UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO

CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA

PROGRAMA DE POS GRADUAÇÃO EM INFORMÁTICA

APLICAÇÃO DE APRENDIZADO DE MÁQUINA PARA INFERÊNCIA DE MODO DE TRANSPORTE EM TRACES DE SMARTPHONES

Carlos Alvaro de Macedo Soares Quintella

Orientadores

Leila Cristina Vasconcelos Andrade

Carlos Alberto Vieira Campos

RIO DE JANEIRO, RJ – BRASIL

SETEMBRO DE 2013

Aplicação de aprendizado de máquina para inferência de modo de transporte em traces de smartphones

Carlos Alvaro de Macedo Soares Quintella

DISSERTAÇÃO APRESENTADA COMO REQUISITO PARCIAL PARA OBTENÇÃO DO TÍTULO DE MESTRE PELO PROGRAMA DE PÓS- GRADUAÇÃO EM INFORMÁTICA DA UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO (UNIRIO). APROVADA PELA COMISSÃO EXAMINADORA ABAIXO ASSINADA.

_____

Leila Cristina Vasconcelos Andrade, D.SC. - UNIRIO

_____

Carlos Alberto Vieira Campos, D.SC. - UNIRIO

_____

_____

RIO DE JANEIRO, RJ – BRASIL

SETEMBRO DE 2013

Quintella, Carlos Alvaro de Macedo Soares.

Q7 Aplicação de aprendizado de máquina para inferência de modo de

transporte em traces de smartphones / Carlos Alvaro de Macedo

Soares Quintella, 2013.

105 f. ; 30 cm

DEDICATÓRIA

Dedico este trabalho a minha filha Julia e a minha esposa Marina.

AGRADECIMENTOS

O processo de produção de uma dissertação requer um grande número de participantes. Devo citar, desde os professores e colegas pesquisadores, até os voluntários que participaram na coleta de informações, todos foram especialmente importantes para elaboração deste trabalho. Assim, sem desejar deixar ninguém de fora, mas reconhecendo que isso é impossível; gostaria primeiramente de agradecer aos meus orientadores, que tanto se empenharam para o sucesso da dissertação e sem o qual por diversos motivos ela não teria acontecido. Agradeço ao Carlos Alberto Campos com o incentivo de buscar novas áreas desafiadoras da pesquisa; e a Leila C.V. Andrade que sempre me manteve motivado a continuar. Em segundo lugar, mas não menos importante, gostaria de agradecer a Simone Bacelar Leal Ferreira, que muito me ensinou como proceder com os textos acadêmicos, dando dicas e estratégias para manter o texto sobre controle e seguir as normas adequadas. A Kate Revoredo por me apresentar a área de mineração de dados, e inspirar-me a investigá-la. Gostaria de agradecer também as professor Sidney e ao Sean Siqueira pela atenção, conselhos e pelas interessantíssimas aulas.

Agradeço ao IBGE, e aos meus colegas de trabalho, em especial aos Doutores Arnaldo Barreto e José Luís Thomazelli, por muito incentivarem, apoiarem e permitirem a flexibilização do meu horário de trabalho para que eu pudesse assistir às aulas vespertinas.

De forma especial, agradeço a todos os voluntários que ajudaram a coletar informações com a ferramenta *CityTracks*.

Por fim gostaria de agradecer a muitos mestres que foram muito importantes ao longo da minha formação e que contribuíram indiretamente para realização deste trabalho.

QUINTELLA, Carlos Alvaro de Macedo Soares. **Aplicação de Aprendizado de máquina para inferência de modo de transporte em traces de smartphones.** UNIRIO, 2013. 105 páginas. Dissertação de Mestrado. Departamento de Informática Aplicada, UNIRIO.

