaoHorarioTrabalho((Math.round((cand.getNumtweetsHorarioTrabalho()/soma
TrabalhoHorario*100.0)))/100.0);
       }else {
       cand.setProporcaoHorarioTrabalho(0);
       }
       if(somaTrabalhoPalavras !=0){

cand.setProporcaoPalavraTrabalho((Math.round((cand.getNumtweetsPalavraTrabalho()/soma
TrabalhoPalavras*100.0)))/100.0);
       }else{
       cand.setProporcaoPalavraTrabalho(0);
       }
       System.out.println(cand.toString());
   }
}

    private void selecionaMoradiaOcupacao(Usuario usu, String nomeTabelaPMORS,
Connection conn) {
       Candidato ocupacao = new Candidato();
       Candidato moradia = new Candidato();
       String sql ="";
       double predicaoTrabalho = 0;
       double predicaoCasa = 0;
       boolean empateCasa=false;
       boolean empateOcupacao=false;
```

```
      for(Candidato cand : usu.getCandidatos()){
         double propOcupacao =
cand.getProporcaoHorarioTrabalho()+cand.getProporcaoPalavraTrabalho();
         double propMoradia =
cand.getProporcaotweetsHorarioCasa()+cand.getProporcaotweetsPalavraCasa();
         cand.setPropOcupacao(propOcupacao);
         cand.setPropMoradia(propMoradia);
         if(propOcupacao> predicaoTrabalho){

            ocupacao = cand;
            predicaoTrabalho=propOcupacao;
            empateOcupacao=false;
         }else{
            if((propOcupacao==predicaoTrabalho) && propOcupacao>0){
               System.out.println("TID: "+usu.getTid()+ " EMPATE OCUPACAO");

if(cand.getProporcaoHorarioTrabalho()==ocupacao.getProporcaoHorarioTrabalho()

&&cand.getProporcaoPalavraTrabalho()==ocupacao.getProporcaoPalavraTrabalho()){
               empateOcupacao=true;

               }else{
               empateOcupacao=false;

if(cand.getProporcaoHorarioTrabalho()>ocupacao.getProporcaoHorarioTrabalho()){
                ocupacao = cand;
                predicaoTrabalho=propOcupacao;


}else{if(cand.getProporcaoHorarioTrabalho()==ocupacao.getProporcaoHorarioTrabalho()){

if(cand.getProporcaoPalavraTrabalho()>ocupacao.getProporcaoPalavraTrabalho()){
                  ocupacao = cand;
                  predicaoTrabalho=propOcupacao;

                  }
            }
         }
         }
         }
         }
         if(propMoradia>predicaoCasa){

            moradia = cand;
            predicaoCasa=propMoradia;
            empateCasa=false;
         }else{
            if((propMoradia==predicaoCasa) && propMoradia>0){
            System.out.println("TID: "+usu.getTid()+ " EMPATE MORADIA");
```

```
if(cand.getProporcaotweetsHorarioCasa()==moradia.getProporcaotweetsHorarioCasa()

&&cand.getProporcaotweetsPalavraCasa()==moradia.getProporcaotweetsPalavraCasa()){
                empateCasa=true;

              }
            else{
              empateCasa=false;

if(cand.getProporcaotweetsHorarioCasa()>moradia.getProporcaotweetsHorarioCasa()){
               moradia = cand;
               predicaoCasa=propMoradia;

}else{if(cand.getProporcaotweetsHorarioCasa()==moradia.getProporcaotweetsHorarioCasa())
{

if(cand.getProporcaotweetsPalavraCasa()>moradia.getProporcaotweetsPalavraCasa()){
                moradia = cand;
                predicaoCasa=propMoradia;

             }
            }
          }
          }
            }
          }
      }
      if(empateOcupacao && empateCasa){
       sql = "INSERT INTO "+nomeTabelaPMORS+" (tid,id_bairro_casa,
nm_bairro_casa,tweets_horario_casa,tweets_palavra_casa,proporcao_casa,"
             + "id_bairro_ocupacao,
nm_bairro_ocupacao,tweets_horario_ocupacao,tweets_palavra_ocupacao,proporcao_ocupaca
o"
             + ") VALUES "+
               "("+usu.getTid()+","+ -1 +",'empate'
"+","+moradia.getNumtweetsHorarioCasa()+","+moradia.getNumtweetsPalavraCasa()+","+m
oradia.getPropMoradia()
             +","+ -1
+",'empate',"+ocupacao.getNumtweetsHorarioTrabalho()+","+ocupacao.getNumtweetsPalavr
aTrabalho()
             +","+ocupacao.getPropOcupacao()+")";

