Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Mineração de Dados de Posição Geográfica e Compras

Everaldo Braga Miranda

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Orientador
Prof. Dr. Jan Mendonça Corrêa

Brasília
2012

Universidade de Brasília — UnB
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Bacharelado em Ciência da Computação

Coordenador: Prof. Dr. Alexandre Zaghetto

Banca examinadora composta por:

Prof. Dr. Jan Mendonça Corrêa (Orientador) — CIC/UnB
Prof. Dr. Li Weigang — CIC/UnB
Prof. Ms. Pedro Antônio Dourado de Rezende — CIC/UnB

CIP — Catalogação Internacional na Publicação

Miranda, Everaldo Braga.

Mineração de Dados de Posição Geográfica e Compras / Everaldo Braga Miranda. Brasília : UnB, 2012.
265 p. : il. ; 29,5 cm.

Monografia (Graduação) — Universidade de Brasília, Brasília, 2012.

1. mineração de dados, 2. mineração dados geográficos, 3. padrões de deslocamento de consumidores, 4. simulação de deslocamento de consumidores, 5. visualização de regras sobre mapas

CDU 004.4

Endereço: Universidade de Brasília
Campus Universitário Darcy Ribeiro — Asa Norte
CEP 70910-900
Brasília–DF — Brasil

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Mineração de Dados de Posição Geográfica e Compras

Everaldo Braga Miranda

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Prof. Dr. Jan Mendonça Corrêa (Orientador)
CIC/UnB

Prof. Dr. Li Weigang
CIC/UnB

Prof. Ms. Pedro Antônio Dourado de Rezende
CIC/UnB

Prof. Dr. Alexandre Zaghetto
Coordenador do Bacharelado em Ciência da Computação

Brasília, 24 de setembro de 2012

Dedicatória

À minha mãe, Ludovina Maria Braga Machado, que esteve sempre ao meu lado, me apoiando, independentemente de quaisquer circunstâncias.

Ao meu pai, Everaldo Miranda Machado, que acreditou na minha capacidade e sempre me ajudou, mesmo na pior das situações.

Agradecimentos

Agradeço primeiramente a Deus, que me deu energia, concetração e persistência para concluir este trabalho.

Agradeço também ao meu orientador, Prof. Dr. Jan Mendonça Corrêa, que mostrou os melhores caminhos para a excução deste projeto.

Resumo

Obter conhecimentos e identificar padrões no grande volume de dados gerados pelo deslocamento de pessoas nas cidades e seus hábitos de consumo é tarefa desafiadora. Para realizá-la, algoritmos de mineração de dados são importantes ferramentas. Este trabalho apresenta uma metodologia para sua utilização nesse contexto, que abrange dados de natureza geográfica. As informações a serem obtidas por meio deste procedimento são as áreas de uma cidade onde, em determinados dias e horários, se concentram compradores de certo produto. São resultados valiosos para empresas e organizações pois dão subsídios para uma melhor escolha de pontos de venda e locais de veiculação de peças publicitárias.

Palavras-chave: mineração de dados, mineração dados geográficos, padrões de deslocamento de consumidores, simulação de deslocamento de consumidores, visualização de regras sobre mapas

Abstract

To find patterns in the great amount of data related to consumers geographic position during their daily activities and their buying behavior, thus obtaining useful knowledge, is a challenging endeavor. Data mining algorithms are important tools that can make it possible. This work shows how to run them on a data set like the one mentioned, which has a geographic context. The results are the locations where consumers of specific products are in a certain day and time, which is very valuable information for those who market such products.

Keywords: *data mining*, *geographic data mining*, *consumers movement patterns*, *consumers movement simulation*, *rules visualization in maps*

Sumário

Lista de Tabelas

Capítulo 1

Introdução

No contexto atual de grande competitividade entre empresas a publicidade tem enorme importância e é fator essencial para o sucesso. É através dela que se pode conferir visibilidade a um produto e convencer um cliente a escolhê-lo ao invés de outro.

Existem várias maneiras de veiculá-la: tv, panfletos, *outdoors*, revistas, *internet*, celulares entre outras. Grandes somas de capital são gastas na sua produção e posicionamento. No entanto, grande parte das propagandas são distribuídas de forma ineficiente: estão em lugares por onde não transita seu público alvo ou são veiculadas em horários nos quais seu público não estará disponível ou atento a elas. Para compensar isso, as organizações geralmente aumentam o volume de publicidade e a distribuem por mais e mais locais.

Publicidade mal posicionada e em quantidade exagerada acarreta desperdício de recursos financeiros e poluição visual, principalmente nos centro urbanos. Também aumenta a demanda por recursos naturais, o que gera, por exemplo, mais desmatamento e poluição industrial. Colabora ainda para o agravamento do problema do lixo nas cidades. Essa realidade só existe por conta do desconhecimento dos hábitos de deslocamento do público alvo das organizações.

Além disso, nos mercados de consumo atuais é imperativo produzir de acordo com a demanda dos clientes ("produção puxada"(4)) em oposição ao antigo modelo da "produção empurrada", na qual produtos eram fabricados de acordo com os parâmetros e possibilidades das empresas e sua venda era forçada pela fábrica aos distribuidores, destes aos varejistas e destes aos clientes. Nesse sentido, para conceber produtos que tenham boa aceitação pelos consumidores, é preciso conhecê-los o máximo possível. Seus hábitos de compra e rotinas de deslocamento são portanto muito úteis também nesse contexto.

Este trabalho visa apresentar uma alternativa às empresas e organizações para que possam identificar e estudar a rotina de circulação espacial do seu público-alvo, aumentando as informações que dispõem sobre seus clientes. Foi sugerida a utilização de técnicas de mineração de dados (descoberta automatizada de informações em bancos de dados). Tais técnicas, além de automatizarem a análise de dados (que devido ao seu grande volume dificultariam muito o trabalho de análise por seres humanos), permitem a descoberta de padrões de difícil identificação através de outros métodos.

Primeiramente foi proposta a forma de obtenção dos dados: aquisição dos mesmos através de aplicativos de celulares (serão inclusive abordadas questões relativas à privacidade das pessoas cadastradas). Os aplicativos citados no trabalho como possíveis fontes de

dados são os que utilizam serviços de GPS (posicionamento global, com dados de latitude e longitude) ou permitem que usuários façam compras através dos seus celulares.

Em seguida foram gerados os bancos de dados necessários à realização do trabalho através de programas escritos para esse fim. Bancos de redes de pontos (cada um representando um cruzamento de vias da cidade considerada), bancos contendo uma população fictícia, bancos de rotas entre várias localidades da cidade e outros contendo rotinas de deslocamento e compras simuladas para a população gerada. Os dados obtidos foram similares aos que poderiam ter sido obtidos no contexto real, pois foram utilizados na geração dos mesmos: informações do IBGE sobre uma determinada área da cidade de Goiânia (para a simulação de uma população numericamente consistente), funcionalidades de geração automática de rotas do *Google Maps* (para a criação de um banco com milhares de rotas), estatísticas de distribuição de classes sociais em grandes metrópoles, pesquisas sobre os produtos mais comprados via *e-commerce*, entre outras informações. A dificuldade de se conseguir dados reais para o trabalho justifica o fato de terem sido gerados bancos de dados fictícios.

O passo seguinte foi a integração dos diversos bancos gerados e o processamento de seus dados a fim de criar tabelas adequadas ao processo de descoberta de conhecimento através de algoritmos de mineração de dados. Tal atividade, conforme realizada neste trabalho, se aplicaria também a dados reais se eles tivessem sido obtidos. Portanto, os modelos de tabelas gerados e a forma de integração e processamento dos dados discutidos neste trabalho são modelos que podem ser seguidos em trabalhos futuros gerando bons resultados. Com as tabelas prontas, foram aplicados vários algoritmos de mineração de dados sobre cada uma. Os resultados foram comparados, avaliando-se os algoritmos mais apropriados e mais eficientes na descoberta de informações no contexto do trabalho (descoberta de conhecimento em bancos de dados com dados geográficos).

Por fim, foi apresentada uma forma eficaz de visualização das informações obtidas pelos algoritmos. Uma forma prática, informativa e que favoreceu a análise dos modelos gerados pela mineração. Isso foi feito através da escrita de um programa que interpreta os modelos gerados pelos algoritmos de mineração e escreve arquivos visualizáveis no *Google Earth*.

Os resultados obtidos foram analisados quanto a sua abrangência de padrões (deliberadamente inseridos nos dados com a finalidade de testar os modelos gerados pelos algoritmos de mineração) e sua relevância para as organizações foi discutida. Todos os procedimentos e programas escritos se encontram detalhados a seguir.

Capítulo 2

Obtendo os dados: *Smartphones*

Neste capítulo será discutida a forma de obtenção de dados reais: através dos *smartphones*. O crescente uso deste equipamento, sua evolução e perspectivas para o futuro. Será tratada também a sensível questão do sigilo dos dados e da privacidade das pessoas cadastradas no sistema proposto.

2.1 O Mercado dos Celulares e *Smartphones*

Há mais de quinze anos atrás a telefonia móvel começava a se desenvolver no Brasil. Celulares eram objetos de valor elevado e poucas pessoas podiam arcar com os custos de possuir um. Só grandes cidades dispunham de rede telefônica capaz de permitir sua utilização, sendo esta rede precária e sujeita a todo tipo de falhas e indisponibilidades.

Atualmente a quantidade de celulares ativos no Brasil ultrapassou a quantidade de habitantes deste país (26). Com o surgimento de celulares pré-pagos, em certos lugares tornou-se mais barato ter um celular do que ter linha de telefone fixo. Muitos planos inclusive diluem os custos dos aparelhos nas mensalidades de forma a permitir a aquisição de aparelhos por virtualmente qualquer pessoa. Há larga disponibilidade de rede mesmo nas menores, mais distantes e isoladas cidades e inclusive ao longo de muitas rodovias. Em menos de 20 anos os celulares deixaram de ser artigo de luxo e passaram a ser acessíveis à quase totalidade da população brasileira, utilizáveis em quase qualquer lugar e possuídos até em mais de uma unidade por algumas pessoas.

Seguindo essa tendência de evolução do mercado de telefonia móvel, há alguns anos já se observa o surgimento de um novo tipo de celular, o *smartphone*. São celulares com sistemas operacionais que possibilitam a criação e execução de programas (19). Dispõem de várias funcionalidades: câmeras fotográficas, sistemas de audio que reproduzem vários formatos de músicas, espaço para armazenamento de arquivos, editores de texto, GPS, entre outras. Eles tem conexão com a internet, serviço que está disponível em grande parte das cidades brasileiras e cujo preço cai a cada dia. A maioria das redes inclusive já se adaptou para permitir que haja conexão com a internet via celulares em alta velocidade.

A popularização dos *smartphones* tem sido rápida, acompanhando as constantes quedas de seu preço. Têm se tornado objeto de desejo da população, que já vê nos celulares um objeto de ostentação que deve ser atualizado a cada novidade do mercado. Por conta disso uma infinidade de modelos foi criada e a cada dia são lançados modelos mais completos e sofisticados, alimentando um ciclo de consumo que se propaga por um mercado

consumidor cada vez mais amplo (tanto de alto poder aquisitivo quanto de baixo) a medida que os preços dos produtos caem.

Paralelamente, novos aplicativos são criados para eles todos os dias e podem ser instalados via internet com grande facilidade e rapidez. Uma infinidade de serviços são oferecidos, desde avisos sonoros para condutores de veículos nas proximidades de radares até jogos, programas de remixagem de músicas e aplicativos de compra online.

Alguns programas são gratuitos para o usuário, outros custam uma quantia determinada. Outros permitem que o usuário escolha se prefere pagar um valor determinado ou receber mensagens publicitárias no seu celular em troca da possibilidade de utilizar o serviço oferecido. Isto criou inclusive um mercado de consumo paralelo para os *smartphones*: o mercado de aplicativos, com lojas online patrocinadas ou pertencentes às próprias empresas fabricantes dos celulares (como o caso da *Apple* e sua *App Store* ou, no caso de aparelhos com sistemas operacionais *Android*, a *Play Store*).

No presente trabalho se supõe que a taxa de crescimento do mercado de celulares observada nas duas décadas passadas se mantenha para os *smartphones*. Espera-se que em poucos anos grande parte da população os utilize, usufruindo de serviços como internet, GPS e aplicativos de compra online. Sobre os dados produzidos por esses dois tipos de aplicativos se propõe a aquisição das informações necessárias para a realização deste projeto.

2.2 Aplicativos de Venda *Online* e GPS

Entre os aplicativos existentes para os *smartphones* estão os que utilizam o serviço de GPS. Eles mostram em um mapa em tempo real a localização geográfica do celular, podendo ser úteis na circulação pelas vias das grandes cidades. Algumas aplicações por exemplo utilizam o GPS dos celulares para emitir avisos sonoros em caso de proximidade de radares em vias públicas, como mostra a figura 2.1. O sistema de GPS funciona através de trocas de informações do aparelho com satélites em órbita da terra. Por meio da análise do atraso da propagação do sinal e da identificação de quais satélites são acessados para cada requisição de posição, é calculada a posição do usuário em determinada unidade e tempo (9). Trata-se de um serviço disponível mundialmente, visto que sua disponibilidade na superfície terrestre é vasta. Muitos *smartphones* vêm prontos para fornecerem informações de posicionamento através de GPS. Tal funcionalidade se tornou comum nos modelos atuais.

Já sobre as vendas de produtos pela internet, o crescimento do *e-commerce* (comércio via internet) tem sido muito grande na última década, tendo estado muito acima do crescimento do comércio usual. Entre os primeiros sites a fornecer a funcionalidade de compra online estavam os de compra de livros, entre eles o da *Amazon*. Atualmente quase todo tipo de produto pode ser encontrado á venda na internet. Inclusive produtos de aplicação industrial. Surgem hoje agregadores (sites que varrem o conteúdo de vários outros sites, apresentando ao usuário uma pesquisa de preços automática poupando-os tempo), sites de compra coletiva (que oferecem desconto na compra de produtos ou serviços desde que um número mínimo de pessoas realize a compra), supermercados *online*, sites que permitem aos usuários participar de leilões, entre outras modalidades de comércio. O crescimento e o ritmo de criação de novas formas de comércio eletrônico é notável. A compra via internet fornece grande comodidade ao comprador visto que este não precisa se deslocar

Figura 2.1: Aplicativo para Celular que Avisa sobre Radares

para pesquisar preços e produtos. Basta acessar os diversos sites de compra online. Os produtos adquiridos chegam por correio na casa do comprador, dias depois de efetuado o pagamento.

No entanto quando se acessa a internet via *smartphone* geralmente o usuário enfrenta dificuldades. Os teclados dos celulares são pequenos, inadequados e as telas são também pequenas para a quantidade de informações apresentada. As letras também são apresentadas em tamanho diminuto, dificultando a leitura. Isso afeta a capacidade de se realizar compras por sites de *e-commerce* via *smartphones*. Os sites de *e-commerce* em geral são projetados para uso em monitores, sendo a tela de um *smartphone* muito menor, o que dificulta a visualização.

Surgiu então a necessidade de se criarem aplicativos para *smartphones* que facilitassem a compra de produtos via internet. Vários já foram criados, como o da *Amazon* (figura 2.2). Esses aplicativos diminuem a necessidade de digitação e privilegiam uma navegação menos baseado em pesquisa de palavras chave e mais baseada em cliques. A apresentação de conteúdo também é mais adequada às telas pequenas dos aparelhos, menos informação é apresentada em cada tela e as fontes são maiores, facilitando a leitura. Esses aplicativos permitem o chamado *m-commerce* ou comércio via celulares.

Com isso tem se tornado cada vez mais comum o surgimento de aplicativos e sites adaptados para o chamado *m-commerce*. Em todo o mundo já existem exemplos de aplicativos como os descritos (7). Grandes redes de supermercados como a rede Pão de Açúcar já anunciaram aplicativos para celulares para facilitar a compra de mercadorias através destes. Espera-se que nos próximos anos o crescimento do *m-commerce* seja semelhante ao crescimento do *e-commcerce* na última década, devido ao grande crescimento do uso

Figura 2.2: Aplicativo de Compras Através de *Smartphones* da *Amazon*

de *smartphones*. O surgimento de muitos aplicativos para este fim evidencia a tendência de crescimento desse mercado.

Este trabalho leva em consideração essa realidade de mercado. Leva em conta a existência dos aplicativos discutidos nesta seção e os dados por eles gerados. Portanto, existe a possibilidade de que, um dia, os métodos a serem apresentados nos capítulos seguintes possam ter aplicação prática.

2.3 Obtenção dos Dados

Os dados a serem trabalhados pelo presente projeto serão simulados, gerados automaticamente por programas em linguagem C. Isso devido à grande dificuldade de obtê-los no mundo real, atividade que estaria fora do foco deste trabalho. Serão no entanto gerados como se estivessem sendo obtidos dos aplicativos descritos na seção anterior. O trabalho supõe que no caso de uma implementação real que vise descobrir os hábitos do público-alvo de determinada organização, tais dados seriam obtidos de aplicativos como os descritos acima. Daí mais um impedimento para obtê-los de fontes reais no contexto da realização deste projeto. Teria sido necessária a colaboração das empresas fabricantes de tais aplicativos para a transmissão de dados que são sensíveis e valiosos para as mesmas. Tal colaboração foi previsivelmente negada pelas empresas contactadas.

Na hipótese da execução deste projeto com dados do mundo real, os aplicativos de compra *online* enviariam a uma central de dados os bens consumidos por cada usuário de *smartphone*, bem como os horários e posições geográficas das compras. O aplicativo que utiliza GPS enviaria por sua vez as diversas posições (em cada horário e data) dos

usuários para a mesma central. Lá, tais dados seriam processados conforme descrito neste trabalho e seriam obtidas as informações desejadas: perfil de circulação do público-alvo de determinada organização, um fabricante de *notebooks* por exemplo.

Vale lembrar que grande parte dos aplicativos a serem instalados em um *smartphone* devem ser adquiridos pelo usuário através de pagamento efetuado em lojas virtuais. Outros são gratuitos, podendo ser obtidos através da internet. Há ainda uma terceira forma de se poder utilizá-los: permitir que seja apresentada publicidade concomitante ao uso do aplicativo. Nesse caso o criador do aplicativo teria receita através da venda do espaço publicitário criado pelo uso do programa.

No contexto deste trabalho os fabricantes de aplicativos para celulares poderiam ter ainda uma terceira opção de receita além das descritas: o fornecimento de informações dos seus usuários. Estes por sua vez para utilizarem os aplicativos poderiam escolher entre pagar uma pequena taxa, receber mensagens publicitárias ou disponibilizar seus dados para mineração de dados.

Com a expectativa de crescimento do mercado de *smartphones* e aplicativos espera-se que a quantidade de dados passíveis de obtenção produzidos da forma descrita seja suficiente para representar com grande fidelidade os hábitos do público-alvo de várias organizações. Dessa forma os procedimentos descritos neste trabalho poderão se tornar importante ferramenta para diversas organizações que venderem seus produtos através de *smartphones* (21).

Deve-se ainda salientar que os bens comercializáveis através da internet e de celulares não se restringem apenas a elementos materiais que chegam aos clientes por correio. Músicas, vídeos, jogos e passagens aéreas são também exemplos de bens vendidos dessa forma. O sistema proposto aplicado à publicidade dos elementos citados também seria eficaz, atingindo rapidamente grande parte público alvo de tais organizações.

2.4 Privacidade

É necessário que se dê privacidade às pessoas cadastradas no projeto quanto à confidencialidade dos seus dados. Muitos dados importantes sobre essas pessoas transitarão pelo sistema central proposto: seus hábitos de compra (o que permitirá deduzir o poder aquisitivo dos cadastrados) e dados muito precisos sobre suas rotas pela cidade com locais, datas e horários. Se corretamente interpretados, tais dados podem fornecer informações íntimas que devem ser protegidas (pois poderiam ser usadas para chantagem ou extorsão), podendo ser acessados por terceiros apenas se autorizados por ordem judicial (no caso da quebra de sigilo durante uma investigação policial, desaparecimento de pessoas, sequestros, entre outras situações), sendo que mesmo nesse caso ainda há o risco de abuso de poder por parte de autoridades que venham a ter acesso a eles.

No caso de uma implementação do sistema proposto para uso efetivo e comercialização de informações com diversas organizações, seria necessária uma abordagem do ponto de vista de segurança de dados. Evitar invasões ao banco da dados central, que acarretariam vazamento de dados e consequente queda no valor dos mesmos, seria crítico para o sucesso da aplicação, pois o sistema seria um grande alvo para tais ações devido ao grande valor comercial dos dados a serem lá depositados.

Entretanto tal abordagem não será realizada neste trabalho visto que o foco do mesmo é a definição de um processo satisfatório para a descoberta de conhecimento nos bancos

de dados propostos, ou seja, a seleção de algoritmos de mineração de dados adequados à aplicação.

Propõe-se, no entanto, o uso de técnicas de desagregação, como por exemplo, a criação de um número identificador para cada cadastrado de forma a ocultar, inclusive no banco de dados central, o número de celular do mesmo. Propõe-se ainda que não se armazene nenhum outro dado que poderia identificar os cadastrados, armazenando-se e utilizando-se única e exclusivamente os dados úteis à aplicação proposta a serem definidos no capítulo de metodologia.

Dessa forma se pretende dificultar, em caso de vazamento de dados, sua utilização indevida, facilitando a preservação do anonimato das pessoas cadastradas.

A política de privacidade de dados do projeto deve ainda ser condensada em forma de contrato para que seja apresentada no ato do cadastro, tendo os usuários que concordarem com a política para só então poderem concretizar seus cadastros. Nesse contrato deverão ser apresentados os riscos reais de vazamento de dados e também o compromisso do projeto com a adoção de métodos de segurança capazes de evitar tais vazamentos e, caso ocorram, dificultar a utilização dos dados.

Apesar dos riscos envolvidos acredita-se que muitas pessoas se cadastrariam devido ao anonimato garantido intrinsecamente no projeto. Propõe-se inclusive que sejam exigidas o menor número possível de informações pessoais que poderiam identificar os papéis sociais dos usuários no ato do cadastro, tranquilizando quem aceitasse compartilhar seus dados.

Capítulo 3

O Método da Mineração de Dados

A cada dia o volume de dados produzidos em tempo real aumenta. A cada nova tecnologia, mais dados são coletados. De informações sobre indivíduos, seus hábitos, características, genoma até informações sobre o estado de conservação de um setor determinado de um oleoduto, a maioria é armazenada digitalmente. Um exame médico pode produzir quantidades incríveis de dados, como é o caso de uma ressonância magnética. Com essa produção acelerada de dados sobre todos os detalhes da vida humana e de tudo o que utilizamos e observamos, surge a possibilidade de integração de diferentes fontes de dados para análise conjunta, possibilitando grande produção de informação e consequentemente, conhecimento.

No entanto um volume tão grande de informações é difícil de ser analisado por seres humanos. Observa-se que a velocidade de produção de dados é muito maior que a velocidade de produção de conhecimento sobre eles (27).

Em conjuntos de dados pequenos e com poucos atributos é possível ao ser humano fazer inferências, descobrir padrões e deduzir propriedades dos fenômenos estudados. Entretanto, quando o conjunto de dados se torna muito grande e cada dado apresenta vários atributos, tal estudo se torna muito difícil do ponto de vista humano. Padrões mais complexos e intrincados envolvendo vários atributos (fatores) diferentes não são tão facilmente identificáveis; e ainda que fossem, seria dispendido muito tempo para isso.

Se um indivíduo desejar estudar padrões e fenômenos relacionados aos dados, será necessária a utilização de artifícios e ferramentas computacionais que possam analisá-los. Com o uso de estatísticas e com a visualização de dados em gráficos e tabelas, é possível obter algum conhecimento sobre um conjunto muito grande de dados. Entretanto, para uma análise mais profunda, métodos automatizados de busca de conhecimento se fazem necessários. Neste contexto surgem as técnicas de mineração de dados.

Segundo Krzysztof (5), mineração de dados são estratégias que têm por meta obter informações que façam sentido em um determinado domínio através de uma grande quantidade de dados não supervisionados. Como exemplo, analisar de forma automatizada, através de algoritmos, um grande banco de dados sobre compras de clientes em uma rede de surpermercados (com produtos comprados em cada horário por cada cliente) e concluir que compradores de fraldas, no período noturno, em geral também compram cerveja, é praticar mineração de dados.

3.1 Passos para a Mineração de Dados

O processo de mineração de dados, segundo Jiawei (14), consiste na execução iterativa dos seguintes passos:

- Limpeza: deletar ruído e dados inconsistentes.
- Integração: combinar múltiplas fontes de dados, o que pode exigir muito tempo de processamento se os bancos de dados forem muito grandes.
- Seleção: selecionar o conjunto dos dados relevantes à análise.
- Transformação: transformar ou consolidar os dados em formas apropriadas para a mineração através de operações de resumo, agregação, entre outras.
- Mineração: onde métodos inteligentes são aplicados na extração de padrões dos dados.
- Avaliação de padrões: identificar os padrões realmente importantes no domínio.
- Apresentação do conhecimento: utilizar técnicas de visualização e representação de conhecimento para apresentar as informações descobertas ao usuário.

Antes de serem aplicados os algoritmos, deve-se ainda dividir o banco de dados disponível em dois conjuntos: de treinamento e de teste. Sobre o conjunto de treinamento serão aplicados os algoritmos para procura de padrões e regras gerando um modelo de classificação das instâncias do banco. Sobre o conjunto de teste o modelo obtido na primeira fase será aplicado, possibilitando avaliação das taxas de erros e acertos do mesmo. Vários indicadores estatísticos devem ser produzidos para que se possa comparar eficientemente vários algoritmos e escolher o mais adequado para cada conjunto de dados analisado.

3.2 Possíveis Fontes de Erros e Dificuldades

São muitas as possíveis fontes de erros no processo de mineração (segundo (27)). Entre elas destaca-se a existência de dados extremos no banco de dados. Podem ser referentes a instâncias em condições raras, especiais ou anormais ou podem até mesmo ser erros de digitação. Para evitar a interferência dos dados extremos o melhor procedimento é sua eliminação na fase de pré-processamento de dados. Algumas vezes no entanto os dados extremos ou pontos isolados são interessantes para avaliação e criação de modelos. Nesses casos eles devem permanecer inalterados. Em todos os demais casos sua permanência provocaria interferências nos algoritmos, no processo de identificação de informações e formação de regras.

Outro fator que pode atrapalhar a obtenção de resultados é a geração de regras muito específicas. Trata-se do *overfitting*, ou adequação exagerada das regras ao conjunto de treinamento de forma que tais regras se tornam inválidas para um novo conjunto de dados que chegue do mesmo domínio. Neste caso a aplicação do modelo gerado sobre o conjunto de dados de teste resulta em altas taxas de erro. Para evitar a ocorrência deste problema um procedimento geralmente eficaz é a alteração dos parâmetros dos algoritmos utilizados de forma a permitir uma margem de erro, ou limitar o seu funcionamento até certo nível.

A observância das taxas de erro produzidas pela aplicação dos modelos sobre os conjuntos de teste ajuda a identificar o problema.

Um problema análogo é o *underfitting*, quando as regras obtidas não se adequam ao conjunto de treinamento (sua aplicação implica muitos erros) no entanto se adequam ao conjunto de teste. Tal fenômeno também deve ser evitado, modificando-se os parâmetros dos algoritmos.

Além disso se deve reconhecer que muitas regras obtidas são óbvias do ponto de vista do usuário (já foram percebidas, como por exemplo um aumento de vendas de chocolate na páscoa). As regras obtidas devem ser selecionadas para que se consiga obter conhecimento relevante sobre o banco de dados analisado. Podem inclusive ser geradas regras inválidas, cabendo ao usuário do sistema discernir dentre os resultados obtidos os mais adequados. Alguns algoritmos como o de redes neurais podem ainda gerar modelos de difícil análise. Além disso muitos algoritmos não são aplicáveis a todo tipo de dados, cabendo ao pesquisador saber quais algoritmos utilizar em cada caso.

Por fim, a visualização de dados também é um fator importante para a facilitação do processo. Muitas vezes a representação das instâncias de um banco em um plano cujos eixos são atributos das instâncias revela a relação entre tais campos e a classificação dos dados, denotando a importância da utilização de métodos de visualização eficientes para a descoberta dos atributos relevantes para a mineração (16). Mesmo depois do processo de mineração de dados, a definição de uma forma de visualização dos resultados também é essencial para favorecer o entendimento pelos seres humanos. A utilização de técnicas visuais geralmente favorece a leitura das informações, permitindo a comunicação de resultados de forma rápida e eficiente. Neste trabalho, uma forma de se visualizar as informações obtidas será também discutida e implementada.

3.3 Alguns Algoritmos

Nesta seção são discutidos alguns tipos de algoritmos que foram utilizados no trabalho. Os detalhes gerais sobre o funcionamento deles foram obtidos em (27) e (17).

Dentre os referidos algoritmos de mineração de dados utilizados destacam-se os algoritmos que utilizam árvores no seu processamento. Obtém-se, através deles, árvores de decisão para classificação dos dados.

Entre os parâmetros de funcionamento desses algoritmos está a definição de uma tolerância a erros para evitar que as árvores de decisão se tornem muito grandes e consequentemente gerem regras muito específicas. Regras muito específicas não são úteis pois se adequam excessivamente aos dados de treinamento do algoritmo produzindo regras que não são aplicáveis a dados futuros. Algumas vezes os algoritmos aceitam como parâmetro a profundidade máxima da árvore, o que seria uma segunda abordagem para o mesmo problema (o de regras muito específicas).

Existem também algoritmos que obtém regras de outras formas, por tentativa e erro avaliando os caminhos através de métricas de tolerância a erros. Também estão sujeitos a *overfitting*, devendo ter seus parâmetros controlados para evitar que tal problema ocorra.

Outra classe de algoritmos são os que usam redes neurais (redes bayesianas) para "aprender a classificar"uma instância observando seus atributos. Tais algoritmos simulam as redes de neurônios humanos e seu método de aprendizado (de forma muito simplificada). Os algoritmos dessa classe fazem um "treinamento"sobre os dados e vão avaliando o grau

de acerto do seu aprendizado. São algoritmos sofisticados que obtém resultados muito bons para determinadas aplicações.

Neste trabalho todos os algoritmos que se aplicarem à mineração dos dados aqui avaliados serão testados.

3.4 Importância da Mineração de Dados

A grande abrangência da computação, a enorme quantidade de dados produzida a todo momento sobre todo e qualquer movimento dos indivíduos na sociedade e as técnicas de mineração de dados existentes levantam ainda outra questão importante: o sigilo, a confidencialidade dos dados.

Se por um lado é importante para o progresso científico que os dados produzidos pela sociedade sejam analisados, por outro lado não se pode violar o direito à privacidade dos cidadãos, via de regra, protegido pelo sigilo dos dados pessoais. As técnicas de mineração de dados permitem cruzamento de bancos de dados e obtenção de informações inimagináveis de forma rápida e automática. Isso facilita a espionagem dos cidadãos principalmente se considerarmos que quase todas as atividades humanas estão sendo digitalizadas (e portanto passando a produzir dados digitais passíveis de mineração). Com o advento da computação ubíqua (presença de sensores e aparatos eletrônico digitais em todos os lugares possíveis) passa a ser virtualmente impossível se mover de forma desapercebida no espaço urbano. Neste sentido Tom M. Mitchell (23) lembra que é essencial considerar a questão da privacidade dos indivíduos na criação e implementação dos algortimos de mineração de dados.

Mas, apesar das implicações sobre a privacidade dos cidadãos, as práticas de mineração de dados são de grande valia para as organizações, como acentua Monte Hancock (15). Elas contribuem para a *business intelligence* (11), conceito que se refere à capacidade das organizações de perceberem rapidamente mudanças no mercado a sua volta e reagirem prontamente com boas soluções. Através da mineração de dados se descobrem informações sobre o público alvo, o que favorece desde o *design* de produtos até sua distribuição. Dessa forma a mineração de dados tende a ser uma tática cada vez mais adotada no mercado, se tornando uma importante ferramenta para o planejamento estratégico das empresas.

Organizações governamentais também podem fazer uso dessas técnicas. Podem ser úteis por exemplo na análise das declarações de imposto de renda (busca por perfis de sonegadores) e na análise das contas públicas (busca por superfaturamento de contratos e outras irregularidades). Podem também ser utilizadas na análise dos dados obtidos com os censos populacionais, gerando conhecimento útil para o planejamento de políticas públicas.

Os governos precisam administrar enormes quantidades de dados sobre seus cidadãos e gastam todos os anos enormes somas para analisar tais dados, dependendo para isso da manutenção de milhares de funcionários. Métodos de análise automáticos e de mineração de dados são portanto muito úteis nesse contexto, permitindo inclusive diminuição dos gastos com contratação de pessoal.

As técnicas discutidas neste trabalho possuem portanto muitas aplicações em vários contextos, podendo ser importantes em vários setores da sociedade.

Capítulo 4

Hipóteses e Objetivos

4.1 Hipóteses

- O uso de *smartphones* irá crescer muito nos próximos anos.
- Uma quantidade considerável de usuários de aplicativos para celulares têm interesse em compartilhar seus dados em troca do uso gratuito de tais aplicativos.
- O *m-commerce* irá se popularizar.
- Aplicativos que utilizam as funcionalidades do GPS continuarão surgindo e serão largamente utilizados.
- É possível simular de forma razoável os dados de deslocamento espacial e compra de produtos por pessoas em uma cidade.
- Existem algoritmos implementados no *Weka* adequados e eficientes para mineração sobre dados geográficos simples como latitude e longitude que geram regras envolvendo posição geográfica e outros atributos como data, hora e região.

4.2 Objetivo Geral

O objetivo do trabalho é mostrar, por meio de simulação, que através do uso de técnicas de mineração de dados existe a possibilidade de se extrair (de bancos de dados com a rotina de movimentação e de compras de uma população) padrões referentes à localização de um público-alvo de determinada empresa, permitindo com isso, possivelmente, descobrir bons pontos de uma cidade para posicionamento de publicidade.

Tal apresentação de informações deverá levar em conta datas e horários nos quais são válidas. Deverá também ser de entendimento intuitivo, realizada sobre um mapa através da plotagem das informações obtidas através dos métodos utilizados.

Pretende-se com isso possibilitar a diminuição da poluição visual e ambiental nas cidades e promover o uso mais eficiente das verbas de publicidade pelas empresas, gerando economia nos gastos.

4.3 Objetivos Específicos

- Descobrir algoritmos eficientes para mineração de dados que incluam informações geográficas.
- Delinear uma forma plausível de obtenção de uma quantidade adequada e representativa de dados numa eventual aplicação prática deste trabalho.
- Gerar dados similares aos reais e em quantidades adequadas que simulem corretamente a realidade de circulação e compra de produtos *online* da população da área mais central de Goiânia - Goiás.
- Definir uma estrutura adequada para um banco de dados, que permita rápido processamento das informações recebidas pela central, enviadas pelos *smartphones*.
- Definir corretamente uma forma de processamento e seleção automática dos dados a fim de se obter tabelas apropriadas à mineração de dados e obtenção de informações adequadas às necessidades das organizações.
- Pré-processar de forma adequada os dados que vão sofrer mineração, selecionando os mais relevantes para a descoberta de padrões.
- Escolher um conjunto de algoritmos eficientes para processar tais dados e descobrir regras e padrões úteis para a aplicação, devendo as regras obtidas serem expressas em função de informações geográficas e horários.
- Apresentar uma boa forma de visualização das informações obtidas que permita rápida e eficiente leitura por parte do usuário.

Capítulo 5

Metodologia do Trabalho

Neste capítulo é discutida em linhas gerais a forma como foi processada a obtenção das informações que se pretende com este trabalho. Desde alguns detalhes dos bancos de dados envolvidos, sua obtenção, seleção, pré-processamento de dados de interesse, mineração de informações, até comparação de resultados através de indicadores estatísticos e visualização de regras.

5.1 Bancos de Dados Necessários

Para que se realize o trabalho proposto deve ser criado um banco de dados com informações geográficas (caminhos percorridos) de cada indivíduo cadastrado no projeto e outro banco de dados com dados sobre os produtos que cada um adquiriu através do celular.

As informações geográficas armazenadas devem ser a latitude e a longitude do celular a cada intervalo de tempo t, a ser definido de acordo com a disponibilidade de memória e processamento do servidor. Cada instância deve conter portanto latitude, longitude e também uma data e horário nos quais o celular esteve naquela posição. Tais informações devem ser enviadas à central de dados pelo celular do cadastrado.

Cada instância do banco de dados deve conter também o número identificador do indivíduo correspondente (ID). Isto é necessário para que se possa relacionar os dados de posição geográfica aos dados de consumo de produtos através do celular. Estes dados também devem ser enviados para a central de dados e armazenados.

Sobre o banco de dados com informações de compra de produtos, suas instâncias devem conter os identificadores dos celulares cadastrados (IDs) idênticos aos do banco de dados de posições geográficas. Além disso, cada instância deve conter um identificador do produto comprado, a posição geográfica onde se deu a compra e o dia e horário da mesma. Tais devem ser as informações enviadas ao servidor pelos celulares cadastrados.

Os registros de compra devem ser atualizados periodicamente, de forma que sejam deletadas as informações de compras antigas (o período de tempo para que um registro seja considerado antigo deve ser definido de acordo com cada produto).

5.2 Bancos de Dados Gerados

Neste trabalho os dados necessários ao banco foram gerados por programas em linguagem C. Obtê-los em quantidade suficiente para a mineração de dados no mundo real demandaria muito tempo e esforço em atividades distantes do foco do trabalho. Seria necessário formar parcerias com empresas que detém tais dados, ou seja, administradoras de aplicativos que utilizam GPS e que promovem venda de produtos através de celulares. Isto se mostrou inviável diante das constantes negativas por parte dessas organizações. Ocorre que os dados necessários ao projeto são de grande valia no mercado e além disso envolvem a privacidade dos usuários dos aplicativos. As organizações que os detém são portanto refratárias à ideia de compartilhá-los com quem quer seja.

Diante da necessidade de gerar dados consistentes, similares aos reais e em quantidade adequada ao processo de mineração de dados, foram seguidas as seguintes etapas:

- Criação de uma rede de pontos (reticulado de pontos), cada ponto representando um cruzamento de ruas, que abrange toda a região a ser considerada no trabalho. Tal reticulado deve ser criado inclusive no caso de se ter acesso a dados reais. Cada par de coordenadas do banco é convertido para o ponto do reticulado que seja mais próximo. Com isso praticamente não se perde informações úteis e ao mesmo tempo se diminui o número de instâncias do banco de dados, possibilitando que os algoritmos de mineração executem suas tarefas muito mais rapidamente e de forma muito mais eficiente.
- Simulação de uma população consistente com a região a ser abordada no trabalho (parte mais central da cidade de Goiânia). Isso foi feito com o auxílio de dados obtidos no site do IBGE relativos ao censo de 2010, de forma que tanto a quantidade de pessoas para cada região da cidade quanto a quantidade de pessoas das classes A e B fossem similares à realidade.
- Requisição automatizada (realizada por programa) de milhares de rotas ao *Google Maps* com pontos de partida e chegada consistentes com a região estudada e a população simulada.
- De posse das rotas e levando-se em consideração a classe social de cada pessoa: simulação do deslocamento da população durante um mês.
- Simulação da compra (e do local da compra) de diversos produtos via celular por parte da população com probabilidade de compra consistente com a classe social de cada indivíduo.
- Gravação de todos os dados obtidos em arquivos para posterior processamento e criação de tabelas adequadas ao processo de mineração de dados.

5.3 Tabelas para Mineração de Dados

Para o processo de mineração de dados devem ser confeccionadas listas de instâncias que contenham atributos úteis para a descoberta de informações. Para tanto é preciso selecionar instâncias dos bancos de dados, cruzando informações de diferentes bancos para se obter o conjunto de dados mais informativo.

Nesse contexto foram escritos programas em C que processam os dados resultantes das simulações anteriores. Esses programas têm como saída alguns tipos básicos de tabelas. A saber:

- Tabelas cujas instâncias são pontos do reticulado seguidos por um valor proporcional à quantidade de pessoas que ali estiveram num intervalo de tempo de meia hora.
- Tabelas de instâncias cujos atributos estão relacionados ao ato da compra: local da compra, dia, hora e produto comprado.
- Tabelas de pontos do reticulado seguidos por um valor proporcional à quantidade de pessoas que ali estiveram em determinado intervalo de tempo levando-se em conta apenas caminhos realizados imediatamente antes do ato de uma compra.
- Tabelas que detalham em cada instância características de cada caminho percorrido por cada uma das pessoas (hora de entrada e identificador de cada um dos setores censitários por onde passa o caminho da instância em questão).

Para cada tipo de tabela foram ainda criadas 14 tabelas. Uma para cada tipo de produto comprado. Assim cada uma das tabelas continha informações relativas apenas a compradores do produto correspondente. Dessa forma durante a mineração de dados se pôde obter informações, regras e padrões relativos exclusivamente ao público consumidor de cada um dos produtos considerado no trabalho.

Este procedimento de processamento de bancos de dados e criação de tabelas, inclusive específicas para cada produto, deve ser realizado também no caso do tratamento de informações reais. Através deste método se obtém tabelas que, aplicadas aos algoritmos de mineração, geram regras que simulam informações muito úteis no contexto do conhecimento dos hábitos do público-alvo de fabricantes de cada tipo de produto.

As informações geradas por essas tabelas são portanto de grande interesse comercial e têm grande valor para as organizações, por gerar informações sobre os hábitos de deslocamento e compra do público-alvo de cada empresa.

5.4 Mineração de Dados

De posse das tabelas descritas foi realizado o processo mineração de dados. Para isso foi utilizado o programa *Weka 3*, a ser descrito nos próximos capítulos.