RESUMO

Este trabalho apresenta o resultado da aplicação de técnicas de aprendizado de máquina, com objetivo de identificar o modo de transporte utilizado pelos usuários de *smartphones* para utilização por sistemas cientes de contexto. Desta forma, apresenta: a definição e implementação de uma arquitetura para coleta de dados de *smartphones*; a coleta de trajetórias de mobilidade de usuários de *smartphones* armazenadas em um banco de dados; a aplicação de técnicas de segmentação de trajetórias e por fim, testes da aplicação de técnicas de mineração de dados para investigar algoritmos de aprendizagem de máquina visando a classificação dos registros de movimentação de acordo com o modo de transporte utilizado na sua geração.

Palavras chaves: smartphones, trajetórias, modo de transporte, sistemas cientes de contexto, contexto, computação ubíqua, mineração de dados, aprendizagem de máquina.

ABSTRACT

This work presents the result of applying machine-learning techniques, to identify the transportation mode of smartphone users, specifically to be applied by context-aware systems. It includes: definition and implementation of an architecture for smartphone data collection for collecting smartphone users trajectories in a database; the application of trajectory segmentation techniques; and finally analyzing results of data mining techniques applied to classify the movement records, according to the transportation mode of their user.

Keywords: smartphone, trajectory, transportation mode, context awareness ubiquitous computing, data mining, machine learning.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

CAPÍTULO I - Introdução

Vivemos num mundo cada vez mais digital, com nossas vidas povoadas por diversos dispositivos computacionais que fazem parte das nossas atividades do dia a dia. Com o advento e o sucesso de mercado dos smartphones e de outros dispositivos móveis, como os *tablets*, novas formas de interação com os dispositivos computacionais foram introduzidas, inaugurando novas aplicações e potencializando os resultados de tarefas rotineiras. Com a expansão da cobertura dos diversos tipos de redes sem fio, a conectividade móvel passou a ser mais utilizada e desejada pelos usuários. Hoje, uma boa parte do consumo de conteúdo digital, ou mesmo da geração de conteúdo, passou a ocorrer com o usuário em movimento, dado pela ampla disponibilidade tanto das redes sem fio, quanto dos dispositivos móveis.

É através dos dispositivos móveis computacionais que a visão da computação ubíqua está se tornando uma realidade. Nessa visão, os usuários interagem com sofisticados sistemas computacionais, praticamente sem notá-los. Esta interação se dá de forma intuitiva, sem que uma bagagem de conhecimento seja necessária para utilizá-los. A computação ubíqua se vale dos sistemas cientes de contexto, tanto para aumentar a eficiência das suas interações, quanto para permitir novas formas de aplicações.

Os sistemas cientes de contexto são aqueles capazes de extrair informações adicionais de contexto além daquela que é informada pelo próprio usuário. No caso específico das aplicações móveis, uma parte importante das informações que compõe o contexto de uso está relacionada ao local e ao meio de transporte utilizado pelo usuário. Nos últimos anos diversos estudos se voltaram para entender as informações produzidas por usuários de smartphones durante suas utilizações diárias com a finalidade de entender a sua movimentação. Estes estudos, buscam responder a perguntas, tais como: qual a sua origem, destino, qual o objetivo e qual meio de transporte está sendo utilizado? Por que esta movimentação ocorreu? Pode vir a acontecer de novo?

O foco do presente trabalho consiste na análise de informações de localização provenientes de smartphones para determinar o modo de transporte utilizado por seus usuários e com objetivo estrito de utilização em sistemas cientes de contexto. Este tipo de aplicação requer a capacidade de aquisição e classificação de informações, de forma rápida e escalável para que possa ser utilizada por um maior número de aplicações possível.

Este Capítulo está estruturado da seguinte forma: na Seção 1.1, são apresentadas as motivações que definiram o campo de pesquisa adotado; em seguida, na Seção 1.2, o objetivo do trabalho é apresentado; na Seção 1.3, é apresentada lista das contribuições deste trabalho; na Seção 1.4, é apresentada a metodologia de pesquisa utilizada e na Seção 1.5 apresenta a organização utilizada neste trabalho.