      }else{
      if(empateOcupacao){
         if(predicaoCasa==0){
            sql = "INSERT INTO "+nomeTabelaPMORS+" (tid,id_bairro_casa,
nm_bairro_casa,tweets_horario_casa,tweets_palavra_casa,proporcao_casa,"
```

```
            + "id_bairro_ocupacao,
nm_bairro_ocupacao,tweets_horario_ocupacao,tweets_palavra_ocupacao,proporcao_ocupaca
o"
            + ") VALUES "+
              "("+usu.getTid()+","+ -1 +",'Nenhum candidato a casa'
"+","+moradia.getNumtweetsHorarioCasa()+","+moradia.getNumtweetsPalavraCasa()+","+m
oradia.getPropMoradia()
            +","+ -1
+",'empate',"+ocupacao.getNumtweetsHorarioTrabalho()+","+ocupacao.getNumtweetsPalavr
aTrabalho()
            +","+ocupacao.getPropOcupacao()+")";

          }else{
           sql = "INSERT INTO "+nomeTabelaPMORS+" (tid,id_bairro_casa,
nm_bairro_casa,tweets_horario_casa,tweets_palavra_casa,proporcao_casa,"
            + "the_geom_casa,id_bairro_ocupacao,
nm_bairro_ocupacao,tweets_horario_ocupacao,tweets_palavra_ocupacao,proporcao_ocupaca
o"
            + ") VALUES "+

"("+usu.getTid()+","+moradia.getRegiao()+",'"+moradia.getNome_regiao()+"',"+moradia.get
NumtweetsHorarioCasa()+","+moradia.getNumtweetsPalavraCasa()+","+moradia.getPropMo
radia()
            +","+this.toSQL(0.001, moradia.getPtsMoradia())+"," + -1
+",'empate',"+ocupacao.getNumtweetsHorarioTrabalho()+","+ocupacao.getNumtweetsPalavr
aTrabalho()
            +","+ocupacao.getPropOcupacao()+")";
          }
       } else{
       if(empateCasa){
        if(predicaoTrabalho==0){
             sql = "INSERT INTO "+nomeTabelaPMORS+" (tid,id_bairro_casa,
nm_bairro_casa,tweets_horario_casa,tweets_palavra_casa,proporcao_casa,"
            + "id_bairro_ocupacao,
nm_bairro_ocupacao,tweets_horario_ocupacao,tweets_palavra_ocupacao,proporcao_ocupaca
o"
            + ") VALUES "+
              "("+usu.getTid()+","+ -1 +",'empate'
"+","+moradia.getNumtweetsHorarioCasa()+","+moradia.getNumtweetsPalavraCasa()+","+m
oradia.getPropMoradia()
            +","+ -1 +",'Nenhum candidato a
ocupacão',"+ocupacao.getNumtweetsHorarioTrabalho()+","+ocupacao.getNumtweetsPalavra
Trabalho()
            +","+ocupacao.getPropOcupacao()+")";

          }else{
          sql = "INSERT INTO "+nomeTabelaPMORS+" (tid,id_bairro_casa,
nm_bairro_casa,tweets_horario_casa,tweets_palavra_casa,proporcao_casa,"
```

```
            + "id_bairro_ocupacao,
nm_bairro_ocupacao,tweets_horario_ocupacao,tweets_palavra_ocupacao,proporcao_ocupaca
o,"
            + "the_geom_ocupacao) VALUES "+
              "("+usu.getTid()+","+ -1
+",'empate','"+moradia.getNumtweetsHorarioCasa()+","+moradia.getNumtweetsPalavraCasa(
)+","+moradia.getPropMoradia()