Vários algoritmos, cada qual com uma estratégia específica, foram utilizados para cada uma das tabelas. Em geral foram obtidas regras envolvendo posições geográficas, regiões, dias da semana e horários para os quais há grande fluxo de pessoas compradoras de determinado produto. Ou seja, por meio de regras, os algoritmos expressaram os pontos no tempo e espaço onde se aglomeram os consumidores de cada tipo de produto, tanto no contexto da rotina semanal das pessoas quanto no contexto dos caminhos realizados imediatamente antes da compra de cada objeto.

Tais eram as informações buscadas desde o início, que motivaram o trabalho. São essas informações que, utilizadas de maneira racional, ajudariam as empresas fabricantes dos produtos consumidos a melhorar a distribuição de publicidade pelas cidades e até mesmo decidir qual o local mais adequado para se ter um ponto de venda dos seus produtos.

Durante o processo de teste dos vários algoritmos para as várias tabelas, foram produzidas pelo próprio programa *Weka 3* várias estatísticas relacionadas às taxas de acertos das regras produzidas por cada um. Com base nessas estatísticas foram comparados os algoritmos utilizados. Foi obtido assim o conjunto de algoritmos de mineração mais eficaz na produção de informação no contexto deste trabalho para cada tipo de tabela utilizada.

Tal informação é valiosa para futuros projetos que envolvam dados reais pois já aponta quais os algoritmos mais indicados para a busca desse tipo de informação nesse tipo de conjunto de dados.

5.5 Visualização de Informações

Foi escrito em linguagem C um programa que lê as regras produzidas pelo algoritmo de mineração de dados denominado PART e escreve arquivos de extensão .KML (12), visualizáveis no programa *Google Earth*. Isso se mostrou necessário pela quantidade de regras produzidas para cada tabela e consequente dificuldade de interpretação das mesmas. Além de numerosas as regras também envolvem coordenadas geográficas o que dificulta ainda mais a visualização no caso da simples leitura das mesmas.

Com a criação do programa mencionado as regras que envolvem coordenadas geográficas são representadas nos mapas do *Google Earth* como quadrados translúcidos. Isso permite visualizar as ruas e regiões da cidade que se incluem na área delimitada por cada regra.

Dessa forma fica facilitada a identificação e deleção das regras óbvias e inválidas, sobrando assim apenas regras úteis. Além disso a visualização concomitante de várias regras também torna-se viável. Isso permite a identificação de padrões nos dados de forma visual, imediata, o que facilita a comparação entre os vários algoritmos utilizados. Fica facilitada a seleção dos algoritmos mais bem sucedidos na identificação de padrões e portanto, mais indicados para cada tipo de tabela.

Além disso a representação gráfica das regras facilita o estudo e a interpretação das mesmas, sendo de grande valia para as empresas que tiverem seu público-alvo analisado. Portanto, no caso da implementação do sistema aqui descrito em contexto real, programas de visualização das regras escritas pelos algoritmos de mineração são imprescindíveis.

Capítulo 6

Serviços WEB e *Softwares* Utilizados

Neste capítulo serão apresentados os serviços WEB que forneceram grande parte das informações utilizadas neste trabalho: *Google Maps* e o CENSO 2010 do IBGE.

Serão também apresentados os principais *softwares* utilizados e mencionadas algumas de suas funcionalidades que foram úteis neste trabalho.

6.1 *Google Maps*

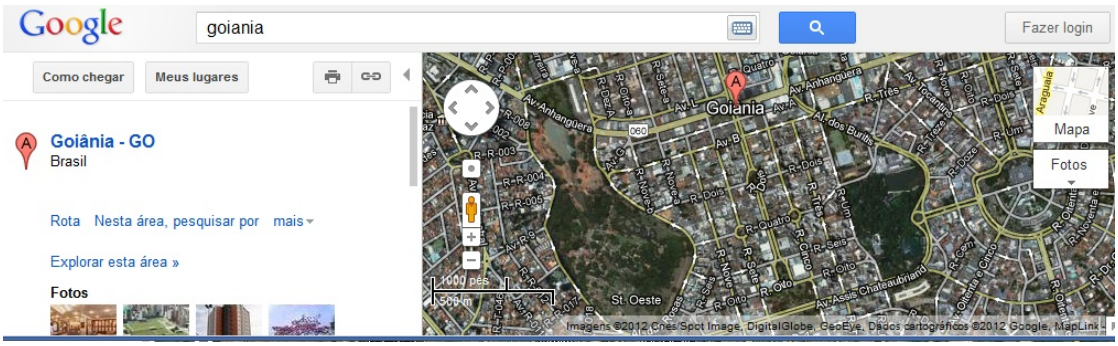

Figura 6.1: Tela do *Google Maps*

Google Maps (13) é um serviço da *Google* oferecido gratuitamente pela internet. Ele apresenta mapas reais de inúmeras localidades do mundo. Mostra ruas, nomes de ruas, empresas, informações sobre as empresas cadastradas (incluindo telefone e localização), faz rotas entre duas localidades escolhidas pelo usuário, mostra imagens obtidas através de satélites dos locais pesquisados e até informações de relevo.

A navegação pelos mapas é simples, bastando o usuário arrastá-los para se mover no espaço e aumentar ou diminuir o *zoom* para aumentar ou diminuir a altitude em relação ao solo.

Foram utilizadas duas funcionalidades desse serviço. A geocodificação reversa e a consulta de rotas.

6.1.1 Geocodificação Reversa

Geocodificação é o processo que permite a consulta de um endereço (país, estado, cidade, bairro, rua e número) e retorna as coordenadas geográficas de um ponto que corresponda a esse endereço. Geocodificação reversa é portanto o processo inverso: consulta de um par de coordenadas geográficas e retorno do endereço correspondente (22).

As consultas pelas informações de cada ponto são realizadas através de URLs com o seguinte formato: http://maps.googleapis.com/ maps/api/ geocode/xml?latlng =latitude,longitude sensor=false onde latitude deve ser o valor numérico da coordenada de latitude do ponto e longitude o valor numérico da longitude. A resposta a essa consulta vem no formato .XML . O nome da rua na qual está localizado o ponto é obtido através da interpretação desse texto .XML .

Tal serviço foi muito utilizado neste trabalho pois foi necessário avaliar o nome da rua a qual pertence um ponto através de suas coordenadas para os milhares de pontos que formam o reticulado de pontos (seção 5.2). Além disso, como cada ponto representa um cruzamento de vias, para se obter o nome das vias que formam o cruzamento foram necessárias várias consultas por ponto.

6.1.2 Consulta de Rotas

O outro serviço utilizado, consulta de rotas, foi disponibilizado pelo *Google Maps* através de outra API. Ela permite a consulta de rotas de um ponto a outro através de URLs contendo as coordenadas do ponto inicial da rota e do ponto final como no exemplo: http://maps.googleapis.com/ maps/api/ directions/ xml?origin= -15.79118, -47.870007 destination=-15.814304,-47.90554 sensor=false. A resposta vem no formato .XML contendo os pontos principais da rota consultada e informações em linguagem natural para auxiliar condutores de veículos a seguir a rota, tais como: Vire a direita na Av. 1. (22)

As informações úteis no contexto deste projeto são os pontos e os nomes das ruas por onde passam cada rota, portanto, as outras informações obtidas nas consultas foram ignoradas.

6.2 Dados do Censo 2010

Várias foram as informações obtidas do site do Censo 2010, realizado pelo IBGE (18). Foram também utilizados mapas interativos disponíveis no site que permitiram a visualização em setores censitários (subdivisões dos bairros das cidades) de variáveis populacionais avaliadas na pesquisa. Arquivos em diversos formatos contendo mais informações na forma escrita e em forma de documentos com informações geográficas também foram obtidos.

Tais informações foram úteis na simulação da população a ser utilizada neste trabalho. Contribuíram para que a população produzida fosse similar à população real, pois foi gerada de acordo com os dados populacionais reais das pessoas que vivem nas regiões a serem exploradas neste trabalho.

6.3 *Weka 3*

O principal programa utilizado no trabalho foi o *Weka 3*. Trata-se de um *software* livre feito em Java e desenvolvido no *Machine Learning Group* da *Waikato University*, Nova Zelândia, que facilita a prática da mineração de dados em todas as suas etapas: desde o pré-processamento dos dados, eliminação de dados extremos, escolha de atributos relevantes para a mineração até a visualização dos resultados e comparação de diferentes algoritmos através de dados estatísticos gerados pelo próprio programa (25) (17).

Na etapa de pré-processamento algumas colunas do banco de dados podem ser apagadas e as colunas que classificam as instâncias também podem mudar de acordo com o comando do usuário. As instâncias podem ser visualizadas de várias formas sendo coloridas de acordo com o parâmetro escolhido para ser o classificador (exemplo: campo sexo das instâncias, sendo sexo masculino cor azul e feminino cor vermelha), conforme figura 6.2.São apresentados histogramas para cada atributo dos dados nos quais para cada valor de atributo é apresentada a quantidade de instâncias que o possui. Cada barra do histograma é dividia por cores de acordo com as classes das instâncias. Isso já permite uma análise superficial sobre quais atributos realmente interferem na classificação das instâncias.

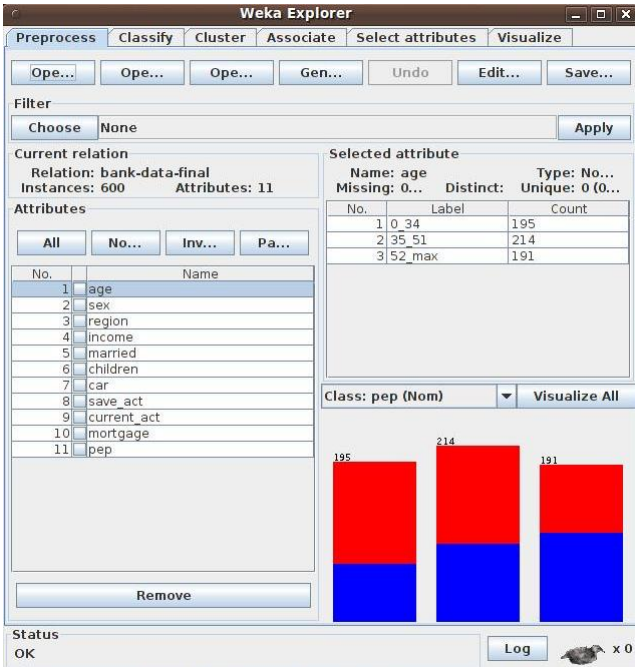

Figura 6.2: Tela de Pré-Processamento

É possível ainda a visualização de todo o banco de dados, instância por instância em uma janela aberta pela programa. Tal janela possibilita inclusive a deleção de instâncias individualmente, atividade útil para a eliminação de dados extremos, irrelevantes para o processo.

Existe também uma funcionalidade que analisa a relevância de um determinado atributo das instâncias para o processo de classificação segundo vários algoritmos vista na figura 6.3. Tal funcionalidade é útil no pré-processamento de dados com muitos atributos

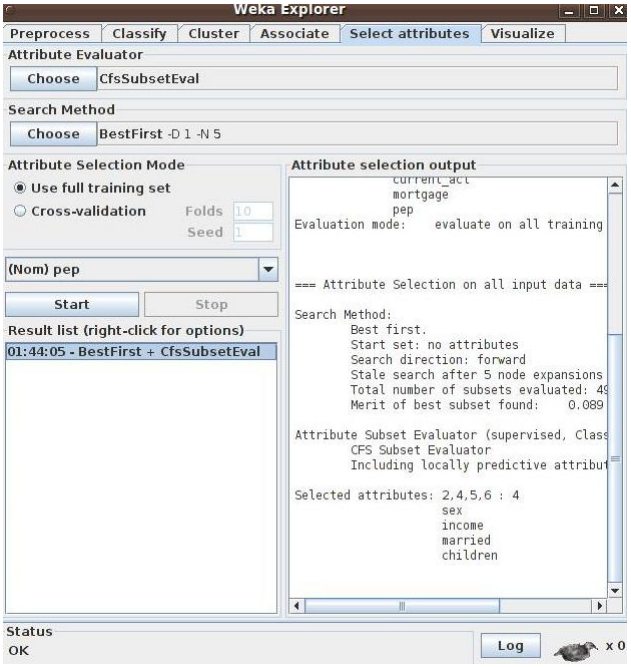

Figura 6.3: Tela de Seleção Automática de Atributos

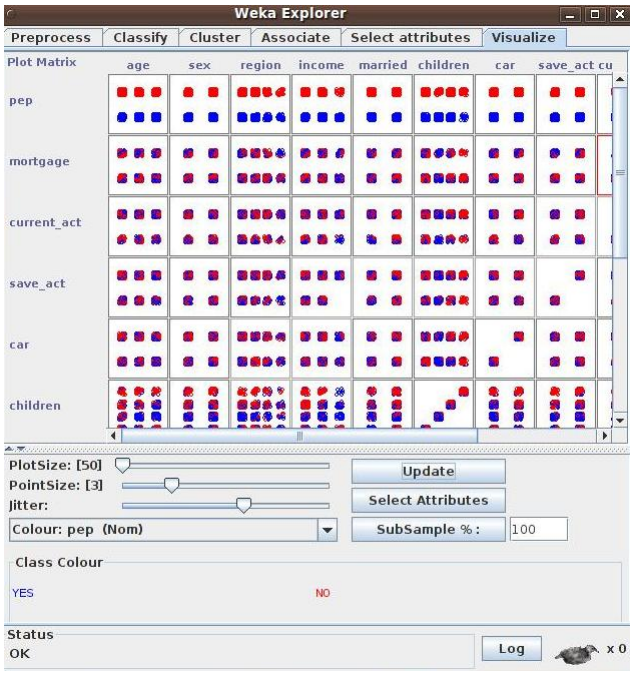

Figura 6.4: Tela de Visualização de Dados

(eliminação de atributos de pouca importância no processo de classificação de instâncias), visando diminuir o tamanho do banco de dados a ser utilizado pelos algoritmos, dinamizando todo o processo.

Os dados podem ainda ser distribuídos num gráfico sendo que os eixo x ou y podem ser quaisquer campos das instâncias. Os erros dos algoritmos podem ser visualizados na mesma interface, conforme figura 6.4 (representando os pontos erroneamente classificados

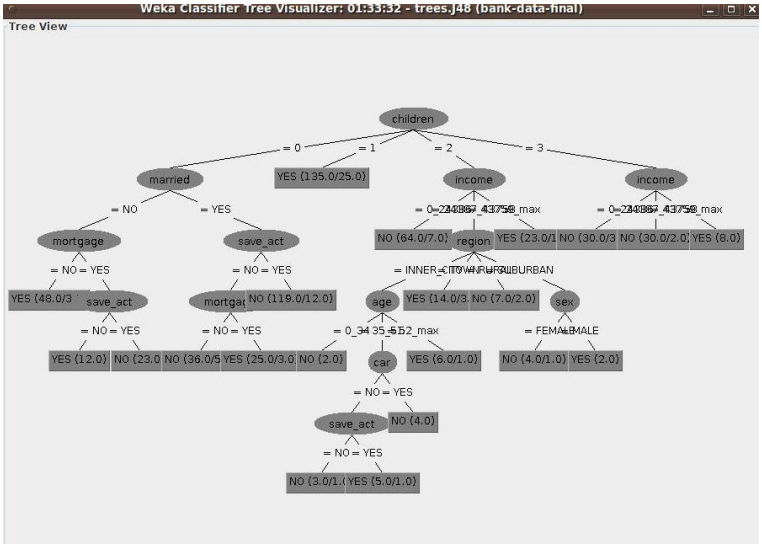

Figura 6.5: Visualização de uma Árvore

com x e os demais com quadrados). Tal interface é útil para a identificação dos atributos mais importantes para a classificação das intâncias, sendo portanto muito útil no estudo dos dados.

Nota-se que o programa têm várias funcionalidades e interfaces que ajudam o usuário a pré-processar os dados, fase essencial no processo de mineração. O programa apresenta ainda várias formas de visualização dos dados e dos resultados dos algoritmos, tornando-se muito prático na atividade de mineração de dados.

O *Weka 3* fornece vários algoritmos de mineração de dados já implementados que podem ser aplicados sobre o banco de dados utilizado. Cada algoritmo tem parâmetros que podem ser ajustados. Após a aplicação de um algoritmo o programa emite um relatório onde constam todos os resultados obtidos incluindo dados estatísticos sobre os erros e acertos e dados relativos à forma de funcionamento e aos passos tomados pelo algoritmo escolhido. Quando os algoritmos se utilizam de árvores elas podem ser visualizadas (figura 6.5). Os relatórios são muito úteis para a comparação de diferentes algoritmos quanto a eficiência na geração de conhecimento sobre determinado banco de dados e serão muito úteis neste trabalho.

Estão presentes no *Weka 3*, dentre muitos, os seguintes algoritmos (17): *ZeroR, Ridor, Prism, PART, DecisionTable, ConjunctiveRule, AdTree, BFTree, J48, LADTree, NBTree, MultilayerPerceptron, VotedPerceptron, RBFNetwork, BayesNet, NaiveBayes,* entre muitos outros. São algoritmos de árvores, regras, redes neurais, algoritmos bayesianos, etc.

O programa oferece ainda a possibilidade de se testar um algoritmo criado ou modificado pelo usuário pois aceita a adição de novos algoritmos aos já implementados. Aceita também a adição de novos algoritmos de quaisquer autores obtidos pela internet sendo portanto possível atualizar o programa com os mais recentes e avançados algoritmos de mineração de dados.

Os dados utilizados devem estar em um formato específico em arquivos .ARFF (17). O formato é simples, o que permite a rápida adequação dos dados de quaisquer bancos de dados ao padrão do *Weka 3*. Deve-se basicamente escrever todos os dados em um arquivo texto, uma instância por linha, com os atributos separados por vírgulas, havendo um

preâmbulo que defina o nome de cada atributo e qual atributo será usado para classificar as instâncias. Para o caso do trabalho, bancos de dados serão produzidos por um programa em linguagem C de forma a obedecer as regras de definição dos arquivos .ARFF . No entanto, caso seja necessário, será utilizada a funcionalidade do programa que permite a conversão de bancos de dados de vários formatos para o formato do programa.

Devido a todas as facilidades que oferece, o *Weka 3* foi o programa escolhido para ser utilizado sobre o banco de dados a ser criado. Serão testados todos os algoritmos nele existentes aplicáveis ao caso do trabalho. Será a ferramenta que ajudará a avaliar a eficácia de vários algoritmos em descobrir regras relevantes no banco de dados proposto.

6.4 *Google Earth*

O programa *Google Earth* se assemelha ao serviço online oferecido pelo *Google Maps*: apresenta imagens de satélite de quase todo o globo, mais detalhadas nos locais mais habitados (cidades), gera rotas entre dois pontos, mostra nomes de ruas, entre outras. A navegação pelos mapas também é semelhante.

No entanto o programa oferece mais funcionalidades que vão além das oferecidas pelo *Google Maps*. A marcação de pontos de interesse é uma delas, tendo sido utilizada para marcar pontos da cidade a serem utilizados na geração das rotas das pessoas. Outra funcionalidade é a de definição de trilhas ponto a ponto, utilizada para se definir o perímetro da cidade a ser considerado neste trabalho. Mas a principal e mais utilizada foi a gravação de dados geográficos em arquivos .KML (12) e a leitura de arquivos .KML com informações geográficas. O programa foi utilizado para visualizar informações produzidas pelos algoritmos de mineração e gravadas em formato .KML, conferir rotas criadas para a simulação das rotinas de deslocamento da população (gravadas em arquivos .KML), gravar os pontos de interesse marcados através do próprio programa, visualizar os setores censitários (regiões da cidade definidas pelo Censo 2010) gravados no formato descrito, entre outras utilizações da funcionalidade.

O *Google Earth* foi portanto muito importante para a realização do trabalho tendo sido utilizado constantemente em diversas tarefas.

6.5 *SAGA GIS*

O programa *SAGA GIS* (10) é um *software* livre desenvolvido para o processamento de dados geográficos. Ele contém várias funcionalidades relacionadas ao processamento dessas informações, oferecendo implementações de diversos algoritmos espaciais. Oferece também variadas opções de visualização de informações e importa diversos formatos de arquivos de dados geográficos.

Neste trabalho foram utilizadas apenas funcionalidades muito básicas deste programa. Entre elas: conversão de arquivos entre diversos formatos, visualização concomitante de diferentes arquivos de dados geográficos e funções simples para relacionar os dados de dois arquivos entre si.

Capítulo 7

Setores Censitários e Reticulado de Pontos

Neste capítulo serão apresentados os procedimentos utilizados para se obter uma rede de pontos que cobre toda a área da cidade de Goiânia a ser considerada. Cada ponto representa um cruzamento de ruas. Assim, todos os cruzamentos internos à área considerada no trabalho foram catalogados formando o reticulado de pontos a ser descrito.

A importância desse reticulado é muito grande, tanto na simulação dos caminhos percorridos pelas pessoas deste trabalho quanto numa aplicação que trate dados reais. Cada ponto referente a uma posição de uma pessoa em determinado horário antes de ser inserido no banco de dados terá suas coordenadas convertidas para as coordenadas do cruzamento mais próximo (ponto do reticulado mais próximo). Com essa medida diminui-se sobremaneira a quantidade de pontos a ser armazenada, agilizando e facilitando a mineração de dados a ser realizada. Caso não fosse utilizada tal estratégia o banco de dados teria pontos muito específicos que se repetiriam muito pouco, dificultando a contagem de pessoas que passam por determinado local. Com o reticulado de pontos se resolve esse problema sem tantas perdas de informações sobre o deslocamento das pessoas.

7.1 Polígonos dos Setores Censitários

Antes da criação do reticulado de pontos foi delimitado um perímetro da cidade de Goiânia a ser considerado no trabalho. Essa região foi escolhida de forma a conter as vias e locais mais movimentados da cidade e as regiões onde moram as pessoas com maior poder de consumo. Dessa forma os pontos de maior interesse para aplicação ficam cobertos pelo trabalho, pois as pessoas de bom poder aquisitivo são as que mais consomem produtos e os locais de maior movimento na cidade são os que concentram mais o público alvo das diversas empresas.

O perímetro foi definido utilizando o programa *Google Earth* e gravado em arquivo .KML (12). O arquivo .KML foi então exportado para o formato .GPX (24). Depois utilizando-se o programa *SAGA* o arquivo em .GPX foi exportado para o formato .SHP (20), mesmo formato do documento contendo os setores censitários a serem descritos no próximo parágrafo. As próximas ações tomadas estão representadas no diagrama da figura 7.1.

Figura 7.1: Diagrama de Definição de Setores Censitários

Foram selecionados todos os setores censitários que estão contidos ou têm intersecção com o perímetro delimitado. Os setores censitários são definidos pelo IBGE para a realização de censos. Cada setor engloba alguns quarteirões da cidade (as vezes apenas um quarteirão), tem uma população numericamente dentro de uma certa faixa de valores e apresenta características relativamente homogêneas. Assim cada parte de uma cidade que guarda determinadas características específicas compõe um setor censitário. A razão dessa seleção é utilizar os dados do censo de 2010 sobre cada um desses setores para tentar simular com realismo tanto em termos numéricos quanto em termos de padrão de consumo, parte da população que reside dentro do perímetro considerado.

Para que fosse feita a seleção dos setores primeiramente foi feito download de um documento digital em formato .SHP do site do Censo 2010 do IBGE (18) contendo polígonos e suas coordenadas, polígonos correspondentes aos setores censitários da cidade de Goiânia.

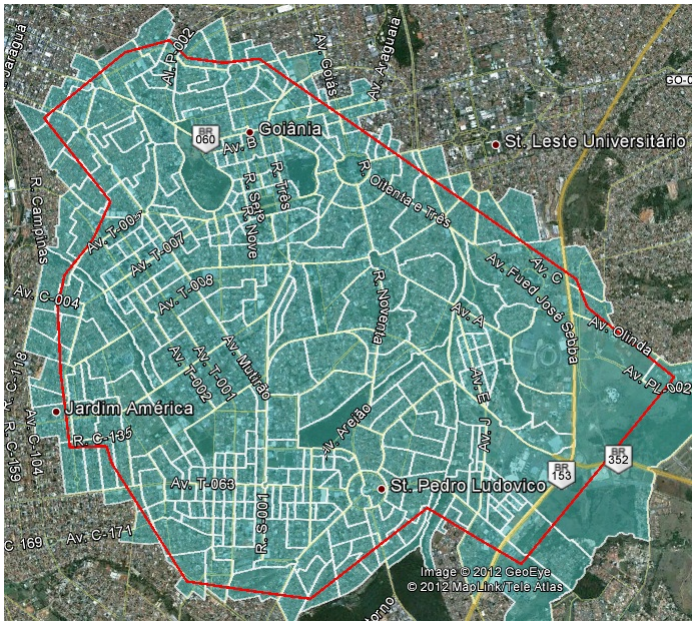

Figura 7.2: Setores Censitários do Perímetro

Utilizando-se o programa *SAGA* foi aberto o documento digital do IBGE e o arquivo .SHP contendo o perímetro mencionado. Sobrepondo as duas imagens foi feita a seleção

de setores censitários que estavam dentro do perímetro de interesse. Os outros setores foram deletados. O resultado dessa seleção, setores dentro do perímetro do trabalho, foi gravado em formado .SHP, podendo ser visualizado (em .KML) na figura 7.2. Cada polígono translúcido representa um setor censitário. O perímetro de interesse é delimitado na figura pela borda vermelha.

7.2 Os Pontos: Cruzamentos de Ruas

Após a definição de cada setor censitário, o próximo passo foi a criação do reticulado de pontos propriamente dito. O diagrama da figura 7.3 demonstra os passos e programas utilizados para tanto.

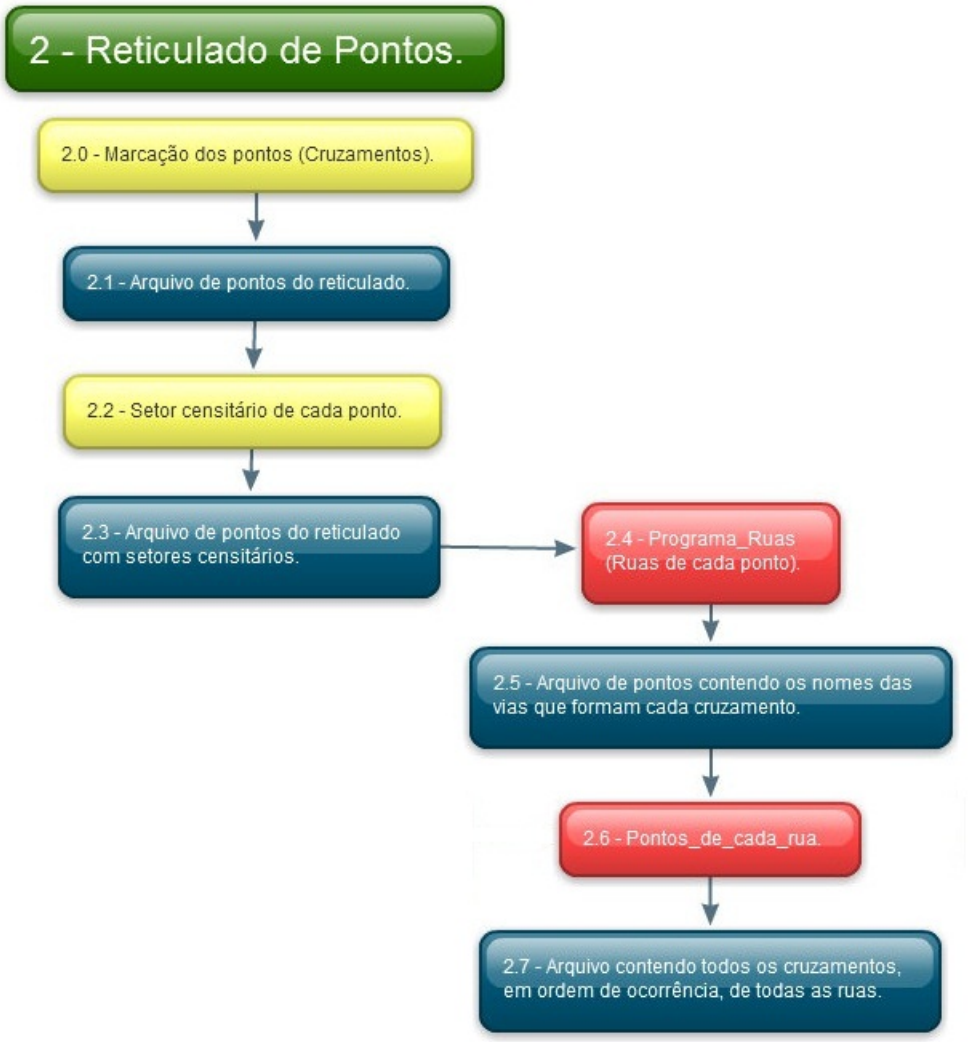

Figura 7.3: Diagrama Representando a Produção do Reticulado de Pontos

Para os diagramas de fluxo de dados deste documento, será utilizada a seguinte codificação. Os retângulos amarelos representam ações realizadas durante o processo, como por exemplo, ordenar uma lista de nomes em uma planilha. Os azuis representam arquivos

que podem ter sido gerados por ações realizadas ou por programas executados. Os vermelhos representam programas escritos e executados, sendo que as setas que se dirigem para tais retângulos são os arquivos lidos pelos programas e as setas que saem dos mesmos são os arquivos gerados por eles. O retângulo verde caracteriza o diagrama por completo quanto ao objetivo da sequência de atividades descritas.

Conforme consta no diagrama citado, foram marcados os pontos do reticulado: um para cada cruzamento de ruas. Utilizando-se o programa *Google Earth* foram criadas trilhas nas quais cada ponto que as compõe correspondeu a um cruzamento de duas ou mais vias. Com isso foram cobertos todos os cruzamentos de vias pertencentes ao perímetro da cidade definido na seção 7.1. Os caminhos foram salvados no formato .KML conforme retratado na figura 7.4.

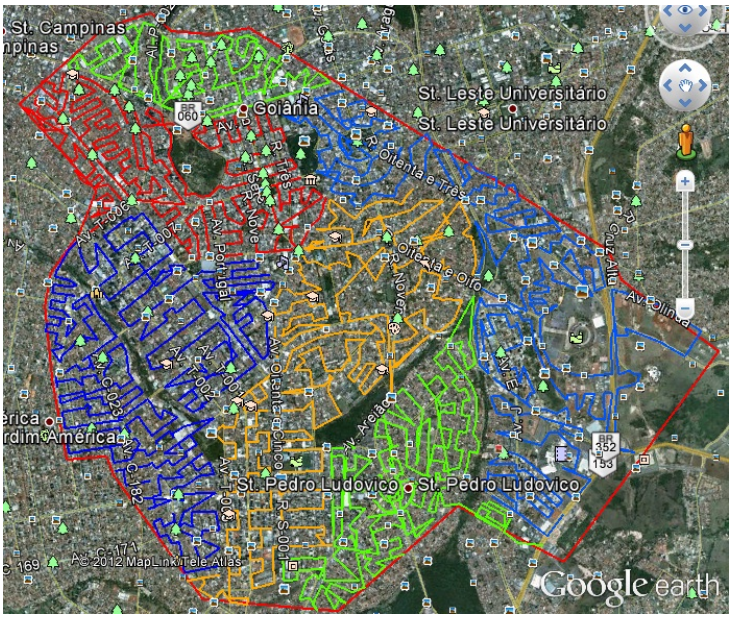

Figura 7.4: Trilhas Abrangendo Todos os Cruzamentos

Utilizando-se o programa *ExpertGPS* foram importados os caminhos. A seguir foi utilizada a opção *join* e o resultado foi gravado em arquivo no formato .SHP (20). Com isso os vários caminhos foram unidos em um só.

O arquivo em seguida foi lido através do programa *SAGA* e foi utilizada a opção *convert lines to points*. Com isso os caminhos formados pelos pontos foram deletados de forma que só os pontos restaram. Os pontos foram gravados em arquivo no formato .SHP.

A seguir o arquivo de setores censitários (apenas com os setores pertencentes ao perímetro definido) foi lido pelo mesmo programa. Os atributos dos setores censitários que não interessavam ao trabalho foram deletados. Em seguida foi utilizada a opção *clip points to polygons*. Dessa forma cada um dos pontos do arquivo que inicialmente era de trilhas foi classificado de acordo com o setor censitário ao qual pertence (figura 7.5). Essa correspondência foi expressa em uma tabela que continha a latitude e a longitude de cada ponto além do setor censitário de cada um. O resultado dessas operações foi gravado em formato .SHP.

A tabela mencionada foi exportada para uma planilha. O código de cada região foi então reduzido de forma que os caracteres relativos ao Estado e à Cidade do setor censitário

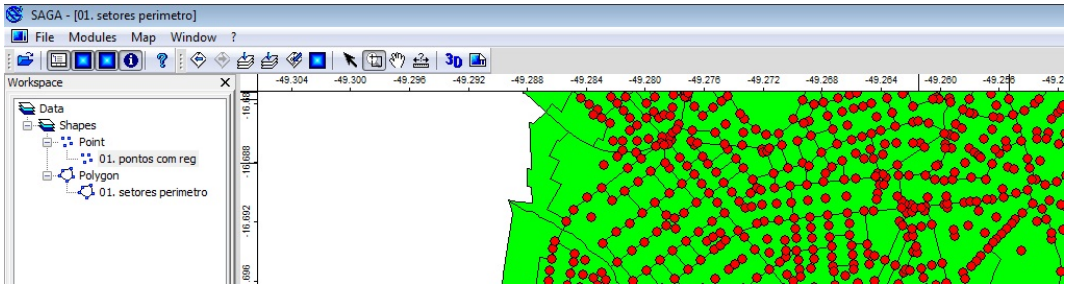

Figura 7.5: Pontos e Setores Censitários

foram deletados (tais caracteres não são necessários pois todos os setores pertencem à mesma cidade). A tabela resultante foi gravada em formato .TXT.

7.3 O Programa *Ruas*

O programa *Ruas* foi criado para ler o arquivo do reticulado de pontos em formato .TXT e criar outro arquivo .TXT que contenha todos os pontos, suas coordenadas, setores censitários e nome das vias que formam o cruzamento que cada ponto representa.

Os nomes das vias são obtidos através de consultas ao serviço de georeferenciamento reverso do *Google Maps*, descrito na seção 6.1.1. Cada consulta é feita através do acesso a uma URL que contém inicialmente o endereço do serviço e no fim da URL o ponto a ser consultado. A resposta vem em formato .XML. O arquivo de resposta é lido pelo programa *Ruas*, interpretado, e o nome da rua obtido é escrito no arquivo de saída em frente ao ponto correspondente. A saída do programa, portanto, é semelhante à tabela 7.1. Cada linha representa um ponto do reticulado, contendo as ruas que formam o cruzamento representado pelo ponto, suas coordenadas, região censitária e número identificador (ID).

ID	Latitude	Longitude	Região Censitária	Primeira Rua	Segunda Rua
1	-16.710590	-49.284679	5160006	Av. C-208	R. C-135
2	-16.709829	-49.284851	5160006	R. C-134	Av. C-208
...	...	...	...	...	...

Tabela 7.1: Tabela de Pontos do Reticulado com Nomes das Ruas

São necessárias algumas consultas por ponto do reticulado para se descobrir o nome das vias que formam o cruzamento a que cada ponto corresponde: as coordenadas de cada ponto são alteradas para que sejam consultados vários pontos próximos ao cruzamento e dessa forma se descubra o nome das duas vias que o formam.

O programa acessa as URLs mencionadas através da utilização da biblioteca *CURL* (6).

7.4 O Programa *Pontos_de_cada_rua*

Foi identificada a necessidade de se organizar os pontos do reticulado por ordem de rua. Ou seja, cada ponto representando cada um dos cruzamentos de cada rua pertencente

ao perímetro definido no trabalho deveria ser ordenado. Assim para cada rua deveriam ser ordenados todos os seus cruzamentos na ordem em que ocorrem na realidade. Um arquivo com essas informações seria útil posteriormente para se corrigir os caminhos a serem requisitados ao *Google Maps*.

Portanto, foi criado outro programa para se executar a tarefa descrita, o programa *Pontos_de_cada_rua*. Esse programa organiza os pontos do arquivo .TXT produzido pelo programa *Ruas* de forma a produzir um arquivo .TXT de saída onde os pontos estão em ordem de rua. Ou seja, o arquivo de saída é organizado de forma que para cada rua todos os pontos (cruzamentos de vias) que pertencem àquela rua são enumerados um em seguida do outro e em ordem. Assim é possível saber para a rua x as coordenadas de todos os seus cruzamentos na ordem que eles acontecem.

A tabela 7.2 representa a saída do programa descrito. Cada linha da tabela representa um ponto de uma rua. A primeira coluna contém a rua cujos pontos estão sendo enumerados, a segunda contém o número identificador (ID) do ponto da linha, as duas próximas contém as coordenadas do ponto, seguidas pela região censitária do ponto e as duas ruas que formam o cruzamento que o mesmo representa.

Rua Considerada	ID	Latitude	Longitude	Região Censitária	Primeira Rua	Segunda Rua
...	...	...	...	...	...	...
R.P-022	2190	-16.674311	-49.280041	5200028	R.P-013	R.P-022
R.P-022	2188	-16.672970	-49.280918	5200028	R.P-015	R.P-022
R.P-022	2175	-16.671740	-49.281761	5200015	R.P-022	Av.Indep.
Av.24deOut.	2189	-16.674660	-49.280548	5200028	R.P-013	Av.24deOut.
Av.24deOut.	2178	-16.673340	-49.281521	5200015	Av.24deOut.	R.P-015
Av.24deOut.	2173	-16.672810	-49.281849	5200015	Av.24deOut.	R.P-023A
Av.24deOut.	2174	-16.672041	-49.282391	5200015	Av.Indep.	Av.24deOut.
...	...	...	...	...	...	...

Tabela 7.2: Tabela de Pontos de Cada Rua

O programa funciona organizando em memória todos os pontos lidos do arquivo .TXT de entrada. Faz-se uma lista de ruas onde cada nó representa uma rua e tem um apontador para a lista de pontos daquela rua (todos os seus cruzamentos). Então cada ponto lido do arquivo de entrada é inserido nessa estrutura de acordo com a rua a qual pertence. Depois, os pontos são dispostos de acordo com a ordem mencionada. A partir dos pontos de uma rua, vai sendo obtida uma lista ordenada dos mesmos. É escolhido, para cada inserção na lista ordenada, o ponto da lista desordenada mais próximo do início ou do fim da lista ordenada. Assim a lista ordenada vai sendo obtida acrescentando-se pontos às suas extremidades. A distância mencionada é a geométrica, em linha reta, de um ponto a outro. O pseudo-código a seguir representa a estratégia de ordenação de pontos de ruas aqui discutida.

```
01  // Retira um nó qualquer da lista_desorganizada.
02  ponta_esquerda <- nó qualquer da lista_desorganizada;
03  // Nó retirado passa a ser a ponta_esquerda da lista_organizada.
04
```

```
05  nó_comparação_ORG <- ponta_esquerda;
06
07  Enquanto existir nó na lista_desorganizada, faça:
08  {
09      nó_retirado <- nó da lista_desorganizada mais próximo do no_comparação_ORG;
10
11      Se houver ponta_direita e a distância do nó_retirado em relação a ela for menor
12      do que a distância em relação à ponta_esquerda:
13      { local_de_inserção <- ponta_direita; }
14      Senão:
15      { local_de_inserção <- ponta_esquerda; }
16
17      Se o nó adjacente ao local_de_inserção for muito mais próximo (3 vezes mais)
18      do local_de_inserção do que o nó a ser inserido (nó_retirado):
19      {
20          // Este é o caso em que há um ponto da rua pertecente a uma alça.
21          Busca na lista_organizada os dois nós mais próximos do nó a ser inserido;
22
23          Se os dois nós forem adjacentes:
24          { Insere o nó_retirado entre os dois nós;}
25          Senão:
26          { Insere nó_retirado na lista de nós sem local;}
27      }
28      Senão: // Nó a ser inserido não pertence a uma alça.
29      {
30          Se local_de_inserção == ponta_direita :
31          {
32              Insere nó_retirado à direita da ponta_direita;
33              ponta_direita passa a ser o no_retirado recém inserido;
34              no_comparação_ORG <- ponta_direita;
35          }
36          Senão :
37          {
38              Insere nó_retirado à esquerda da ponta_esquerda;
39              ponta_esquerda passa a ser o no_retirado recém inserido;
40              no_comparação_ORG <- ponta_esquerda;
41
42              Se ponta_direita não estiver inicializada:
43              { ponta_direita <- antiga ponta_esquerda; }
44          }
45      }
46  }
```

Nas linhas 17 a 27 é mencionado o caso em que há "alça"em uma rua. Tais casos são como o apresentado na figura 7.6, rua em cinza. Esses casos foram tratados corretamente pelo programa escrito, resultando em listas de pontos corretamente ordenados. Os pontos

que formam as alças são inseridos na lista de pontos ordenados por último, no local correto, ou seja, entre os dois pontos da lista que forem mais próximos do ponto a ser inserido.