1.1. Motivação

Cada vez mais utilizamos os dispositivos computacionais móveis durante nossos deslocamentos diários. Considerando que a computação ubíqua se tornou uma realidade através dos *tablets* e smartphones, e que para funcionar de forma eficaz precisa dos sistemas cientes de contexto. Nesse ambiente, o contexto do modo de transporte passa a ser um importante componente do contexto de uso das aplicações. Considerando também, que quanto mais rico o conhecimento do contexto, melhor a previsão do contexto futuro. Este autor acredita que os recentes avanços tecnológicos em áreas de redes sem fio, processamento de grandes massas de dados, a computação em nuvem e inteligência artificial estão estabelecendo alicerces renovados que vão alavancar a utilização de sistemas cientes de contexto distribuídos e compartilhados. Com isso, novas aplicações e pesquisas no campo de redes e de computação ubíqua podem fazer uso das contribuições deste trabalho.

1.2. Objetivo

Até o presente momento, pelo melhor conhecimento do autor desta dissertação, apesar de já existirem mecanismos para se identificar o meio de transporte utilizado pelos usuários de smartphones; quando se considera a possibilidade de se utilizar somente o sensor de localização e a necessidade de identificar o meio de transporte, enquanto o deslocamento ainda está acontecendo, ainda não existe uma palavra definitiva sobre o

assunto. Além disso, deve-se considerar que uma determinada localidade geográfica podem influenciar nos meios de transportes disponíveis e na forma que estes se movimentam dentro de suas bordas. Além disso, é preciso considerar que os diferentes tipos de sensores utilizados nos celulares podem ter influência direta na qualidade dos dados obtidos e assim nos resultados produzidos pelas aplicações que os utilizam. Por consequência dos fatores supracitados, o problema abordado neste trabalho é: a ausência de mecanismos precisos para obter informação de contexto de modo de transporte utilizado para sistemas cientes de contexto que utilizem somente dados provenientes do sensor de localização. O objetivo deste trabalho é definir um mecanismo para identificar o meio de transporte utilizado por usuários de *smartphones*.

Assim, os seguintes requisitos devem ser observados no processo, para que o objetivo possa ser alcançado com plenitude:

(a) o intervalo de tempo entre a captura da informação e a classificação deve ser compatível com os requisitos de interatividade dos sistemas cientes de contexto (assim, quanto mais rapidamente for possível definir o contexto, maior o grau de utilidade esta informação para os mais diversos tipos de aplicações interativas);
(b) deve-se considerar a limitação de recursos de armazenamento, processamento, capacidade da rede sem fio e consumo de energia dos dispositivos móveis;
(c) a solução deve ser escalável, para que possa ser usada por um grande número de usuários.

1.3. Contribuições

O objetivo primário deste trabalho é apresentar uma análise da utilização dos algoritmos de aprendizado de máquina para classificar traces produzidos por smartphones das movimentações diárias dos usuários de acordo com o modo de transporte utilizado. É importante ressaltar que a finalidade dessa análise é a aplicação por sistemas cientes de contexto, o que introduz uma série de restrições além das já citadas na seção anterior:

(a) a precisão deve ser alta, espera-se resultados que sejam acima de 90%, este valor e sugerido por REDDY, MUN, et al. [25] (de acordo com as restrições do aplicativo utilizado no projeto PEIR [25]);

(b) a classificação deve acontecer próxima ao tempo real, no mínimo enquanto se dá a movimentação, para que assim esta informação possa ser utilizada por sistemas cientes de contexto;
(c) e por consequência, as informações da trajetória corrente completa não estarão disponíveis, embora o histórico de trajetórias possa ser utilizado.

As seguintes contribuições adicionais também fazem parte deste trabalho, são elas:

(a) a definição de uma arquitetura escalável para coleta de dados de smartphones, incluindo traces e notas geo-localizadas;
(b) a definição de um modelo de banco de dados para a coleta de traces de smartphones;
(c) desenvolvimento de um banco de dados de trajetórias de usuários de smartphones com movimentações;
(d) um sistema para coleta de dados genéricos de smartphones, que poderá ser utilizado para diferentes tipos de pesquisas que envolvam a coleta de amostras e/ou o sensoriamento com uso de smartphones;
(e) a definição de um processo de coleta de traces por usuários e a análise crítica de sua aplicação;
(f) a definição de um mecanismo para segmentação de traces e transformação destes em dados para descoberta de conhecimento usando técnicas de mineração de dados.