+","+ocupacao.getRegiao()+",'"+ocupacao.getNome_regiao()+"','"+ocupacao.getNumtweetsH
orarioTrabalho()+","+ocupacao.getNumtweetsPalavraTrabalho()
            +","+ocupacao.getPropOcupacao()+","+this.toSQL(0.001,
ocupacao.getPtsOcupacao())+")";
       }
      }else{
      if(predicaoTrabalho==0){
           sql = "INSERT INTO "+nomeTabelaPMORS+" (tid,id_bairro_casa,
nm_bairro_casa,tweets_horario_casa,tweets_palavra_casa,proporcao_casa,"
            +"nm_bairro_ocupacao, the_geom_casa) VALUES "+

"("+usu.getTid()+","+moradia.getRegiao()+",'"+moradia.getNome_regiao()+"','"+moradia.get
NumtweetsHorarioCasa()+","+moradia.getNumtweetsPalavraCasa()+","+moradia.getPropMo
radia()
            +", 'Nenhum candidato a trabalho' ,"+this.toSQL(0.001,
moradia.getPtsMoradia())+")";

        System.out.println("Nenhum candidato a trabalho");

      }else{
      if(predicaoCasa ==0){
      sql = "INSERT INTO "+nomeTabelaPMORS+" (tid,nm_bairro_casa,
id_bairro_ocupacao,
nm_bairro_ocupacao,tweets_horario_ocupacao,tweets_palavra_ocupacao,proporcao_ocupaca
o,"
            + "the_geom_ocupacao) VALUES "+
              "("+usu.getTid()+", 'Nenhum canditado a moradia',
"+ocupacao.getRegiao()+",'"+ocupacao.getNome_regiao()+"','"+ocupacao.getNumtweetsHora
rioTrabalho()+","+ocupacao.getNumtweetsPalavraTrabalho()
            +","+ocupacao.getPropOcupacao()+","+this.toSQL(0.001,
ocupacao.getPtsOcupacao())+")";

      }else{
       sql = "INSERT INTO "+nomeTabelaPMORS+" (tid,id_bairro_casa,
nm_bairro_casa,tweets_horario_casa,tweets_palavra_casa,proporcao_casa,"
            + "the_geom_casa,id_bairro_ocupacao,
nm_bairro_ocupacao,tweets_horario_ocupacao,tweets_palavra_ocupacao,proporcao_ocupaca
o,"
            + "the_geom_ocupacao) VALUES "+

"("+usu.getTid()+","+moradia.getRegiao()+",'"+moradia.getNome_regiao()+"','"+moradia.get
```

```
NumtweetsHorarioCasa()+","+moradia.getNumtweetsPalavraCasa()+","+moradia.getPropMo
radia()
            +","+this.toSQL(0.001,
moradia.getPtsMoradia())+","+ocupacao.getRegiao()+",'"+ocupacao.getNome_regiao()+"',"+
ocupacao.getNumtweetsHorarioTrabalho()+","+ocupacao.getNumtweetsPalavraTrabalho()
            +","+ocupacao.getPropOcupacao()+","+this.toSQL(0.001,
ocupacao.getPtsOcupacao())+")";

      System.out.println ("TID: "+usu.getTid()+" Moradia: "+moradia.getRegiao()+"
Ocupação: "+ocupacao.getRegiao());

      }
   }}}}
      try {
         Statement s = conn.createStatement();
         s.execute(sql);
      } catch (SQLException ex) {

Logger.getLogger(ExplorarRedesSociaisMetodos.class.getName()).log(Level.SEVERE, null,
ex);
      }
   }

   private void criatabelaPredicaoMoradiaOcupacao(String nomeTabelaPMORS, Statement s)
throws SQLException {
      System.out.println("\t\t criando tabela de predição de moradia e ocupação...");
      try {
         s.execute("DROP TABLE "+nomeTabelaPMORS);
         s.execute("DELETE FROM geometry_columns WHERE f_table_name =
'"+nomeTabelaPMORS+"'");

      }catch (SQLException ex) {
      }finally {
        s.execute(
            "CREATE TABLE "+nomeTabelaPMORS+"("+
            "    gid serial NOT NULL,"+
            "    tid integer NOT NULL,"+
            "    id_bairro_casa integer,"+
            "    nm_bairro_casa character varying(100),"+
            "    tweets_horario_casa double precision,"+
            "    tweets_palavra_casa double precision,"+
            "    proporcao_casa double precision,"+
            "    the_geom_casa geometry,"+
            "    id_bairro_ocupacao integer,"+
            "    nm_bairro_ocupacao character varying(100),"+
            "    tweets_horario_ocupacao double precision,"+
            "    tweets_palavra_ocupacao double precision,"+
            "    proporcao_ocupacao double precision,"+
            "    the_geom_ocupacao geometry,"+
            "    CONSTRAINT "+nomeTabelaPMORS+"_gidkey PRIMARY KEY (gid)"+
```