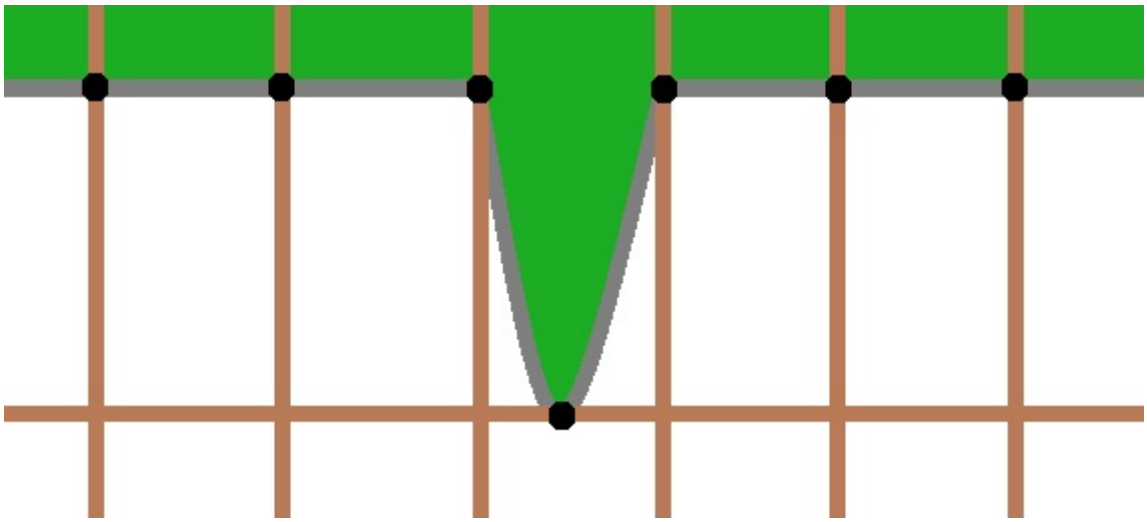

Figura 7.6: Rua com Presença de "Alça"

O pseudo-código apresentado foi implementado no programa discutido nesta seção. Conseguiu-se organizar os pontos de cada rua de forma condizente com a ordem real de ocorrência de cada cruzamento. O arquivo gerado foi convertido para o formato .KML e visualizado no *Google Earth* para conferência, não tendo sido identificado qualquer caso de ordenação incorreta. Dessa forma, o uso da estratégia descrita no pseudo-código acima foi apropriado para a situação deste trabalho.

Capítulo 8

Simulação de População

Para obter os dados necessários a este projeto foi preciso simular a população residente do perímetro considerado da cidade de Goiânia. A população simulada deveria ser similar à real, tanto numericamente quanto qualitativamente. Assim, cada região da cidade deveria conter uma quantidade de pessoas proporcional à sua população real. Da mesma forma a quantidade de pessoas de alto poder aquisitivo de cada setor também deveria ser proporcional à quantidade real. Com isso as rotas e caminhos a serem atribuídos a essas pessoas, em conjunto, formariam um banco de dados plausível em comparação à realidade.

Para atingir esses objetivos foram utilizadas informações do censo de 2010, disponíveis no site do IBGE (18), relativas à cidade de Goiânia. Dentre todas as características avaliadas no censo, foram selecionadas as que melhor e mais diretamente permitem deduzir a condição de renda dos moradores de determinada região, além é claro dos valores numéricos de população de cada setor da cidade. Os dados do censo foram especificados para cada setor censitário (seção 7.1), portanto foi possível traçar um perfil populacional para cada setor considerado no trabalho.

Infelizmente, dentre os dados disponíveis até a data deste trabalho não constavam informações de renda familiar específicas para cada setor censitário. Portanto, as informações utilizadas para se tentar deduzir o perfil de renda (classe econômica) foram: porcentagem de pessoas residentes de cada setor censitário que mora em domicílio quitado, porcentagem que mora em domicílio em aquisição e porcentagem que mora em domicílio alugado. Tais informações foram as mais adequadas, dentre todas as disponíveis, para se tentar avaliar a condição financeira das pessoas, visto que estão relacionadas a um bem (residência) cujo valor médio pode ser avaliado de acordo com a região em que se encontra.

Com os dados mencionados, para cada setor censitário e para cada situação de domicílio, foram atribuídas probabilidades para que uma pessoa moradora seja da classe econômica A, B ou C. Essas probabilidades levaram em consideração o valor dos imóveis novos e antigos de cada região, além dos valores de aluguel. Exemplificando: para a região 51200, caso a pessoa more em domicílio alugado, as probabilidades de ela ser das classes A, B e C poderiam ser 60, 30 e 10 % respectivamente, imaginando que a região 51200 seja nobre, com imóveis de alto valor e altas taxas de aluguel. Probabilidades como as mencionadas foram atribuídas para pessoas de cada região, de acordo com a situação e domicílio e o valor geral dos imóveis da região.

A noção geral dos preços de imóveis e aluguéis de cada região foi obtida através de

consultas ao site de anúncios de imóveis Lugar Certo (3). O método para se tentar deduzir as classes econômicas das pessoas foi portanto impreciso. Depende, até certo ponto, de uma avaliação subjetiva da probabilidade de pessoas de determinada condição financeira viverem em residências de determinada situação de aquisição em determinada região da cidade. Portanto os resultados não são idênticos à realidade. Mas são plausíveis, visto que, diante da impossibilidade de se obter os dados reais, foram baseados em informações reais.

Um resumo das atividades a serem descritas neste capítulo pode ser visualizado no diagrama da figura 8.1. Conforme a explicação localizada no início da seção 7.2, cada retângulo de determinada cor tem um significado específico, descrito na seção mencionada.

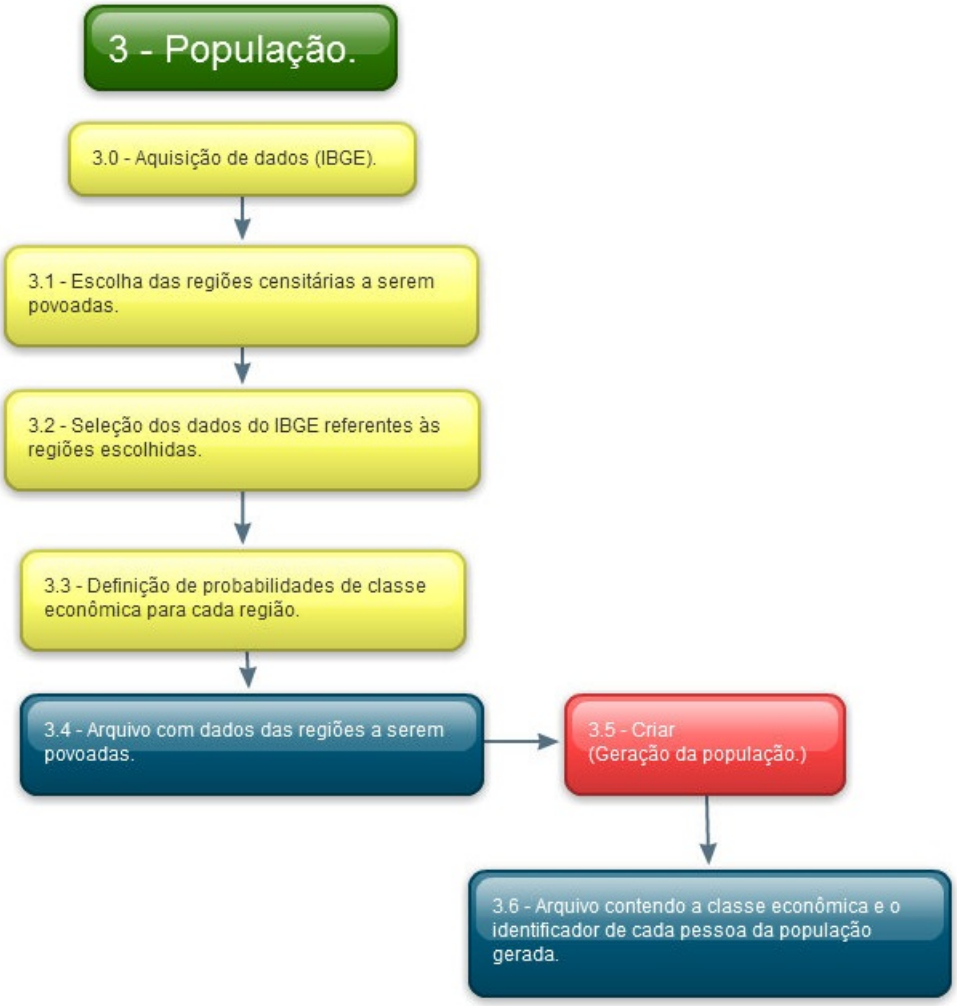

Figura 8.1: Diagrama de Simulação de População

8.1 Procedimentos Iniciais

Primeiramente foi aberto no programa *SAGA* o arquivo contendo as regiões censitárias que estão dentro do perímetro considerado neste trabalho. Em seguida, foi acessada uma aplicação oferecida pelo site do Censo 2010 (18) . Ela apresenta todas as regiões censitárias

de uma cidade e em código de cores denota quais regiões têm maior concentração de pessoas de acordo com a variável escolhida (que pode ser desde situação do domicílio até idade dos habitantes). A variável utilizada nesse caso foi a população absoluta. Com isso obteve-se a representação visual de quais setores censitários da cidade de Goiânia são mais densamente habitados (figura 8.2).

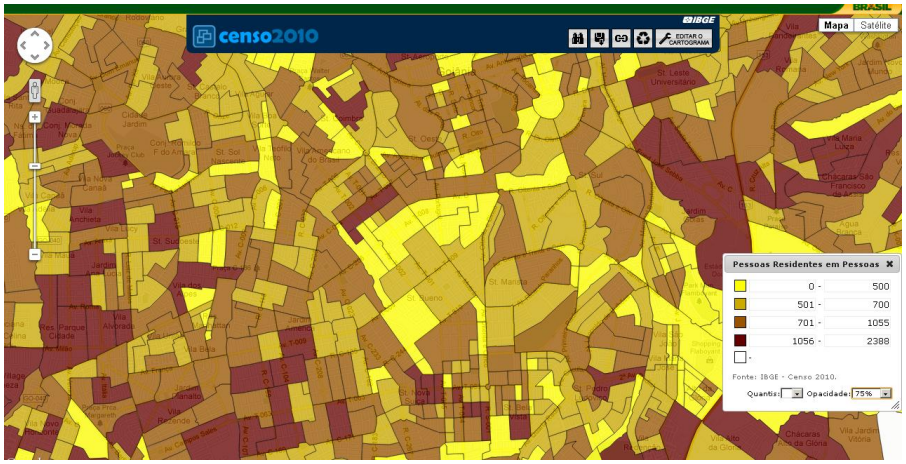

Figura 8.2: Aplicativo IBGE: Densidade Populacional de Setores Censitários

Com ajuda do aplicativo descrito as regiões mais populosas pertencentes ao perímetro da cidade considerado neste trabalho foram selecionadas. As outras regiões foram deletadas (isto apenas para o contexto da produção de uma população). O resultado dessa operação foi gravado em arquivo .SHP e foi também exportado para um arquivo em formato .KML para visualização no programa *Google Earth*. Além disso uma tabela contendo os nomes de todas as regiões escolhidas foi gravada em formato .CSV.

Em seguida foi feito download de um arquivo do site do Censo 2010 em formato .CSV contendo informações (todas as variáveis produzidas na pesquisa) sobre cada um dos setores censitários da cidade de Goiânia. Este arquivo foi aberto, assim como também foi aberto o arquivo contendo a lista das regiões escolhidas, através de um programa de planilhas. Essa lista de regiões escolhidas foi copiada para o arquivo de informações censitárias e por meio de funções do programa foi feita uma tabela cuja primeira coluna continha as regiões escolhidas e as demais colunas continham informações sobre cada região.

As informações ou variáveis escolhidas foram as já mencionadas: população, porcentagem de pessoas que vivem em domicílios alugados, domicílios em aquisição e domicílios quitados para cada setor censitário. Além dessas informações, para cada região e para cada tipo de status de aquisição de domicílio foram atribuídas probabilidades de os moradores serem da classe A, B ou C, conforme já discutido. Para uma correta atribuição de probabilidades foram utilizadas informações presentes no trabalho realizado por C. Neri, Marcelo e C. C. de Melo, Luisa (2) acerca da quantidade de pessoas de cada classe social existente nas capitais brasileiras. Foram também consultados anúncios de imóveis para venda e aluguel presentes no site Lugar Certo (3), como discutido no início deste capítulo.

Os dados resultantes foram gravados em um arquivo .TXT conforme representado na tabela 8.1. Nessa tabela, as probabilidades de cada pessoa ser de uma classe econômica dependem da região censitária onde ela reside e da situação de aquisição da residência

Região	Parcela da População	Imóveis Alugados /Cedidos	Chance Classes A B C	Imóveis em Aquisição	Chance Classes A B C	Imóveis Quitados	Chance Classes A B C
...	...	...	...	...	...	...	...
5090004	.011	.29	15 75 10	.13	25 65 10	.57	15 70 15
5090007	.010	.29	25 70 05	.16	35 60 05	.54	20 75 05
5090008	.009	.28	60 35 05	.22	80 15 05	.48	87 08 05
5090011	.012	.23	05 45 50	.02	07 68 25	.73	05 65 30
5100002	.014	.34	00 60 40	.02	03 72 25	.62	05 65 30
...	...	...	...	...	...	...	...

Tabela 8.1: Tabela de Informações sobre Regições de Moradia

onde ela mora. Cada linha apresenta os dados de probabilidade de uma região específica. Assim, na primeira linha, para a região 5090004: 1,1 % da população a ser gerada deve residir lá, 29 % das pessoas que lá residirem devem morar em residências alugadas ou cedidas e devem ter chance de 15, 75 e 10 % de serem das classes A, B e C repectivamente, etc.

8.2 Programa que Simula a População

Foi escrito um programa (*Criar*) que lê o arquivo descrito na tabela 8.1 e escreve em um outro arquivo .TXT uma lista com todas as pessoas geradas de acordo com as probabilidades constantes no arquivo lido. O arquivo de saída contém em cada linha um número identificador de uma pessoa gerada (ID), o código do setor censitário onde reside e sua classe econômica, conforme representado pela tabela 8.2.

Identificador de Cada Pessoa	Região Censitária de Moradia	Classe Econômica
...	...	...
40	5190002	A
41	5190002	B
42	5190002	B
43	5120010	B
...	...	...

Tabela 8.2: Tabela Representando o Arquivo que Contém a População

O programa funciona lendo cada linha do arquivo de entrada: as probabilidades e o número de pessoas de cada setor para cada status de aquisição de residência. Para cada região censitária são então criadas um número de pessoas correspondente ao número total de habitantes para aquele setor constante na linha lida. Dessas pessoas criadas, o número de residentes de imóveis quitados, alugados e em aquisição também segue as cifras constantes na linha lida, correspondentes às informações do Censo 2010. É então decidida a classe econômica de cada pessoa, utilizando-se as probabilidades também escritas na linha lida do arquivo de entrada e a função RAND() que produz números aleatórios.

Assim, de acordo com a região de residência, a condição de moradia e a probabilidade relacionada, cada pessoa recebe uma classificação de renda.

8.3 Classes A, B e C

De acordo com as rendas familiares características de cada classe econômica publicadas pela Associação Brasileira de Empresas de Pesquisa (ABEP) (1) e sobre os três perfis aquisitivos mencionados neste trabalho, foi definido que:

- A denominação Classe A se refere a pessoas de renda familiar superior a seis mil reais por mês. Ou seja, inclui pessoas das classes A1, A2 e parte das pessoas de classe B1 de acordo com o que foi publicado pela ABEP.
- Classe B diz respeito a pessoas de renda familiar superior a dois mil reais e inferior a seis mil reais, incluindo assim algumas pessoas da classe C1, pessoas da classe B2 e grande parte das pessoas da classe B1.
- Classe C se refere a pessoas de renda familiar inferior a dois mil reais, compreendendo pessoas das classes E, D, C2 e parte das pessoas da classe C1.

As pessoas de classe C (conforme definido acima no contexto deste trabalho), geradas pelo programa descrito, foram desconsideradas no trabalho. Isso foi feito porque se trata de pessoas que consomem menos produtos e praticamente ainda não fazem compras através de smartphones, em sua maioria. Foram portanto excluídas dos bancos de dados pois apenas ocupariam mais espaço neles, dificultando o processo de mineração de dados sem em contrapartida gerarem informações tão significativas quanto as produzidas pelas demais pessoas.

Capítulo 9

Requisição de Rotas e Simulação de Rotinas

Para que se pudesse obter os dados de deslocamento da população gerada, foi necessário criar um banco de dados de rotas ou caminhos. Cada caminho compreende uma sequência de pontos que se percorridos do primeiro ao último condizem com o caminho que uma pessoa faria se utilizasse um automóvel para ir de um ponto a outro na cidade de Goiânia, os pontos inicial e final do caminho. Tais rotas foram solicitadas ao *Google Maps* por meio de URLs contendo os pontos final e inicial de cada caminho.

Os procedimentos seguidos para a obtenção de um banco de rotas ou caminhos suficiente para a simulação do deslocamento da populção gerada serão discutidos a seguir.

Ao final do capítulo será também detalhada a forma como o banco de rotas foi utilizado na simulação da rotina semanal de deslocamento de cada pessoa da população. Os procedimentos utilizados na simulação dos hábitos de consumo de cada pessoa serão também apresentados.

9.1 Banco de Rotas

Os procedimentos descritos nesta seção têm por objetivo a aquisição de rotas em número e qualidade suficientes para permitir a simulação do deslocamento da população gerada anteriormente. Isso levando-se em conta a distribuição espacial das residências das pessoas e suas classes sociais. As atividades e programas necessários para tanto estão representados no diagrama da figura 9.1 e serão explicados com detalhes nas subseções a seguir.

O significado da maioria das cores presentes no diagrama está explicitado na seção 7.2. Quanto aos retângulos verde-claro, representam diagramas discutidos em capítulos anteriores e as setas que saem deles representam arquivos gerados nos diagramas por eles representados. Cada uma dessas setas tem inscrições que identificam o arquivo de que tratam.

9.1.1 Pontos de Interesse

O primeiro passo para a solicitação das rotas ou caminhos necessários foi a identificação de diversos pontos de interesse dentro do perímetro considerado da cidade. Para uma

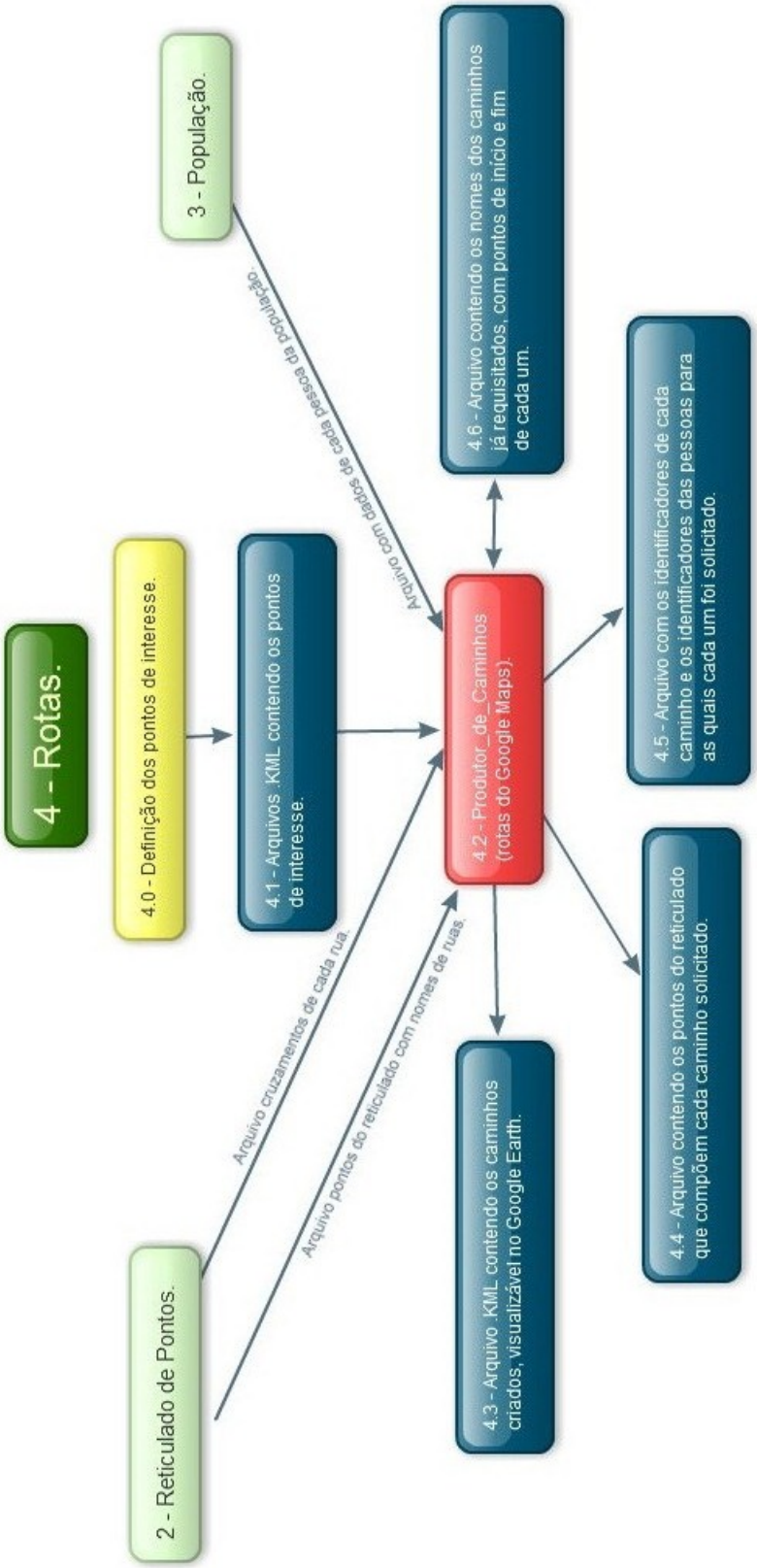

Figura 9.1: Diagrama de Aquisição de Rotas

simulação válida do deslocamento da população considerada foi necessário marcar pontos correspondentes à localização geográfica:

- Dos restaurantes pertencentes ao perímetro considerado no trabalho, pois são locais muito frequentados pela população tanto para lazer no fim de semana quanto para almoço durante a semana e *happy hour* após o trabalho.
- Dos locais pertencentes ao perímetro considerado com maior concentração de escritórios, empresas e órgãos públicos, pois são nessas regiões que trabalham a maior parte da população.
- Dos pontos da cidade internos à região considerada no trabalho com estruturas de lazer, pois são esses pontos que representam o destino da maioria das pessoas que saem de casa a lazer nos fins de semana.
- De cada uma das regiões censitárias cujas populações foram simuladas. Cada região recebeu um ponto (localizado em sua área mais central) que representa o local de residência das pessoas geradas para aquela região. Estes pontos são o início de grande parte dos caminhos a serem requisitados ao *Google Maps*, pois representam a residência de cada uma das pessoas.

Figura 9.2: Locais de Trabalho, Classe B

A maior parte dessas informações (localização de restaurantes, órgãos públicos e locais de lazer) foram obtidas na lista telefônica da cidade de Goiânia. No próprio programa *Google Earth*, bastante utilizado em várias etapas deste projeto, já se encontram indicados muitos desses locais.

Os pontos foram marcados com auxílio do *Google Earth* e gravados em arquivos de extensão .KML (figura 9.2). Foram realizadas marcações de pontos (locais de trabalho,

lazer e restaurantes) no contexto de deslocamento das pessoas de classe A e também no contexto de deslocamento das pessoas de classe B, gerando com isso arquivos diferentes para cada classe. No entanto, apesar de diferentes, tais arquivos têm grande intersecção (muito pontos em comum) visto que vários restaurantes e locais de lazer são frequentados por ambas as classes e vários locais de trabalho oferecem opções de trabalho também para ambas as classes econômicas.

9.1.2 Programa Produtor de Caminhos

O próximo passo para a criação do banco de dados de caminhos foi a escrita de um programa para requisitá-los um a um ao *Google Maps*. O programa também processa cada caminho obtido de forma a gerar caminhos que contenham apenas pontos pertencentes ao reticulado de pontos. Por fim, grava em arquivos .TXT os resultados gerando também arquivos .KML para visualização no programa *Google Earth*.

Arquivos de Entrada

- Arquivo contendo a população gerada (com identificadores, classes econômicas e regiões censitárias de residência de cada pessoa, conforme tabela 8.2).
- Arquivo contendo o reticulado de pontos (com coordenadas e região censitária de cada ponto, conforme tabela 7.1).
- Arquivo contendo todos os pontos, ordenados, de cada rua pertencente ao perímetro do trabalho, conforme tabela 7.2.
- Arquivos .KML contendo os pontos de interesse para a criação de caminhos (localização de restaurantes, locais de lazer e locais de trabalho relativos a ambas as classes econômicas tratadas).

Funcionamento Geral

Inicialmente são lidos os arquivos .KML com os pontos de interesse. Cada um dos pontos dos arquivos .KML é inserido em um vetor apropriado (vetor de locais de trabalho da classe A, por exemplo).

Em seguida o arquivo com os pontos do reticulado é lido e armazenado em uma matriz. Cada célula da matriz contém uma lista pontos pertencentes a uma faixa específica de coordenadas. Isso foi feito para agilizar a busca por um ponto determinado: primeiro se encontra a faixa de coordenadas a qual pertence o ponto buscado e depois se analisa a lista de pontos da célula correspondente.

É também criada uma lista geral de pontos, contendo apontadores para cada um dos pontos da matriz descrita. Através dessa lista e da leitura do arquivo de pontos de cada rua (seção 7.4) é criada uma outra lista, dessa vez de ruas. Cada nó dessa lista final contém o nome de uma rua e a lista de pontos a ela pertencentes, em ordem.

A partir daí começa a leitura do arquivo de pessoas. A cada três pessoas lidas que sejam da mesma classe econômica e moradoras da mesma região censitária são selecionados aleatoriamente (através da função RAND()) pontos escolhidos nos vetores de pontos de

interesse (locais de trabalho, restaurantes, etc). A cada ponto escolhido são solicitadas ao *Google Maps* as rotas correspondentes.

Os tipos de pontos de interesse selecionados aleatoriamente e os tipos de caminhos requisitados para cada três pessoas de mesma classe econômica e região são os seguintes:

- Um ponto relativo ao local padrão de residência da região censitária das três pessoas selecionadas. Este ponto é localizado no vetor de pontos que representam os locais de residências das pessoas para cada região censitária. Representará portanto o ponto de moradia das três pessoas na cidade.
- Um ponto representando o local de trabalho das três pessoas e um caminho do local de moradia até esse ponto.
- Quatro pontos diferentes entre si representando restaurantes da cidade compatíveis com a classe econômica das três pessoas e os caminhos entre a residência das pessoas e cada um desses restaurantes.
- Quatro pontos distintos relativos a locais de lazer da cidade compatíveis com a classe econômica das pessoas e os caminhos entre eles e o local de moradia das mesmas.
- Dois pontos diferentes, não muito distantes do local de trabalho escolhido (as distâncias são avaliadas pelo programa), representando locais para almoço das pessoas e os caminhos entre o local de trabalho e cada ponto.
- Quatro pontos, relativos a restaurantes, compatíveis com a classe econômica das pessoas que representarão opções de *happy hour* e os caminhos de ida do trabalho a esses lugares e de volta desses lugares até a residência das pessoas.

Na maior parte dos casos não foram requisitados caminhos de volta, apenas caminhos de ida. Tal simplificação foi feita devido ao fato de a maioria das ruas da cidade de Goiânia ser de mão dupla, ou seja, as rotas de ida são quase idênticas às rotas de volta. Outra vantagem dessa estratégia é a diminuição das consultas ao *Google Maps*, gerando economia de tempo.

Os caminhos descritos são então gravados em arquivos e referenciados posteriormente por outro programa. Esse programa define através de escolhas aleatórias (usando a função RAND()) a rotina de deslocamento de cada pessoa do arquivo de pessoas durante um mês. Como a quantidade de caminhos acima descrita é criada a cada três pessoas com pontos de destino diferentes, as rotas obtidas são suficientemente variadas para permitir a geração de uma rotina plausível de deslocamento para a população.

Obtenção e Correção das Rotas

As rotas são solicitadas ao *Google Maps* utilizando-se bibliotecas do *CURL* (6) que permitem o acesso a *URLs* através de programas na linguagem C. Cada resposta do serviço web é gravada em um arquivo .XML que é lido pelo programa em tempo de execução. Cada ponto de cada rota é então gravado em uma lista.

No entanto ao plotar os pontos obtidos em cada consulta de rota observou-se que seu número era insuficiente para esta aplicação. No contexto deste trabalho seria necessário obter um ponto para cada cruzamento de vias de cada rota. O que o *Google Maps* retorna

é simplesmente um ponto para cada mudança de via, ou seja, o ponto de entrada em uma e rua e o ponto de saída da mesma, para cada rua percorrida na rota requisitada.

Dessa forma foi necessário criar funções que corrigissem as rotas retornadas pelo *Google* inserindo nelas os pontos necessários que não foram retornados pelo serviço. Assim é enviada para essas funções a lista inicial de pontos obtidos através do *Google Maps* e elas completam a lista com os pontos que faltaram.

São duas as estratégias utilizadas por essas funções. A primeira estratégia consiste em varrer a lista de pontos obtida através da consulta ao *Google Maps*. Os pontos são considerados dois a dois, na ordem que foram retornados pelo *Google*. Para cada par de pontos consecutivos da lista são encontrados os dois pontos do reticulado que mais se aproximam deles, ou seja, os dois cruzamentos de vias mais próximos. São então comparados os nomes das vias que se intersectam nesses dois cruzamentos a fim de se descobrir qual rua foi percorrida no trajeto entre os dois pontos considerados da lista inicial. A rua que pertencer aos dois cruzamentos simultaneamente terá sido a rua percorrida entre os dois pontos da lista. Os nomes das vias de cada cruzamento (ou ponto do reticulado) estão organizados em memória em um vetor.

Descoberto o nome da via percorrida são então consultados todos os seus cruzamentos (constantes na lista de pontos de cada rua que foi armazenada em memória). Todos os cruzamentos da rua identificada que estiverem localizados entre os dois pontos inicialmente considerados são adicionados à lista de pontos da rota corrente em ordem de ocorrência. O processo se repete até que todos os pontos retornados pelo *Google* sejam analisados e todos os cruzamentos contidos na rota consultada sejam adicionados à lista.

O pseudo-código abaixo representa a estratégia discutida acima. Os pontos inicial e final de uma parte de uma rota (a ser complementada em uma iteração) são representados respectivamente por ponto_anterior e ponto_atual. São procurados os dois pontos da lista de pontos da rua identificada que mais se aproximam dos pontos mencionados (ponto_anterior e ponto_atual), ponto_LR_anterior e ponto_LR_atual respectivamente. Assim, no tocante à lista de pontos de uma rua determinada, os pontos ponto_LR_anterior e ponto_LR_atual são respectivamente o início e o fim da fração da rota considerada.

```
01  // Percorre a lista de pontos da rota retornada pelo Google, dois a dois.
02
03  ponto_anterior <- lista; // Primeiro ponto da lista.
04  ponto_atual <- lista->prox; // Segundo ponto da lista.
05
06  Enquanto não chegar ao fim da lista de pontos:
07  {
08      Identifica a rua que passa pelos pontos atual e anterior através dos nomes das
09      ruas que formam os cruzamentos representados por esses pontos;
10
11      Localiza a lista de pontos (cruzamentos) da rua no vetor de ruas em memória;
12      Localiza na lista de pontos da rua o ponto mais próximo do ponto_atual e o
13      mais próximo do ponto_anterior: ponto_LR_atual e ponto_LR_anterior;
14
15      Verifica o sentido de percorrimento dos pontos da rua, ou seja, se o sentido do
```

```
16      ponto anterior ao atual é o mesmo do que se observa na lista de pontos da rua
17      ou oposto;
18
19      Se ponto_LR_atual == ponto_LR_anterior:
20      {
21          Se ponto_LR_atual != ponto_anterior E ponto_LR_atual != ponto_atual:
22          {
23              Se ponto_LR_atual estiver entre os pontos atual e anterior:
24              { Insere ponto_LR_atual, ordenadamente, na lista de pontos da rota;}
25              Senão, não insere ponto_LR_atual na lista.
26
27              // Vai avançar a execução para os próximos pontos da lista de pontos
28              // da rota do Google.
29          }
30      }
31      Senão: //ponto_LR_atual != ponto_LR_anterior
32      {
33          Se ponto_LR_anterior != ponto_anterior :
34          {
35              Se ponto_LR_anterior é cruzamento posterior ao ponto_anterior:
36              { Adiciona ponto_LR_anterior à rota; }
37              Senão, não adiciona ponto_LR_anterior à rota.
38          }
39
40
41          Percorre os pontos (cruzamentos) da lista referente à rua identificada, no
42          sentido de percorrimento identificado (oposto ou não ao da lista de pontos
43          da rua), de ponto_LR_anterior até o ponto anterior ao ponto_LR_atual:
44          {
45              Adiciona cada ponto, ordenamente, à lista de pontos da rota;
46          }
47
48
49          Se ponto_LR_atual é anterior ao ponto_atual:
50          {
51              Adiciona ponto_LR_atual antes de ponto_atual na lista de pontos da
52              rota;
53          }
54          Senão, não adiciona ponto_LR_atual à lista.
55      }
56
57      // Avança para o próximo ponto.
58      ponto_anterior <- ponto_atual;
59      ponto_atual <- ponto_atual->prox;
60  }
```

A segunda estratégia é a de procurar na matriz que contém o reticulado todos aqueles pontos que estiverem a uma distância pequena (previamente determinada) de cada uma das retas que passam por cada dois pontos retornados pelo *Google Maps*. Ou seja, a cada dois pontos consecutivos da lista de pontos enviada às funções é feita uma reta. Todos os pontos próximos dessa reta são inseridos entre os dois pontos que formaram a reta na lista de pontos, em ordem. Tal ação é repetida para todos os pontos dois a dois na ordem que ocorrem na lista inicial de pontos de cada caminho retornada pelo *Google*. Ao final é obtida uma lista mais completa de pontos. A lista contém não só os pontos de entrada e saída de cada rua, como também um ponto para cada cruzamento de vias percorrido pertencente a cada rua transitada.

A estratégia discutida acima está representada pelo pseudo-código abaixo.

```
01  // Percorre a lista de pontos da rota retornada pelo Google, dois a dois.
02
03  ponto_anterior <- lista; // Primeiro ponto da lista.
04  ponto_atual <- lista->prox; // Segundo ponto da lista.
05
06  Enquanto não chegar ao fim da lista de pontos:
07  {
08      Calcula coeficientes da reta que passa por ponto_anterior e ponto_atual;
09
10      Para todos os pontos do reticulado que estão na faixa de coordenadas dos pontos
11      atual e anterior:
12      {
13          Calcula a distância de cada ponto à reta;
14
15          Se o ponto corrente estiver a menos de 10 metros da reta:
16          {
17              Adiciona o ponto, ordenadamente, à lista de pontos da rota enviada
18              pelo Google entre os pontos atual e anterior;
19          }
20      }
21
22      // Avança para o próximo ponto da lista de pontos da rota enviada pelo Google.
23      ponto_anterior <- ponto_atual;
24      ponto_atual <- ponto_atual->prox;
25  }
```

A primeira estratégia é eficaz para os casos em que se consegue identificar o nome da rua percorrida a cada dois pontos da rota retornada pelo *Google Maps*. No caso em que não é identificada a rua é utilizada a segunda estratégia de busca de pontos.

Portanto, a segunda estratégia em grande parte dos casos não é utilizada. Isso se deve ao fato de que ela só é eficaz para ruas retas. No caso de ruas curvas são encontrados pontos que não pertencem ao traçado da via (a busca de pontos é feita em relação a uma linha reta). Assim esse método só é utilizado em duas circunstâncias. Quando a primeira estratégia falha ou quando analisada a lista final obtida pelo primeiro método se identifica

um espaço muito grande entre dois pontos. Nesse caso final utiliza-se a segunda estratégia para tentar encontrar cruzamentos entre os dois pontos distantes.

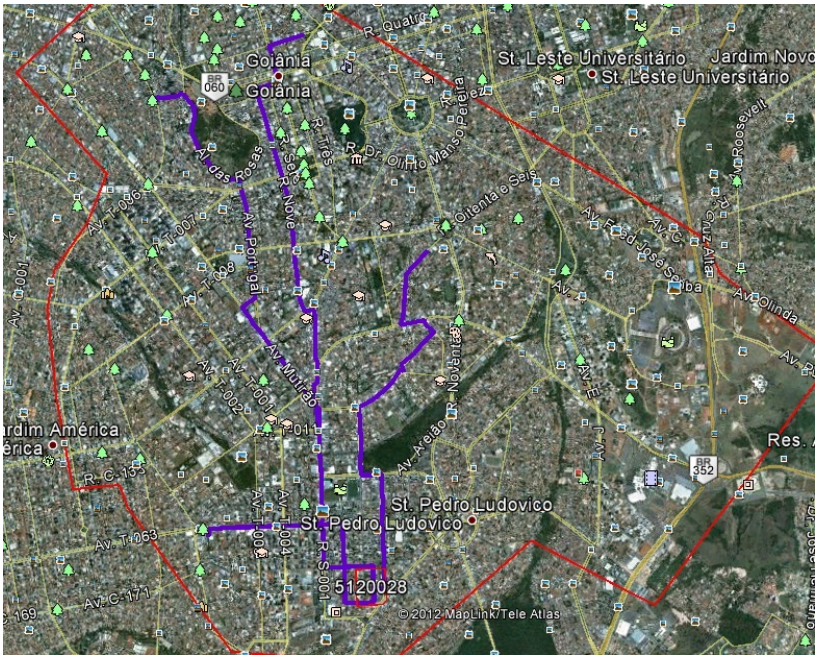

Figura 9.3: Exemplos de Caminhos, Moradores da Região 5120028

Assim se obtém uma lista final de pontos completa, contendo todos os cruzamentos percorridos em cada rota requisitada, como na figura 9.3.

Arquivos de Saída

Identificador do Caminho	Nome	Longitude do Ponto	Latitude do Ponto	Região Censitária do Ponto
...	...	...	...	...
8035	5040002BR300000	-49.28153	-16.70048	5160039
8035	5040002BR300000	-49.28215	-16.70098	5160039
8035	5040002BR300000	-49.28215	-16.70098	5160039
8035	5040002BR300000	-49.28153	-16.70048	5160039
8036	5040002BL200000	-49.26434	-16.68495	5040002
8036	5040002BL200000	-49.26434	-16.68495	5040002
...	...	...	...	...

Tabela 9.1: Tabela que Representa o Arquivo de Pontos de cada Caminho

- Arquivo no formato .KML contendo todas as rotas requisitadas, visualizável no programa *Google Earth*, utilizado para conferir por amostragem das rotas recebidas.
- Arquivo .TXT contendo a lista de pontos, em ordem, referentes a cada caminho requisitado. Cada linha do arquivo contém o identificador e o nome do caminho,

ID da Rota	Coordenadas do Ponto Inicial	Coordenadas do Ponto Final	Nome do Caminho	Interpretação do Nome do Caminho
...	...	...	...	...
4	-16.681557 -49.280346	-16.689238 -49.263641	5190002BR3	B Restaurante 3
5	-16.681557 -49.280346	-16.712336 -49.259018	5190002BR4	B Restaurante 4
6	-16.681557 -49.280346	-16.674917 -49.264389	5190002BL1	B Lazer 1
7	-16.681557 -49.280346	-16.671625 -49.269531	5190002BL2	B Lazer 2
...	...	...	...	...

Tabela 9.2: Tabela Representando o Arquivo de Nomes dos Caminhos

Região de Moradia de Cada Pessoa	Identificador da Pessoa	Classe Econômica	Identificador do Caminho
...	...	...	...
5130030	907	A	4521
5130030	900	B	4522
...	...	...	...

Tabela 9.3: Tabela de Pessoas para as quais cada Caminho foi Requisitado

as coordenadas e a região censitária à qual o ponto pertence, conforme tabela 9.1. Para cada caminho o número de linhas é igual ao número de pontos que formam o mesmo, sendo cada linha referente a um ponto do caminho.

- Arquivo .TXT contendo o identificador e o nome de cada caminho, as coordenadas dos seus pontos final e inicial e a interpretação do nome de cada caminho (a classe econômica das pessoas para as quais ele foi feito, o tipo do caminho, ou seja, trabalho, restaurante, lazer, entre outras informações), representado na tabela 9.2.
- Arquivo .TXT contendo o identificador de cada pessoa, a região onde reside, sua classe econômica e o identificador dos caminhos requisitados para o grupo de pessoas ao qual ela pertence (cada grupo com três integrantes moradores da mesma região censitária e da mesma classe econômica), conforme demonstrado na tabela 9.3.

9.2 Simulação de Rotinas

Nesta seção será discutido o método utilizado para gerar o banco de dados contendo todos os caminhos realizados por todas as pessoas durante um mês. Será também apresentada a estratégia utilizada na simulação dos hábitos de consumo da população, ou seja, na decisão de quais e quantos produtos cada pessoa comprou através do celular.

O diagrama da figura 9.4 resume a sequência de execução de programas necessários à simulação das rotinas mencionadas. Explicita também quais arquivos são utilizados para tanto e os arquivos gerados pelos programas em questão. As cores de cada retângulo do diagrama têm a mesma semântica das cores presentes nos diagramas das seções 7.2 e 9.1.