1.4. Método de pesquisa

Quanto ao método de pesquisa empregado, primeiramente, deve-se considerar que uma pesquisa pode ser classificada segundo diferentes aspectos, como apresentado nos trabalhos: [33], [32] e [28]. O presente trabalho pode ser classificado em relação aos seguintes aspectos:

(a) em relação ao objeto de pesquisa, como uma pesquisa tecnológica - conforme apresentado no trabalho de SOUZA, MULLER, et al. [28] e na apresentação de JUNG [33];
(b) já, quanto aos seus objetivos, pode ser classificado como uma pesquisa experimental, conforme abordado por WAZLAWICK [32]; o que é comum a pesquisas de tecnologia, como citado por SOUZA, MULLER, et al. [28];

(c) quanto a abordagem, tem características quantitativa [28], quando se utiliza de parâmetros numéricos e estatísticos em processos de análise de dados;

Além disto, de forma mais especifica, este trabalho se enquadra na categoria de pesquisa exploratória experimental - conforme classificação apresentada no estudo [13]. Deve-se considerar, que neste tipo de pesquisa, busca-se entender um fenômeno, aumentando o conhecimento sobre ele de forma gradual, conforme apresentado em [14]. Por consequência, pesquisa apresenta um amplo escopo e dificilmente responde definitivamente as questões abordadas em sua totalidade, embora contribua para o aumento do conhecimento do campo e das próprias questões em si.

O método de pesquisa aplicado está estruturado em cinco grandes etapas, onde: (i) a primeira etapa é a revisão bibliográfica e aquisição do conhecimento de computação ubíqua, redes sem fio e aprendizado de máquina, sistemas de informação geográficas e processamento de trajetórias, (ii) a segunda etapa refere-se a aquisição de dados, onde foi proposta uma arquitetura, o desenvolvimento e a implementação de ferramentas para coleta de dados; (iii) a terceira etapa seleção e implementação de mecanismos de segmentação das trajetórias; (iv) a experimentação usando técnicas de descoberta de conhecimento em banco de dados; (v) e finalmente a análise dos resultados obtidos.

1.5. Estrutura deste trabalho

Este trabalho está estruturado da seguinte forma: o Capítulo II apresenta o problema, seus desafios e a literatura relacionada; o Capítulo III são apresentados os conceitos, métodos, e ferramentais empregados; o Capítulo IV apresenta a proposta de arquitetura para coleta de informações dos smartphones, a mesma empregada para efetuar a coleta de dados utilizados neste trabalho; o Capítulo V apresenta a execução da coleta de dados e da etapa de mineração de dados; o Capítulo VI apresenta os resultados obtidos, assim como a análise crítica do processo; o Capítulo VII apresenta a conclusão deste trabalho e os trabalhos futuros.

CAPÍTULO II - -Definição do Problema

Este capítulo tem o objetivo de apresentar o problema abordado, iniciando com uma contextualização superficial, cujo o objetivo é facilitar tanto a leitura quanto a compreensão, para em seguida apresentar o problema e os principais desafios envolvidos em busca de uma solução.

2.1. Contextualização

Com objetivo de subsidiar a compreensão do problema estudado, torna-se necessário apresentar o contexto onde este se apresenta, incluindo um breve histórico da área de estudo e importantes desdobramentos.

2.1.1. Computação ubíqua

O termo computação ubíqua foi inicialmente utilizado por Mark Weiser, pesquisador da Xerox nos anos 1990. O termo, Ubíquo[1], se refere ao que está presente em toda parte ao mesmo tempo. Já, segundo POSLAD [23], o termo computação ubíqua é usado para designar sistemas de TIC (Tecnologia da Informação e Comunicação) que permitem que informações e atividades estejam disponíveis extensivamente, de forma que possam ser utilizadas de forma intuitiva, se apresentando de forma transparente para o usuário.