```
            ") WITHOUT OIDS;"
          );

       }


    }

    private String toSQL(double buffer, ArrayList<GPSPoint> pts) {

       String ret1 = "ST_PointFromText('POINT(";
       if (pts.size()==1) {
       ret1 += pts.get(0).point.getX()+ " "+ pts.get(0).point.getY();
       ret1 = ret1.substring(0,ret1.length()-2) + ")','"+-1+"')";
          ret1 = "ST_Buffer("+ret1+","+buffer+")";
       return ret1;

                  }else{
       String ret = "ST_LineFromText('LINESTRING(";

       for (int i=0;i<pts.size();i++) {
          ret += pts.get(i).point.getX() + " " + pts.get(i).point.getY() + ",";
       }
       ret = ret.substring(0,ret.length()-2) + ")','"+-1+"')";

          ret = "ST_Buffer("+ret+","+buffer+")";

       return ret;
    }
    }
    }


/**
 *
 * @author Jana
 */

public class ExplorarRedesSociais extends javax.swing.JDialog {
   private Connection conn;
  private Config config = new Config();
  ArrayList<Usuario> usuarios = new ArrayList<Usuario>();


    /**
     * Creates new form ExplorarRedesSociais
     */
    public ExplorarRedesSociais() {
       initComponents();
    }
```

```
ExplorarRedesSociais(Connection conn) {
 this.setTitle("Generate Arff File");
    this.conn = conn;
    initComponents();
    carregaSchemas();
    this.pack();
                this.setVisible(true);
}

/**
 * This method is called from within the constructor to initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is always
 * regenerated by the Form Editor.
 */
@SuppressWarnings("unchecked")
// <editor-fold defaultstate="collapsed" desc="Generated Code">
private void initComponents() {

    jSeparator1 = new javax.swing.JSeparator();
    jLabelSchema = new javax.swing.JLabel();
    jLabelRIPs = new javax.swing.JLabel();
    jcomboBoxSchema = new javax.swing.JComboBox();
    jComboBoxRIPs = new javax.swing.JComboBox();
    jButtonLoad = new javax.swing.JButton();
    jLabel1 = new javax.swing.JLabel();
    jComboBoxCandidatos = new javax.swing.JComboBox();
    jButtonPredizerMoradia = new javax.swing.JButton();
    jLabelNumMin = new javax.swing.JLabel();
    jComboBoxPostagens = new javax.swing.JComboBox();

    setDefaultCloseOperation(javax.swing.WindowConstants.DISPOSE_ON_CLOSE);
    setTitle("Data mining in social networks");
    setPreferredSize(new java.awt.Dimension(600, 300));
    setResizable(false);

    jLabelSchema.setText("Schema:");

    jLabelRIPs.setText("Pre-processed Trajectories:");

    jcomboBoxSchema.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jcomboBoxSchemaActionPerformed(evt);
        }
    });

    jButtonLoad.setText("Load");
    jButtonLoad.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButtonLoadActionPerformed(evt);
```

```
        }
    });

    jLabel1.setText("Maximum number of candidates:");

    jComboBoxCandidatos.setModel(new javax.swing.DefaultComboBoxModel(new
String[] { "2", "3", "4", "5" }));
    jComboBoxCandidatos.setSelectedIndex(1);

    jButtonPredizerMoradia.setText("OK");
    jButtonPredizerMoradia.setEnabled(false);

jButtonPredizerMoradia.setHorizontalTextPosition(javax.swing.SwingConstants.CENTER);
    jButtonPredizerMoradia.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jButtonPredizerMoradiaActionPerformed(evt);
        }
    });

    jLabelNumMin.setText("Minimum number of posts per candidate:");

    jComboBoxPostagens.setModel(new javax.swing.DefaultComboBoxModel(new String[]
{ "2", "3", "4", "5", "6", "7", "8", "9", "10" }));
    jComboBoxPostagens.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            jComboBoxPostagensActionPerformed(evt);
        }
    });