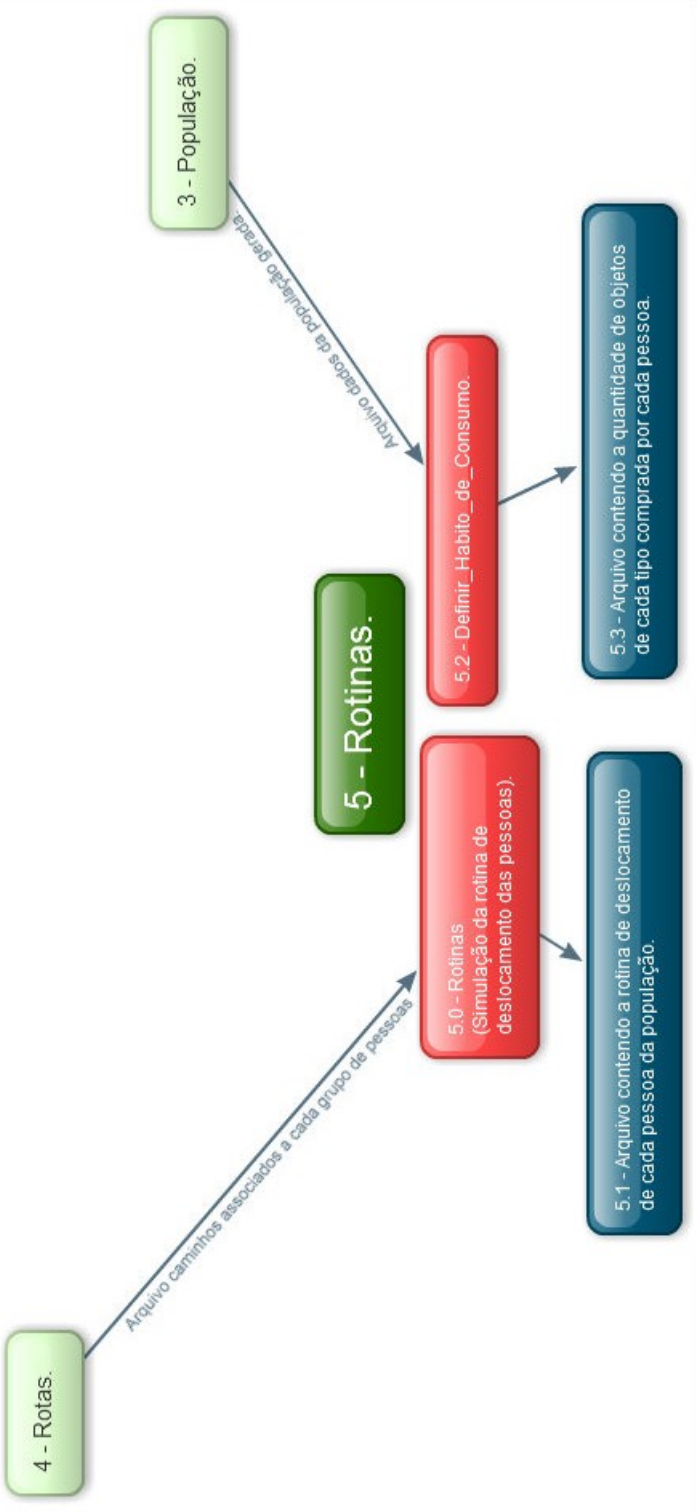

Figura 9.4: Diagrama de Simulação de Rotinas

9.2.1 Rotinas de Deslocamento

Para gerar um conjunto de dados similar à realidade acerca do deslocamento das pessoas que compõem a população do trabalho, foi necessária a utilização de um dos

principais arquivos gerados pelo Programa *Produtor_ de_ Caminhos*. O arquivo utilizado foi o que reúne o ID de cada uma das pessoas seguido do ID dos caminhos ou rotas atribuíveis a cada uma delas (tabela 9.3).

A simulação das rotinas foi feita através de um programa escrito em linguagem C de nome *Rotinas*. Ele escolhe caminhos (utilizando a função RAND()) de acordo com certas regras e probabilidades definidas diferentemente para pessoas de classe A e pessoas de classe B. Escolhidos os caminhos ele os atribui a cada uma das pessoas em determinado dia da semana e horário. Dessa forma é simulada a rotina de um mês inteiro das pessoas.

Mais especificamente o arquivo utilizado, aberto logo no início do programa, contém em cada linha o ID de cada uma das pessoas da população, suas classes econômicas e os caminhos requisitados ao *Google Maps* relativos a cada grupo de três pessoas moradoras da mesma região censitária e de mesma classe econômica. Os IDs dos caminhos são lidos ordenamente para cada grupo de pessoas.

A partir daí o programa gera a rotina mensal de cada uma das pessoas pertencentes ao grupo considerado utilizando os caminhos lidos também relativos ao grupo atual. De segunda a sexta feira são atribuídos caminhos de ida e volta ao local de trabalho escolhido para o grupo, caminhos relativos aos locais de almoço e restaurantes frequentados como forma de lazer após o expediente de trabalho. Nos fins de semana podem ou não ser atribuídos caminhos para o trabalho no sábado e são atribuídos caminhos para locais de lazer e restaurantes.

A atribuição de um caminho é feita de acordo com alguma probabilidade, definida em função do dia da semana e da classe econômica da pessoa que o executa (algumas dessas probabilidades serão discutidas na seção 10.6). A chance de uma pessoa de classe A sair a lazer para um restaurante na noite de quinta feira por exemplo é maior do que a chance de uma pessoa de classe B fazer o mesmo. Cada uma das opções de caminhos (quando há opções) para um determinado horário de determinado dia é escolhida aleatoriamente, utilizando-se a função RAND(), mas obedecendo as probabilidades associadas.

A cada caminho atribuído também está associado um horário. Este horário é decidido dentro de uma faixa de possibilidades também através da utilização da função RAND().

Dessa forma, para cada pessoa e a cada dia da semana, uma quantidade de caminhos é escolhida com uma probabilidade que varia de acordo com a classe econômica da mesma. Os horários de realização dos caminhos também variam aleatoriamente dentro de faixas definidas.

Assim, as rotinas criadas procuram ser similares à realidade, sendo definidas de acordo com cada pessoa de cada classe econômica. São rotinas que envolvem variados caminhos executados em horários variáveis, simulando de maneira razoavelmente adequada a rotina real de um morador da cidade de Goiânia.

As rotinas são gravadas em um arquivo .TXT de saída do programa que contém em cada linha o ID da pessoa que executou o caminho, a região censitária onde essa pessoa mora, sua classe econômica, o dia da semana e horário de partida, ID do caminho, se ele é o caminho de ida ou de volta (caracter I ou V no fim da denominação do caminho) e uma cadeia de caracteres com a denominação do caminho ("trab"para a rota residência - trabalho; "almcs"para a mesma rota, mas em horário de almoço; "hh"para a rota entre o trabalho e um restaurante a título de *happy hour*; entre outras denominações), conforme representado pela tabela 9.4.

Região Censitária da Pessoa	ID e Classe da Pessoa	ID da Rota	Dia da Semana e Horário	Natureza da Rota
...	...	...	...	...
5120010	64 A	366	4 0740	N trabI
5120010	64 A	370	4 1230	N almcsI
5120010	64 A	370	4 1430	I almcsV
5120010	64 A	366	4 1810	I trabV
5120010	64 A	366	5 0720	N trabI
5120010	64 A	370	5 1230	N almcsI
5120010	64 A	370	5 1410	I almcsV
5120010	64 A	376	5 1840	N hhI
5120010	64 A	377	5 2130	N hhV
...	...	...	...	...

Tabela 9.4: Tabela de Rotinas de Deslocamento das Pessoas

9.2.2 Rotinas de Consumo

Segundo levantamento realizado pela empresa EBIT (que é referência no fornecimento de informações sobre o *e-commerce* no Brasil) e referenciado no site (8), os produtos mais consumidos pela internet são eletrodomésticos (15 %), produtos de informática (12 %), eletrônicos (8 %) entre outros gêneros.

Baseado nas informações mencionadas foi decidido que os produtos a serem consumidos pelas pessoas no contexto deste trabalho seriam: livros, câmeras fotográficas, celulares, videogames, relógios, *notebooks*, cupons de sites de compra coletiva, cafeteiras, televisões, fogões, geladeiras, microondas, máquinas de lavar roupas e liquidificadores.

Foi criado um arquivo em formato .TXT contendo o nome de cada produto seguido por uma matriz de 3 colunas e 4 linhas (tabela 10.7).

As colunas representam as compras realizadas por cada classe: primeira coluna, compras da classe A, segunda coluna, compras da classe B e terceira coluna, compras da classe C. Cada linha representa um quantidade de produtos daquele gênero comprada no período de um ano. primeira linha representa zero produtos comprados, a segunda linha representa um produto comprado e assim por diante.

As células de cada matriz representam a probabilidade de uma pessoa da classe social correspondente à coluna comprar um quantidade correspondente à linha de produtos do gênero descrito pela matriz no período de 1 ano. Essas probabilidades foram arbitradas em parte com base nas pesquisas citadas, levando em conta a classe social das pessoas e o tipo de produto de cada matriz. Tais probabilidades serão mais discutidas a frente na seção sobre regras e padrões dos dados (10.6).

Em seguida foi escrito o programa *Definir_ Hábito_ de_ Consumo*. Ele lê o documento .TXT de matrizes e armazena em uma lista cada uma delas. O programa também lê o arquivo .TXT contendo cada uma das pessoas (seus números de identificação, região onde moram e classe social, conforme tabela 8.2) consideradas neste trabalho. Em seguida grava um terceiro arquivo contendo os identificadores de cada uma das pessoas, o identificador de cada um dos produtos por elas comprados e a quantidade comprada de cada produto. Cada linha do arquivo contém portanto um identificador de uma pessoa, um identificador

de um produto e quantidade de produtos daquele tipo comprados por aquela pessoa no último ano, conforme representado na tabela 9.5.

ID da Pessoa	ID do Objeto	Quantidade Comprada
...	...	...
1	7	3
1	10	1
1	13	1
1	14	1
2	3	2
2	4	1
...	...	...

Tabela 9.5: Tabela do Hábito de Consumo da População

A quantidade de cada produto comprada por cada pessoa é decidida com base nas probabilidades escritas nas células da matriz do produto utilizando-se a função RAND(). O programa grava ainda um segundo arquivo .TXT no qual cada linha contém um identificador de um produto e seu nome, ou seja, uma lista de todos os produtos e seus identificadores.

Dessa forma se definiu os hábitos de consumo de cada uma das pessoas da população gerada: de acordo com o tipo de produto comprado e a classe econômica do comprador.

Capítulo 10

Tabelas para Mineração de Dados

Depois de simular os hábitos de deslocamento e consumo da população gerada para este trabalho o próximo passo foi processar os dados obtidos. Primeiramente foi realizada o processamento concomitante dos múltiplos bancos de dados, reunindo as informações mais relevantes em um conjunto limitado de tabelas (integração). Em seguida, durante a conversão dessas tabelas para o formato .ARFF, legível ao programa *Weka 3*, foram desconsiderados os dados inconsistentes ou dispensáveis (limpeza), como rotas excessivamente longas e que ocorrem com baixa frequência ou pontos de movimento zero, e amostrados os dados mais relevantes ao processo de mineração de dados (seleção) (14).

As ações e programas executados para se obter tais resultados estão resumidas nos diagramas das figuras 10.1 e 10.2 e serão discutidos com mais detalhes nas seções a seguir.

As cores de cada retângulo dos diagramas seguintes indicam o mesmo que as cores dos diagramas das seções 7.2 e 9.1. Quanto aos retângulos roxos, eles agregam arquivos, atribuem códigos a eles e os distribuem para os programas que os utilizam. Cada seta que sai dos retângulos roxos tem um código que identifica a quais arquivos a mesma se refere.

No caso hipotético deste trabalho ser aplicado em situações reais, a maioria dos procedimentos descritos neste capítulo poderiam ser utilizados. As tabelas descritas, resultantes da reunião dos dados mais relevantes no contexto de cada uma delas, também poderiam ser geradas e utilizadas na mineração de dados reais, o que poderia levar à obtenção de importantes resultados.

Ao final deste capítulo serão também apresentados os padrões deliberadamente inseridos nos dados e tabelas geradas. Eles serão enumerados e será discutida a relevância da existência de cada um deles para o processo de avaliação dos algoritmos a serem utilizados na fase de mineração dos dados.

10.1 Tabelas de Fluxo de Pessoas em Cada Ponto

O primeiro tipo de tabela de dados produzido foi resultante da contagem do número de pessoas que transitaram por cada ponto do reticulado (capítulo 7) em cada dia da semana e horário.

Foram produzidas 14 tabelas, cada uma levando em consideração a movimentação dos compradores de determinado produto. Dessa forma a tabela 1 foi feita relativa ao movimento dos compradores do produto de identificador 1, e assim por diante.

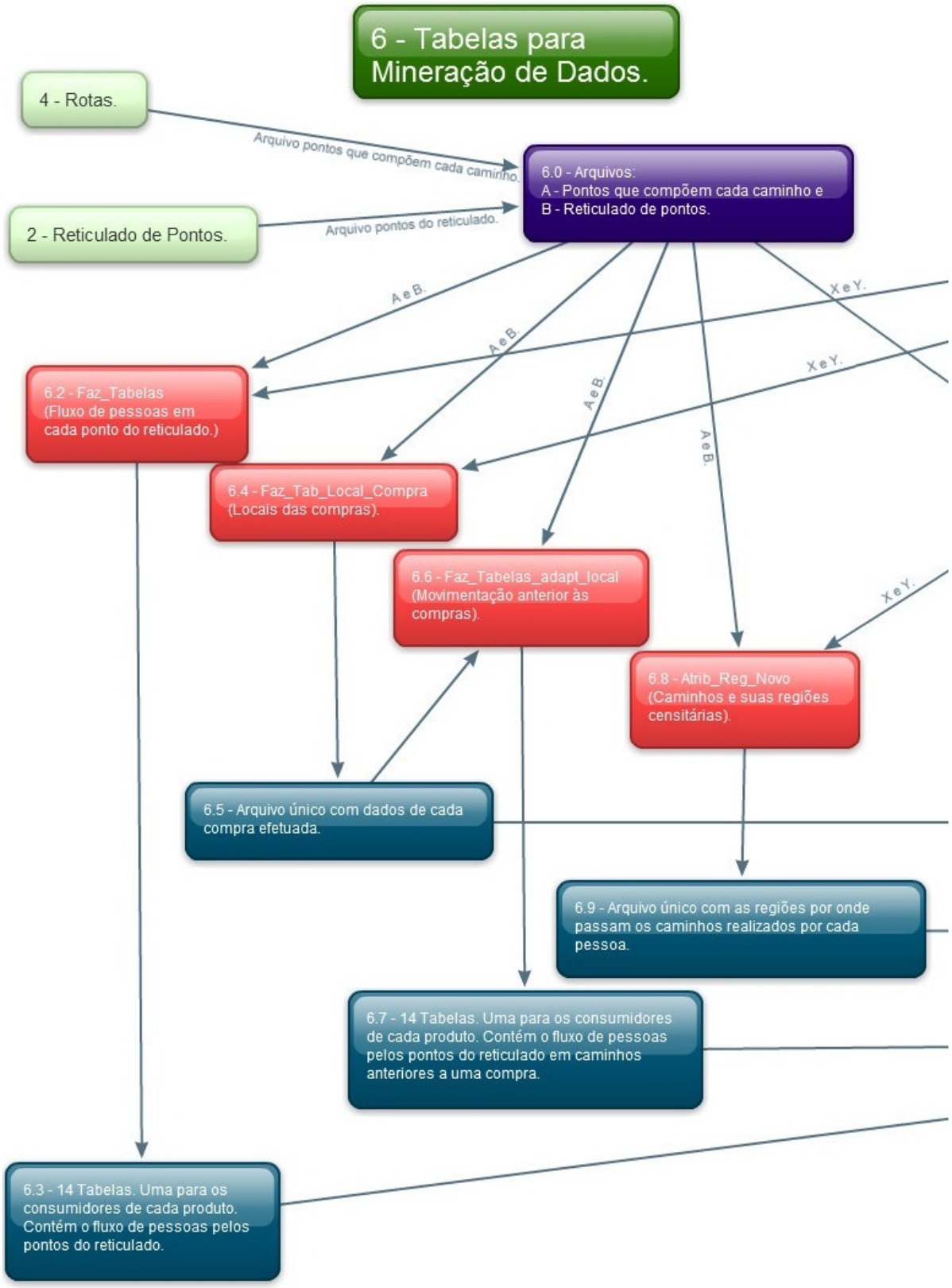

Figura 10.1: Diagrama de Geração de Tabelas para Mineração

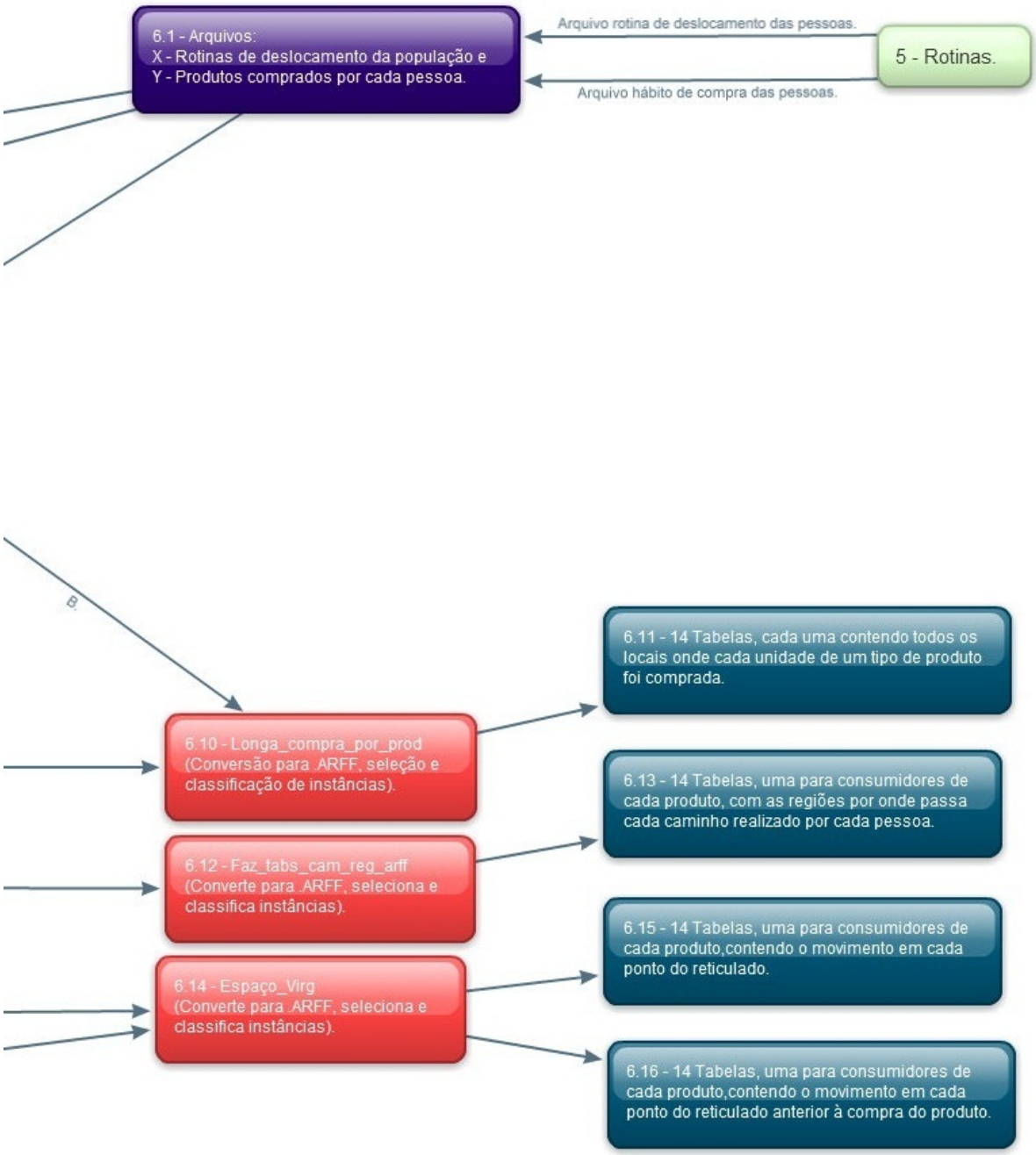

Figura 10.2: Continuação

A contagem e a gravação das tabelas em arquivos foi executada pelo programa *Faz_ Tabelas*. O programa abre e lê os seguintes arquivos:

- Arquivo contendo as rotinas de deslocamento da população: cada linha com o dia, hora, identificador da pessoa e o caminho realizado (conforme tabela 9.4).
- Arquivo das compras realizadas pelas pessoas: cada instância composta por identificador da pessoa, identificador do produto e quantidade de produtos daquele tipo comprados por aquela pessoa (conforme tabela 9.5).
- Arquivo contendo todos os pontos de todos os caminhos: cada linha com o identificador do caminho, as coordenadas do ponto daquele caminho e o identificador de cada um dos pontos, além da região censitária na qual se insere cada um dos pontos (conforme tabela 9.1).
- Arquivo do reticulado de pontos: cada instância com as coordenadas de um ponto, seu identificador, região e ruas que formam o cruzamento (conforme tabela 7.1).

Após a abertura dos arquivos citados a próxima ação do programa foi criar vetores de índice para alguns dos arquivos, evitando assim acessos desnecessários para busca de instâncias.

Foi criado um vetor de aproximadamente 23000 posições que guarda em cada uma delas o ponteiro para o local do arquivo de caminhos que contém informações sobre o caminho cujo identificador é numericamente igual ao índice da posição do vetor. Assim para obter os pontos que compõem cada caminho basta acessar o vetor descrito na posição que corresponde ao identificador do caminho procurado, ler seu conteúdo e acessar o arquivo de pontos dos caminhos na posição numericamente igual ao conteúdo acessado do vetor.

Outro vetor de índice foi criado para o arquivo de hábitos de consumo, com aproximadamente 5500 posições (correspondente ao número de pessoas da população gerada). Cada posição do vetor guarda o *offset* do arquivo correspondente à posição em que estão os dados sobre as compras realizadas pela pessoa cujo identificador é igual à posição acessada do vetor de índices.

O próximo vetor criado armazenou para cada um de seus índices as coordenadas do ponto de reticulado cujo identificador corresponde ao índice do vetor. Ele foi usado para a consulta de coordenadas de pontos baseado nos seus identificadores.

Após a criação dos vetores de índices foi gerada uma estrutura (de nome: Aglomerado) para armazenar (e contar) a quantidade de pessoas que passa por cada ponto de reticulado em cada faixa de horário e dia da semana. A estrutura foi uma matriz 7 x 29, o primeiro índice referente a cada dia da semana e o segundo referente a cada faixa de horário definida. Cada célula da matriz contendo um vetor de aproximadamente 2200 posições, cada posição do vetor representando um ponto do reticulado cujo identificador coincide com o valor de índice de cada posição do vetor. Assim se pôde incrementar cada posição do vetor de forma a se armazenar a quantidade de pessoas que passaram por determinado ponto em determinado dia da semana e horário.

Foi lido linha a linha o arquivo de rotinas das pessoas. Para cada linha lida (que contém o caminho percorrido por uma pessoa em um dia de semana e horário) foi pesquisado no arquivo de hábitos de consumo os produtos comprados pela pessoa referida na linha. Como o programa é executado uma vez para cada produto, a cada execução

são avaliados os movimentos apenas das pessoas que consumiram aquele produto. Assim, verifica-se se a pessoa referida na linha lida consumiu o produto avaliado na execução corrente do programa. Em caso afirmativo são acessados no arquivo que guarda os pontos de cada caminho (através do índice armazenado em vetor) os pontos que compõem os caminhos percorridos pela pessoa lida. A cada ponto percorrido a quantidade de pessoas armazenada na matriz Aglomerado para aquele ponto naquele dia da semana e horário é incrementada. Os horários de cada ponto são avaliados a partir do horário do início do caminho (disponível no arquivo de hábitos de deslocamento das pessoas) que é incrementado de 35 segundos a cada ponto percorrido.

Deve ainda ser mencionado que no caso das pessoas que compraram dois ou mais produtos iguais a contagem referente a esta pessoa é multiplicada pelo número de produtos. Dessa forma a importância de cada consumidor também é levada em conta. Assim se a pessoa consome três vezes mais que outra isso é levado em consideração.

Dessa forma, ao final da execução do programa, são escritos em arquivo todos os pontos do reticulado e as respectivas quantidades de pessoas que transitaram por cada um e que compraram determinado produto. Ao todo 14 arquivos, um para cada tipo de produto, são gerados. A tabela 10.1 representa um desses arquivos.

Quantidade de Pessoas	Região Censitária do Ponto	ID do Ponto	Coordenadas do Ponto	Dia da Semana e Horário
...	...	...	...	...
230	5120011	153	-16.701139 -49.245560	5 730
002	5120011	154	-16.701380 -49.245708	5 730
031	5120011	155	-16.702379 -49.245312	5 730
144	5120011	156	-16.701811 -49.244560	5 730
...	...	...	...	...

Tabela 10.1: Tabela de Fluxo pelo Reticulado de Pessoas Consumidoras do Produto 1

10.2 Tabelas de Locais das Compras

Além dos dados gerados já discutidos (população, hábitos de compra e deslocamento) foi observado que seria útil para o processo de mineração de dados que fossem geradas informações sobre o momento em que cada produto foi adquirido. Com isso poderiam ser buscadas mais informações sobre os hábitos de consumo das pessoas: locais onde a compra de cada produto geralmente é concretizada, locais da cidade por onde passaram os compradores imediatamente antes da compra de um produto, entre outras.

Portanto, numa aplicação real, além do registro dos produtos comprados por cada pessoa através do celular poderiam também ser registrados o horário da compra e as coordenadas geográficas do local. Com isso poderiam ser descobertas muitas informações importantes através da mineração de dados.

Devido a impossibilidade de obtenção de dados reais, já discutida em capítulos anteriores, foi escrito um programa (*Faz_ Tab_ Local_ Compras*) que simulou tais informações. Sua estrutura é semelhante à do programa descrito na seção anterior (*Faz_ Tabelas*).

O programa *Faz_ Tab_ Local_ Compras* lê inicialmente os mesmos arquivos que o programa *Faz_ Tabelas* (discutido na seção anterior): arquivo de rotinas de deslocamento, produtos comprados por cada pessoa, pontos de cada caminho e o arquivo com o reticulado de pontos. Em seguida cria os mesmos vetores índice: um vetor para se encontrar os pontos de um caminho no arquivo de pontos dos caminhos, outro para se acessar no arquivo de produtos comprados os relativos a cada pessoa, outro para se acessar as coordenadas de um ponto através de seu identificador.

Em seguida é acessado o arquivo de produtos consumidos e extraídas as quantidades de cada produto comprado. Isto é feito ordenadamente para cada pessoa da população gerada. A partir daí é decidido, de acordo com regras predeterminadas (a serem discutidas na seção 10.6), o dia e horário a serem registrados como sendo o momento da compra de cada produto comprado por cada pessoa. Uma dessas regras seria, por exemplo, que relógios são comprados por pessoas de classe econômica B entre segunda e quarta-feira no trabalho no período da manhã. O dia e hora exatos da compra são decididos utilizando-se a função RAND().

Sabendo-se o dia e horário da compra de um produto por uma pessoa o arquivo de rotinas das pessoas é acessado na posição que contém a rotina da pessoa analisada. É feita uma busca na rotina da pessoa pelo último caminho realizado por ela antes do momento da compra. Obtido o identificador desse caminho o programa grava uma instância no arquivo de saída.

ID e Classe da Pessoa	Região Censitária da Pessoa	ID do Produto e da Rota	Dia da Semana, Horário, ID, Coordenadas e Região Censitária do Ponto Inicial	Dados Sobre o Ponto Final	Tipo de Rota
...	...	...	...	...	...
3 B	5190002	07 01 N	6 0750 082 -16.681 -49.280 5190002	6 0817 ...	trabI
3 B	5190002	09 69 N	7 1210 728 -16.697 -49.262 5030007	7 1236 ...	hhI
3 B	5190002	11 69 N	7 1210 728 -16.697 -49.262 5030007	7 1236 ...	hhI
3 B	5190002	14 69 N	7 1210 728 -16.697 -49.262 5030007	7 1236 ...	hhI
1 B	5190002	01 01 N	3 0730 082 -16.681 -49.280 5190002	3 0757 ...	trabI
1 B	5190002	03 01 I	6 1800 728 -16.697 -49.262 5030007	6 1827 ...	trabV
...	...	...	...	...	...

Tabela 10.2: Tabela com Informações sobre cada Compra Realizada

Cada instância do arquivo de saída contém o identificador da pessoa compradora, classe econômica, região censitária onde mora, identificador do produto comprado, identificador do caminho percorrido antes da compra, dia da semana, horário, identificador do ponto onde foi efetuada a compra, coordenadas desse ponto, região do mesmo, informações sobre a localização e horário do ponto inicial do caminho percorrido e por fim, a natureza desse caminho (almoço - alm, trabalho - trab, lazer - laz, textithappy hour - hh,etc). O horário do início do caminho e do fim é obtido acessando o arquivo de pontos dos caminhos (através do vetor de índices e do identificador do caminho) e calculando a duração deste (foi convencionado que o deslocamento entre dois pontos dura em média 35 segundos).

Com isso é gerado e gravado em arquivo um dia, horário e local para cada compra efetuada por cada pessoa. São também gravados dados sobre o último caminho percorrido antes da compra. Esse arquivo está representado na tabela 10.2.

10.3 Tabelas de Movimentação Anterior às Compras

A partir dos dados acerca dos caminhos realizados antes de cada compra foi criado um terceiro tipo de tabela. A exemplo das tabelas de fluxo de pessoas em cada ponto foram criados 14 arquivos contendo tabelas (uma para cada produto) com conteúdo semelhante, com a diferença que o número de pessoas que transitou por cada ponto refere-se apenas a rotas executadas antes da compra de um produto.

Dessa forma, tornou-se possível a descoberta, através de mineração de dados, de informações relativas ao ato da compra: locais mais frequentados em determinados dias da semana e horários por compradores de cada produto antes de efetivarem a compra do mesmo.

Para criar os arquivos o programa *Faz_ Tabelas* (que criou as tabelas iniciais de fluxo de pessoas) foi adaptado, resultando no Programa *Faz_ Tabelas_ Adapt_ Local*. A principal alteração foi a leitura do arquivo de caminhos percorridos antes de cada compra, tendo se tornada desnecessária a leitura do arquivo de rotinas das pessoas, antes realizada. Com essa alteração o programa passou a funcionar diferentemente, avançando a execução linha por linha do arquivo de caminhos anteriores às compras e não mais de acordo com cada instância do arquivo de rotinas das pessoas.

Foi feita também a exclusão do código que gerava a tabela de índices para o arquivo de consumo das pessoas. O acesso a esse arquivo tornou-se desnecessário visto que já consta no arquivo de caminhos anteriores às compras o produto comprado a que se refere cada instância.

Já a contagem do número de pessoas que passou por cada ponto e seu armazenamento na matriz denominada *Aglomerado* continuou sendo feita da mesma forma, com acesso ao arquivo de pontos dos caminhos para avaliação do tempo dispendido no trajeto e identificação de cada ponto transitado.

O resultado do procedimento foram 14 arquivos contendo instâncias com os mesmos atributos das tabelas anteriores mencionadas (tabela 10.1): quantidade de pessoas que passou pelo ponto, região censitária, identificador e coordenadas do ponto, dia da semana e horário da contagem. Porém no caso dessas tabelas os dados são relativos apenas a rotas percorridas imediatamente antes da compra de algum produto.

10.4 Tabelas de Caminhos e suas Regiões Censitárias

O quarto tipo de tabela criado foi uma tentativa de analisar as rotas percorridas pela população de uma forma diferente. Sem fragmentá-las em seus pontos constituintes. Mantendo para cada uma a informação acerca de qual ponto veio depois do outro. Os arquivos com as tabelas desse novo tipo contém para cada linha um caminho realizado por alguma pessoa em algum dia e horário e nos atributos desses caminhos estão todas as regiões censitárias pelas quais tal caminho passou. Também como atributos de cada

caminho foram gravados os horários de entrada em cada região percorrida pela pessoa que o executou.

Dessa forma cada caminho executado por cada pessoa da população é representado em uma linha específica. São 60 atributos que guardam o identificador das regiões transitadas e 60 que guardam os horários de entrada em cada região, além de um atributo que guarda o dia em que foi realizada a rota. É também registrado em cada linha o identificador do produto comprado logo após a realização do caminho e o destino deste ("alm"para o local do almoço, "rest"para um restaurante frequentado a lazer, "hh"para um local de *happy hour* e caracteres "I"para ida e "V"para volta do local). A tabela 10.3 exemplifica o arquivo produzido.

Dia da Semana	ID do Produto	1a Região	Hora	2a Região	Hora	...	60a Região	Hora	Tipo de Rota
...	...	...	...	...	...	...	...	...	...
7	13	5040014	1220	5040035	1221	...	0	0	hhI
7	14	5040014	1220	5040035	1221	...	0	0	hhI
3	01	5040021	0740	5040022	0742	...	0	0	trabI
1	01	5040021	1230	5040022	1231	...	0	0	restI
...	...	...	...	...	...	...	...	...	...

Tabela 10.3: Tabela de Regiões Censitárias por Onde Passa cada Caminho

Essa nova perspectiva possibilita a descoberta de novos tipos de informações. Torna-se possível pesquisar nos dados por que outras regiões passaram consumidores de um determinado produto que passam por determinada região da cidade em determinado dia da semana e faixa de horário. Isso pode resultar na descoberta de padrões sobre o destino desses consumidores (uma região de restaurantes por exemplo) ou de onde saíram (do trabalho por exemplo).

Uma aplicação interessante seria a de descobrir padrões nos hábitos de deslocamento de um determinado segmento de consumidores de um produto específico. Ou seja, seria possível identificar por exemplo por onde passam pessoas que trabalham nos Tribunais da cidade e são consumidoras de artigos esportivos. A descoberta de tais informações seria muito útil na segmentação do público consumidor, prática muito importante para as organizações do ponto de vista de *marketing*. Para utilizar os dados dessa forma seria no entanto imprescindível ter uma informação que caracterize inequivocamente as pessoas do segmento pesquisado em termos de localização geográfica. No exemplo, tal informação poderia ser: as pessoas do segmento pesquisado estão na região dos tribunais no período de 9:30 *a.m.* até 11:00 *a.m.* nas terças feiras.

Definidas as posições geográficas e horários que caracterizam o segmento a ser pesquisado, o próximo passo é a classificação das instâncias que atendem a esses padrões, ou seja, que são relativas ao público pesquisado. Tais instâncias seriam classificadas positivamente e as demais negativamente. Outro passo importante seria a deleção em todas as instâncias das regiões que caracterizam tal público. Devem ser deletadas também as regiões vizinhas à região que gera a classificação das instâncias. Dessa forma se evita que os algoritmos de mineração identifiquem a classe das instâncias através de um dos seus atributos. Se evita também a descoberta de regras muito óbvias como: passa pela região definida quem passa pelas regiões vizinhas a ela.

Para gerar a tabela descrita foi criado o programa de nome *Atrib_Reg_Novo*. Sua estrutura se assemelha muito à estrutura dos programas descritos até agora neste capítulo.

São lidos inicialmente os arquivos de hábito de consumo das pessoas, rotinas de deslocamento da população, reticulado de pontos e o arquivo que contém os pontos de cada caminho. São criados vetores de índices, como os descritos na seção 10.1, para os arquivos de pontos de cada caminho, hábitos de consumo e é criado também um vetor contendo as coordenadas de cada ponto do reticulado ordenado pelos identificadores dos pontos. Outro vetor é criado para armazenar as regiões censitárias pelas quais cada caminho passa. A cada caminho analisado esse vetor é reinicializado. Dessa forma, a cada iteração ele será usado para armazenas as regiões pelas quais passa o caminho da iteração corrente. Cada posição do vetor tem campos para armazenar o identificador de uma região, o horário de entrada e o horário de saída da mesma.

O programa lê um a um cada caminho percorrido por cada pessoa. A cada caminho lido o arquivo de pontos dos caminhos é acessado e são analisados todos os pontos do caminho corrente. A cada ponto analisado o horário é atualizado, simulando a passagem do tempo decorrente do deslocamento de um ponto a outro pela pessoa que realiza o caminho. As regiões censitárias nas quais se insere cada ponto são consideradas. Sempre que há mudança de uma região para outra no decorrer da análise dos pontos do caminho corrente o vetor de regiões censitárias é atualizado. É armazenado o identificador da região nova, o horário de entrada nela e o horário de saída da região anterior. Isso é feito até o fim do caminho. No fim da análise da rota corrente, todas as regiões pelas quais passa o caminho estão corretamente armazenadas no vetor de regiões.

A seguir é acessado o arquivo de hábitos de consumo na posição relativa aos hábitos da pessoa que realizou o caminho recém analisado. Para cada produto adquirido pela pessoa em questão é gravada no arquivo de saída uma cópia do vetor de regiões e demais informações sobre o caminho percorrido e a pessoa que o percorreu, além do identificador de um produto comprado pela pessoa.

Assim é escrito um arquivo (conforme tabela 10.3) contendo uma linha para cada caminho realizado por cada pessoa da população. Cada linha contém, em ordem de percorrimento, todas as regiões pelas quais passou o caminho, seguidas dos horários de entrada em cada região. Cada caminho é replicado tantas vezes quantos forem os produtos comprados pela pessoa que o realizou.

Na próxima seção será descrito de que forma tal arquivo foi utilizado na geração de tabelas para mineração de dados, cada uma específica para um público consumidor de um produto, todas em formato .ARFF (formato das tabelas passíveis de leitura pelo *Weka 3*).

10.5 Conversão de Tabelas para o Formato .ARFF

Depois do processamento dos diversos bancos de dados gerados e gravação dos dados selecionados para a descoberta de informações, foi necessário converter os arquivos gerados para o formato ARFF.

Como a mineração de dados foi realizada no programa *Weka 3*, todos os arquivos que foram utilizados no processo e que foram lidos pelo programa tiveram que ser convertidos para o formato padrão de leitura do mesmo: .ARFF . Essa extensão diz respeito a arquivos com instâncias cujos atributos são descritos no cabeçalho. Assim cada um dos atributos é

nomeado e classificado de acordo com o seu tipo: numérico, nominal, entre outros. No caso dos atributos nominais os valores possíveis para eles devem ser enumerados na declaração do atributo. Os atributos das instâncias devem ser separados por vírgula simples (17).

Para converter os arquivos contendo as tabelas descritas neste capítulo foram escritos três programas. O primeiro, de nome *Espaço-Virg* converte as tabelas que contém a quantidade de pessoas que passou por cada ponto do reticulado em cada dia da semana e horário (tabela 10.1) e também o arquivo que contém os dados relativos a cada compra (dados sobre a pessoa que a realizou, o produto comprado e o caminho realizado antes dela, conforme a tabela 10.2). São 14 arquivos relativos ao movimento nos pontos considerando todos os caminhos, 14 arquivos contendo o movimento nos pontos considerando apenas caminhos anteriores a compras e 1 arquivo com informações gerais sobre cada compra efetuada. Todos são convertidos para o formato .ARFF, cada qual com o cabeçalho adequado e cada atributo separado por vírgula simples.

Esse programa também seleciona os atributos mais adequados para a mineração de dados, desconsidera atributos desnecessários e gera um atributo de classificação das instâncias, no caso, dos 28 arquivos relativos ao movimento nos pontos do reticulado. Em relação a esses 28 arquivos, cada linha representa um ponto do reticulado em determinado dia da semana e faixa de horário. Para cada instância, os atributos selecionados ou gerados foram: quantidade de pessoas que por lá transitaram no período considerado, região censitária onde o ponto se encontra, coordenadas do ponto, dia da semana e horário considerados e, como varivél de classificação, o grau de movimento, ou seja, se o movimento no ponto é considerado pequeno (p), médio (m) ou grande (g).

A variável de classificação é muito utilizada pelos algoritmos de mineração de dados. São geradas regras para cada classe p,m e g da variável de classificação, sendo portanto imprescindível sua existência. Quanto à quantidade de pessoas que transitam pelo ponto, no momento da mineração de dados tal atributo foi excluído, permanecendo apenas a variável qualitativa relativa ao movimento nos pontos (p,m e g). A tabela 10.4 representa os arquivos aqui discutidos.

Quantidade de Pessoas	Região Censitária do Ponto	Coordenadas do Ponto	Dia da Semana e Horário	Grau de Movimento *(p, m, g)*
...	...	...	...	...
230	5120011	-16.701139 -49.245560	5 730	g
002	5120011	-16.701380 -49.245708	5 730	p
031	5120011	-16.702379 -49.245312	5 730	m
144	5120011	-16.701811 -49.244560	5 730	g
...	...	...	...	...

Tabela 10.4: Fluxo pelo Reticulado de Consumidores de um Produto Específico

Devido à limitada quantidade de memória RAM disponível no computador utilizado para a mineração de dados e devido à grande exigência de memória para a execução de certos algoritmos, as tabelas que tinham tamanho superior a 2 *megabytes* precisaram ter instâncias deletadas. As tabelas que precisaram sofrer tal procedimento foram as que continham dados sobre o fluxo de pessoas pelos pontos do reticulado referente a todos os caminhos realizados por todas as pessoas.

Para proceder a deleção de instâncias foi feita uma seleção aleatória (de acordo com a função RAND()) das que seriam escritas. A vantagem desse método é que ele procura manter a frequências das variáveis, ou seja, não altera as características estatísticas dos dados e portanto tem impacto quase nulo na qualidade das regras e informações obtidas através da mineração de dados.

Apesar de a escrita de um ponto ter sido ao acaso, a chance de escrita de um ponto com pouco movimento foi definida como sendo menor que a chance de escrita dos pontos com maior movimento. Isso foi feito porque o objetivo não é a descoberta de regras que indiquem os locais menos movimentados e sim o oposto. Daí a menor importância atribuída aos pontos de pouco movimento e sua chance menor de escrita nos arquivos.