2.1.2. Sistemas cientes de contexto

Um dos campos de estudo mais importantes da computação ubíqua é a computação ciente de contexto, conforme apontado por BALDAUF, DUSTDAR, et al. [4]:

[1] Segundo o dicionário Michaelis:
adj (*lat ubiquu*) Que está ou pode estar em toda parte ao mesmo tempo; onipresente.
(http://michaelis.uol.com.br/moderno/portugues/index.php?lingua=portugues-portugues&palavra=ub%EDquo , acesso em 01/03/2013).

> *"...são sistemas capazes de adaptar seu modo de operação, sem a necessidade de intervenção explícita do usuário, assim buscando aumento da usabilidade e efetividade levando em consideração o contexto do ambiente."*

A computação ciente de contexto é uma tecnologia emergente, conforme lembra MEHRA [18], e que quando utilizada com dispositivos pessoais móveis, tem a capacidade de otimizar os resultados da utilização, tanto no aspecto de usabilidade quanto do resultado desejado. Ela tem o potencial de revolucionar experiência de uso dos dispositivos móveis, permitindo completar a informação que foi dita pela que não foi dita.

Contexto é qualquer informação que possa ser usada para caracterizar a situação das entidades que são relevantes para a interação de um usuário com um sistema. Estas entidades podem ser: uma pessoa, um lugar ou um objeto, incluindo o próprio usuário ou o sistema. O contexto pode ser classificado como externo e interno. O contexto externo pode ser obtido com o uso de sensores. Já o contexto interno, pode ser obtido com informações providas pelo usuário e pelo monitoramento da utilização do sistema. O modo de transporte utilizado é um importante componente do contexto de um usuário móvel, conforme é lembrado por STENNETH, WOLFSON, et al. [29].

Quando se trabalha com contexto, pode-se distinguir três tipos de entidades agrupando as informações de contexto[4]: pessoas, lugares e coisas. Cada uma dessas entidades pode ser representadas por um conjunto de atributos, que por sua vez podem ser agrupados em quatro categorias: identidade, localização, estado (ou status) e tempo. Identidade é o atributo identificador da entidade. Localização é o local onde a entidade se encontra, seja ele expresso em relação a um objeto ou em forma de coordenadas. Estado caracteriza a atividade e/ou outras informações intrínsecas da entidade. Tempo pode ser em formato de *timestamp* ou apenas expressando a ordem de eventos envolvendo a entidade. Na maioria dos casos, somente a informação de contexto não é suficiente para a utilização com um sistema ciente de contexto, atributos adicionais são necessários, como: a fonte (ou origem) da informação, quando ocorreu a medição (um *timestamp*), e o grau de confiança dessa medição.

A informação de contexto deve estar disponível em tempo hábil (compatível com a aplicação), é sensível a um deadline, além do qual ela pode perder o seu valor. Entre os próximos desafios relacionados a informação de contexto pode-se relacionar: a

obtenção de uma caracterização mais rica do contexto, utilizando dados dos sensores de forma combinada, ou ainda, obter a capacidade de inferir contexto futuro, explorando a informação histórica de contexto.

A representação do contexto requer um modelo eficiente, conforme cita BALDAULF, DUSTDAR, et al. [4] que precisa suportar o processamento, compartilhamento, armazenamento e gerenciamento de informação de contexto, tanto para operações de leitura, quanto de atualização. Entre as abordagens para a modelagem do contexto, as mais relevantes são as baseadas em: (i) modelo chave-valor (*Key-Value Model)*; (ii) *Markup scheme model*; (iii) Modelos Gráficos (como por exemplo o *Unified Modeling Language* - UML); (iv) modelos orientados a objetos; (v) modelos baseados em lógica (regras, expressões e fatos são usados para modelar o contexto); (vi) modelos baseados em ontologias (representam modelos de conceitos e relacionamentos). Os modelos baseados em ontologias são os mais expressivos [4].