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addComponent(jSeparator1)
        .addGroup(layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(jLabelRIPs)
            .addGap(0, 0, Short.MAX_VALUE))
        .addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addGroup(layout.createSequentialGroup()
                            .addComponent(jLabelSchema)
                            .addGap(26, 26, 26)
```

```
                        .addComponent(jcomboBoxSchema,
javax.swing.GroupLayout.PREFERRED_SIZE, 117,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGap(38, 38, 38)
                        .addComponent(jButtonLoad))
                    .addGroup(layout.createSequentialGroup()
                        .addComponent(jLabel1)
                        .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING,
false)
                            .addComponent(jComboBoxRIPs,
javax.swing.GroupLayout.PREFERRED_SIZE, 386,
javax.swing.GroupLayout.PREFERRED_SIZE)
                            .addGroup(layout.createSequentialGroup()
                                .addComponent(jComboBoxCandidatos,
javax.swing.GroupLayout.PREFERRED_SIZE, 48,
javax.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                                .addComponent(jLabelNumMin)
                                .addGap(18, 18, 18)
                                .addComponent(jComboBoxPostagens,
javax.swing.GroupLayout.PREFERRED_SIZE, 50,
javax.swing.GroupLayout.PREFERRED_SIZE))))))
                .addGroup(layout.createSequentialGroup()
                    .addGap(238, 238, 238)
                    .addComponent(jButtonPredizerMoradia,
javax.swing.GroupLayout.PREFERRED_SIZE, 98,
javax.swing.GroupLayout.PREFERRED_SIZE)))
            .addContainerGap(28, Short.MAX_VALUE))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabelSchema)
                    .addComponent(jcomboBoxSchema,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(jButtonLoad))
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(jSeparator1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
```

```
                .addComponent(jLabelRIPs)
                .addComponent(jComboBoxRIPs,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(jLabelNumMin)
                .addComponent(jComboBoxPostagens,
javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(jLabel1)
                .addComponent(jComboBoxCandidatos))
            .addGap(34, 34, 34)
            .addComponent(jButtonPredizerMoradia)
            .addGap(134, 134, 134))
        );

        pack();
    }// </editor-fold>

    private void jcomboBoxSchemaActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    private void jButtonLoadActionPerformed(java.awt.event.ActionEvent evt) {
      try{
            Statement s = conn.createStatement();
            ResultSet vTableName = s.executeQuery("SELECT f_table_name as tableName, type
"+
                                        "FROM geometry_columns " +
                                        "WHERE f_table_schema=trim('"+(String)
jcomboBoxSchema.getSelectedItem()+"') "+
                                        "ORDER BY tableName");
            jComboBoxRIPs.removeAllItems();
            while ( vTableName.next() )  {/* creates a new table for each table that has objects
with topological relation to vRegion */
              if(vTableName.getString(1).startsWith("ri_")){
               jComboBoxRIPs.addItem(vTableName.getString(1));
            }
             }
            jButtonPredizerMoradia.setEnabled(true);
        }catch (Exception vErro){
            vErro.printStackTrace();
        }
    }

    private void jButtonPredizerMoradiaActionPerformed(java.awt.event.ActionEvent evt) {
      ExplorarRedesSociaisMetodos explorar = new ExplorarRedesSociaisMetodos();
      String nomeTabelaTrajetorias = "";
```

```
        String nomeTabelaRF = "";
        String nomeTabela=jComboBoxRIPs.getSelectedItem().toString();
        try {
            Statement s = conn.createStatement();
            String sql ="SELECT * FROM relacao_rip_rf WHERE
nometabela='"+nomeTabela+"';";
            ResultSet rs = s .executeQuery(sql);
        while(rs.next()){
        nomeTabelaTrajetorias =rs.getString("tabelatrajetorias");
        nomeTabelaRF = rs.getString("tabelaRF");
        }
         } catch (SQLException ex) {
            Logger.getLogger(ExplorarRedesSociais.class.getName()).log(Level.SEVERE, null,
ex);
         }


        int numeroCandidato
=Integer.parseInt(jComboBoxCandidatos.getSelectedItem().toString());
        int numeroPostagens
=Integer.parseInt(jComboBoxPostagens.getSelectedItem().toString());

        try {
            usuarios = explorar.SelecionarCadidatos (conn, nomeTabela, numeroCandidato,
numeroPostagens);
            explorar.montarTabelaAux(conn, nomeTabelaTrajetorias, nomeTabelaRF ,usuarios);