Mesmo assim, a quantidade de pontos de pouco movimento nas tabelas ficou muito superior à quantidade de pontos de grande movimento, o que está de acordo com a distribuição das informações originais. Tal realidade foi mantida apesar da maior importância dos pontos de grande movimento porque, mantidos os pontos de pouco movimento, os algoritmos de mineração de dados foram forçados a gerar regras mais específicas para os pontos de grande movimento. Se fossem criadas regras muito amplas para classificar as instâncias de interesse, o erro relacionado a elas seria grande pois tais regras englobariam muitos pontos de pouco movimento. Dessa forma, para obter taxas de erro menor, os algoritmos acabam procurando gerar regras mais específicas, que englobam somente as áreas de pontos de grande movimento. Os modelos obtidos dessa maneira acabam por ter mais qualidade, englobando os pontos de grande movimento de forma mais precisa, sem envolver áreas paralelas que não têm interesse. O resultado são regras mais fáceis de serem interpretadas e que facilitam a obtenção de informações sobre os padrões dos dados.

O segundo programa escrito foi o *Longa_ compra_ por_ prod*. Ele lê o arquivo contendo dados gerais sobre os caminhos realizados antes das compras (tabela 10.2, convertido para .ARFF pelo programa discutido nos parágrafos anteriores) e escreve 14 arquivos contendo os atributos mais relevantes do arquivo original (dados sobre o ponto final dos caminhos percorridos antes da compra de um produto, ou seja, dados sobre o ponto onde efetivamente ocorreu a compra, conforme a tabela 10.5). Cada um dos arquivos contém informações relativas exclusivamente à compra de um dos 14 produtos considerados neste trabalho. Todos os arquivos no formato .ARFF.

Vale ressaltar que, no caso dessas tabelas, foi preciso adicionar pontos do reticulado onde não ocorreram compras de forma que tais pontos foram classificados negativamente e os pontos onde ocorreram compras foram classificados positivamente (variável de classificação: *s* para sim e *n* para não). Só assim pôde ser realizado o processo de mineração desses dados, ou seja, a busca pelos padrões das instâncias classificadas, no caso, positivamente (locais de compra dos produtos), pois os algoritmos de classificação necessitam de ao menos duas classes. Portanto, adicionar pontos do reticulado onde não ocorreram compras permitiu a identificação dos padrões de localização dos pontos onde ocorreram as compras.

A tabela 10.5 mostra um esquema geral do formato dos arquivos gerados. Os atributos gravados para cada instância foram dia da semana e hora da compra, coordenadas do ponto onde ocorreu a compra, região censitária do mesmo e se de fato ocorreu uma compra naquele ponto / data / horário ou se a instância referida na linha foi adicionada pelo motivo discutido no parágrafo anterior.

Dia da Semana e Hora	Coordenadas do Ponto	Região Censitária do Ponto	Ocorrência da Compra (s, n)
...	...	...	...
7 2021	-16.693190 -49.269791	5030002	s
2 0804	-16.710011 -49.260529	5120037	s
7 2021	-16.693190 -49.269791	5030002	s
1 2423	-16.704170 -49.283371	5160009	n
1 2418	-16.708120 -49.279678	5160010	n
...	...	...	...

Tabela 10.5: Tabela Referente ao Local de Cada Compra Realizada

O terceiro programa escrito *Faz_tabs_cam_reg_arff_Novo* cria e converte para o formato .ARFF tabelas contendo caminhos e as regiões censitárias por onde passa cada caminho. Ele lê o arquivo que contém para todos os caminhos de todas as pessoas tais regiões (tabela 10.3) e gera 14 outros arquivos relativos cada um aos caminhos realizados por pessoas que consumiram determinado produto. Assim os caminhos são separados de acordo com o público consumidor que os realiza.

Cada caminho gravado é classificado positivamente ou negativamente. Se passa por determinada região censitária do Setor Marista entre 17:55 e 18:40 ele é classificado positivamente (s). Caso contrário é classificado negativamente (n). Essa região foi escolhida por se tratar de um local com grande concentração de clínicas médicas que visam pessoas da classe A, o que favorece a investigação dos hábitos de deslocamento de pessoas que trabalham na área da saúde nessa região especificamente. O horário foi escolhido para se considerar o fluxo de pessoas que saem do trabalho, portanto, favorecendo a análise das rotas de pessoas que trabalham na região citada. Trata-se assim de uma tentativa de estudo de um segmento específico do público-alvo: compradores de um determinado produto que trabalham em uma região determinada caracterizada pela alta concentração de determinado tipo de empresa.

Todos os caminhos que foram positivados são gravados. Dentre os negativados apenas 1 a cada 55 também são gravados. Isso foi necessário para que os arquivos gerados não fossem grandes a ponto de inviabilizar a mineração de dados por falta de memória RAM. As regiões vizinhas à região citada e a própria foram excluídas de todos os caminhos gravados para evitar que fossem geradas regras sem utilidade, como explicado na seção anterior sobre tabelas desse tipo.

Os arquivos resultantes desse processo seguem o padrão da tabela 10.6. Cada instância representa um caminho realizado por um comprador do produto tratado pelo arquivo. São gravados como atributos o dia da semana, o atributo de classificação da rota (s ou n) e as regiões censitárias pelas quais a rota passa, além dos horários de entrada em cada região. São N campos para guardar as informações de horário e região. Quando o caminho passa por um número menor que N regiões, os campos restantes recebem valor 0.

Assim, através dos programas discutidos nesta seção, foram obtidas tabelas adequadas para os processos a serem realizados no *Weka 3*, tanto em termos de formato quanto em termos de dados. Os três formatos finais dos quatro grupos de tabela geradas foram: quatorze tabelas conforme a tabela 10.6, quatorze do tipo da tabela 10.5, quatorze do

Dia da Semana	1a Região	Hora	2a Região	Hora	...	N-ésima Região	Hora	Rota de Interesse
...	...	...	...	...	...	...	...	...
7	5040014	1220	5040035	1221	...	0	0	s
7	5040014	1220	5040035	1221	...	0	0	s
3	5040021	0740	5040022	0742	...	0	0	n
1	5040021	1230	5040022	1231	...	0	0	s
...	...	...	...	...	...	...	...	...

Tabela 10.6: Tabela de Regiões por Onde Passam as Rotas

modelo da tabela 10.4 e mais quatorze deste mesmo último modelo para deslocamento de pessoas anterior ao ato de uma compra.

10.6 Padrões e Regras das Tabelas de Dados

Os dados gerados para este trabalho foram definidos utilizando-se regras e padrões que admitem várias possibilidades para cada instância. Dentre as possibilidades, cada instância foi escolhida aleatoriamente. Portanto, nem tudo foi decidido ao acaso.

A população, por exemplo, foi criada com base em informações do Censo 2010 do IBGE e também com informações do mercado imobiliário obtidas em um site de anúncios de imóveis. Regras subjetivas também foram utilizadas. Nesse caso foi atribuída uma probabilidade, dependendo da região censitária e status de aquisição de residência, para que cada pessoa fosse de determinada classe econômica. Com base na probabilidade foi então decidida (através de números aleatórios) a classe econômica de cada pessoa.

Várias são as razões para que as escolhas não tenham sido feitas de forma exclusivamente aleatória. Como os dados precisavam ser similares à realidade, todas as informações disponíveis sobre o contexto real deveriam ser utilizadas (rotas do *Google Maps*, marcação de pontos de interesse da cidade, pesquisa de produtos mais comprados via internet). Além disso, caso os dados fossem totalmente aleatórios seria difícil avaliar a qualidade das informações descobertas através de mineração de dados. Qualquer regra ou informação descoberta poderia ser tida como válida, mesmo quando não fizesse sentido no contexto real. Dessa forma, para diferenciar os algoritmos de mineração quanto a qualidade de cada um foi necessária a inserção de alguns padrões nos dados, além daqueles utilizados por realismo. Os algoritmos que apontaram esses padrões foram considerados mais confiáveis tendo aumentada a credibilidade das informações por eles obtidas.

Os padrões inseridos exclusivamente para teste de algoritmos (além daqueles já discutidos utilizados para se obter dados similares à realidade) serão apresentados nas seções a seguir.

10.6.1 Padrões de Consumo

Em relação aos hábitos de consumo da população gerados foram inseridos os seguintes padrões:

- Todas as pessoas de classe A compram cafeteiras e nenhuma pessoa da classe B as compra. Esse padrão é útil para se avaliar exclusivamente os hábitos de deslocamento de pessoas da classe A, caso sejam considerados apenas os compradores de cafeteiras na mineração de dados. Os padrões relativos a essa classe serão portanto revelados.
- Todas as pessoas de classe B compram muitos cupons de sites de compra coletiva e ninguém da classe A compra tais cupons. Padrão útil para se avaliar o deslocamento e os padõres exclusivamente de pessoas da classe B, caso sejam utilizadas as tabelas relativas aos compradores de cupons.
- Pessoas de classe A e B compram fogões e relógios com a mesma probabilidade quaisquer que sejam as quantidades de produtos comprados (colunas iguais na matriz de probabilidades). Padrão útil para se observar os hábitos de deslocamento das duas classes sociais sem levar em conta os produtos que elas consomem.

A escolha dos produtos acima foi aleatória. Visou apenas o isolamento de uma classe econômica ou de outra, de forma a permitir a análise por meio da mineração de dados de classes específicas, ou a geração de um banco de dados onde a classe econômica pudesse ser irrelevante. Assim, o fato de pessoas de classe A comprarem cafeteiras ou pessoas de classe B comprarem cupons, possivelmente não corresponde à realidade, visto que tais produtos foram escolhidos ao acaso para serem comprados por uma classe ou outra.

As demais probabilidades de compra dos outros produtos em diversas quantidades foram arbitradas diferentemente de acordo com o produto, mas sem a atribuição de padrões específicos e nem foram baseadas em fontes reais. Apenas representam cenários possiveis de consumo de tais produtos, sendo portanto arbitrárias. Todas as probabilidades de compra de cada produto por uma pessoa de determinada classe econômica em determinada quantidade estão escritas na tabela 10.7. Cada linha das matrizes diz respeito a uma quantidade de produtos comprados (0, 1, 2, ou 3 produtos). A primeira coluna é referente à classe A e a segunda à classe B. Assim cada célula das matrizes contém a probabilidade de uma pessoa da classe equivalente à coluna comprar o produto relativo a matriz na quantidade especificada para aquela linha.

10.6.2 Padrões Quanto ao Dia da Semana e Horário das Compras

Os padrões arbitrários seguidos, relativos ao dia da semana e horário de compra de cada tipo de produto de acordo com a classe econômica do comprador, serão descritos a seguir. Não são baseados em dados reais. Eles visam apenas possiblitar a avaliação do poder dos algoritmos de mineração de dados (verifição se os padrões serão apontados pelos algoritmos), sendo, ao mesmo tempo, opções plausíveis para o momento em que são efetivadas as compras de cada produto através de aplicativos de *m-commerce*.

- Livros e relógios: Comprados entre segunda e quarta-feira, no trabalho, na parte da manhã, no caso de pessoas da classe B. Para pessoas da classe A podem também ser comprados sábado durante a tarde ou domingo.
- Câmeras e celulares: Comprados entre quinta e sexta feira após o trabalho por pessoas de ambas as classes econômicas.

1 - Livros			2 - Câmeras			3 - Celulares		
0	30	70	0	40	70	0	25	15
1	40	20	1	50	30	1	50	65
2	20	10	2	10	00	2	20	20
3	10	00	3	00	00	3	05	00
4 - Videogames			5 - Relógios			6 - *Notebooks*		
0	50	40	0	70	70	0	35	50
1	40	55	1	25	25	1	55	45
2	05	05	2	05	05	2	10	05
3	05	00	3	00	00	3	00	00
7 - Cupons			8 - Cafeteiras			9 - Televisões		
0	100	00	0	00	100	0	35	25
1	00	00	1	00	00	1	45	70
2	00	50	2	50	00	2	15	05
3	00	50	3	50	00	3	05	00
10 - Fogões			11 - Geladeiras			12 - Microondas		
0	60	60	0	55	65	0	60	60
1	30	30	1	30	30	1	30	40
2	08	08	2	10	05	2	05	00
3	02	02	3	05	00	3	05	00
13 - Máquinas			14 - Liquidificadores					
0	70	45	0	60	50			
1	25	55	1	30	45			
2	05	00	2	10	05			
3	00	00	3	00	00			

Tabela 10.7: Probabilidades de Compra

- Videogames e *notebooks*: Comprados entre segunda e quarta-feira, no trabalho, na parte da manhã ou sábado durante a tarde ou domingo por ambas as classes econômicas.
- Cupons de compra coletiva para restaurantes: Comprados entre quarta e sexta-feira, de manhã, por ambas as classes.
- Cafeteiras, televisões, fogões, geladeiras, microondas, máquinas de lavar e liquidificadores: Comprados no sábado a tarde ou no domingo por pessoas da classe B, podendo também ser comprados entre segunda e quarta-feira, no trabalho, na parte da manhã, por pessoas de classe econômica A.

10.6.3 Padrões de Deslocamento

As diferenças de rotinas de deslocamento entre as classes A e B foram geradas pelas diferentes probabilidades de execução de determinados caminhos pelas diferentes classes (por exemplo, a chance de alguém da classe A visitar um restaurante depois do trabalho na quinta feira é bem maior que a chance de uma pessoa da classe B fazer o mesmo). Tais probabilidades, apesar de terem sido escolhidas arbitrariamente, foram escolhidas de forma a serem plausíveis e foram utilizadas como parâmetros na execução do programa que definiu as rotas da população gerada neste projeto (discutido na seção 9.2.1), daí a existência de diferenças entre as rotas de cada classe. Outro fator que gerou tais diferenças foi também a diversidade dos pontos de interesse relacionados a cada classe (diferentes restaurantes, locais de trabalho e lazer).

Além dos padrões descritos, foram definidos mais alguns, não necessariamente baseados na realidade. Isso foi feito para diferenciar ainda mais as rotinas entre as classes e possibilitar a avaliação do poder dos algoritmos de mineração a serem utilizados (verificar quais apontariam tais padrões). São eles:

- Todas as pessoas de classe A frequentam restaurantes de uma região específica do Setor Marista com chance de 75 % cada vez que saem de casa para irem a um restaurante em fins de semana. Padrão útil para se avaliar quais algoritmos de mineração de dados o identificarão corretamente, sendo também relativamente similar à realidade, pois a maioria dos restaurantes frequentados por pessoas de alto poder aquisitivo em Goiânia se encontram nessa região.
- Todas as pessoas de classe B frequentam com probabilidade de 75 % o *Shopping Flamboyant* em fins de semana como opção de lazer. Trata-se de uma regra relativamente similar à realidade, visto ser esse o maior e o mais frequentado *shopping* da cidade. No entanto esta regra foi criada realmente para se avaliar os algoritmos de mineração.
- Pessoas de classe A voltam para casa para almoçar durante dias úteis com probabilidade de apenas 10 %. Regra criada para se tentar avaliar seu impacto no banco como um todo.

Capítulo 11

Resultados da Mineração de Dados

Após o processamento e a organização dos dados gerados em tabelas no formato .ARFF, foi iniciada a Mineração de Dados através do programa *Weka 3*. Na geração de cada tipo de tabela foram selecionados os atributos mais úteis ao processo, eliminando-se atributos desnecessários (como descrito capítulo 10) e mantendo-se os atributos em função dos quais deveriam ser obtidas regras (horários, dias da semana, posições geográficas e regiões censitárias). Portanto, não foi necessária a utilização das funcionalidades de seleção de atributos relevantes que o *Weka 3* oferece. Todos os atributos das tabelas foram escolhidos por sua relevância e pela necessidade de se obter regras em função dos mesmos.

Neste capítulo serão discutidos os passos tomados até a aplicação dos algoritmos de mineração e também serão apresentados e analisados os resultados obtidos.

O procedimento geral seguido para a mineração de dados e análise dos modelos gerados pode ser visualizado no diagrama da figura 11.1. A cor de cada retângulo deve ser interpretada da mesma forma que nos diagramas do início do capítulo 10.

11.1 Tabelas Efetivamente Utilizadas

Foram quatro os tipos de tabela gerados. Um tipo contém o fluxo de público consumidor nos pontos do reticulado em todos os dias de suas rotinas (como na tabela 10.4). Outro tipo contém o fluxo pelos mesmos pontos mas considerando apenas caminhos realizados imediatamente antes das compras (conforme tabela 10.4). Um terceiro reúne as coordenadas geográficas do local onde cada compra foi efetuada (tabela 10.5). O último tipo contém para cada caminho realizado por um público consumidor as regiões censitárias pelas quais passa tal caminho (como na tabela 10.6).

Como são 14 os produtos considerados neste trabalho, são também 14 públicos alvo diferentes. Com isso foram criadas 14 tabelas diferentes para cada tipo de tabela descrito no parágrafo anterior, cada tabela contendo o contexto de movimentação de consumidores de um determinado produto, totalizando 56 tabelas.

Tornou-se portanto inviável realizar o processo de mineração e análise de regras produzidas para todas essas tabelas, pois alguns algoritmos podem demorar até 10 minutos para gerar todos os resultados, como o *lazy.Kstar*. Além disso cada algoritmo deve ser utilizado várias vezes sobre uma mesma tabela com parâmetros diferentes, a fim de se descobrir qual sua melhor configuração para aquele tipo de tabela e para que se possa produzir o melhor modelo possível (conjunto de regras que gera o maior número de acertos).

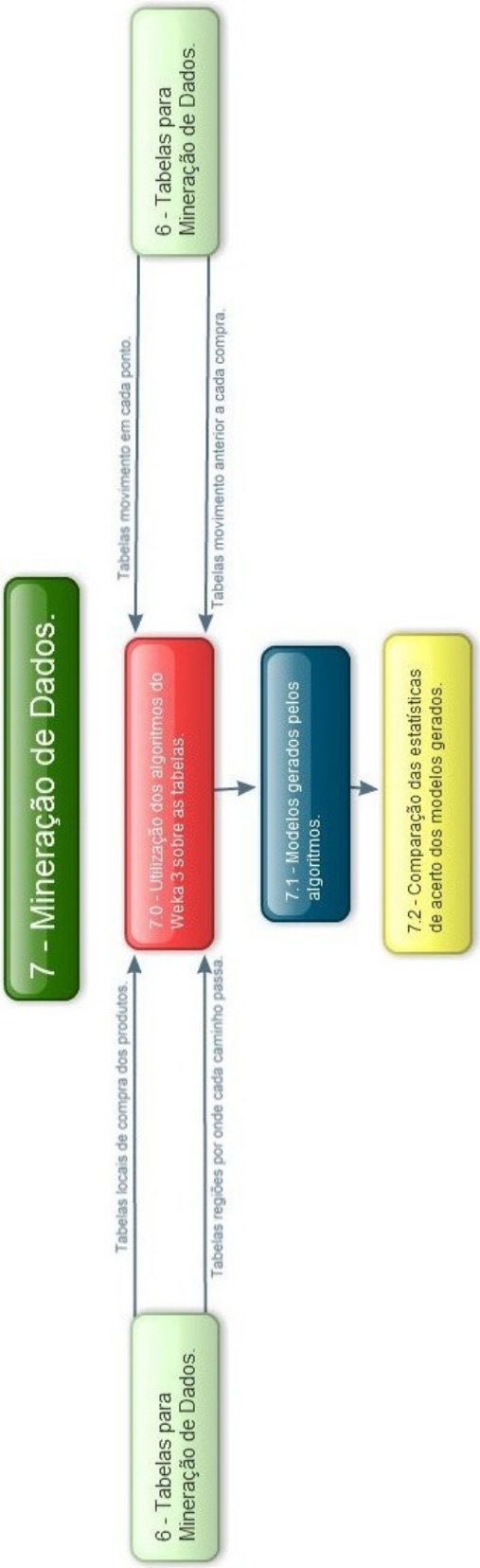

Figura 11.1: Diagrama do Processo de Mineração de Dados

Diante desse fato nem todas as tabelas geradas foram utilizadas. Apesar disso, as tabelas foram selecionadas de forma que o grupo escolhido para um determinado tipo de tabela contivesse todos os padrões gerados no projeto (descritos na seção 10.6). Assim, de acordo com os padrões de local e horário de compra de produtos e também de acordo com as diferentes probabilidades de compra de cada produto por pessoas de cada classe econômica, os consumidores de cada um foram semanticamente agrupados. Com isso o grupo de tabelas escolhidas contendo por exemplo o fluxo de pessoas pelos pontos do reticulado em todos os momentos de suas rotinas abrange todos os padrões inseridos no projeto que poderiam se manifestar nesse contexto. Isso possibilitou a posterior análise das regras produzidas em busca de tais padrões, permitindo validar todo o processo, ou seja, verificar se realmente a mineração de dados identificou os padrões inseridos nos dados.

Em relação às probabilidades de compra de cada produto por pessoas de cada classe econômica foram considerados 5 grupos. Cada grupo reúne produtos com padrões de probabilidade (tabela 10.7) semelhantes entre as classes, a saber:

- Grupo P1, formado por consumidores dos produtos 1 (livros), 2 (câmeras), 4 (videogames), 6 (*notebooks*), 11 (geladeiras), 12 (microondas), 13 (máquinas de lavar) e 14 (liquidificadores). São produtos comprados em menores quantidades e de forma menos frequente, com uma tendência geral a serem consumidos em maiores quantidades por pessoas da classe econômica A.
- Grupo P9, formado por consumidores dos produtos 9 (televisões) e 3 (celulares). As probabilidades de compra atribuídas a pessoas de ambas as classes são igualmente altas no caso desses dois produtos.
- Grupo IG, consumidores dos produtos 10 (fogões) e 5 (relógios). Probabilidades de compra iguais entre as classes econômicas.
- Grupo A, compradores de cafeteiras (produto 8), por serem compradas exclusivamente por pessoas da classe econômica A.
- Grupo B, compradores de cupons de sites de compra coletiva (produto 7), por serem compradas exclusivamente por pessoas da classe econômica B.

Em relação ao local, dia da semana e horário de compra de cada produto (seção 10.6.2) foram também considerados 5 grupos de padrões semelhantes:

- Grupo 1, contendo os produtos 8 (cafeteiras), 9 (televisões), 10 (fogões), 11 (geladeiras), 12 (microondas), 13 (máquinas de lavar) e 14 (liquidificadores), por serem comprados na segunda, terça e quarta feiras de manhã (no trabalho) por pessoas da classe econômica A e no sábado a tarde e domingo (em casa, restaurantes ou locais de lazer) por pessoas da classe B.
- Grupo 2, formado pelo produto 7 (cupons de compra coletiva) pelo fato de serem comprados nas quarta, quinta e sexta feiras no período da manhã (nos locais de trabalho dos compradores).
- Grupo 3, contendo os produtos 4(videogames) e 6(*notebooks*). São produtos comprados na segunda, terça e quarta feiras de manhã (trabalho), sábado a tarde e domingo (locais de lazer, restaurantes e residências) por ambas as classes.

- Grupo 4, com os produtos 2(câmeras) e 3 (celulares), por serem comprados na quinta e na sexta feira durante a noite (restaurantes e residências) por pessoas de ambas as classes.
- Grupo 5, produtos 1(livros) e 5(relógios), comprados por pessoas da classe B e A na segunda, terça e quarta feiras de manhã (no trabalho) e também comprados no sábado a tarde e no domingo por pessoas de classe A.

Portanto, quando se considerou tipos de tabelas que contém informações que desconsideram os locais de compra (fluxo de pessoas pelos pontos do reticulado durante todo o período considerado, sem restrições, por exemplo), foram selecionadas uma tabela de cada grupo definido em relação às probabilidades de compra de cada classe(duas tabelas no caso do grupo P1, para verificar se há diferenças substanciais entre as tabelas de um mesmo grupo).

Já para os tipos de tabelas que também consideram os locais de compra apenas foram desconsideradas no processo de mineração de dados as tabelas que continham padrões dos mesmos grupos. Dentre as tabelas dos produtos 4 e 6 por exemplo (grupo 3 quanto a local de compra e P1 quanto a probabilidades de compra de cada classe), apenas uma foi escolhida no caso por exemplo da mineração de tabelas de locais de compra.

11.2 Utilização de Algoritmos

O programa *Weka 3* oferece mais de 50 opções de algoritmos de mineração de dados. No entanto muitos deles se aplicam apenas a tabelas com determinadas características (atributo de classificação numérico, por exemplo). Outros não produzem estatísticas que favoreçam a comparação de sua eficiência com outros algoritmos (como o *functions.LeastMedSq*). Além disso, devido à grande quantidade de tabelas a serem pesquisadas pelos algoritmos e ao tempo limitado disponível para tanto, foi necessário selecionar, dentre os que funcionam para os modelos de tabelas do trabalho, os mais adequados para a utilização.

A primeira decisão quanto a redução do número de algoritmos a serem utilizados foi em relação aos algoritmos da pasta *meta* do *Weka 3*, num total de 23. A tarefa básica deles é tentar melhorar os demais algoritmos do programa que atuam puramente na mineração de dados. Cada *meta* utiliza uma estratégia diferente, podendo atuar sobre um ou vários outros algoritmos ao mesmo tempo procurando selecionar as melhores regras ou modelos, buscando taxas de acerto maiores na classificação das instâncias.

Neste trabalho o objetivo foi descobrir os algoritmos que geram os melhores conjuntos de regras para determinados tipos de dados e tabelas e não gerar um único modelo melhor possível para um conjunto de dados específico. Portanto, os algoritmos *meta* não foram utilizados, visto que visam apenas melhorar o resultado de outros algoritmos. O foco foi encontrar os algoritmos que utilizados isoladamente geram as regras mais relevantes.

Outro motivo para a não utilização dos *meta* é o de que eles multiplicariam o tempo necessário para a conclusão da pesquisa. Cada um deles teria que ser utilizado sobre um ou mais algoritmos ao mesmo tempo, podendo ser utilizados em todos. Além disso em alguns testes realizados não foi identificada melhora significativa nas taxas de acertos (foi utilizado o *meta.vote*).

Muitos algoritmos da pasta *functions* também não foram utilizados, entre eles *Logistic*, *MultilayerPerceptron*, *SimpleLogistic*, *SMO* e *SMOreg*. Houve grande demora na execução desses algoritmos. Testados em tabelas dos quatro tipos utilizados neste trabalho, em todos os casos a execução foi interrompida depois de um intervalo de tempo muito grande (geralmente em torno de meia hora e excepcionalmente de até uma hora e meia). Já o *Winnow* não foi utilizado por não suportar atributos numéricos (como coordenadas geográficas e horários). *LinearRegression*, *LeastMedSq*, *PaceRegression* e *SimpleLinearRegression* não foram usados por exigir um atributo de classificação numérico ao invés de nominal, como em *g* para grande movimento e *p* para pequeno, como é o caso das tabelas deste trabalho. *VotedPerceptron* também foi excluído por apresentar tempo de execução muito grande, suas tentativas de utilização eram interrompidas depois de meia hora sem sucesso na obtenção de regras ou modelos.

Dentre os algoritmos de árvores (pasta *trees* do *Weka 3*) o *UserClassifier* e o *LMT* também não foram utilizados devido a grande demora na execução (que foi interrompida depois de meia hora de espera, em média). O *Id3* não foi utilizado por exigir apenas atributos nominais nas instâncias (não podendo haver coordenadas nem horários). Já o *M5P* não foi utilizado por exigir atributo de classificação numérico.

Alguns algoritmos da pasta *rules* também foram excluídos: *M5Rules*, por exigir atributo de classificação numérico, *Prism* por exigir apenas atributos nominais e *ZeroR* por sempre gerar regras que resultavam em zero % de acertos para uma das classes.

Da pasta bayes do *Weka 3* foram excluídos os algoritmos *ComplementNaiveBayes* e *NaiveBayesMultinomial*,por exigirem que todos os atributos fossem numéricos, e o *AODE* por exigir que todos os atributos fossem nominais.

Da pasta *lazy* foi excluído o *LBR* por não aceitar atributos numéricos e o *LWL* porque além de apresentar uma execução excessivamente demorada é também um algoritmo como os *meta*, que atua sobre outros algoritmos na tentativa de melhorar seus resultados.

Os algoritmos selecionados foram utilizados nos quatro tipos de tabelas geradas. Nas três primeiras tabelas de cada tipo foram utilizados todos. Suas estatísticas foram então comparadas e para a mineração dos dados das demais tabelas daquele tipo foram descartados os algoritmos que geraram consistentemente modelos ruins para as três primeiras. Com isso a busca pelos melhores foi concentrada apenas nos algoritmos que geravam regras com taxas maiores de acerto.

Neste trabalho foi sempre utilizado o valor de 70 para o *Percentage Split*. Isto significa que de 100 % das instâncias de cada tabela utilizada na mineração de dados 70 % foi utilizada na construção do modelo (geração de regras, descoberta de padrões) e 30 % foi utilizada para teste do modelo ou conjunto de regras gerado. Assim foram produzidas estatísticas acerca dos erros e acertos dos modelos de classificação de instâncias gerados por cada algoritmo com a utilização de parte dos dados, dados estes isolados daqueles utilizados na geração dos modelos.

Os dados de acertos e erros (estatísticas) dos modelos gerados pelos algoritmos são representados por meio de tabelas de confusão como as presentes na tabela 11.1. Cada coluna contém o número de instâncias que foram classificadas como sendo da classe a qual a coluna faz referência. Cada linha contém o número de instâncias da classe correspondente à linha.

No caso das tabelas referenciadas são três as classes: *p* (pequeno), *m* (médio) e *g* (grande). Então o número presente na primeira linha, primeira coluna, diz respeito às

instâncias classificadas como sendo p que são realmente da classe p. Já o número presente na primeira linha, terceira coluna, é a quantidade de pontos p classificados pelas regras como se fossem g. Assim, os números das primeira e segunda linhas da terceira coluna são referentes ao número de instâncias classificadas como g mas que não são g e os números da terceira linha, primeira e segunda colunas são as instâncias da classe g que foram erroneamente classificadas.

Por meio desses números presentes nas tabelas de confusão foram selecionados os melhores algoritmos para cada tipo de tabela. São considerados melhores os algoritmos que produzem regras que abrangem mais instâncias classificadas como sendo de determinada classe de interesse, ou seja, que identificam corretamente o maior número de instâncias daquela classe.

11.3 Tabelas de Trânsito nos Pontos do Reticulado

As tabelas de trânsito nos pontos do reticulado são referentes aos pontos da cidade por onde passam mais consumidores de cada tipo de produto em determinado dia e horário. Cada instância se refere a um ponto em determinado dia e horário. Os atributos são: quantidade (movimento de pessoas), região censitária do ponto, latitude e longitude, dia da semana, horário e atributo de classificação: grande, pequeno ou médio, relativo ao fluxo de pessoas pelo ponto, conforme tabela 10.4. O atributo quantidade foi deletado antes da mineração porque o atributo de classe já resume suas informações.

Foram escolhidas para a mineração de dados as tabelas relativas aos produtos 1 (livros), 2 (câmeras), 3 (celulares), 7 (cupons de compra coletiva), 8 (cafeteiras) e 10 (fogões). Foram escolhidas de forma a cobrir todos os padrões descritos na seção 11.1, relativos às probabilidades de compra de cada produto por cada classe econômica.

Nas tabelas 1, 2 e 3 foram utilizados os seguintes algoritmos. Da pasta bayes o *BayesNet* (*ICSSearch*, *GeneticSearch* e *HillClimber*), *NaiveBayes*, *NaiveBayesSimple*, *NaiveBayesUpdatable*. Da pasta *functions* o *RBFNetwork*. Da pasta *lazy* o *IB1*, *IBK* e *KStar*. Da pasta *misc* o *VFI* e o *HyperPipes*. Da pasta *rules* o *ConjunctiveRule*, *DecisionTable*, *JRip*, *NNge*, *OneR*, *PART* e *Ridor*. Da pasta *trees* o *DecisionStump*, *J48*, *NBTree*, *RandomForest*, *RandomTree* e *REPTree*.

Nas demais, por apresentarem melhores resultados, foram utilizados apenas os algoritmos: *BayesNet* (*ICSSearch* e *HillClimber*), *NaiveBayes*, *NaiveBayesUpdatable*, *KStar*, *VFI*, *DecisionTable*, *PART*, *Ridor*, *J48*, *NBTree*, *RandomForest* e *RandomTree*. Os outros algoritmos não foram utilizados por apresentarem repetidamente resultados insatisfatórios para as três tabelas iniciais.

As tabelas 11.1, 11.2, 11.3, 11.4, 11.5 e 11.6 reuniram as tabelas de confusão dos 6 melhores algoritmos utilizados para cada público consumidor dos produtos citados, por ordem de melhor resultado. É considerado "melhor resultado"uma taxa maior de acerto na classificação das instâncias da classe g, ou seja, quanto maior a abrangência das regras sobre o conjunto de instâncias g, melhor o algoritmo.

Os algoritmos *Ridor*, *VFI*, *PART* e *KStar* tiveram desempenho satisfatório, apresentando regras que englobam grande parte dos pontos de grande movimento. Nota-se que muitos pontos foram classificados erradamente como g, o que até certo ponto não representa um grande problema pois as regras produzidas geralmente cobrem áreas da cidade dentro das quais é possível identificar os cruzamentos das vias principais e das vias secun-

dárias. Como individualmente uma regra cobre uma área pequena, mesmo que metade dos seus pontos estejam erradamente classificados é possível observar na área coberta as vias mais movimentadas.

O algoritmo *VFI* no entanto classificou uma quantidade muito grande de pontos *p* como sendo *g*, o que na prática indica que seu modelo produziu regras relativas a áreas muito amplas da cidade, possivelmente pouco específicas. Isso indica que o *VFI* não apresenta resultados tão bons quanto os outros. Neste quesito os algoritmos *PART* e *KStar*, em geral, apresentaram bons resultados.

O algoritmo *Ridor* foi, em geral, o melhor em termos de identificação de instâncias *g*, mas os algoritmos *PART* e *KStar* produziram regras mais específicas. Os três algoritmos foram portanto os melhores para este modelo de tabela.

11.4 Tabelas de Movimento Anterior ao Ato da Compra

As tabelas de movimento anterior ao ato da compra são referentes aos pontos do reticulado e o fluxo de consumidores de cada tipo de produto em determinado dia e horário desde que os caminhos por eles realizados sejam imediatamente anteriores ao ato da compra do produto. Cada instância se refere a um ponto em determinado dia e horário. Os atributos são: quantidade (movimento de pessoas), região censitária do ponto, latitude e longitude, dia da semana, horário e atributo de classificação: grande, pequeno ou médio, relativo ao fluxo de pessoas pelo ponto, conforme tabela 10.4. O atributo quantidade foi deletado antes da mineração porque o atributo de classe já resume suas informações.

As tabelas desse tipo escolhidas foram a 1 (livros), 2 (câmeras), 3(celulares), 5 (relógios), 6 (*notebooks*), 7 (cupons de compra coletiva), 8 (cafeteiras), 9 (televisões), 10 (fogões) e 11 (geladeiras). Elas englobam todos os padrões relativos às probabilidades de compra de cada produto por cada classe econômica e também à data e ao local das compras.

Nas tabelas 1, 2 e 3 foram utilizados os seguintes algoritmos. Da pasta bayes o *BayesNet* (*ICSSearch*, *GeneticSearch* e *HillClimber*), *NaiveBayes*, *NaiveBayesSimple*, *NaiveBayesUpdatable*. Da pasta *functions* o *RBFNetwork*. Da pasta *lazy* o *IB1*, *IBK* e *KStar*. Da pasta *misc* o *VFI* e o *HyperPipes*. Da pasta *rules* o *ConjunctiveRule*, *DecisionTable*, *JRip*, *NNge*, *OneR*, *PART* e *Ridor*. Da pasta *trees* o *DecisionStump*, *J48*, *NBTree*, *RandomForest*, *RandomTree* e *REPTree*.

Excepcionalmente, na tabela 1, os algoritmos da pasta *functions LeastMedSq* e *LinearRegression*, o algoritmo da pasta *trees M5P* e o algoritmo da pasta *rules M5Rules* foram utilizados. Para tanto foi necessário deletar o atributo de classificação nominal (*p, m, g*). O atributo da quantidade, no caso apenas desses algoritmos, foi mantido e considerado como atributo de classificação. O resultado foram algoritmos muito lentos (alguns tiveram a execução interrompida depois de meia hora) e os que executaram até o final não produziram estatísticas próprias para a avaliação do modelo criado e nem para a comparação dos resultados com outros algoritmos. Por esse motivo os algoritmos citados não foram mais utilizados neste trabalho.

Nas tabelas referentes aos consumidores dos outros produtos, por apresentarem melhores resultados, foram utilizados apenas os algoritmos: *BayesNet* (*ICSSearch*, *HillClimber* e

GeneticSearch), *NaiveBayes*, *NaiveBayesSimple*, *NaiveBayesUpdatable*, *KStar*, *IBK*, *VFI*, *PART*, *Ridor*, *J48*, *RandomForest*, *RandomTree* e *REPTree*.

As tabelas 11.7, 11.8, 11.9, 11.10, 11.11, 11.12, 11.13, 11.14, 11.15 e 11.16, reuniram as tabelas de confusão dos 6 melhores algoritmos para cada produto por ordem de melhor resultado. É considerado "melhor resultado"uma taxa maior de acerto na classificação das instâncias da classe *g*.

Os algoritmos *Ridor* e *VFI* identificaram muito bem a grande maioria das instâncias *g* em todas as tabelas. *PART*, *RandomTree* e *RandomForest*, em geral, identificaram medianamente bem tais instâncias. No entanto o *VFI* continuou classificando erradamente muitas instâncias *p* como se fossem *g*, permitindo deduzir que a qualidade de seu modelo não é muito boa. Provavelmente obtém altas taxas de acerto de instâncias *g* por considerar regras muito amplas (e portanto com alta taxa de erro).

Assim, *Ridor*, em geral, foi o melhor algoritmo, seguido por *PART*, *RandomTree* e *RandomForest*, que foram medianos. Esporadicamente o *BayesNet* também gerou bons resultados.

11.5 Tabelas de Caminhos e Regiões

As tabelas de Caminhos e Regiões contém os caminhos realizados por todas as pessoas. A rotina completa de todas as pessoas da população gerada, desde que as pessoas sejam compradoras do produto ao qual a tabela faz referência. Cada linha se refere a uma caminho realizado por uma pessoa em um determinado dia e horário. São 70 atributos. O primeiro é o dia em que foi percorrido o caminho, seguido por 34 pares região/hora. Cada par indica a passagem do caminho da linha por uma determinada região em uma determinada hora. Assim o primeiro par contém o identificador da primeira região por onde passa o caminho e o horário que isso ocorre, o segundo par contém os dados sobre a segunda região transitada e assim por diante, conforme a tabela 10.6.

Por último há o atributo de classificação. No caso das tabelas do trabalho um determinado caminho foi classificado como *s* caso passasse dentro de uma região determinada do Setor Marista numa determinada faixa de horário. Portanto, foi necessário deletar o atributo indicando a passagem por essa região pois a existência ou não desse atributo define a classe da instância. Caso permanecesse na tabela os algoritmos revelariam essa regra de classificação e acertariam 100 % das instâncias, deixando de produzir os resultados esperados. Dessa forma todas as instâncias que continham essa região em um de seus atributos tiveram os dados desse atributo apagados. Além disso foram também apagadas as regiões limítrofes da região de classificação com intuito de produzir regras com informações mais úteis. Caso as regiões limítrofes permanecessem as regras produzidas não teriam utilidade (pois já é conhecido o fato de que para se chegar a determinada região é necessário transitar por suas regiões vizinhas).

Dessa forma, devido às alterações realizadas nas tabelas descritas acima pôde-se garantir que a classe das instâncias não é função linear dos seus atributos. Com isso o processo de mineração de dados pôde ser realizado sem o risco de descoberta de regras sem utilidade.

As tabelas escolhidas para a mineração de dados foram as relativas aos produtos 5 (relógios), 7 (cupons de compra coletiva), 8 (cafeteiras), 9 (televisões), 11 (geladeiras) e 12 (micro-ondas). Elas englobam todos os padrões relativos às probabilidades de compra

de cada produto por cada classe econômica, não sendo necessário nesse contexto escolher produtos que cubram os padrões de locais de compra visto que todos os caminhos da rotina das pessoas foram considerados.

Nas tabelas 5, 7 e 8 foram utilizados os seguintes algoritmos. Da pasta bayes o *NaiveBayes* e o *NaiveBayesUpdatable*. Da pasta *functions* o *RBFNetwork*. Da pasta *lazy* o *IB1*, *IBK* e *KStar*. Da pasta *misc* o *VFI* e o *HyperPipes*. Da pasta *rules* o *ConjunctiveRule*, *DecisionTable*, *JRip*, *NNge*, *OneR*, *PART* e *Ridor*. Da pasta *trees* o *ADTree*, *DecisionStump*, *J48*, *NBTree*, *RandomForest*, *RandomTree* e *REPTree*.

O algoritmo *BayesNet* não foi utilizado por falta de memória no computador usado para a mineração. Devido à natureza das tabelas descritas nesta seção é necessária mais memória quando utilizado tal algoritmo. Já o *NaiveBayesSimple* não foi utilizado por um problema com um dos atributos (desvio padrão 0 para o atributo hora31).

Por terem apresentado resultados insatisfatórios para as três primeiras tabelas os algoritmos seguintes não foram mais utilizados: *RBFNetwork*, *NBTree*, *ConjunctiveRule* e *DecisionStump*.

As tabelas 11.17, 11.18, 11.19, 11.20, 11.21 e 11.22 reuniram as tabelas de confusão dos 7 melhores algoritmos para cada produto por ordem de melhor resultado. É considerado "melhor resultado"uma taxa maior de acerto na classificação das instâncias da classe *s*, ou seja, quanto maior a abrangência das regras sobre o conjunto de instâncias *s*, melhor o algoritmo.