Os primeiros sistemas cientes de contexto foram desenvolvidos para funcionar de forma fechada, em um ambiente de redes restrito. O aumento das áreas cobertas por redes móveis sem fio trouxe o desejo e a necessidade dos usuários estarem conectados a todo momento, inclusive durante seus deslocamentos e acessando os diversos serviços disponíveis na nuvem, mesmo quando passando por diferentes infraestruturas de redes sem fio, conforme citado por BELLAVISTA, CORRADI, et al. [5].

Através do uso de middleware para sistemas ciente de contexto é possível que várias aplicações possam ter acesso às informações de contexto, permitindo com um grau maior de escalabilidade. A distribuição de dados de contexto (*context data distribution*), denomina a capacidade de obter e entregar informações relevantes de contexto para todas as entidades interessadas.

A obtenção do contexto é uma tarefa que pode ser distribuída entre o dispositivo e o middleware na nuvem. A quantidade de recursos alocados a essa tarefa, seja no dispositivo móvel ou no servidor na rede deve ser ajustada para não comprometer a escalabilidade nem drenar processamento, memória e bateria dos dispositivos móveis que são recursos escassos.

2.2. O problema

Esta seção tem o objetivo de apresentar os detalhes relacionados ao problema em estudo, iniciando-se pela caracterização do problema, para em seguida apresentar as possíveis aplicações e assim, justificando a relevância do estudo. Por fim, são apresentados os desafios a serem endereçados para definir uma solução satisfatória.

2.2.1. Caracterização do problema

A detecção do meio de transporte em tempo de uso requer um alto nível de acurácia, conforme aponta REDDY, BURKE, et al. [24], sob pena de que os usuários percam a confiança na aplicação e por consequência deixem de utilizá-la.

O problema abortado neste estudo á a classificação de padrões de movimentação dos usuários de smartphones, com objetivo de identificar o modo de transporte por eles utilizados, visando à aplicação por sistemas cientes de contexto.

Considerando a existência de diferentes modelos de smartphones, a diversidade de sensores disponíveis em cada um deles - não somente em relação ao tipo, mas também em relação as características técnicas de cada um deles, as aplicações que necessitam utilizar dados destes sensores de localização e precisam lidar com um considerável grau de heterogeneidade entre os diferentes aparelhos e sensores. Por outro lado, o uso de um sensor de localização baseado em *sensor fusion* de GPS, WiFi e rede de telefonia celular, mesmo com características diferentes entre fabricantes, está presente na grande maioria dos equipamentos, incluindo iOS, WindowsPhone e Android. Este é então, o componente que apresenta melhores resultados, quando utilizado de forma isolada, para identificar o modo de deslocamento utilizados por seus usuários [25], principalmente quando consideramos os diversos modos de deslocamentos tanto motorizados quanto os não motorizados.

2.2.2. Aplicações

A capacidade de identificar o contexto de modo de transporte se prova valiosa para diferentes campos de aplicação. Para sistemas de *e-health*, permite calcular o gasto calórico e a quantidade de esforços que um usuário utilizou em seu deslocamento. Aplicações similares ao *Nike Running*[2] e ao *Map My Ride*[3] podem se beneficiar da

[2] Nike Running App (“http://nikeplus.nike.com/plus/products/gps_app/”)