            System.out.println("OK");
         } catch (SQLException ex) {
            Logger.getLogger(ExplorarRedesSociais.class.getName()).log(Level.SEVERE, null,
ex);
         }


    }

    private void jComboBoxPostagensActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /*
         * Set the Nimbus look and feel
         */
        //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">
        /*
         * If Nimbus (introduced in Java SE 6) is not available, stay with the
```

```
        * default look and feel. For details see
        * http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html
        */
       try {
           for (javax.swing.UIManager.LookAndFeelInfo info :
javax.swing.UIManager.getInstalledLookAndFeels()) {
               if ("Nimbus".equals(info.getName())) {
                   javax.swing.UIManager.setLookAndFeel(info.getClassName());
                   break;
               }
           }
       } catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(ExplorarRedesSociais.class.getName()).log(java.util.loggi
ng.Level.SEVERE, null, ex);
       } catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(ExplorarRedesSociais.class.getName()).log(java.util.loggi
ng.Level.SEVERE, null, ex);
       } catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(ExplorarRedesSociais.class.getName()).log(java.util.loggi
ng.Level.SEVERE, null, ex);
       } catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(ExplorarRedesSociais.class.getName()).log(java.util.loggi
ng.Level.SEVERE, null, ex);
       }
       //</editor-fold>

       /*
        * Create and display the form
        */
       java.awt.EventQueue.invokeLater(new Runnable() {

           public void run() {
               new ExplorarRedesSociais().setVisible(true);
           }
       });
   }
   // Variables declaration - do not modify
   private javax.swing.JButton jButtonLoad;
   private javax.swing.JButton jButtonPredizerMoradia;
   private javax.swing.JComboBox jComboBoxCandidatos;
   private javax.swing.JComboBox jComboBoxPostagens;
   private javax.swing.JComboBox jComboBoxRIPs;
   private javax.swing.JLabel jLabel1;
   private javax.swing.JLabel jLabelNumMin;
   private javax.swing.JLabel jLabelRIPs;
   private javax.swing.JLabel jLabelSchema;
```

```
    private javax.swing.JSeparator jSeparator1;
    private javax.swing.JComboBox jcomboBoxSchema;
    // End of variables declaration

    private void carregaSchemas() {
   try {
            Statement smnt = conn.createStatement();
            ResultSet rs = smnt.executeQuery("SELECT DISTINCT f_table_schema FROM
geometry_columns");
            while (rs.next()) {
               jcomboBoxSchema.addItem(rs.getString(1));
            }
        }catch(SQLException e) {
            e.printStackTrace();
            JOptionPane.showMessageDialog(this,"Falha ao carrehar schemas");
        }
    }
}


/**
 *
 * @author Jana
 */

public class Candidato {
    private int regiao;
    private int numeropostagens;
    private double numtweetsHorarioCasa;
    private double numtweetsHorarioTrabalho;
    private double proporcaotweetsHorarioCasa;
    private double proporcaoHorarioTrabalho;
    private double numtweetsPalavraCasa;
    private double numtweetsPalavraTrabalho;
    private double proporcaotweetsPalavraCasa;
    private double proporcaoPalavraTrabalho;
    private double propOcupacao;
    private double propMoradia;
    private String nome_regiao;
    private ArrayList<GPSPoint> ptsMoradia = new ArrayList<>();
    private ArrayList<GPSPoint> ptsOcupacao = new ArrayList<>();


    public double getNumeropostagens() {
        return numeropostagens;
    }

    public void setNumeropostagens(int numeropostagens) {
        this.numeropostagens = numeropostagens;
    }
```