O algoritmo *RandomForest*, foi o melhor, gerando modelos que classificavam bem as instâncias *s* para todas as tabelas, seguido pelo *J48* que, em geral, teve bom desempenho. Os algoritmos *Ridor*, *VFI* e *ADTree* foram os que geraram as regras de menores taxas de acerto de instâncias *s*, apesar de serem taxas altas mesmo assim. Já os algoritmos *HyperPipes*, *OneR*, *NaiveBayesUpdatable* e *NaiveBayes* geraram regras com taxas menores de acerto em relação às instâncias *n*, ou seja, suas regras para classificação das instâncias *s* englobaram muitas instâncias *n*, indicando que eles geraram regras pouco específicas e por isso de qualidade duvidosa. Os demais algoritmos *IB1*, *IBK*, *KStar*, *DecisionTable*, *JRip*, *NNge*, *PART*, *RandomTree* e *REPTree*, em geral, geraram bons resultados.

Assim, *RandomForest* foi o melhor algoritmo, seguido pelo *J48*.

Observou-se nesse tipo de tabela uma taxa de acerto muito alta dos modelos gerados por grande parte dos algoritmos utilizados. Isso ocorreu talvez pelo fato de as instâncias terem muitos atributos e serem por esse motivo facilmente diferenciáveis umas das outras, além do fato de a quantidade de instâncias ser pequena. Também é possível que as regiões excluídas vizinhas à região de classificação não tenham sido suficientes, ou seja, alguma região próxima pela qual todos que passavam na região de classificação transitavam deve ter sido utilizada nos modelos criados.

11.6 Tabelas de Locais das Compras

As tabelas de locais das compras reunem os pontos do reticulado onde cada tipo de produto foi comprado em um dia e horário específicos. Cada instância se refere a um ponto onde a compra de um produto ocorreu. Os atributos são: dia da semana, horário, coordenadas do ponto, região censitária e atributo de classificação: *s* caso tenha ocorrido uma compra naquele ponto no dia e horário especificado ou *n* caso contrário, conforme tabela 10.5.

Os produtos escolhidos para a mineração foram 1 (livros), 2 (câmeras), 3(celulares), 4 (videogames), 5 (relógios), 7 (cupons de compra coletiva), 8 (cafeteiras), 9 (televisões), 10 (fogões) e 12 (microondas). Incluem todos os padrões relativos às probabilidades de compra de cada produto por cada classe econômica e também os padrões relativos à data e ao local das compras.

Nas tabelas 1, 2 e 3 foram utilizados os seguintes algoritmos. Da pasta bayes o *BayesNet* (*ICSSearch* e *HillClimber*), *NaiveBayes*, *NaiveBayesSimple*, *NaiveBayesUpdatable*. Da pasta *functions* o *RBFNetwork*. Da pasta *lazy* o *IB1*, *IBK* e *KStar*. Da pasta *misc* o *VFI* e o *HyperPipes*. Da pasta *rules* o *ConjunctiveRule*, *DecisionTable*, *JRip*, *NNge*, *OneR*, *PART* e *Ridor*. Da pasta *trees* o *ADTree*, DecisionStump, *J48*, *NBTree*, *RandomForest*, *RandomTree* e *REPTree*. Não houve memória suficiente para viabilizar a utilização do *BayesNet GeneticSearch*.

Nas demais tabelas *ConjunctiveRule*, *NNge*, *OneR*, DecisionStump e *REPTree* não foram utilizados por terem gerado resultados pouco expressivos para as três primeiras tabelas.

As tabelas 11.23, 11.24, 11.25, 11.26, 11.27, 11.28, 11.29, 11.30, 11.31 e 11.32, reuniram as tabelas de confusão dos 7 melhores algoritmos para cada produto por ordem de melhor resultado. É considerado "melhor resultado"uma taxa maior de acerto na classificação das instâncias da classe *s*.

Os algoritmos *Ridor* e *IBK*, em geral, foram os melhores na identificação de regras para a classe *s* conseguindo classificar corretamente a maioria dessas instâncias. *KStar* e *RandomForest* também foram bons. *HyperPipes* e *VFI* apesar de identificarem corretamente a maioria das instâncias *s*, classificaram muitas *n* como se fossem *s*. Isso sugere que seus modelos não foram bons, pois possivelmente atribuíram regras pouco específicas às instâncias *s*.

Os algoritmos que tiveram as piores taxas de acerto de instâncias de classe *s* foram *NaiveBayes*, *NaiveBayesSimple*, *NaiveBayesUpdatable*, *BayesNet*, *RBFNetwork*, *JRip* e *ADTree*. Os demais algoritmos testados, em geral, geraram modelos de bom desempenho.

Rules.Ridor
-F 4 -S 1 -N 2.0

p	m	g	
6870	183	1083	p
830	250	956	m
175	41	1732	g

Misc.VFI -B 0.6

p	m	g	
4351	1719	2066	p
484	666	886	m
187	386	1375	g

Rules.PART
-B -M 2 -C 0.05 -Q 1

p	m	g	
7358	514	264	p
751	888	397	m
225	367	1356	g

BayesNet - Q *ICSSearch*
-mbc -S AIC .SE – -A 5.0

p	m	g	
5581	950	1605	p
749	534	753	m
321	275	1352	g

lazy.Kstar
-B 20 -E -M a

p	m	g	
6776	868	492	p
685	933	418	m
277	334	1337	g

BayesNet -Q *local.HillClimber*
-R -P 1-mbc -S AIC-E SE -A 60.0

p	m	g	
6992	273	871	p
1193	185	658	m
565	103	1280	g

Tabela 11.1: Movimento nos Pontos - P1

Rules.Ridor
-F 8 -S 1 -N 2.0

p	m	g	
7612	120	856	p
653	152	638	m
103	26	992	g

Misc.VFI -B 0.6

p	m	g	
4801	1765	2022	p
356	486	601	m
96	269	756	g

Rules.PART
-B -M 2 -C 0.05 -Q 1

p	m	g	
8048	368	172	p
539	676	228	m
148	231	742	g

BayesNet -Q *ICSSearch*
-mbc -S AIC SE -A 5.0

p	m	g	
5790	1234	1564	p
497	533	413	m
194	233	694	g

Trees.J48
-C 0.4 -B -M 2

p	m	g	
7966	402	220	p
687	565	191	m
248	203	670	g

Rules.DecisionTable
-X 1 -S 5 -I -R

p	m	g	
8025	372	191	p
739	522	182	m
308	188	625	g

Tabela 11.2: Movimento nos Pontos - P2

Rules.Ridor
-F 7 -S 1 -N 2.0

p	m	g	
7211	188	1079	p
1045	290	1208	m
189	42	2321	g

Trees.J48
-C 0.05 -B -M 2

p	m	g	
7617	376	485	p
1291	726	526	m
446	265	1841	g

Rules.PART
-B -M 2 -C 0.25 -Q 1

p	m	g	
7478	696	304	p
811	1194	538	m
253	492	1807	g

Misc.VFI -B 0.6

p	m	g	
4396	1909	2173	p
611	857	1075	m
272	504	1776	g

BayesNet -Q *ICSSearch*
-mbc -S AIC -c 2 -E SE -A 0.5

p	m	g	
7329	541	608	p
1258	608	677	m
495	341	1716	g

lazy.Kstar
-B 20 -E -M a

p	m	g	
7220	750	508	p
1034	1053	456	m
478	375	1699	g

Tabela 11.3: Movimento nos Pontos - P3

Rules.Ridor
-F 7 -S 1 -N 2.0

p	m	g	
5456	687	356	p
1800	1510	878	m
161	158	1413	g

Misc.VFI -B 0.6

p	m	g	
3315	1564	1620	p
941	1495	1752	m
145	327	1260	g

lazy.Kstar
-B 2 -M a

p	m	g	
4792	1427	280	p
936	2751	501	m
143	422	1167	g

Rules.PART
-B -M 2 -C 0.5 -Q 1

p	m	g	
5435	966	98	p
1065	2716	407	m
85	498	1149	g

Trees.J48
-C 0.25 -B -M 2

p	m	g	
5308	1026	165	p
1293	2548	347	m
179	474	1079	g

Rules.DecisionTable
-X 1 -S 5 -I -R

p	m	g	
5174	1126	199	p
1214	2684	290	m
203	470	1059	g

Tabela 11.4: Movimento nos Pontos - P7

Rules.Ridor
-F 7 -S 1 -N 2.0

p	m	g	
4946	556	702	p
1623	1175	1276	m
115	119	1533	g

lazy.Kstar
-B 2 -M a

p	m	g	
4463	1418	323	p
989	2584	501	m
139	410	1218	g

Misc.VFI -B 0.6

p	m	g	
3379	1397	1428	p
1263	1402	1409	m
226	368	1173	g

Rules.PART
-B -M 2 -C 0.05 -Q 1

p	m	g	
5056	978	170	p
1111	2496	467	m
125	518	1124	g

Trees.J48
-C 0.6 -B -M 2

p	m	g	
4869	1155	180	p
1182	2496	396	m
191	489	1087	g

Trees.RandomTree
-K 6 -M 1.0 -S 1

p	m	g	
4490	1416	298	p
1257	2331	486	m
245	495	1027	g

Tabela 11.5: Movimento nos Pontos - P8

Rules.Ridor
-F 3 -S 1 -N 2.0

p	m	g	
7984	184	415	p
677	228	467	m
124	65	1010	g

Misc.VFI -B 0.6

p	m	g	
4502	1850	2231	p
242	475	655	m
59	232	908	g

lazy.Kstar
-B 2 -M a

p	m	g	
7523	711	349	p
478	643	251	m
184	190	825	g

Rules.PART
-B -M 2 -C 0.05 -Q 1

p	m	g	
8021	389	173	p
535	587	250	m
136	247	816	g

Trees.J48
-C 5.0 -B -M 2

p	m	g	
7831	526	226	p
591	569	212	m
243	215	741	g

BayesNet -Q *ICSSearch*
-mbc -S AIC -c 2 -E SE -A 0.05

p	m	g	
7793	409	381	p
684	393	295	m
284	178	737	g

Tabela 11.6: Movimento nos Pontos - P10

Rules.Ridor
-F 22 -S 1 -N 2.0

p	m	g	
3553	40	728	p
524	22	370	m
109	15	623	g

Rules.Ridor
-F 15 -S 1 -N 2.0

p	m	g	
3648	49	624	p
531	39	346	m
133	14	600	g

Misc.VFI -B 0.5

p	m	g	
2277	888	1156	p
270	258	388	m
66	156	525	g

BayesNet -Q *GeneticSearch*
-L 15 -A 150 -U 15 -mbc -S BAYES -E

p	m	g	
3076	247	998	p
445	189	282	m
196	80	471	g

Rules.PART
-M 2 -C 0.3 -Q 1

p	m	g	
3839	320	162	p
475	307	134	m
152	145	450	g

BayesNet -Q *ICSSearch*
-mbc -S AIC -c 2 -E SE -A 11.0

p	m	g	
3373	318	630	p
480	195	241	m
203	100	444	g

Tabela 11.7: Movimento Compras - P1

Rules.Ridor
-F 17 -S 1 -N 2.0

p	m	g	
1408	10	389	p
184	10	259	m
48	2	464	g

lazy.Kstar
-B 2 -M a

p	m	g	
1241	296	270	p
136	187	130	m
32	91	391	g

BayesNet -Q local.*HillClimber*
-R -P 1 -mbc -S AIC -E SE -A 70.0

p	m	g	
1354	174	279	p
196	83	174	m
94	43	377	g

BayesNet -Q *GeneticSearch*
-L 20 -A 200 -U 20 -mbc -S BAYES -E

p	m	g	
1350	74	383	p
225	32	196	m
114	28	372	g

Trees.RandomForest
-I 50 -K 0 -S 1

p	m	g	
1575	176	56	p
183	188	82	m
56	86	372	g

NaiveBayesSimple

p	m	g	
1508	56	243	p
244	50	159	m
112	34	368	g

Tabela 11.8: Movimento Compras - P2

Rules.Ridor
-F 9 -S 1 -N 2.0

p	m	g	
1125	36	468	p
210	37	337	m
66	19	900	g

NaiveBayesSimple

p	m	g	
1147	31	451	p
273	11	300	m
157	2	826	g

NaiveBayes

p	m	g	
1116	27	486	p
280	11	293	m
174	2	809	g

NaiveBayesUpdateable

p	m	g	
1116	27	486	p
280	11	293	m
174	2	809	g

lazy.Kstar
-B 40 -M a

p	m	g	
982	311	336	p
160	191	233	m
71	131	783	g

BayesNet -Q *GeneticSearch*
-L 10 -A 100 -U 10 -mbc -S BAYES -E

p	m	g	
1271	66	292	p
310	40	234	m
178	35	772	g

Tabela 11.9: Movimento Compras - P3

Rules.Ridor
-F 15 -S 1 -N 2.0

p	m	g	
3227	27	349	p
287	21	217	m
23	5	273	g

Misc.VFI
-C -B 0.5

p	m	g	
2094	747	762	p
133	153	239	m
29	58	214	g

Trees.J48
-C 0.7 -B -M 2

p	m	g	
3421	149	33	p
274	201	50	m
58	57	186	g

Rules.PART
-M 2 -C 0.25 -Q 1

p	m	g	
3415	142	46	p
275	196	54	m
58	79	164	g

Trees.RandomTree
-K 5 -M 1.0 -S 1

p	m	g	
3227	265	111	p
266	172	87	m
79	75	147	g

Trees.REPTree
-M 1 -V 0.0010 -N 7 -S 1 -L -1 -P

p	m	g	
3227	265	111	p
266	172	87	m
79	75	147	g

Tabela 11.10: . Movimento Compras - P5

Rules.Ridor
-F 7 -S 1 -N 1.0

p	m	g	
4157	69	853	p
384	78	438	m
47	12	530	g

Misc.VFI -B 0.5

p	m	g	
2239	1364	1476	p
179	329	392	m
36	125	428	g

Rules.PART
-B -M 2 -C 0.05 -Q 1

p	m	g	
4693	319	67	p
445	351	104	m
92	127	370	g

Trees.RandomTree
-K 2 -M 1.0 -S 1

p	m	g	
4451	484	144	p
443	323	134	m
121	147	321	g

Trees.RandomForest
-I 50 -K 0 -S 1

p	m	g	
4506	410	163	p
515	252	133	m
151	128	310	g

Trees.J48
-C 0.5 -B -M 2

p	m	g	
4722	267	90	p
488	297	115	m
176	110	303	g

Tabela 11.11: . Movimento Compras - P6

Rules.Ridor
-F 7 -S 1 -N 2.0

p	m	g	
570	10	601	p
95	15	493	m
19	7	1310	g

BayesNet -Q *ICSSearch*
-mbc -S BDeu -c 8 -E SE -A 0.5

p	m	g	
864	40	277	p
282	44	277	m
147	36	1153	g

Trees.J48
-C 0.05 -B -M 2

p	m	g	
951	56	174	p
274	125	204	m
128	71	1137	g

Trees.RandomForest
-I 60 -K 0 -S 1

p	m	g	
940	158	83	p
180	276	147	m
72	140	1124	g

BayesNet -Q *GeneticSearch*
-L 10 -A 100 -U 10 -mbc -S AIC -E

p	m	g	
872	77	232	p
259	105	239	m
170	68	1098	g

Rules.PART
-M 2 -C 0.25 -Q 1

p	m	g	
935	141	105	p
209	256	138	m
80	169	1087	g

Tabela 11.12: . Movimento Compras - P7

Rules.Ridor
-F 5 -S 1 -N 2.0

p	m	g	
3532	71	632	p
539	116	519	m
179	34	1123	g

Misc.VFI -B 0.8

p	m	g	
1768	1033	1434	p
313	321	540	m
156	241	939	g

BayesNet -Q *ICSSearch*
-mbc -S BAYES -c 4 -E SE -A 3.0

p	m	g	
3410	314	511	p
558	281	335	m
298	140	898	g

Rules.PART
-B -M 2 -C 0.7 -Q 1

p	m	g	
3718	350	167	p
500	467	207	m
222	226	888	g

Trees.RandomTree
-K 5 -M 1.0 -S 1

p	m	g	
3374	590	271	p
469	496	209	m
243	220	873	g

BayesNet -Q *GeneticSearch*
-L 10 -A 100 -U 10 -mbc -S BAYES -E

p	m	g	
3472	264	499	p
654	222	298	m
343	125	868	g

Tabela 11.13: . Movimento Compras - P8

Rules.Ridor
-F 8 -S 1 -N 2.0

p	m	g	
4672	46	636	p
503	78	332	m
132	15	686	g

Misc.VFI -B 0.2

p	m	g	
2349	1456	1549	p
192	375	346	m
62	223	548	g

Trees.RandomTree
-K 10 -M 1.0 -S 1

p	m	g	
4568	548	238	p
404	376	133	m
155	169	509	g

Rules.PART
-B -M 2 -C 0.05 -Q 1

p	m	g	
4932	312	110	p
492	282	139	m
175	149	509	g

BayesNet -Q *ICSSearch*
-mbc -S AIC -c 2 -E se -a 30.0

p	m	g	
4209	50	1095	p
565	57	291	m
315	33	485	g

Trees.RandomForest
-I 50 -K 0 -S 1

p	m	g	
4687	417	250	p
507	279	127	m
221	154	458	g

Tabela 11.14: . Movimento Compras - P9

Rules.Ridor
-F 14 -S 1 -N 2.0

p	m	g	
4262	9	566	p
357	20	261	m
66	6	322	g

Misc.VFI -B 0.2

p	m	g	
2486	1086	1265	p
197	190	251	m
32	89	273	g

BayesNet -Q *ICSSearch*
-mbc -S AIC -c 2 -E SE -A 30.0

p	m	g	
3777	48	1012	p
407	48	183	m
146	28	220	g

Rules.PART
-B -M 2 -C 0.25 -Q 1

p	m	g	
4553	217	67	p
373	197	68	m
82	101	211	g

Trees.RandomTree
-K 5 -M 1.0 -S 1

p	m	g	
4299	396	142	p
323	238	77	m
88	97	209	g

Trees.RandomForest
-I 50 -K 0 -S 1

p	m	g	
4403	316	118	p
385	184	69	m
119	82	193	g

Tabela 11.15: . Movimento Compras - P10

Rules.Ridor
-F 15 -S 1 -N 2.0

p	m	g	
4156	9	704	p
391	15	338	m
61	4	386	g

Misc.VFI -B 0.6

p	m	g	
2368	1233	1268	p
157	270	317	m
28	91	332	g

Trees.RandomTree
-K 5 -M 1.0 -S 1

p	m	g	
4311	420	138	p
377	271	96	m
115	91	245	g

lazy.Kstar
-B 1 -M a

p	m	g	
4103	541	225	p
376	240	128	m
116	91	244	g

Rules.PART
-B -M 2 -C 0.0050 -Q 1

p	m	g	
4562	227	80	p
423	229	92	m
114	105	232	g

Trees.RandomForest
-I 80 -K 0 -S 1

p	m	g	
4380	351	138	p
447	208	89	m
152	80	219	g

Tabela 11.16: . Movimento Compras - P11

Rules.ConjunctiveRule
-N 8 -M 2.0 -P -1 -S 1

s	n	
752	1	s
337	757	n

Lazy.IBK
-K 1 -W 0

s	n	
749	4	s
51	1043	n

Lazy.IB1

s	n	
749	4	s
51	1043	n

Trees.RandomForest
-I 20 -K 0 -S 1

s	n	
748	5	s
21	1073	n

Trees.J48
-C 0.05 -B -M 2

s	n	
748	5	s
61	1033	n

Trees.DecisionStump

s	n	
746	7	s
328	766	n

Rules.NNge
-G 6 -I 6

s	n	
741	12	s
18	1076	n

Tabela 11.17: . Regiões dos Caminhos - P5

Trees.RandomTree
-K 1 -M 1.0 -S 1

s	n	
3010	0	s
71	3402	n

Trees.RandomForest
-I 23 -K 0 -S 1

s	n	
3010	0	s
15	3458	n

Misc.HyperPipes

s	n	
3010	0	s
222	3251	n

Lazy.IBK
-K 1 -W 0

s	n	
3010	0	s
45	3428	n

Trees.J48
-C 0.25 -B -M 2

s	n	
3007	3	s
68	3405	n

NaiveBayesUpdateable -K

s	n	
3005	5	s
406	3067	n

lazy.Kstar
-B 3 -M a

s	n	
983	3	s
33	1142	n

Tabela 11.18: . Regiões dos Caminhos - P7

Trees.RandomTree
-K 1 -M 1.0 -S 1

s	n	
2321	0	s
123	4184	n

Trees.RandomForest
-I 19 -K 0 -S 1

s	n	
2321	0	s
50	4257	n

Trees.J48
-C 0.15 -B -M 2

s	n	
2321	0	s
139	4168	n

Misc.HyperPipes

s	n	
2321	0	s
400	3907	n

Lazy.IBK
-K 1 -W 0

s	n	
2321	0	s
78	4229	n

Rules.NNge
-G 5 -I 5

s	n	
2318	3	s
60	4247	n

Rules.ConjunctiveRule
-N 7 -M 2.0 -P -1 -S 1

s	n	
2319	2	s
1429	2878	n

Tabela 11.19: . Regiões dos Caminhos - P8

Trees.J48
-C 0.05 -B -M 2

s	n	
1942	5	s
117	2484	n

Trees.RandomForest
-I 20 -K 0 -S 1

s	n	
1938	9	s
45	2556	n

Rules.DecisionTable
-X 1 -S 5 -I -R

s	n	
1937	10	s
84	2517	n

Misc.HyperPipes

s	n	
1938	9	s
332	2269	n

Rules.PART
-B -M 2 -C 0.1 -Q 1

s	n	
1934	13	s
75	2526	n

Rules.NNge
-G 7 -I 7

s	n	
1930	17	s
56	2545	n

Lazy.IBK
-K 1 -W 0

s	n	
1930	17	s
99	2502	n

Tabela 11.20: . Regiões dos Caminhos - P9

Rules.DecisionTable
-X 1 -S 6 -I -R

s	n	
1094	7	s
124	1579	n

Misc.HyperPipes

s	n	
1094	7	s
155	1548	n

Trees.RandomForest
-I 25 -K 0 -S 1

s	n	
1091	10	s
35	1668	n

Rules.OneR -B 6

s	n	
1091	10	s
276	1427	n

Rules.JRip
-F 5 -N 2.0 -O 3 -S 1

s	n	
1090	11	s
111	1592	n

lazy.Kstar
-B 4 -M a

s	n	
546	6	s
65	785	n

Lazy.IBK
-K 2 -W 0

s	n	
1089	12	s
138	1565	n

Tabela 11.21: . Regiões dos Caminhos - P11

Trees.RandomForest
-I 28 -K 0 -S 1

s	n	
1010	5	s
35	1510	n

Trees.J48
-C 0.15 -B -M 2

s	n	
1009	6	s
104	1441	n

Rules.DecisionTable
-X 1 -S 6 -I -R

s	n	
1009	6	s
108	1437	n

Misc.HyperPipes

s	n	
1007	8	s
177	1368	n

Lazy.IBK
-K 1 -W 0

s	n	
1005	10	s
80	1465	n

Rules.NNge
-G 4 -I 4

s	n	
1003	12	s
35	1510	n

lazy.Kstar
-B 3 -M a

s	n	
640	9	s
84	974	n

Tabela 11.22: . Regiões dos Caminhos - P12

Misc.HyperPipes

s	n	
1244	3	s
3065	5512	n

Rules.Ridor
-F 7 -S 1 -N 2.0

s	n	
1204	43	s
441	8136	n

Lazy.IBK
-K 2 -W 0

s	n	
1194	53	s
533	8044	n

lazy.Kstar
-B 3 -M a

s	n	
1163	84	s
331	8246	n

BayesNet -Q *ICSSearch*
-mbc -S AIC -c 2 -E SE -A 0.5

s	n	
1159	88	s
619	7958	n

Trees.RandomTree
-K 1 -M 1.0 -S 1

s	n	
1156	91	s
271	8306	n

Lazy.IB1

s	n	
1156	91	s
285	8292	n

Tabela 11.23: . Local da Compra - P1

Misc.VFI -B 0.4

s	n	
837	0	s
191	8401	n

Misc.HyperPipes

s	n	
837	0	s
189	8403	n

Rules.Ridor
-F 3 -S 1 -N 2.0

s	n	
827	10	s
195	8397	n

Lazy.IBK
-K 2 -W 0

s	n	
825	12	s
244	8348	n

lazy.Kstar
-B 3 -M a

s	n	
816	21	s
230	8362	n

NaiveBayesUpdateable -K

s	n	
812	25	s
111	8481	n

NaiveBayes -K

s	n	
812	25	s
111	8481	n

Tabela 11.24: . Local da Compra - P2

Misc.HyperPipes

s	n	
1640	2	s
162	8321	n

Rules.Ridor
-F 3 -S 1 -N 2.0

s	n	
1637	5	s
254	8229	n

Misc.VFI -B 0.34

s	n	
1635	7	s
156	8327	n

Lazy.IBK
-K 2 -W 0

s	n	
1628	14	s
253	8230	n

Trees.RandomForest
-I 30 -K 0 -S 1

s	n	
1619	23	s
82	8401	n

Rules.JRip
-F 7 -N 2.0 -O 2 -S 1

s	n	
1619	23	s
208	8275	n

Lazy.IB1

s	n	
1619	23	s
137	8346	n

Tabela 11.25: . Local da Compra - P3

Misc.HyperPipes

s	n	
1077	10	s
3394	5102	n

Rules.Ridor
-F 8 -S 1 -N 2.0

s	n	
1034	53	s
550	7946	n

Lazy.IBK
-K 2 -W 0

s	n	
1009	78	s
574	7922	n

BayesNet -Q *ICSSearch*
-mbc -S AIC -c 2 -E SE -A 0.5

s	n	
988	99	s
824	7672	n

lazy.Kstar
-B 4 -M a

s	n	
983	104	s
394	8102	n

Rules.DecisionTable
-X 2 -S 5 -I -R

s	n	
973	114	s
255	8241	n

Lazy.IB1

s	n	
961	126	s
313	8183	n

Tabela 11.26: . Local da Compra - P4

Misc.HyperPipes

s	n	
562	9	s
2589	5918	n

Misc.VFI -B 0.4

s	n	
537	34	s
1863	6644	n

Lazy.IBK
-K 2 -W 0

s	n	
501	70	s
326	8181	n

lazy.Kstar
-B 2 -M a

s	n	
500	71	s
191	8316	n

Rules.Ridor
-F 3 -S 1 -N 2.0

s	n	
497	74	s
246	8261	n

Trees.J48
-C 0.5 -B -M 2

s	n	
482	89	s
76	8431	n

Trees.RandomForest
-I 25 -K 0 -S 1

s	n	
481	90	s
93	8414	n

Tabela 11.27: . Local da Compra - P5

Trees.RandomForest
-I 20 -K 0 -S 1

s	n	
1880	0	s
50	8470	n

Misc.VFI -B 0.5

s	n	
1880	0	s
129	8391	n

Misc.HyperPipes

s	n	
1880	0	s
126	8394	n

lazy.Kstar
-B 2 -M a

s	n	
1880	0	s
179	8341	n

Lazy.IBK
-K 2 -W 0

s	n	
1880	0	s
212	8308	n

NaiveBayesUpdateable

s	n	
1880	0	s
203	8317	n

NaiveBayesSimple

s	n	
1880	0	s
203	8317	n

Tabela 11.28: . Local da Compra - P7

Misc.HyperPipes

s	n	
2265	3	s
2863	5573	n

Lazy.IBK
-K 2 -W 0

s	n	
2235	33	s
551	7885	n

Rules.Ridor
-F 7 -S 1 -N 2.0

s	n	
2217	51	s
467	7969	n

lazy.Kstar
-B 4 -M a

s	n	
2218	50	s
386	8050	n

Trees.RandomForest
-I 35 -K 0 -S 1

s	n	
2215	53	s
173	8263	n

Lazy.IB1

s	n	
2211	57	s
292	8144	n

Trees.RandomTree
-K 1 -M 1.0 -S 1

s	n	
2190	78	s
260	8176	n

Tabela 11.29: . Local da Compra - P8

Misc.HyperPipes

s	n	
1328	7	s
3336	5238	n

Rules.Ridor
-F 8 -S 1 -N 2.0

s	n	
1282	53	s
704	7870	n

Lazy.IBK
-K 2 -W 0

s	n	
1240	95	s
614	7960	n

lazy.Kstar
-B 3 -M a

s	n	
1229	106	s
387	8187	n

Trees.RandomForest
-I 20 -K 0 -S 1

s	n	
1221	114	s
208	8366	n

Rules.DecisionTable
-X 1 -S 5 -I -R

s	n	
1216	119	s
274	8300	n

Lazy.IB1

s	n	
1209	126	s
356	8218	n

Tabela 11.30: . Local da Compra - P9

Misc.HyperPipes

s	n	
848	4	s
3238	5297	n

Rules.Ridor
-F 8 -S 1 -N 2.0

s	n	
738	114	s
498	8037	n

Lazy.IBK
-K 2 -W 0

s	n	
730	122	s
496	8039	n

Misc.VFI -B 0.3

s	n	
722	130	s
1370	7165	n

lazy.Kstar
-B 2 -M a

s	n	
709	143	s
276	8259	n

Lazy.IB1

s	n	
704	148	s
254	8281	n

Rules.DecisionTable
-X 1 -S 5 -I -R

s	n	
700	152	s
182	8353	n

Tabela 11.31: . Local da Compra - P10

Misc.HyperPipes

s	n	
819	10	s
3053	5398	n

BayesNet -Q *ICSSearch*
-mbc -S AIC -c 2 -E SE -A 0.5

s	n	
745	84	s
799	7652	n

Rules.Ridor
-F 4 -S 1 -N 2.0

s	n	
727	102	s
389	8062	n

Lazy.IBK
-K 2 -W 0

s	n	
721	108	s
453	7998	n

Misc.VFI -B 0.4

s	n	
697	132	s
1277	7174	n

lazy.Kstar
-B 4 -M a

s	n	
693	136	s
260	8191	n

Rules.DecisionTable
-X 1 -S 5 -I -R

s	n	
687	142	s
170	8281	n

Tabela 11.32: . Local da Compra - P12

Capítulo 12

Visualização dos Resultados

Após a mineração de dados propriamente dita foi iniciada a análise dos modelos produzidos pelos melhores algoritmos. Logo se percebeu que a quantidade de regras dos modelos era muito grande (mais de 200 para cada tabela em muitos casos). As regras envolviam todos os atributos: regiões censitárias, coordenadas geográficas, dias da semana e horários (não envolvem coordenadas apenas no caso das tabelas que têm as regiões pelas quais passam os caminhos).

Nesse contexto o entendimento das regras e padrões descobertos se tornou muito difícil sem a associação de um mapa contendo as regiões censitárias e sem um interpretador de posições geográficas. Não era possível portanto avaliar se os modelos obtidos, além de serem eficazes na classificação de instâncias, eram também eficazes na descoberta dos padrões dos dados. Não se podia também comparar os modelos entre si, buscando as diferenças que porventura existiam entre eles.

Com o intuito de facilitar a visualização e interpretação das regras produzidas foi escrito um programa que lê os modelos gerados pelo algoritmo *PART* e gera um arquivo visualizável no *Google Earth*. Dessa forma ao invés de identificadores de região e coordenadas geográficas é possível visualizar para cada regra as localidades a que faz referência.

Um esquema geral dos procedimentos realizados pode ser observado na figura do diagrama 12.1. A cor de cada retângulo indica o mesmo que nos diagramas do início do capítulo 10.

As regras do algoritmo *PART* foram escolhidas para isso porque esse foi um algoritmo que obteve bom desempenho nos tipos de tabela a serem visualizados (gerou bons modelos). Ele apresenta limitações na correta identificação das faixas de horário e dias da semana quando cada regra é válida, gerando por exemplo regras que são válidas só durante dias úteis como se fossem válidas também aos sábados e domingos. Mas apesar disso ele gerou regras mais simples de serem interpretadas e bem delimitadas pois quase sempre foram baseadas em regiões censitárias ao invés de simplesmente em faixas de coordenadas como o *Jrip* por exemplo. Regras baseadas em faixas de coordenadas geralmente englobam uma região muito ampla, pouco específica, cobrindo grandes áreas da cidade. Portanto, apesar de suas limitações, foi o algoritmo ideal para se realizar o processo descrito neste capítulo.

O algoritmo *Ridor* não foi escolhido porque em muitos casos as regras por ele produzidas dizem respeito às instâncias cuja classe é diferente da classe que se deseja analisar. Por exemplo, ele faz regras para a classe n ao invés de fazê-las para a classe s. Assim

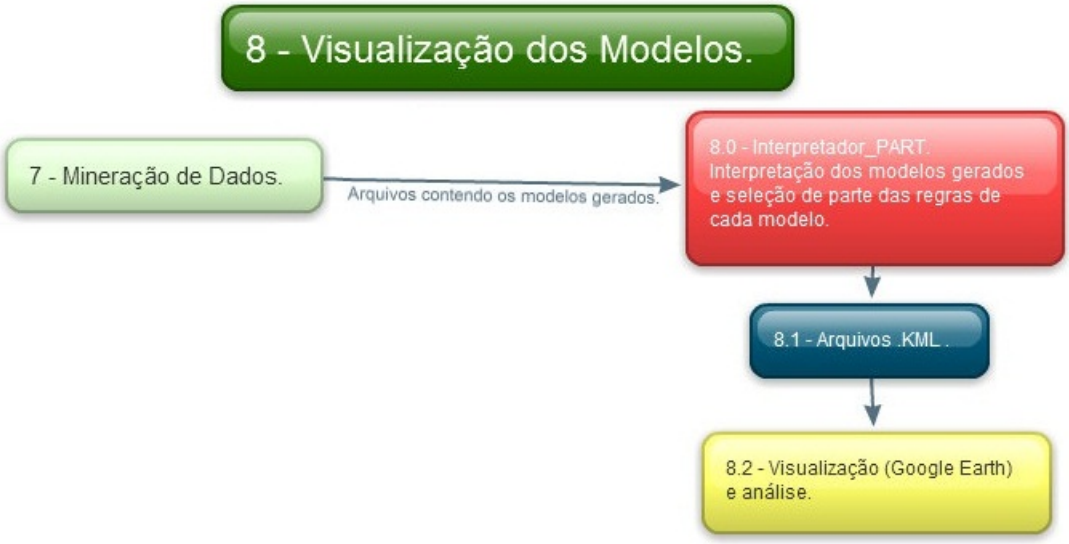

Figura 12.1: Diagrama de Interpretação de Modelos

todos os pontos que se inserem nas regras são n e os demais são s. Em casos como esse fica muito mais difícil visualizar as regiões s. O arquivo .KML produzido teria que ser escrito com regiões complementares às regras do *Ridor*. Como o algoritmo *PART* foi eficiente na maioria das tabelas e também obteve bons resultados foi dada preferência a ele.

Idealmente deveria ter sido escrito um interpretador para cada um dos algoritmos que produziram boas regras (cada algoritmo diferente gera regras em um padrão específico), mas isso apenas será feito em trabalhos futuros. Além do grande período de tempo necessário para desenvolvimento de tais programas seria também necessário muito tempo para a análise dos arquivos .KML produzidos. Assim, tal etapa, neste projeto, foi realizada apenas para o algoritmo citado.

Na produção do arquivo .KML muitas vezes foi feita uma seleção das regras lidas. A visualização de todas de uma só vez dificulta a análise, pois em muitos casos elas envolvem várias áreas da cidade. Se considerada a semana inteira todas as avenidas principais são movimentadas em algum horário e, portanto, haveria regras cobrindo tais avenidas. Isso gera uma imagem poluída que dificulta a análise. Assim as regras escritas no arquivo .KML foram, em muitos casos, selecionadas, ou seja, eram escritas apenas as que tratavam de determinado dia e horário, facilitando a avaliação de padrões característicos de determinado período do dia.

O objetivo da análise das imagens foi avaliar se os modelos identificaram os padrões inseridos nos dados, discutidos na seção 10.6 . A meta foi observar se o algoritmo realmente foi eficaz na descoberta de padrões, ou seja, se realmente produz modelos válidos, de acordo com a realidade dos dados.

As tabelas de regiões de cada caminho não foram analisadas porque não haviam padrões observáveis inseridos no contexto dessas tabelas. Os algoritmos procuraram avaliar que características dos caminhos os levaram a passar pela região de classificação e não foram inseridos padrões nesses caminhos em relação à região censitária na qual passavam. Portanto, não há subsídios para dizer se um determinado modelo acerca dessas tabelas está de acordo com a realidade dos dados. Dessa forma não foram geradas imagens para

esse tipo de tabela.

12.1 Programa de Visualização de Modelos

Como discutido no início deste capítulo, diante da necessidade de se analisar as centenas de regras que compõem cada modelo gerado pelo algoritmo *PART* através da mineração de dados, foi escrito um programa de visualização, o *Interpretador_PART*.

Tal programa, em termos gerais, funciona da seguinte forma. Primeiramente lê um arquivo no formato .GPX contendo todos os pontos que formam os polígonos que representam os setores censitários. Tal arquivo foi obtido pela conversão de um outro arquivo .SHP para o formato .GPX através do programa *SAGA*. O arquivo .SHP foi obtido diretamente do IBGE e contém todas as informações geográficas sobre os setores censitários de Goiânia.

Ao ler o arquivo de pontos dos setores censitários o programa grava tais dados organizadamente em memória: ordena em um vetor, por ordem de identificador de região, as estruturas contendo todos os dados de cada região. Assim, cada posição do vetor criado recebe os dados sobre uma região censitária, inclusive o apontador para a lista de pontos que delimitam essa região.

Em seguida, são abertos arquivos contendo as partes que um arquivo .KML deve ter e copiadas tais partes no arquivo de saída que conterá o modelo *PART* no formato .KML. Os trechos de sintaxe são copiados no arquivo de saída na ordem que devem aparecer, de forma que o arquivo resultante possa ser lido pelo *Google Earth*.

A próxima ação do programa é ler o arquivo .TXT que contém o modelo *PART*. Os modelos gerados por esse algoritmo são compostos por regras que envolvem os atributos das instâncias e as classificam de acordo com os atributos de classificação das mesmas. Um exemplo de regra seria: Hora $>$ 1200.0 *AND* Long $<=$ 251819.0 *AND* Região != 5100022 *AND* Região!= 5090007 *AND* Região = 5100001 *AND* Lat $<=$ 714639.0: *m* (esta regra significa que a área definida pela região 5100001 e por latitudes maiores ou iguais a -16.714639 e longitudes maiores ou iguais a -49.251819, em horários superiores a 12:00 *a.m.*, tem um fluxo de consumidores de determinado produto considerado médio, ": *m*"no final da regra, ou seja, nem grande nem pequeno). A cada regra lida é avaliado se é possível definir um polígono fechado referente à área da regra. Muitas vezes isso não é possível, como no exemplo: Hora $>$ 1200.0 *AND* Long $<=$ 251819.0 *AND* Região != 5100022 *AND* Região != 5090007 *AND* Lat $<=$ 714639.0: *m*. Tal regra faz referência a uma área aberta (Longitudes maiores ou iguais a -49.251819, latitudes maiores ou iguais a -16.714639 e locais fora das regiões 5100022 e 5090007). Uma área aberta não pode ser contida por um polígono, não podendo ser representada de forma prática. Regras assim são descartadas.

A cada regra lida que define uma região fechada, esta região é escrita no arquivo de saída .KML. Para tanto é criado um polígono segundo a sintaxe .KML e são escritos todos os pontos que o limitam. Tais pontos são obtidos através de consultas ao vetor de regiões censitárias, quando a regra faz referência a uma região, ou através do cálculo dos pontos que delimitam as faixas de coordenadas da regra (latitude e longitude) o que gera um quadrilátero.

Há no entanto casos de regras que definem uma região limitada tanto por regiões censitárias quanto por faixas de coordenadas, devendo ser escrito um polígono que per-

tença à região censitária mas que também obedeça às restrições de coordenadas (latitude maior ou igual a x e menor ou igual a y, etc). Nesses casos os pontos que delimitam a região censitária que estão dentro das faixas de coordenadas válidas são considerados e o restante é desconsiderado. Os pontos de intersecção entre as faixas de coordenadas e as regiões censitárias também são inseridos, de forma que o polígono resultante cubra realmente toda a região a que faz referência a regra. Um exemplo visual do caso aqui descrito pode ser observado na figura 12.2. Em azul está a região censitária à qual a regra faz referência e em vermelho a região que efetivamente delimita a área onde a regra é válida. A possibilidade da existência de múltiplas restrições de latitude e longitude foi uma dificuldade que teve que ser superada para que o programa pudesse funcionar a contento. O pseudo-código a seguir representa a estratégia utilizada para se obter as áreas onde as regras realmente são válidas.