identificação de que o usuário está usando formas de deslocamento de interesse. Para sistemas de gerenciamento de cidades, no campo de cidades inteligentes permite acompanhar as opções de transportes dos usuários, tanto para identificar problemas quanto para projetar a capacidade e sistemas de transportes futuros. Para sistemas de informações pessoais, pode atuar fornecendo opções inteligentes de agendamento e deslocamentos para economia de tempo e recursos, em busca de deslocamentos mais inteligentes tanto individuais quanto de grupo. A fim de ilustrar esta aplicação, imagine um sistema capaz de guardar o histórico das movimentações de seu usuário, seria possível para o sistema fazer sugestões para otimizar tantos os seus compromissos quanto seus deslocamentos, de acordo com as preferências e os padrões de movimentação e também levando em consideração os padrões de trânsito da sua cidade. Para sistemas de gerenciamento de informações ambientais, permite calcular a pegada de carbono relacionada a movimentação de um usuário específico ou um determinado grupo de pessoas, como usado no projeto PEIR [24]. Para redes sem fio permite obter informações sobre o meio de transporte utilizado para planejamento de capacidade. Para redes DTN pode permitir uma proposta de roteamento ciente do modo transporte utilizado, como uma evolução do protocolo de roteamento que levam em consideração o histórico de movimentação de seus usuários como o NECTAR, proposto por OLIVEIRA e DE ALBUQUERQUE [21]. A velocidade de deslocamento pode influenciar no tamanho da janela de contato e o tipo do deslocamento pode acelerar a entrega das mensagens e influenciar nas oportunidades de contato. Para sistemas de monitoramento de segurança pessoal e auditoria, permite identificar padrões de locomoção atípicos e assim disparar notificações para que providências possam ser tomadas. Para sistemas de marketing permitem a entrega de propaganda voltada para o meio de transporte utilizado pelo usuário ou baseado no seu histórico de movimentação, como por exemplo, a oferta de bicicletas para pessoas que fazem deslocamentos curtos de ônibus ou a oferta de um leasing de automóvel para pessoas que se deslocam em maiores distâncias com mais de um meio de transporte. Além disso, para os sistemas de busca de informações, o modo de transporte utilizado pode ser usado como parâmetro de buscas customizadas na web, assim como a localização já é utilizada.

[3] Map My Ride (“http://www.mapmyride.com”)

2.2.3. Desafios

Entre os principais desafios na busca de uma solução para inferência do modo de transporte para uso de sistemas interativos cientes de contexto, podemos citar: (i) obter a classificação ainda em tempo de utilização; (ii) não consumir de forma significante os recursos limitados do celular; (iii) utilizar um mínimo de sensores; (iv) e ser capaz de escalar a solução para uma utilização em nível global. Quando este número de requisitos é inserido na busca por uma solução, estes afetam na forma com que o dado terá que ser coletado e representado, além disso limita o conjunto de informação que poderá ser extraído.

Considerando que a tarefa de classificação dos modos de movimentação torna-se mais "difícil" a cada modo de deslocamento adicionado; seja por que existe um grau de similaridade nos padrões de movimentação entre muitos deles, e que estes padrões podem ter tanto suas características quanto frequências influenciadas por características locais a cada cidade: condições do tráfego e do clima.

2.3. Hipótese

Considerando que o melhor sensor para obter a localização e os padrões de movimentação usando um smartphone é a fusão de sensores de localização. Considerando ainda que este tipo de sensor está presente na maioria dos smartphones modernos. Se for possível melhorar os resultados para classificação dos registros de movimentações, para determinar o meio de transporte utilizado pelo usuário de smartphone enquanto seu deslocamento ainda está acontecendo, é possível obter benefícios para as aplicações cientes de contexto e possibilitar novas aplicações que utilizem a informação do modo de transporte utilizado pelo usuário.

2.4. Solução proposta

O presente trabalho propõe revisitar os processos de inferência de modo de transporte já apresentados pelos trabalhos anteriores, na busca de oportunidades para melhorar os resultados obtidos pelos métodos anteriores. O objetivo principal é conseguir um método eficiente para classificar os deslocamentos baseados apenas no sensor de localização, para que possa ser usado para classificar os padrões de movimentação de acordo com o meio de transporte utilizado pelos usuários de smartphones. A abordagem

a ser utilizada consiste na obtenção de movimentação em uma área geográfica urbana delimitada e utilizar aparelhos de uma mesma plataforma com o mesmo tipo de sensor e o mesmo tipo de informação. Com isso, é esperado um aumento na qualidade da classificação e a possibilidade de abordar as etapas envolvidas no processo de mineração e coleta de dados para identificar possíveis alternativas e oportunidade de melhorias para o processo como um todo.