```
public int getRegiao() {
   return regiao;
}

public void setRegiao(int regiao) {
   this.regiao = regiao;
}

public double getNumtweetsHorarioCasa() {
   return numtweetsHorarioCasa;
}

public void setNumtweetsHorarioCasa(double numtweetsHorarioCasa) {
   this.numtweetsHorarioCasa = numtweetsHorarioCasa;
}

public double getNumtweetsHorarioTrabalho() {
   return numtweetsHorarioTrabalho;
}

public void setNumtweetsHorarioTrabalho(double numtweetsHorarioTrabalho) {
   this.numtweetsHorarioTrabalho = numtweetsHorarioTrabalho;
}

public double getNumtweetsPalavraCasa() {
   return numtweetsPalavraCasa;
}

public void setNumtweetsPalavraCasa(double numtweetsPalavraCasa) {
   this.numtweetsPalavraCasa = numtweetsPalavraCasa;
}

public double getNumtweetsPalavraTrabalho() {
   return numtweetsPalavraTrabalho;
}

public void setNumtweetsPalavraTrabalho(double numtweetsPalavraTrabalho) {
   this.numtweetsPalavraTrabalho = numtweetsPalavraTrabalho;
}

public double getProporcaoHorarioTrabalho() {
   return proporcaoHorarioTrabalho;
}

public void setProporcaoHorarioTrabalho(double proporcaoHorarioTrabalho) {
   this.proporcaoHorarioTrabalho = proporcaoHorarioTrabalho;
}

public double getProporcaoPalavraTrabalho() {
```

```
        return proporcaoPalavraTrabalho;
    }

    public void setProporcaoPalavraTrabalho(double proporcaoPalavraTrabalho) {
        this.proporcaoPalavraTrabalho = proporcaoPalavraTrabalho;
    }

    public double getProporcaotweetsHorarioCasa() {
        return proporcaotweetsHorarioCasa;
    }

    public void setProporcaotweetsHorarioCasa(double proporcaotweetsHorarioCasa) {
        this.proporcaotweetsHorarioCasa = proporcaotweetsHorarioCasa;
    }

    public double getProporcaotweetsPalavraCasa() {
        return proporcaotweetsPalavraCasa;
    }

    public void setProporcaotweetsPalavraCasa(double proporcaotweetsPalavraCasa) {
        this.proporcaotweetsPalavraCasa = proporcaotweetsPalavraCasa;
    }

    @Override
    public String toString() {
        return "Candidato{" + "R=" + regiao + ", NP=" + numeropostagens + ",\n NHC=" +
numtweetsHorarioCasa + ", NHT=" + numtweetsHorarioTrabalho + ", PHC=" +
proporcaotweetsHorarioCasa + ", PHT=" + proporcaoHorarioTrabalho + ",\n NPC=" +
numtweetsPalavraCasa + ", NPT=" + numtweetsPalavraTrabalho + ", PPC=" +
proporcaotweetsPalavraCasa + ", PPT=" + proporcaoPalavraTrabalho + '}';
    }

    public ArrayList<GPSPoint> getPtsMoradia() {
        return ptsMoradia;
    }

    public void setPtsMoradia(ArrayList<GPSPoint> ptsMoradia) {
        this.ptsMoradia = ptsMoradia;
    }

    public ArrayList<GPSPoint> getPtsOcupacao() {
        return ptsOcupacao;
    }

    public void setPtsOcupacao(ArrayList<GPSPoint> ptsOcupacao) {
        this.ptsOcupacao = ptsOcupacao;
    }

    public double getPropMoradia() {
        return propMoradia;
```

```
    }

    public void setPropMoradia(double propMoradia) {
        this.propMoradia = propMoradia;
    }

    public double getPropOcupacao() {
        return propOcupacao;
    }

    public void setPropOcupacao(double propOcupacao) {
        this.propOcupacao = propOcupacao;
    }

    public String getNome_regiao() {
        return nome_regiao;
    }

    public void setNome_regiao(String nome_regiao) {
        this.nome_regiao = nome_regiao;
    }

}

/**
 *
 * @author Jana
 */
public class Usuario {
    private int tid;
    private ArrayList<Candidato> candidatos = new ArrayList<>();

    public Usuario() {
    }

    public ArrayList<Candidato> getCandidatos() {
        return candidatos;
    }

    public void setCandidatos(ArrayList<Candidato> candidatos) {
        this.candidatos = candidatos;
    }

    public int getTid() {
        return tid;
    }

    public void setTid(int tid) {
        this.tid = tid;
    }
```

```
}
```