Figura 12.2: Regra Relativa à Região 5120011

```
01  Interpreta uma regra e descobre os intervalos de valores de atributos que
02  a definem, inclusive a região censitária a qual faz referência;
03
04  Localiza a lista de pontos da região censitária da regra por busca binária no vetor de
05  regiões;
06
07  Considera o primeiro ponto da lista como "ponto atual"a ser analisado;
08
09  Sinal <- 0;
10  // Variável que sinaliza se o ponto atual está dentro (0) ou fora (1) da área na qual
11  // a regra é válida.
12
13  Enquanto não forem percorridos todos os pontos que formam o polígono da região,
14  faça:
15  {
16        Verifica se o ponto atual da lista está fora da região da regra;
17
```

```
18      Se o ponto estiver dentro da região onde a regra é válida:
19      {
20          Se sinal == 1: // Primeiro ponto depois de sair de região de exclusão
21          {
22              Calcula coeficientes da reta que passa pelos pontos atual e anterior;
23
24              Calcula as coordenadas do ponto da reta que estiver entre o ponto atual
25              e o anterior e estiver sobre a fronteira entre a área da regra e a área
26              fora da mesma;
27
28              Insere o ponto de fronteira encontrado na lista de pontos que delimita
29              a área onde regra é válida;
30          }
31
32          Sinal <- 0; // Sinaliza que está dentro da área da regra.
33
34          Insere ponto atual na lista de pontos da região onde a regra é válida;
35      }
36      Senão, se o ponto atual estiver fora da região da regra, a lista de pontos da regra
37      for não vazia e Sinal == 0 (primeiro ponto fora da área da regra):
38      {
39          Sinal <- 1; // Sinaliza que está fora da área da regra.
40          Calcula os coeficientes da reta que passa pelo ponto atual e o anterior;
41
42          Encontra ponto da reta que estiver sobre a fronteira entre a área fora e
43          dentro da regra;
44
45          Adiciona ponto encontrado à lista de pontos que definem a região da regra;
46      }
47      Senão, além do ponto atual estar fora da regra, a lista de pontos da regra está
48      vazia (caso em que o ponto é o primeiro analisado) ou o ponto atual não é o
49      primeiro fora da regra.
50      { Sinal <- 1; // Sinaliza que está fora da área da regra.}
51
52      // Abaixo é tratado o caso em que há uma faixa de latitude e outra de longitude
53      // a partir das quais a regra não é válida e há um ponto que delimita o polígono
54      // dentro da área de dupla exclusão.
55
56      Se o ponto estiver fora da regra devido ao fato de ambas as suas coordenadas
57      estarem além das limitações de latitude e longitude:
58      {
59          Calcula coordenadas do ponto de encontro entre as duas regiões de exclusão
60          da regra;
61
62          Adiciona ponto obtido à lista de pontos que delimitam a região de validade
63          da regra;
```

```
64          }
65
66          Atualiza o ponto atual para o próximo ponto da lista que delimita a região
67          censitária que a regra referencia;
68  }
```

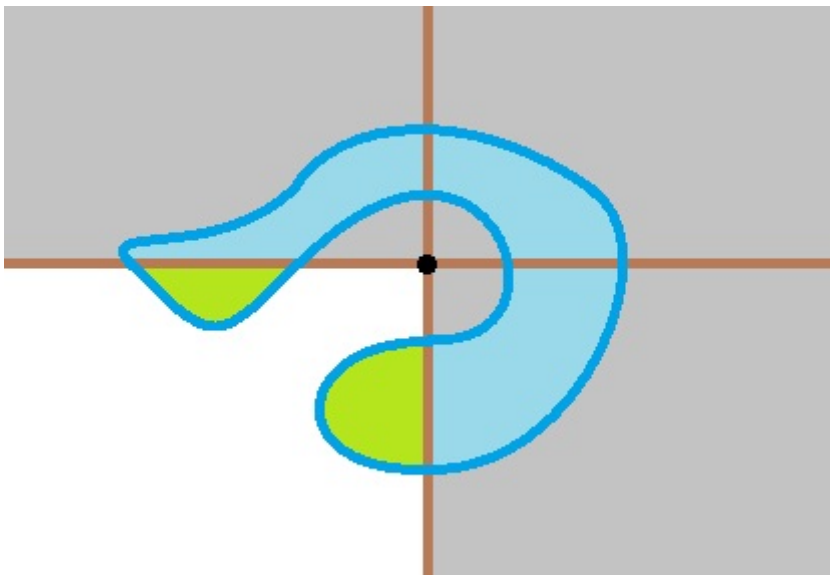

Figura 12.3: Caso de Erro da Heurística

Deve-se observar que a estratégia acima não funcionaria no caso representado pela figura 12.3 (a região censitária representada em azul e verde com borda azul e as regiões de exclusão em cinza, sendo a área verde equivalente à área de validade da regra). Nesse caso seriam gerados resultados inverídicos, pois o ponto preto observado na figura mencionada seria inserido na lista de pontos que definem a região de validade da regra (linhas 56 a 64). No entando, a situação observada na figura citada não ocorreu no contexto deste projeto devido à regularidade das áreas de cada região censitária. Possivelmente, outras situações também gerariam erros, como por exemplo a ausência de pontos (que definem a área de uma região) localizados em uma região de exclusão de uma regra, o que resultaria na inobservância de uma região onde a regra não é válida. No entanto, devido à natureza das regiões utilizadas neste trabalho (compostas por vários pontos, correspondentes a cada mudança sutil na curvatura das ruas) e devido às características peculiares das regras geradas pelo algoritmo *PART* (que apresentam limitações de latitude e longitude de acordo com um determinado padrão), tais situações não foram observadas no decorrer deste projeto. A heurística apresentada acima foi implementada no programa discutido nesta seção e, durante a conferência das áreas resultantes, não foi observada ocorrência de erro de interpretação de regra alguma. Assim, a utilização da mesma foi válida e adequada às necessidades do trabalho.

Escritos os polígonos referentes à área de cada regra, o nome de tais polígonos foi gerado pelo programa escrevendo-se a faixa de horário e dias da semana nos quais as regras em questão são válidas. Foi escrita também a quantidade de instâncias classificadas através da regra.

Vale também ressaltar que os polígonos efetivamente escritos no arquivo .KML são selecionados de acordo com as condições de consulta às regras, definidas no programa para cada modelo a ser processado. Por exemplo, se para o modelo atualmente analisado são requisitadas apenas regras válidas exclusivamente nos fins de semana, somente os polígonos gerados para essas regras são escritos no arquivo de saída. Cada modelo lido pelo programa tem suas regras selecionadas de acordo com parâmetros definidos com o objetivo de ressaltar determinado padrão nos dados que geraram aquele modelo.

Dessa forma, utilizando-se o programa aqui descrito, foram gerados diversos arquivos .KML referentes aos modelos produzidos pelo *PART* para os dados deste trabalho. Cada arquivo foi gerado com a seleção das regras mais adequadas (em termos de dias e horários nos quais eram válidas) para a busca dos padrões que se desejava verificar em cada caso.

Os arquivos .KML gerados foram visualizados através do programa *Google Earth* e foram geradas figuras do que se podia observar no programa. Nas figuras, como na figura 12.4, cada polígono translúcido representa uma regra diferente. Suas bordas foram retiradas para que fosse facilitada a visualização e análise das áreas cobertas pelas regras. Em praticamente todas as figuras analisadas neste trabalho observa-se áreas menos e mais transparentes. As áreas menos transparentes, de cores mais intensas, são resultado da cobertura da mesma área por mais de uma regra (polígono). Isso sugere que as áreas de cor mais intensa são de maior importância, ou seja, são áreas de maior interesse para o posicionamento de publicidade. Seja pelo fato de possivelmente reunirem mais consumidores (se forem englobadas por várias regras referentes aos mesmos dias da semana e horários), seja pelo fato de reunirem grande quantidade de consumidores durante um período mais prolongado (se forem englobadas por várias regras referentes a diversos horários e dias da semana).

Algumas dentre as diversas figuras geradas são apresentadas e discutidas nas próximas seções.

12.2 Tabelas de Trânsito nos Pontos do Reticulado

Foram analisados alguns modelos produzidos pelo algoritmo *PART* para as tabelas de trânsito nos pontos do reticulado selecionadas para a mineração. As tabelas em questão foram geradas considerando-se toda a rotina semanal das pessoas, sem restrições.

No caso da tabela referente ao produto 1 (livros) foi gerado um arquivo .KML envolvendo todas as regras produzidas pelo algoritmo, figura 12.4, um polígono por regra, conforme explicado no fim da seção 12.1. Nota-se que várias são as localidades de grande movimento durante o período da semana inteira. Se vistas todas ao mesmo tempo as regras não fornecem informações úteis, além do fato já conhecido de que as principais vias da cidade são as mais movimentadas (e também do fato discutido no fim da seção 12.1 sobre as áreas de coloração mais intensa). O mesmo foi feito para a tabela dos consumidores do produto 2 (câmeras) gerando semelhantes resultados (figura 12.5).

A princípio esperava-se que as regras obtidas, se visualizadas todas juntas, gerariam o mesmo efeito de cobertura das principais vias da cidade independentemente do público consumidor analisado. No entanto a visualização simultânea das regras dos dois modelos citados acima (figura 12.6) sugere que isto não é verdade. Mesmo para produtos consumidos com probabilidades parecidas pelas mesmas classes econômicas, como é o caso dos

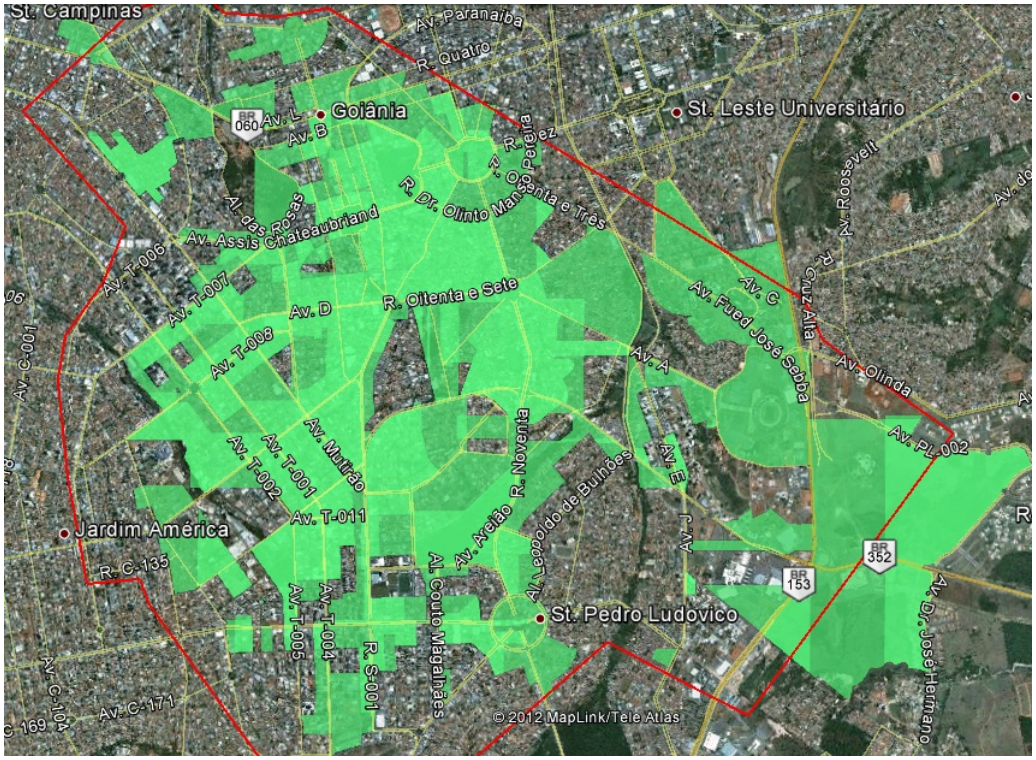

Figura 12.4: Trânsito pelo Reticulado Produto 1

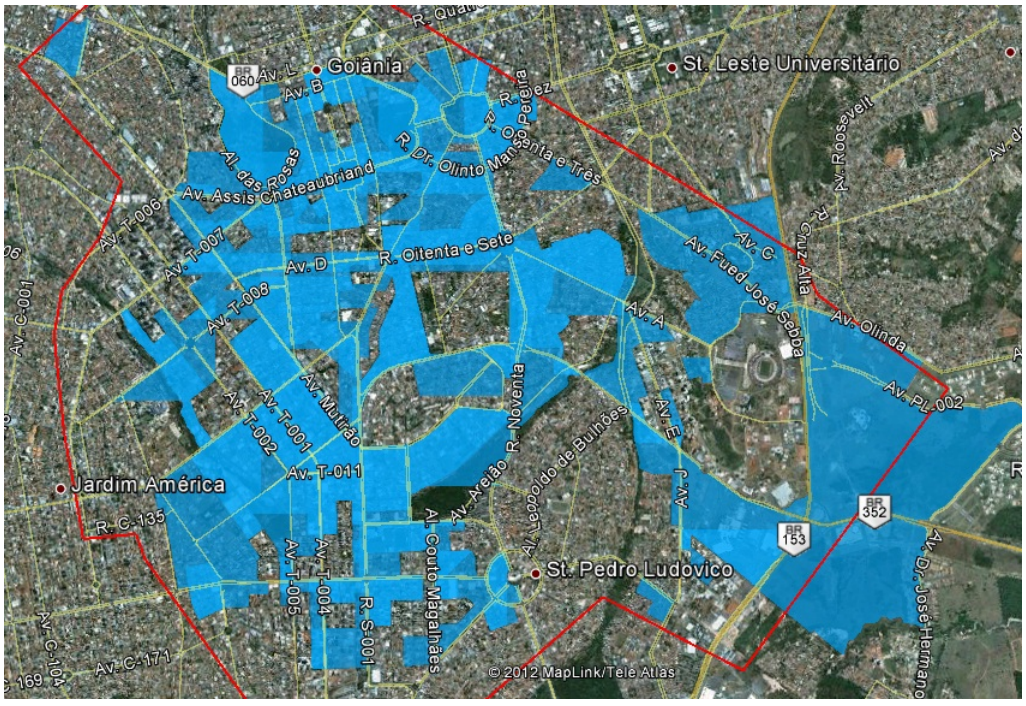

Figura 12.5: Trânsito pelo Reticulado Produto 2

produtos 1 e 2, os padrões de circulação pela cidade de cada público difere, ainda que coincida em muitas regiões.

É possível que o mesmo ocorra no caso da análise de dados reais. Assim nota-se que é

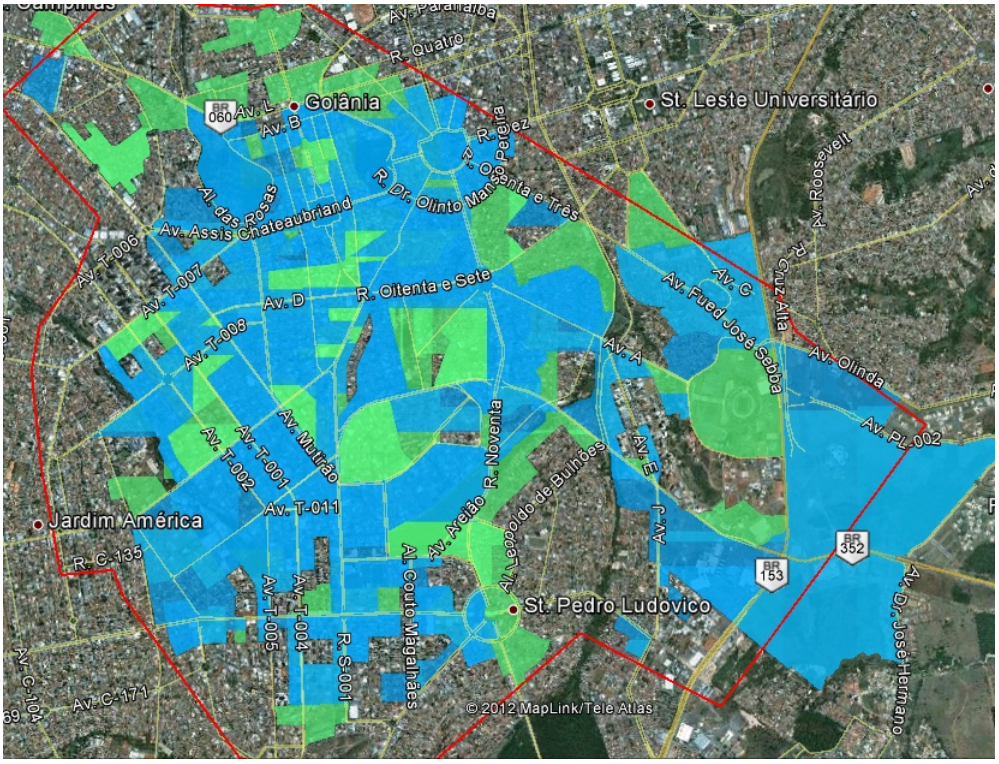

Figura 12.6: Trânsito pelo Reticulado Produtos 1 e 2

importante a separação dos dados de acordo com os produtos consumidos por cada pessoa pois os padrões de circulação de consumidores de cada produto provavelmente difere. Mesmo considerando-se as principais vias da cidade o comportamento de consumidores de diferentes produtos provavelmente gera padrões de deslocamento diferentes. Assim, algumas vias principais seriam melhores que outras para o posicionamento de publicidade de acordo com o público consumidor que se deseja atingir.

Outro padrão importante notado nesses e em todos os outros modelos gerados para as outras tabelas é que o movimento se concentra nas áreas mais centrais da cidade e também nas áreas próximas ao cruzamento das BR352 e BR153. Quanto mais afastada é a área da cidade desses pontos, menos movimento ocorre na região e, portanto, menos interessante é a região no contexto deste trabalho. O perímetro da cidade foi escolhido com esse fato em mente, de forma a conter as áreas de maior densidade populacional e comercial e pelas quais passam as avenidas mais movimentadas da cidade.

O próximo modelo visualizado foi o referente aos consumidores do produto 3 (celulares). Para efeito de melhor visualização e análise foram selecionadas apenas as regras que diziam respeito exclusivamente ao movimento em sábados e domingos. Assim a imagem obtida, figura 12.7, se refere ao deslocamento de pessoas de ambas as classes econômicas durante o fim de semana. Os marcadores amarelos são os locais dos restaurantes frequentados por pessoas das classes A e também dos frequentados por pessoas de ambas as classes. Os pretos se referem a restaurantes do Setor Marista. O verde se refere ao *Shopping Flamboyant*. Tais marcadores condizem com os pontos de interesse mencionados na seção 9.1.1.

É possível observar na figura que o fluxo de pessoas é alto na maioria das áreas de

restaurantes, sendo particularmente alto nas vias de acesso aos restaurantes do Setor Marista. Isso condiz com o padrão (descrito na seção 10.6) inserido nos dados referente ao fato de que pessoas da classe A frequentam restaurantes do Setor Marista aos fins de semana com chance de 76 %. Também é possível identificar grande movimento na região do *Shopping Flamboyant* e vias de acesso. Isso remete a outro padrão inserido nos dados que institui chance de 76 % de pessoas da classe B frequentarem o *Shopping Flamboyant* como fonte de lazer nos fins de semana. Nota-se portanto que o movimento nas vias da cidade durante o fim de semana, indicado pelo modelo obtido, condiz com os padrões inseridos nos dados. Portanto, apesar da dificuldade de se avaliar um modelo através de sua visualização (dificuldade que é consideravelmente maior se a análise for feita sobre as classificações expressas objetivamente no modelo em formato textual, conforme discutido no início deste capítulo), o modelo obtido foi de boa qualidade.

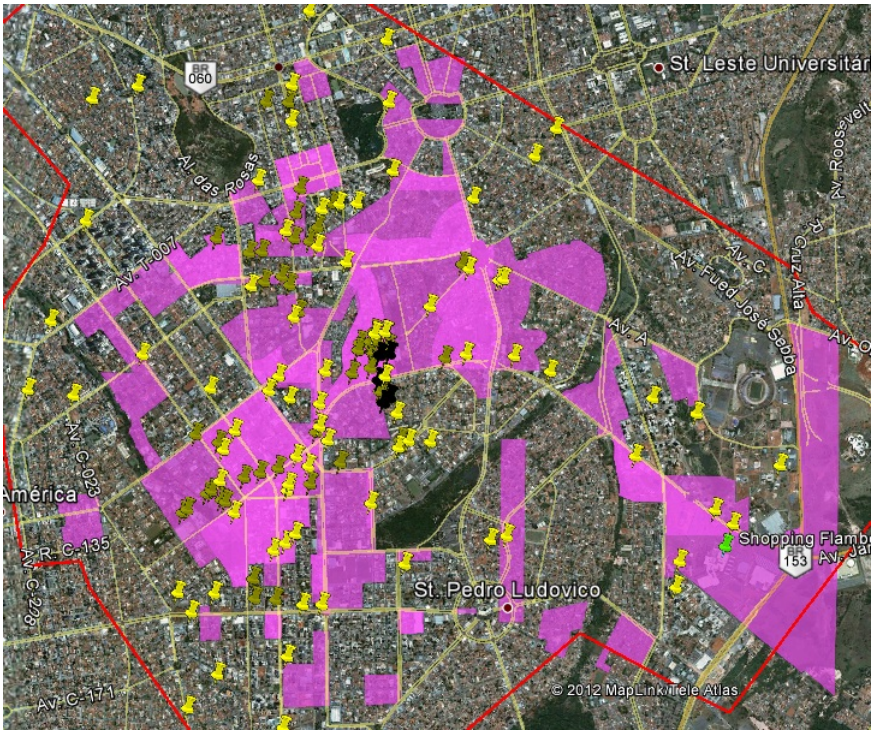

Figura 12.7: Trânsito pelo Reticulado Produto 3

Vale lembrar que nem todos os restaurantes da cidade foram apontados pelo modelo citado no parágrafo anterior. Isso porque nem todas as pessoas frequentam restaurantes no fim de semana. Frequentar um restaurante está associado a uma probabilidade inserida no programa que gerou a rotina de movimentação das pessoas (seção 9.3.1). Além disso nem todas as pessoas são compradoras do produto 3. Portanto, é possível que desconsiderar as pessoas que não compram tal produto seja outro fator que gerou a ausência de movimento nas proximidades desses estabelecimentos.

O próximo padrão, anteriormente inserido nos dados, que foi identificado pelos modelos construídos foi o de pessoas da classe A terem chance pequena de voltar para suas residências no horário de almoço em dias de trabalho, almoçando em locais próximos do trabalho, enquanto pessoas de classe B geralmente voltam para almoçar em casa. Para

identificá-lo foram gerados arquivos .KML selecionando-se regras que dizem respeito exclusivamente a dias úteis nos horários entre 11:00 a.m. e 16:00 p.m.

Um dos arquivos .KML (figura 12.8) foi gerado através da leitura das regras produzidas para a tabela referente aos consumidores do produto 7 (cupons de compra coletiva). Isto foi feito para esse produto porque ele é consumido exclusivamente por pessoas da classe B. Assim, as regras obtidas e o movimento retratado diz respeito exclusivamente ao movimento de pessoas da classe B.

Outro arquivo .KML (figura 12.9) foi gerado através do modelo referente aos consumidores do produto 8 (cafeteiras). É um produto consumido somente por pessoas da classe A. Portanto, o movimento retratado diz respeito unicamente a pessoas da classe A.

Nos dois arquivos .KML os polígonos sem preenchimento de borda vermelha indicam as regiões censitárias para as quais foi gerada a população.

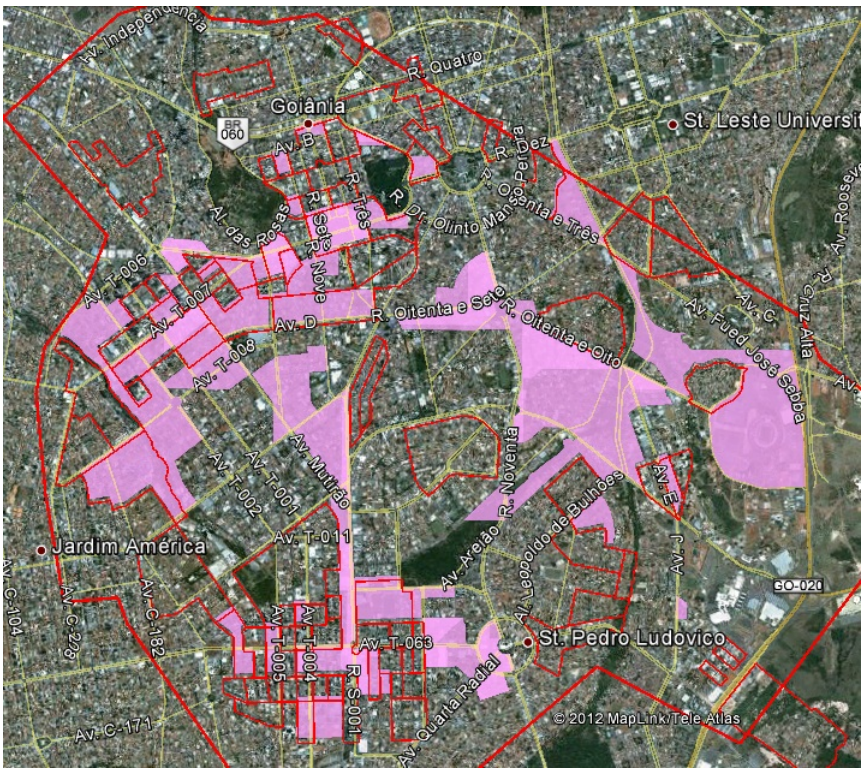

Figura 12.8: Trânsito pelo Reticulado Produto 7

Nota-se que o arquivo referente a pessoas da classe B, figura 12.8, indica movimento em regiões menos centrais, coincidentes com vias de acesso às áreas residenciais. Tal fato sugere que o padrão referente a essa classe foi identificado pela mineração de dados pois demonstra grande circulação da população em áreas residenciais e circulação bem menor nas áreas centrais, que concentram a maioria dos locais de trabalho. A volta para casa em horário de almoço por parte da população de classe B foi portanto apontada pelo modelo gerado.

Já no caso do arquivo referente a pessoas de classe A, figura 12.9, o movimento ocorre em vias mais centrais da cidade, em regiões onde se concentram a maioria dos locais de trabalho das pessoas. Lembrando que o almoço ocorre em restaurantes próximos aos locais de trabalho, tal fato sugere que a população de classe A raramente volta para almoçar em

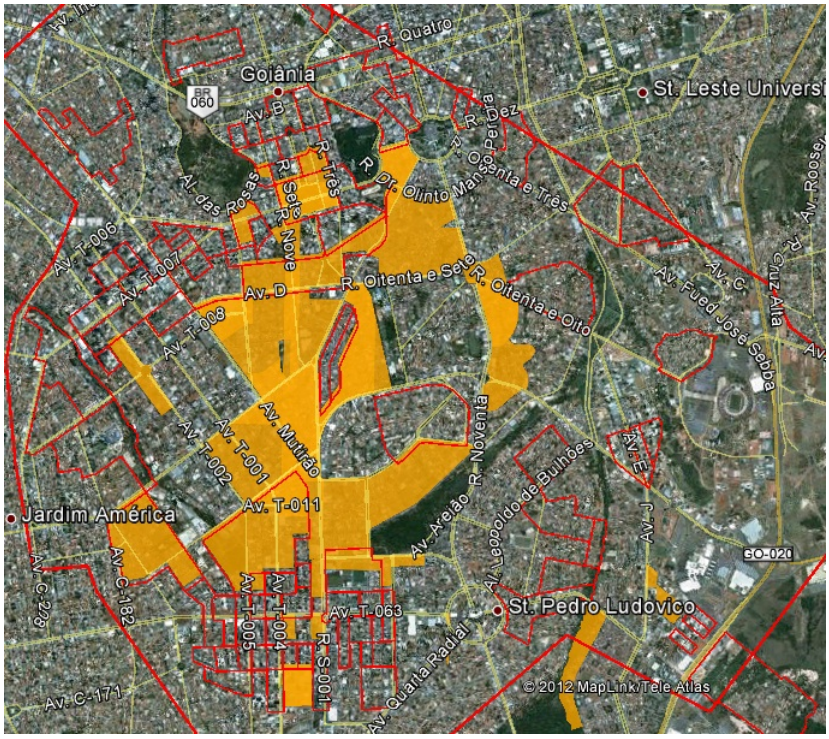

Figura 12.9: Trânsito pelo Reticulado Produto 8

casa durante a semana. Isso condiz com o padrão de deslocamento inserido nos dados, mencionado anteriormente (seção 10.6.3). A tendência de não voltar do trabalho para casa em horário de almoço por parte de pessoas da classe A foi portanto identificada pelo algoritmo.

12.3 Tabelas de Movimento Anterior ao Ato da Compra

As tabelas nas quais foi realizada mineração de dados tratadas nesta seção dizem respeito ao trânsito nos pontos do reticulado realizado imediatamente antes da compra do produto ao qual cada uma faz referência.

O primeiro arquivo .KML gerado foi obtido pela leitura do modelo referente aos consumidores do produto 1 (Livros). Foram selecionadas apenas as regras que diziam respeito exclusivamente ao movimento nos sábados e domingos. Segundo padrões discutidos na seção 10.6, livros podem ser comprados no sábado a tarde ou no domingo apenas por pessoas da classe A. Portanto, a seleção em questão resultou em regras relativas unicamente ao movimento de pessoas da classe A em fins de semana, imediatamente antes da compra do produto 1.

A figura 12.10 retrata o resultado da visualização do referido arquivo .KML. Os marcadores vermelhos indicam os restaurantes do Setor Marista. Os polígonos sem preenchimento de bordas vermelhas são as regiões de residência da população.

Segundo o padrão de deslocamento em fins de semana de pessoas da classe A, que institui que 76 % dos restaurantes frequentados estão no Setor Marista, esperava-se obter

grande movimento na região em questão e nas vias de acesso a tal localidade. Observando-se a figura 12.10 conclui-se que foi identificado certo movimento na região e nas vias próximas. Foi também identificado movimento em algumas das principais vias da cidade.

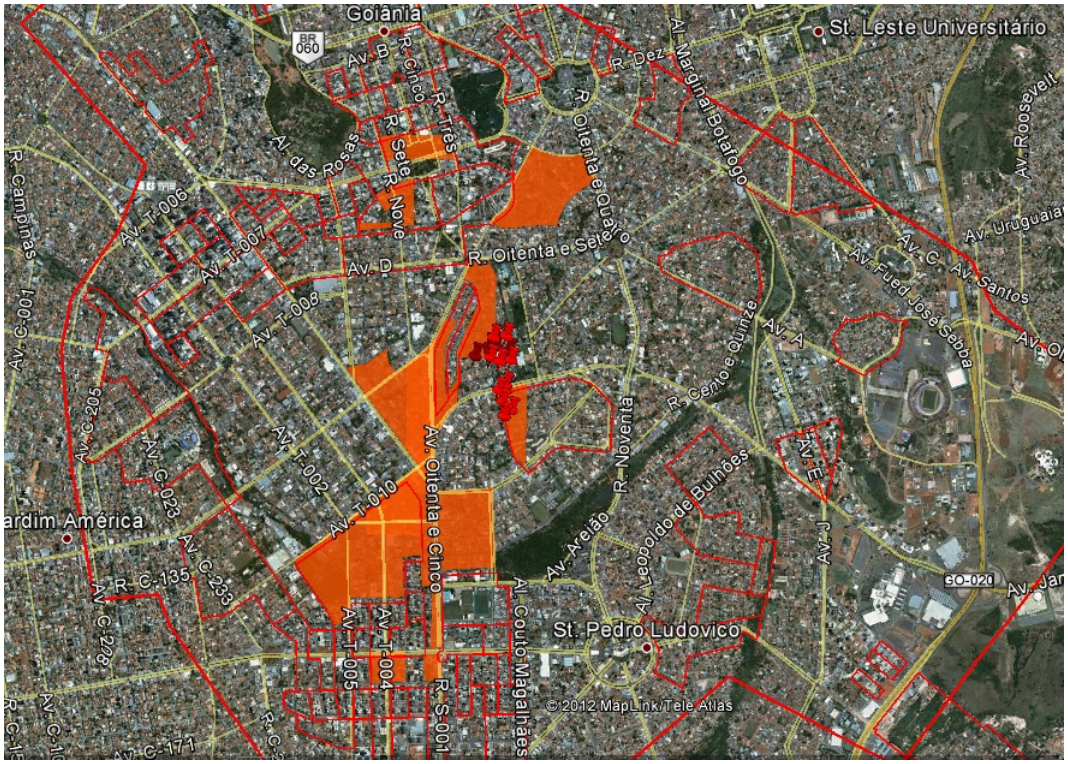

Figura 12.10: Caminhos Antes da Compra Produto 1

No entanto o movimento identificado parece ter sido restrito e aquém das expectativas. Possivelmente isso se deve ao fato de que as pessoas da classe A também realizam compras de livros durante a semana. Portanto, a quantidade de compras no fim de semana se torna menor. Além disso, nem todas as pessoas compram livros. A tabela de probabilidades (tabela 10.7) de compra desse produto por pessoas de cada classe prevê chance de 30 % de pessoas da classe A não os terem comprado. Assim, a movimentação a ser identificada com a seleção de regras descrita torna-se pequena. Mesmo assim o padrão em questão foi identificado.

Outro arquivo .KML foi gerado através da leitura das regras produzidas para a tabela de consumidores do produto 8 (Cafeteiras). Foram lidas apenas as regras que diziam respeito aos sábados e domingos unicamente. Segundo padrões discutidos anteriormente (seção 10.6), consumidores do produto 8 são exclusivamente da classe A. Assim as áreas azuis observadas na figura 12.11 referente a esse arquivo mostram exclusivamente o movimento anterior ao ato da compra de todas as pessoas da classe A que compraram cafeteiras nos fins de semana.

Segundo os mesmos padrões de probabilidade (também apresentados na seção 10.6) o consumo de tal produto ocorre em grande quantidade. A chance de uma pessoa da classe A comprar pelo menos 2 produtos do tipo é de 100 %. As áreas destacadas na figura são portanto relativas a grande parte da população de classe A pois apesar de as pessoas também comprarem cafeteiras na segunda, terça e quarta feiras, grande parte as compra

no fim de semana. Isso justifica a grande quantidade de áreas movimentadas no período considerado. Quantidade essa muito maior que no caso da tabela anterior referente aos mesmos dias da semana e ao mesmo público.

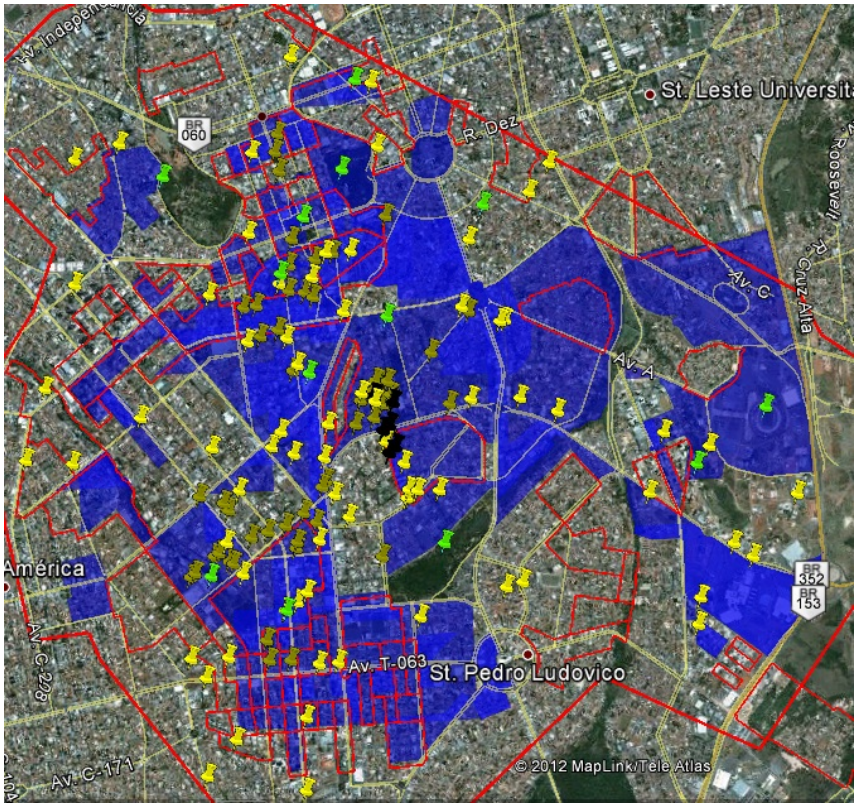

Figura 12.11: Caminhos Antes da Compra Produto 8

Sobre a figura 12.11, os marcadores amarelos representam todos os restaurantes frequentados por pessoas da classe A. Os marcadores verdes representam locais de lazer na cidade. Os pretos indicam o local dos restaurantes do Setor Marista. Tais marcadores foram obtidos como descrito na seção 9.3.1. Os polígonos de bordas vermelhas indicam as regiões censitárias onde mora a população gerada.

Observa-se que praticamente todas as regiões que contém os restaurantes mencionados apresentaram grande movimento, inclusive as regiões ao redor dos restaurantes do Setor Marista (local com maior chance de visitação nos fins de semana por pessoas de classe A, como discutido anteriormente nesta seção). A maioria das áreas de lazer também apresenta movimento, assim como as regiões mais densamente habitadas da cidade.

Segundo os padrões obedecidos durante a criação dos hábitos de compra da população, as compras efetuadas nos fins de semana se processam após a primeira rota executada do dia (no caso do sábado, após o trabalho, caso a pessoa trabalhe nesse dia). Portanto, no caso retratado pela figura 12.11 era esperado movimento correspondente às rotas entre as residências das pessoas e seus locais costumeiros de destino no fim de semana: restaurantes, locais de lazer e, no caso de pessoas da classe A, principalmente restaurantes do Setor Marista. Através da observação da figura tal expectativa foi confirmada, o que leva a crer que o modelo retratado cobriu corretamente as áreas da cidade que deveria, identificando assim os padrões dos dados.

O próximo arquivo .KML foi escrito através da leitura do modelo criado para os dados de movimento dos consumidores do produto 9 (televisões). Foram selecionadas regras relativas apenas aos sábados e domingos com o objetivo de analisar o movimento dos consumidores exclusivamente durante os fins de semana. Com isso, de acordo com os padrões de compra do produto em questão, selecionou-se o movimento anterior ao ato das compras realizadas no fim de semana por pessoas de ambas as classes sociais.

Na figura obtida, figura 12.12, os marcadores pretos indicam os locais dos restaurantes do Setor Marista e o marcador verde indica o local do *Shopping Flamboyant*. Observa-se movimento considerável nessas duas regiões e nas vias que dão acesso a elas. Pode-se supor que grande parte das compras do produto 9 se dão após realizadas rotas que passam por essas regiões. Isto está de acordo com o deslocamento esperado para esse público em fins de semana, tanto em relação a pessoas da classe A e restaurantes frequentados quanto em relação a pessoas da classe B e o *Shopping Flamboyant* (frequentado com chance de 76 % por toda pessoa da classe B que sai a lazer em fins de semana).

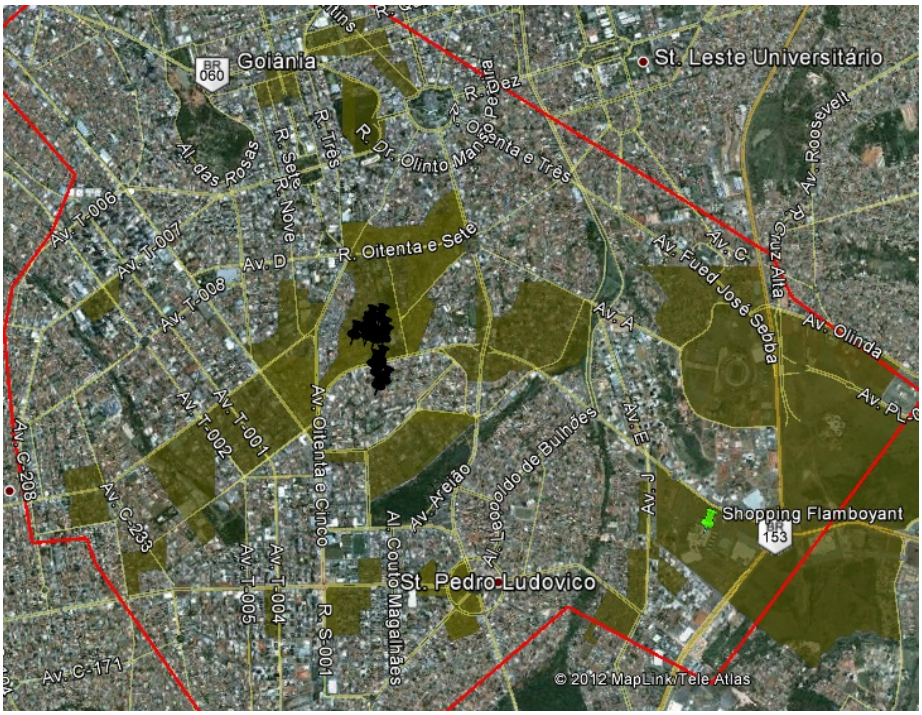

Figura 12.12: Caminhos Antes da Compra Produto 9

Nota-se ainda a diferença em relação ao modelo do produto 8 (figura 12.11). O fluxo de pessoas pela cidade no caso do produto 9 é bem menor, o que se justifica pelos mesmos motivos mencionados a respeito do produto 1. Nem todas as pessoas compram televisões (de acordo com as regras de probabilidade de compra do produto) e dentre as que compram, muitas o fazem durante a semana. Assim a quantidade de pessoas que fazem compras no fim de semana se torna menor, resultando na diferença aqui discutida.

É também importante mencionar a diferença entre os modelos referentes ao produto 8 (figura 12.11) e 1 (figura 12.10). O produto 8 é consumido por ambas as classes no mesmo período que o produto 1. O resultado disso é um movimento maior, por mais áreas e por locais como o *Shopping Flamboyant*, muito frequentados por pessoas de classe

B. Nota-se também um movimento mais descentralizado, resultante da ampliação das origens das rotas que agora englobam mais locais de residência das pessoas que antes, quando englobavam apenas locais de residência de pessoas de classe A.

As próximas análises foram feitas em relação à identificação dos locais de trabalho das pessoas da classe econômica A. Tais locais são onde ocorre a compra dos produtos mencionados a seguir, nos dias consultados.

Foi escrito um arquivo .KML, retratado na figura 12.13, que diz respeito ao modelo gerado pela análise da tabela de consumidores do produto 10 (fogões). Na figura são retratadas regras que são válidas nas segundas-feiras. Analisando-se os padrões referentes ao consumo do produto 10 é possível notar que só quem compra tal produto durante a semana (nas segundas, terças e quartas) são pessoas da classe econômica A, concretizando a aquisição após a chegada ao trabalho no período da manhã. Portanto, o movimento retratado na figura mencionada refere-se exclusivamente a pessoas da classe A, em segundas feiras e nos seus caminhos para o trabalho (são considerados apenas caminhos imediatamente anteriores às compras).

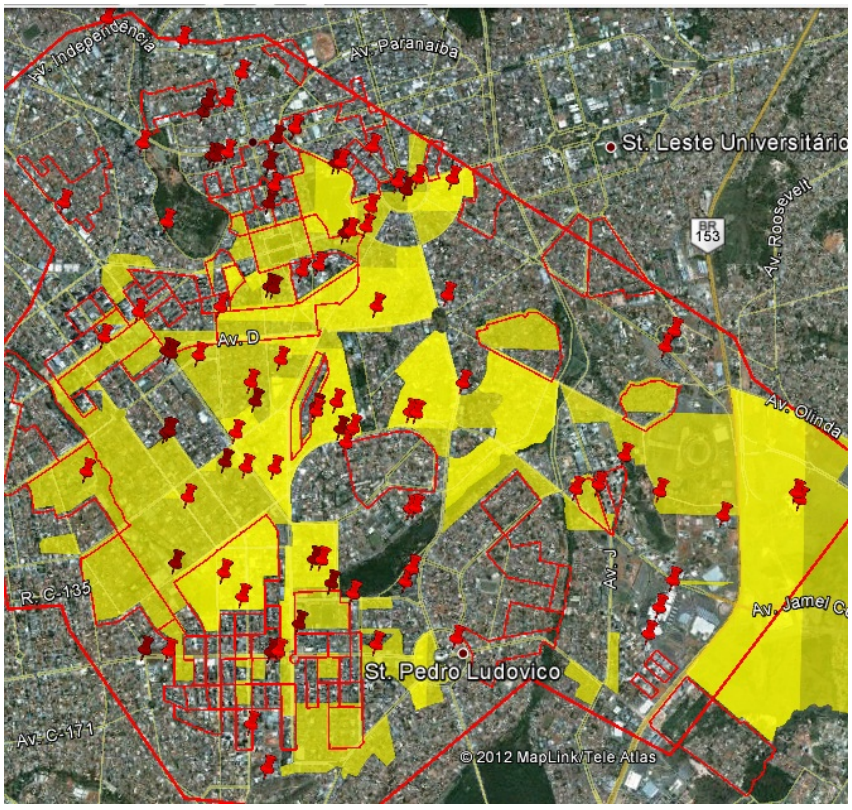

Figura 12.13: Caminhos Antes da Compra Produto 10

Outro arquivo .KML foi gerado em seguida, desta vez pela leitura das regras resultantes da mineração dos dados da tabela referente ao produto 11 (Geladeiras). Assim como no caso do arquivo descrito no parágrafo anterior foram consideradas apenas regras válidas nas segundas feiras. O produto 11 também é comprado nas segundas, terças e quartas apenas por pessoas da Classe A e logo após a chegada ao trabalho na parte da manhã. Assim os movimentos retratados na figura resultante 12.14 são referentes apenas a esse

público, no dia da semana mencionado e gerados por caminhos das residências dessas pessoas aos seus locais de trabalho, do mesmo modo que na figura 12.13.

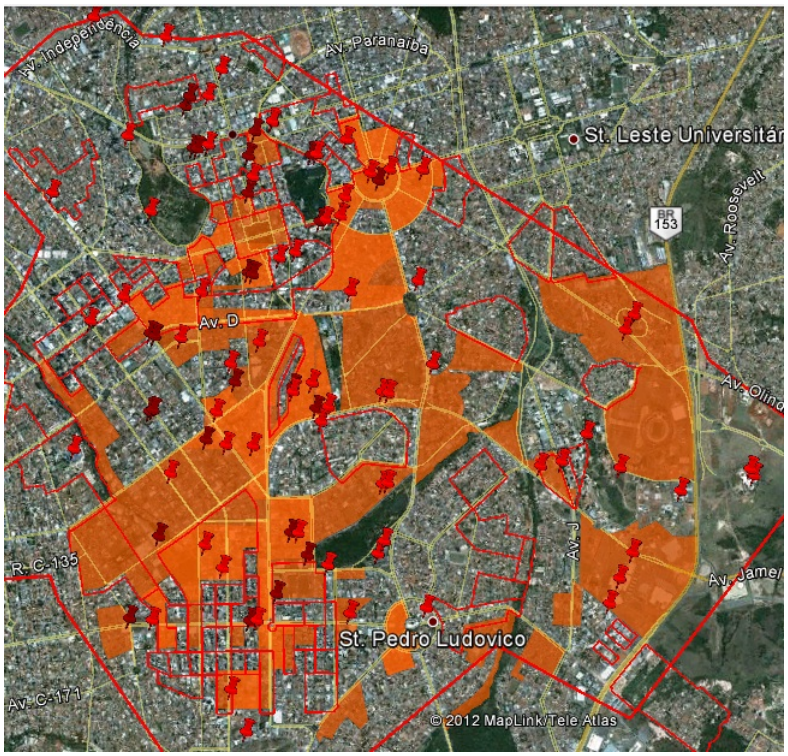

Figura 12.14: Caminhos Antes da Compra Produto 11

Em ambas as imagens mencionadas os marcadores (obtidos como descrito em 9.1.1) representam locais de trabalho de pessoas da classe A e os polígonos de borda vermelha delimitam as áreas dos setores censitários onde se localizam as residências da população gerada.

Nota-se que nos dois casos a área da cidade considerada movimentada é relativamente ampla. Só não é mais ampla, envolvendo todos os locais de trabalho e mais áreas da cidade, porque também nos dois casos a probabilidade de uma pessoa não comprar o produto (fogão ou geladeira) varia entre 55 e 60 %. Portanto, muitas pessoas não consomem tais produtos, o que resulta em algumas áreas não cobertas, correspondentes aos locais frequentados por pessoas que não realizaram compras. Apesar disso a maioria dos locais de trabalho de pessoas da classe A foram cobertos pelas regras nos dois casos, o que coincide com o esperado para segundas feiras no contexto dos dados das tabelas em questão. Isso indica que os modelos conseguiram identificar corretamente mais essa tendência dos dados nesse dia da semana e horário.

Devido a grande semelhança dos padrões obedecidos ao serem gerados os dados referentes aos produtos 10 e 11, esperava-se obter modelos quase idênticos. Apesar de as regras para os dois produtos terem identificado os mesmos padrões de deslocamento é possível perceber observando-se a figura 12.15 que existem algumas diferenças.

Algumas regras de um modelo fazem referência a áreas que contém locais de trabalho não referenciados pelas regras do outro modelo. Isso demonstra mais uma vez que existem diferenças entre os padrões de deslocamento de consumidores de produtos di-

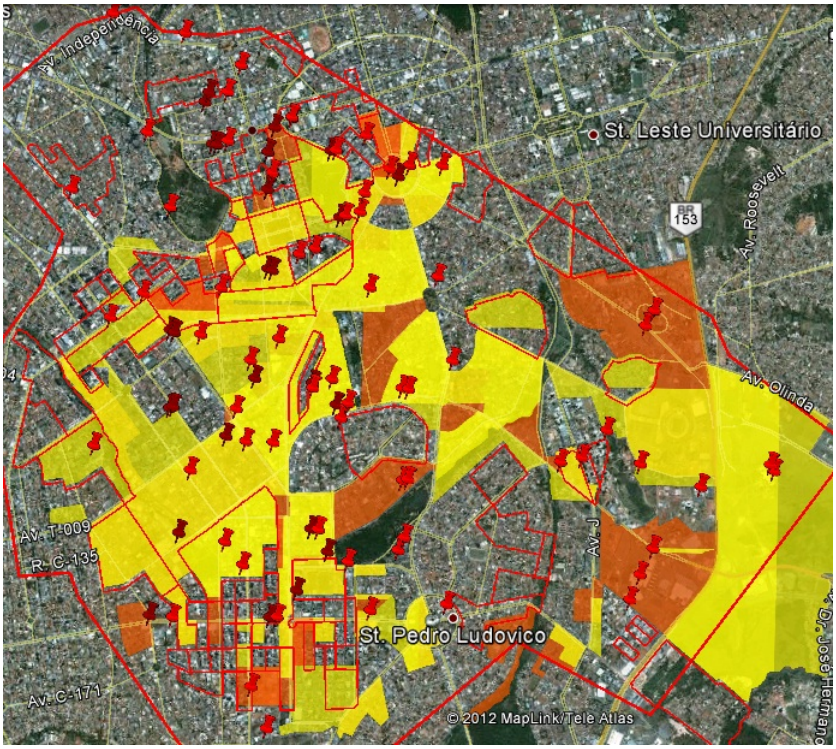

Figura 12.15: Caminhos Antes da Compra Produtos 11 e 10

ferentes mesmo quando tais produtos têm público consumidor pertencente a realidades semelhantes (mesmo padrão de consumo, mesmas regiões de residência e mesmo conjunto de possíveis locais de trabalho). Possivelmente isso também ocorrerá ao se analisar dados reais. Tal fato aponta que a divisão dos dados relativos a consumidores de diferentes tipos de produtos se justifica pois seus padrões de deslocamento possivelmente têm diferenças. Tais diferenças são de grande valia para o correto posicionamento de publicidade no ambiente urbano.

12.4 Tabelas de Locais das Compras

Os dados das tabelas referidas nesta seção são relativos aos locais onde foram realizadas as compras de cada produto por cada pessoa. Os caminhos ou rotas percorridos pelas pessoas não foram considerados.

O primeiro modelo utilizado na criação de um arquivo .KML foi gerado através da tabela de consumidores do produto 1 (livros). Para a geração de tal arquivo .KML foram consideradas todas as regras do modelo, independente do dia da semana ou horário em que eram válidas. Foram obtidas assim todas as regiões onde ocorreram compras de livros na cidade realizadas por pessoas de ambas as classes econômicas (figura 12.16). Os marcadores vermelhos (obtidos como descrito em 9.1.1) dizem respeito aos locais de trabalhos de pessoas das classes A e B. As áreas de cor mais intensa têm sua relevância considerada conforme o discutido no final da seção 12.1.

O local de compra de um livro foi definido pelos padrões seguidos durante a geração dos dados (discutidos na seção 10.6) de forma que nas segunda, terça e quarta feiras eles

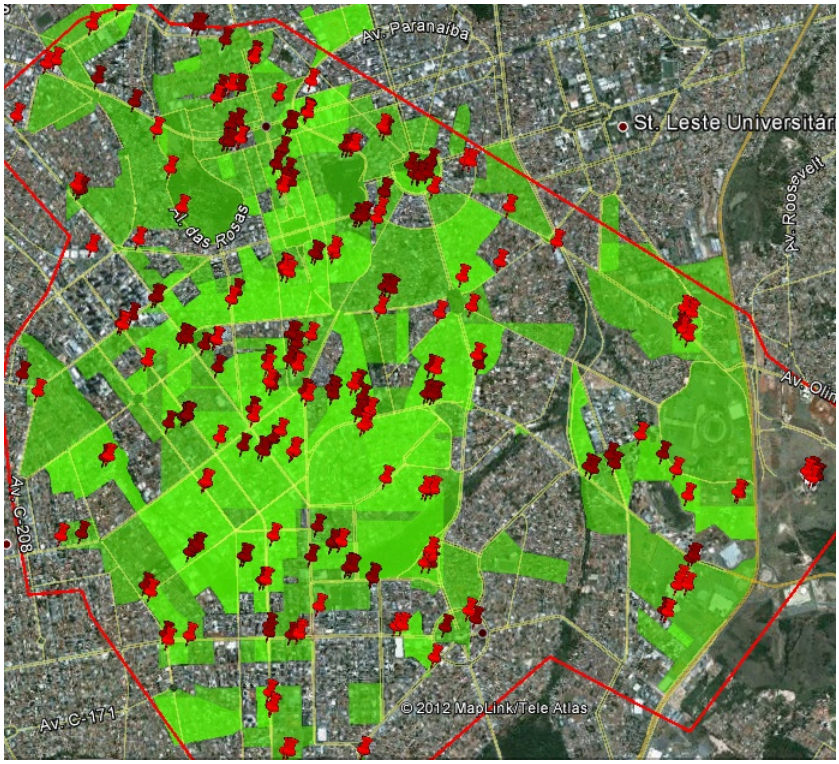

Figura 12.16: Local da Compra Produto 1

fossem comprados no último ponto do primeiro caminho do dia realizado pelo comprador, na parte da manhã. Seriam portanto comprados no trabalho. Para pessoas da classe A foi também facultada a compra nos fins de semana. Como a classe B só poderia consumir nos dias úteis a maioria das compras se concentrou nesses dias, que foram justamente os escolhidos para representação gráfica.

Dessa forma, se esperava obter através da seleção de regras realizada regiões que cobrissem a maioria dos locais de trabalho de pessoas das classes A e B. Foi o que de fato ocorreu (figura 12.16). Praticamente todas as regiões de trabalho de pessoas de ambas as classes foram cobertas pelas regras. O modelo produzido conseguiu mais uma vez identificar o padrão que os dados seguem.

O próximo documento .KML gerado representa o modelo obtido através da mineração dos dados da tabela referente ao produto 2 (câmeras). Foram consideradas todas as regras produzidas. Com isso foi considerado o movimento das duas classes econômicas tratadas neste trabalho. A figura produzida, 12.17, contém marcadores amarelos que representam tanto os restaurantes frequentados pela classe A quanto os frequentados pela classe B.

Os locais de compra do produto 2 foram decididos conforme regras que definem que tais produtos são comprados por pessoas de ambas as classes, na quinta ou na sexta feira, no período noturno, no último ponto do caminho seguido logo após a saída do trabalho, ou seja, na residência das pessoas ou nos restaurantes por elas escolhidos como local de *happy hour*. Dessa forma se esperava obter regras que cobrissem as regiões onde se localizam os restaurantes e também várias das regiões mais densamente habitadas. Como pessoas da classe A têm chance muito alta de frequentar restaurantes após o expediente (regra definida no programa que gerou os hábitos de consumo da população) esperava-se

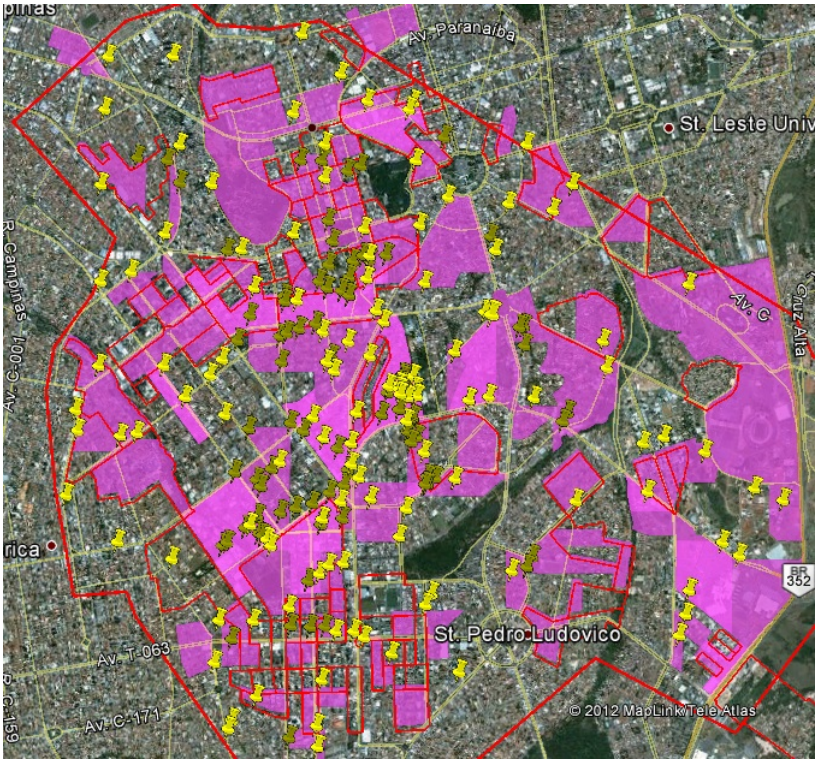

Figura 12.17: Local da Compra Produto 2

também que nem todas as regiões caracterizadas por habitantes de alto poder aquisitivo fossem cobertas pelas regras (seus habitantes teriam se dirigido aos restaurantes e não simplesmente voltado para casa depois do trabalho).

Observando -se a figura 12.17 é possível notar que realmente quase todos os restaurantes foram cobertos por alguma regra, assim como quase todas as regiões censitárias onde reside a população. As regiões que foram menos cobertas se localizam na parte sul da figura (ao lado do St. Pedro Ludovico), em áreas com alto índice de moradores da classe A segundo o IBGE.

Os padrões que se esperava observar devido à forma como foram gerados os dados foram portanto identificados e representados com sucesso pelo modelo produzido, mais uma vez indicando a eficiência do algoritmo *PART*.

A seguir, as regras do modelo construído sobre os dados relativos aos consumidores do produto 4 (videogames) foram representadas em arquivo .KML . Foram selecionadas apenas regras que se aplicam unicamente a sábados e domingos. O resultado foi retratado na figura 12.18. Marcadores amarelos dizem respeito à localização de restaurantes, marcadores verdes indicam locais de lazer e os pretos marcam a posição dos restaurantes do Setor Marista.

Segundo os padrões seguidos na criação dos hábitos de compra da população, no caso do produto 4 pessoas de ambas as classes sociais podem efetuar a compra nas segundas, terças e quartas feiras ou no fim de semana. A seleção das regras referentes apenas a fins de semana restringe portanto o número de compras consideradas. Isso explica os padrões de cobertura de regiões relativamente esparsos da figura 12.18.

Ainda segundo os padrões de consumo definidos para o produto 4, suas compras em fins

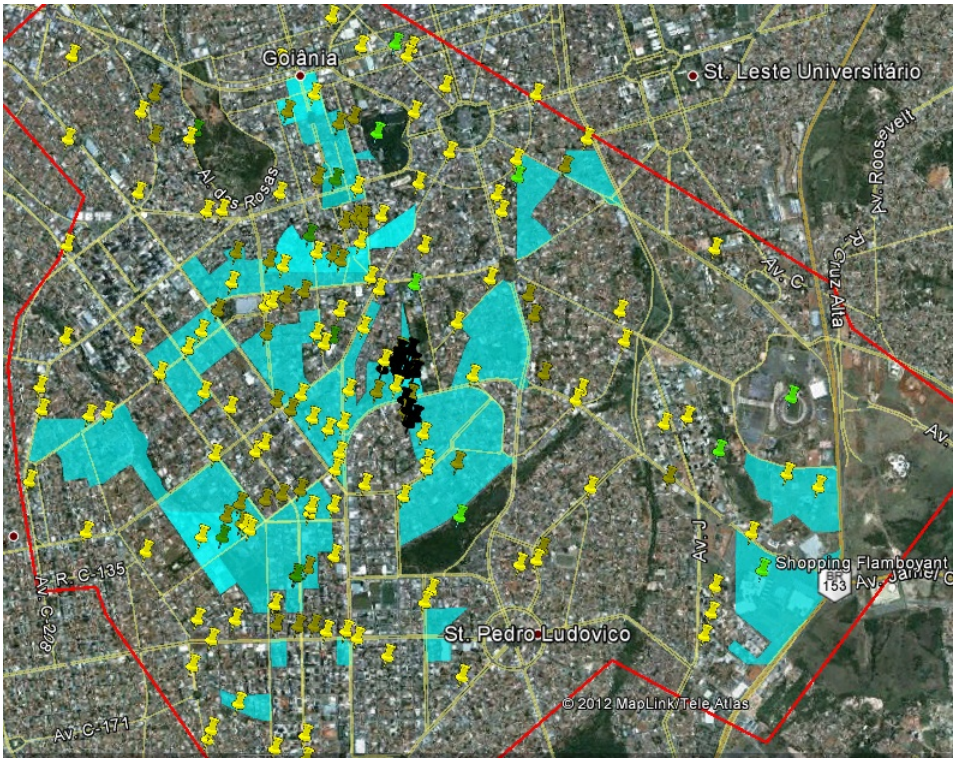

Figura 12.18: Local da Compra Produto 4

de semana são efetivadas no último ponto do primeiro caminho realizado pelo comprador, após o trabalho no sábado (caso trabalhe) ou após a realização do primeiro caminho do dia. Dessa forma era esperado que as regiões cobertas pelas regras produzidas englobassem restaurantes e locais de lazer. No tocante a pessoas da classe B a expectativa era que as regras cobrissem a região do *Shopping Flamboyant* (frequentada em 75 % das vezes que tais pessoas saem de casa em fins de semana a lazer). Já em relação a pessoas da classe A era esperado que as regras cobrissem a área dos restaurantes do Setor Marista (frequentados em 76 % das vezes que tais pessoas se dirigem a restaurantes em fins de semana).

Analisando a figura 12.18 observa-se que todos os padrões esperados foram identificados, tanto em relação a cobertura de grande parte dos restaurantes da cidade, quanto em relação ao Setor Marista e ao *Shopping Flamboyant* (canto inferior direito da imagem). Mais uma vez o modelo produzido foi adequado aos padrões dos dados.

O último arquivo .KML escrito foi referente ao modelo gerado para os dados de consumo do produto 8 (cafeteiras). Foram selecionadas as regras válidas nas segundas feiras. A imagem resultante foi a figura 12.19. Nessa figura os marcadores vermelhos indicam os locais de trabalho das pessoas da classe A.

Como já mencionado anteriormente (seção 10.6), o produto 8 é comprado exclusivamente por pessoas da classe A e em quantidades consideráveis, sendo que a chance de uma pessoa dessa classe comprar ao menos dois produtos é de 100 %. Assim as regras observadas na figura 12.19 dizem respeito somente a pessoas da classe A, fazendo referência ao comportamento da maioria delas devido ao alto nível de consumo do produto em questão. Isso justifica a grande quantidade de áreas cobertas pelas regras.

Figura 12.19: Local da Compra Produto 8

Ainda sobre os hábitos de consumo desse produto, eles são comprados no último ponto do primeiro caminho realizado pelos compradores dos dias úteis. Portanto, são comprados no local e trabalho dessas pessoas. Dessa forma esperava-se obter regras que cobrissem a grande maioria dos locais de trabalho das pessoas em questão (pessoas da classe A). De fato isso é o que se observa na figura 12.19. Praticamente todos os locais de trabalho foram cobertos, demonstrando que o modelo gerado para a tabela em questão realmente tem boa qualidade.

12.5 Considerações Finais

Pode-se perceber que o algoritmo *PART* realmente identificou os padrões inseridos nos dados no caso dos três tipos de tabelas analisados e não simplesmente gerou modelos com altas taxas de acerto na classificação de instâncias. Pela visualização de suas regras foi possível inclusive perceber o impacto que diferentes quantidades de compradores gera na área coberta pelas regras, além de comparar as diferenças de deslocamento existentes entre pessoas de diferentes classes econômicas.

Trata-se portanto de um algoritmo eficiente na identificação de padrões, sendo altamente indicado no estudo de dados reais para a busca de padrões não óbvios de deslocamento dos compradores de cada tipo de produto. Basta para isso analisar as regras produzidas uma a uma de acordo com o contexto dos consumidores do produto. A presença de pessoas que já tenham informações de marketing acerca desse público específico seria de grande valia nesse processo de busca de informações.

Foi também observado que mesmo produtos como geladeiras e fogões que têm público consumidor com características e padrões de consumo muito semelhantes neste trabalho, sendo consumidos em dias, locais e horários semelhantes, podem ter consumidores cujo comportamento difere. Isso justifica a divisão dos dados por tipo de consumidor. Tal divisão pode gerar informações importantes no contexto do marketing de tais produtos, diferenciando os hábitos dos diferentes públicos.

Por último é importante observar que o algoritmo de regras *PART* tem limitações quanto a correta identificação dos dias e horários quando cada regra gerada é válida. Durante toda a análise descrita neste capítulo, foram observados os horários e dias da semana das regras e percebeu-se que, apesar de uma tendência geral de seguirem os horários e dias nos quais os padrões inseridos nos dados são válidos, muitas vezes tais períodos se estendiam por datas quando as regras não eram mais válidas. Foi comum observar regras que se diziam válidas em todos os dias da semana (incluindo sábados e domingos) quando na verdade só o eram nas segundas, terças e quartas feiras por exemplo. Por esse motivo muitas análises discutidas neste capítulo se processaram sobre uma seleção de regras aplicáveis exclusivamente a determinados dias da semana, na tentativa de se eliminar regras válidas somente em dias úteis que se estendiam indevidamente aos fins de semana, por exemplo.

Os modelos gerados pelo algoritmo *PART* identificam muito bem padrões geográficos, sendo portanto imprescindíveis na busca de informações. Mas a análise somente deles não é suficiente, sendo importante a análise de modelos gerados por outros algoritmos que porventura produzam regras mais bem definidas quanto ao período em que são consideradas válidas.

Capítulo 13

Conclusões

Com a perspectiva de crescimento do mercado de *smartphones* e, por consequência, de aplicativos para os mesmos, é provável que brevemente seja viável a aquisição em grande volume dos dados necessários para a realização de projetos semelhantes a este. É possível até que grande parte do público consumidor de determinados produtos possa ter seus movimentos analisados das formas descritas, gerando informações precisas e confiáveis sobre os padrões de deslocamento das pessoas compradoras desses produtos.

Alguns tipos de tabelas, descritas neste trabalho, possibilitam inclusive a segmentação de determinado público consumidor (divisão por exemplo entre consumidores que moram na periferia da cidade e consumidores que moram nas partes mais centrais).

Portanto, a possiblidade de se definir com precisão os locais mais adequados para a veiculação de publicidade no espaço urbano deve tornar-se real em breve, visto que os dados poderão estar disponíveis e os métodos descritos neste trabalho podem ser usados para se obter as informações necessárias para tanto.

Os benefícios para as organizações que se utilizarem de tais métodos e das informações por eles produzidas são muitos. Incluem economia de recursos, aumento da eficiência de cobertura de público-alvo das campanhas publicitárias, melhor adequação da publicidade aos diversos tipos de público, entre outros. Para a sociedade o benefício seria a provável diminuição do excesso de publicidade, contribuindo para a queda nos níveis de poluição ambiental urbana (tanto em termos visuais quanto em termos de lixo gerado).

No tocante às técnicas utilizadas para realizar o projeto, o objetivo principal foi a viabilização da mineração de dados com atributos de natureza geográfica, espacial. Obter modelos que indicassem áreas do espaço físico urbano que englobassem os padrões dos dados. O comportamento dos algoritmos disponibilizados pelo *Weka 3* era incerto em relação a este tipo de proposta. Assim, foi necessário um amplo conjunto de testes para concluir se tais algoritmos eram apropriados para o que se pretendia.

Mas, para a realização dos testes foi preciso primeiramente obter os dados. Diante da impossibilidade de reunir dados reais eles foram gerados localmente através de programas em linguagem C. Era necessário que fossem o mais similares possível à realidade para que se pudesse estender as conclusões do trabalho a situações envolvendo dados reais. Para tanto foram utilizados serviços de obtenção de rotas do *Google Maps*, dados do Censo 2010 do IBGE e outras informações obtidas de outras fontes.

A qualidade do conjunto de dados obtido foi verificada pelos modelos gerados através da mineração de dados. Durante a visualização dos modelos foi possível perceber que as

áreas apontadas como mais movimentadas foram as regiões contendo as vias principais do perímetro urbano considerado, o que coincide com a realidade. Além disso foram observados vários padrões condizentes com o esperado no contexto real, por exemplo, o fluxo de pessoas por regiões ora mais ora menos residenciais de acordo com o dia e horário das regras que englobaram tais áreas. Tais observações apontam para a validade do conjunto de dados. Sugerem ser possível simular dados de deslocamento de uma população que sejam similares à realidade, desde que utilizadas as fontes citadas neste trabalho.

Gerados os vários conjuntos de dados (população e seus hábitos de consumo e deslocamento) a próxima fase foi o seu processamento. Eles foram integrados para a criação de quatro tipos de tabelas que foram utilizadas na fase de mineração. Cada tipo favoreceu a obtenção de informações de determinada natureza. Tais formatos de tabela, discutidos no corpo do trabalho, são sugeridos para o caso do tratamento de dados reais. Levaram a ótimos resultados do ponto de vista do acerto na classificação de instâncias dos modelos gerados pelos algoritmos de mineração de dados. Além disso, os modelos identificaram informações importantes sobre os dados, como será discutido a seguir, denotando que o formato das tabelas criadas favoreceu a descoberta de conhecimento.

Quanto à mineração de dados propriamente dita, foram utilizados todos os algoritmos do *Weka 3* que se aplicavam aos dados de cada tipo de tabela. Alguns algoritmos apresentaram grande demora na sua execução, tornando-se inviáveis para a utilização neste trabalho. Muito provavelmente terão comportamento igualmente ruim em cenário real, até porque nesse caso o volume de dados, idealmente, seria muito maior. São eles o *Logistic, MultilayerPerceptron, SimpleLogistic, SMO, SMOreg, VotedPerceptron, UserClassifier* e o *LMT*.

Além dos citados, os algoritmos da pasta *meta* também não foram utilizados. A atuação deles é apenas para a melhora de modelos obtidos por outros algoritmos, sendo que o foco deste projeto foi encontrar bons algoritmos e não simplesmente bons modelos.

Outros algoritmos também não foram utilizados por não se adequarem aos tipos dos dados do trabalho (discutidos na seção 11.2). É esperado que ocorra o mesmo no caso de dados reais se forem geradas tabelas com o mesmo padrão das discutidas neste trabalho.

Excetuando-se os mencionados, foram utilizados vários dos algoritmos disponíveis no *Weka 3* sobre várias das tabelas geradas. Através da análise das taxas de acerto dos modelos produzidos por cada um nas várias tabelas deste trabalho foi identificado o seguinte:

- Os algoritmos de árvore *RandomTree, RandomForest* e muitas vezes também o *j48* geram modelos que têm altos índices de acerto, podendo tais modelos serem interpretados com o auxílio de programas de visualização. Seu uso é indicado em projetos futuros desta natureza.

- O algoritmo de regras *PART* também gera modelos com altas taxas de acertos. Suas regras são especialmente apropriadas para a interpretação gráfica, sendo sua utilização muito importante no processo de descoberta dos padrões espaciais dos dados. É portanto recomendado para projetos de mineração de dados geográficos.

- O algoritmo *Ridor* foi o que gerou os mais altos índices de acerto na classificação de instâncias. No entanto, em muitos casos, gera modelos com regras de difícil representação gráfica pois, na configuração que resulta em maiores índices de acerto, gera regras para as classes que não são de interesse para a busca de informações.

Sua utilização em projetos futuros é recomendada com ressalva quanto a difícil interpretação de alguns dos seus modelos gerados.

- O algoritmo *Kstar* gerou muitos bons modelos, mas o teste dos mesmos sobre o conjunto de dados de teste geralmente consumia muito tempo. Isto leva a crer que os modelos gerados são complexos. A aplicação dos mesmos sobre um conjunto de dados muito grande seria demorada. Além disso, a análise dos modelos ou sua interpretação não foi possível através do *Weka 3*. A mesma análise vale para o *IBK*, sendo no entanto um algoritmo menos bem sucedido em relação ao *Kstar*. Na ausência de uma forma de representar os modelos por eles produzidos, sua utilização não é recomendada.

- Os algoritmos bayesianos geraram modelos bons apenas em alguns casos e de difícil representação. Dessa forma, sua utilização não é tão recomendável quanto a dos outros algoritmos citados.

- Quanto aos algoritmos *VFI* e *HyperPipes*, é recomendado muito cuidado na observação de suas taxas de acerto. Geralmente geram modelos que acertam grande parte das instâncias de uma classe e apresentam altas taxas de erro para a outra classe, ou seja, geram regras pouco específicas, muitas vezes sem utilidade na obtenção de informações sobre os dados. Sua utilização não é recomendada.

Os demais algoritmos disponíveis no *Weka 3* geraram modelos bons em apenas uma parte das tabelas de dados. Portanto, sua utilização não é imprescindível.

Foi também realizada uma avaliação qualitativa dos modelos produzidos. Para tanto, foram inseridos padrões específicos nos dados que gerariam consequências observáveis do ponto de vista espacial. Através da detecção ou não desses padrões pelos modelos gerados, pôde ser avaliado se eles de fato identificam padrões espaciais nos dados. Entre esses padrões estava por exemplo a tendência de pessoas da classe econômica B frequentarem determinada região em fins de semana (os demais estão descritos na seção 10.6).

Como os modelos produzidos pelos algoritmos são de difícil interpretação (em alguns casos são compostos por centenas de regras envolvendo coordenadas geográficas e regiões censitárias), foi escrito um programa que os lê e gera um arquivo visualizável no *Google Earth*. A visualização de regras mencionada nesta conclusão faz referência aos arquivos gerados por esse programa, aplicados sobre os modelos do algoritmo *PART*. A escolha desse algoritmo foi feita pelo fato de ele gerar modelos com bons índices de acerto em todos os tipos de tabela deste trabalho e também por gerar regras específicas e apropriadas para a visualização. Suas regras envolvem regiões censitárias, o que as torna mais específicas e fáceis de interpretar.

Aplicado aos modelos *PART*, o programa descrito gerou diversas imagens que ilustraram vários dos padrões inseridos nos dados. Isso mostra que o algoritmo em questão além de criar regras com boas taxas de acerto, gera modelos de qualidade, que apontam corretamente os padrões espaciais dos dados. Dessa forma, pelo menos no caso do algoritmo citado, a mineração de dados de natureza geográfica é viável e gera bons resultados. Nos casos dos outros algoritmos, cujas regras não foram visualizadas, apenas se pode afirmar que geram modelos com altas taxas de acerto na classificação de instâncias. Possivelmente também constroem modelos capazes de identificar padrões geográficos pois muitos geram regras semelhantes às do *PART*, segundo constatado pela leitura das mesmas.

Quanto aos modelos *PART* vale lembrar que foram identificadas limitações. As datas e horários de validade das regras não foram tão precisas quanto o necessário. Daí a necessidade da escrita de outros programas visualizadores que interpretem modelos de outros tipos de algoritmos. Em projetos futuros tais programas serão necessários, tanto para possibilitar a avaliação de modelos mais precisos em relação a datas e horários quanto para comparar a qualidade dos modelos de algoritmos diferentes. É também possível que cada algoritmo seja mais favorável à visualização de determinado padrão, hipótese a ser investigada em projetos futuros através da escrita dos programas mencionados.

Pela análise das imagens relativas aos modelos *PART* foi também identificado que mesmo no caso de produtos consumidos com probabilidades e de acordo com regras semelhantes, houve diferenças nas áreas cobertas pelos modelos produzidos. Isso significa que mesmo em casos que envolvem a busca de padrões de públicos semelhantes a separação dos dados de cada público consumidor é importante pois geralmente resulta em modelos diferentes. Por essa razão, tal procedimento é indicado sempre que for possível a sua realização.

A divisão do perímetro urbano estudado em diversas regiões menores (envolvendo uma ou algumas poucas quadras) também foi muito importante, tendo sido um dos principais fatores responsáveis pelo sucesso na detecção dos padrões espaciais dos dados. Foi essa divisão que tornou possível obter regras mais bem delimitadas, mais específicas, em modelos produzidos por algoritmos como o *PART*. Algoritmos que desconsideraram essas divisões obtiveram regras com taxas de erro maiores, abrangendo áreas maiores da cidade, o que dificultou a análise das mesmas em busca de informações. Dessa forma é imprescindível que não se considere simplesmente as coordenadas dos pontos armazenados mas também que eles sejam inseridos em subdivisões do espaço urbano considerado.

O caso das tabelas contendo as regiões por onde passa cada caminho realizado demonstrou ainda a importância de se delimitar de forma racional tais regiões. Apesar dos modelos criados para essas tabelas não terem sido visualizados, ao ler as regras produzidas foram identificadas algumas relativas apenas a horários. Elas são atribuídas ao fato de as regiões censitárias terem tamanho muito diversificado, criando a possibilidade de se mapear uma rota por tais regiões de acordo com o tempo levado para percorrer cada uma. Assim, uma regra envolvendo somente horários engloba um número limitado de possibilidades, de acordo com o tamanho e a ordem das regiões por onde passam os caminhos. Por esse motivo são geradas tais regras (resultam em taxas de erro pequenas). A criação de regiões mais uniformes, sem diferenças muito grandes no tamanho das áreas que envolvem, resolveria esse problema. Daí, conclui-se ser necessário em projetos como este, a divisão das áreas urbanas consideradas em regiões de tamanhos compatíveis uns com os outros, não muito distintos.

A escrita de tabelas do tipo tratado no parágrafo anterior é muito importante em trabalhos como este. Isso porque permite a divisão do público consumidor de um produto em segmentos, favorecendo a descoberta de informações sobre cada um. Tais informações são de extrema importância para as organizações responsáveis pela produção e comercialização dos produtos pois auxiliam na segmentação do mercado consumidor, prática útil e necessária no contexto atual de marketing e publicidade. A forma de organização de dados dessas tabelas pode inclusive ser o foco de trabalhos futuros. A representação gráfica das regras produzidas para esse tipo de tabela também seria de grande valia.

Por fim, a última etapa da mineração de dados que compreende a identificação e a

separação das regras úteis (em meio às que já são conhecidas e às que são equivocadas), não pôde ser realizada. Para tanto seriam necessários dados reais e conhecimento de alguns fatos sobre o público consumidor analisado, ou seja, conhecimento sobre a área da aplicação. Só assim poderia ser avaliada cada regra no tocante à sua relevância e validade na prática.

Nesse caso, seria recomendável inclusive o uso dos algoritmos *meta* para a obtenção de modelos melhorados.

Um trabalho com dados reais que possibilitasse a realização dessa última etapa poderia produzir excelentes resultados e informações valiosas, tanto para a organização que se dispusesse a participar do projeto, quanto para os pesquisadores nele envolvidos.

Referências

[1] Associação Brasileira de Empresas de Pesquisa ABEP. *Critério de Classificação Econômica Brasil*. 2012. 37

[2] Marcelo C. Neri and Luisa C. C. de Melo. *Atlas do Bolso dos Brasileiros*. FGV/IBRE, Centro de Políticas Sociais, Setembro 2009. 35

[3] Lugar Certo. http://www.lugarcerto.com.br acessado em 15/01/2012. 34, 35

[4] H.Y.U.H. Ching. *Gestão de Estoques na Cadeia Logística Integrada*. Atlas, 2010. 1

[5] Krzysztof J. Cios, Witold Pedrycz, Roman W. Swiniarski, and Lukasz A. Kurgan. *Data Mining A Knowledge Discovery Approach*. Springer Science and Business Media, 2007. 9

[6] CURL. http://curl.haxx.se/ acessado em 06/01/2012. 29, 42

[7] N. Dholakia, M. Rask, and R.R. Dholakia. *M-commerce: global experiences and perspectives*. E-Libro. Idea Group Pub., 2006. 5

[8] EBIT. http://www.e-commerce.org.br/stats.php acessado em 20/01/2012. 50

[9] A. El-Rabbany. *Introduction to GPS: the Global Positioning System*. Artech House, 2002. 4

[10] SAGA GIS. http://www.saga-gis.org/en/index.html acessado em 13/11/2011. 24

[11] C.F.S. GOMES. *Gestão da Cadeia de Suprimentos Integrada: Tecnologia da Informação*. Pioneira Thomson Learning, 2004. 12

[12] Google. http://developers.google.com/kml/documentation/kmlreference, acessado em 10/12/2011. 18, 24, 25

[13] Google. http://maps.google.com/support/ bin/answer.py?hl=pt-br&answer=7060 acessado em 27/6/2011. 19

[14] Jiawei Han and Micheline Kamber. *Data Mining Concecpts and Techniques*. Morgan Kaufmann Publishers, second edition, 2006. 10, 52

[15] Monte Hancock and Rhonda Delmater. *Data Mining Explained, A Manager's Guide to Customer-Centric Business Intelligence*. Butterworth-Heinemann, 2001. 12

[16] David Hand, Heikki Mannila, and Padhraic Smyth. *Principles of Data Mining*. The MIT Press, 2001. 11

[17] G. Holmes, A. Donkin, I.H. Witten, and University of Waikato. Dept. of Computer Science. *WEKA: a machine learning workbench*. Working paper series. Dept. of Computer Science, University of Waikato, 1994. 11, 21, 23, 61

[18] Censo 2010 IBGE. http://www.ibge.gov.br/censo2010/ acessado em 04/01/2012. 20, 26, 33, 34

[19] M. Ilyas and S.A. Ahson. *Smartphones: Research Report*. International Engineering Consortium, 2006. 3

[20] Environmental Systems Research Institute. Esri shapefile technical description. 1998. 25, 28

[21] Gordon S. Linoff and Michael J. A. Berry. *Mining the Web, Transforming Customer Data into Customer Value*. John Wiley & Sons, Inc., 2001. 7

[22] Google Maps. https://developers.google.com/maps/documentation/directions/?hl=pt-br acessado em 28/6/2011. 20

[23] Tom M. Mitchell. Mining our reality. *Science*, 326(5960), 2009. 12

[24] Topografix. http://www.topografix.com/gpx/1/1/, acessado em 15/12/2011. 25

[25] Waikato University. http://www.cs.waikato.ac.nz/ml/weka/ acessado em 27/6/2011. 21

[26] Revista Veja. http://veja.abril.com.br/noticia/economia/brasil-ultrapassa-a-marca-de-um-celular-por-habitante acessado em 23/10/2011. 3

[27] Ian H. Witten and Eibe Frank. *Data mining: practical machine learning tools and techniques*. Morgan Kaufmann Publishers, second edition, 2005. 9, 10, 11