2.5. Trabalhos relacionados

Diversos trabalhos tratam da inferência de modo de transporte, muitos deles utilizam GPS, acelerômetro e outros sensores para atingir seu objetivo. Neste trabalho o foco é estrito em dados de localização, mais especificamente de smartphones modernos que possuem características intrínsecas, como: co-localização com seu usuário e o uso de fusão de sensores para localização. Dessa forma os trabalhos relacionados selecionados são aqueles que utilizam tipos de dados de localização/GPS.

Foi após o trabalho de ZHENG, LIU, et al. [37] que uma série de trabalhos surgiram, abordando a inferência do meio de transporte através de traces de GPS ou smartphones. No trabalho de ZHENG, LIU, et al. [37], os autores abordam um processo para classificar as movimentações dos usuários baseado na coleta de dados de GPS provenientes de telefones celulares, foram utilizados dados de 45 usuários coletados ao longo de seis meses. Foi proposto um mecanismo de segmentação de viagens baseada em limiares (de tempo e ausência de sinal) e na possibilidade de segmentar em mais uma etapa de acordo com pontos de mudança, tempo ou distância do deslocamento. Os atributos investigados foram velocidade, aceleração, mudança de direção e paradas. A aplicação do estudo não foi voltada para a identificação dos deslocamentos incompletos, apenas para os deslocamentos que já foram concluídos. Foram investigados a possibilidade de classificação de estruturas de subsegmentos de acordo com sua duração e a classificação do último segmento, através do uso de *Conditional Random Fields* (CRF) [46]. Neste trabalho é possível notar que o mecanismos de segmentação aplicados ainda não estão maduros, e que a forma de calcular a velocidade e distância estão sujeitas a erro. Em alguns casos, os erros nos cálculos de atributos como velocidade máxima e aceleração máxima, foram reduzidos com o uso da média entre três leituras.

No trabalho de ZHENG, LI, et al. [38] os autores também utilizam-se de dados de trajetórias completas de GPS para inferir modos de transporte por eles utilizados, embora houve um aumento dos dados coletados, que incluíram trajetórias completas de 65 usuários durante 10 meses. Para inferir o modo de transporte utilizado pelos usuários, foi um processo sequencial baseado em quatro etapas: segmentação, extração de características, inferência e pós processamento. O processo de segmentação proposto no trabalho anterior [37] foi detalhado, apresentando a segmentação dos traces em trajetórias em segmentos baseado em pontos de mudança [37] e [38], isto é, o ponto onde o tipo de meio de transporte muda. Os pontos de mudança são detectados, levando-se em consideração que entre a troca de meios de transportes deve haver uma pausa; mesmo que pequena e que entre meios de transporte, deve haver um de tipo andando. Pelo uso de árvores de decisão, ocorre a classificação dos segmentos em tipo andando e em tipo não-andando. A seguir os segmentos são classificados quanto ao modo de transporte: carro, ônibus, andando ou bicicleta, de acordo com as características de cada um em relação ao modelo de inferência com uso de árvores de decisão. Os autores consideram que quanto maior o segmento, mais rico serão as suas características; e que por consequência, aumenta a acurácia da classificação. Neste mesmo trabalho [37], uma etapa adicional de pós processamento avalia a mudança entre os modos de transporte, efetuando um ajuste que leva em consideração as probabilidades de mudanças de modo de transporte, pelo o uso de redes Bayesianas.

Enquanto no trabalho “Understanding mobility based on GPS data” [38], os autores mostram o processo de segmentação e na utilização de informações adicionais extraídas da própria característica da movimentação; neste trabalho [37] os autores focam na análise do uso de diferentes técnicas de divisão do segmento de viagem, fixa em tempo e fixa em distância.

Já no trabalho de ZHENG, CHEN, et al. [40], os autores apresentam em maiores detalhes o mecanismo de classificação baseado em pontos de mudança. Neste trabalho, também foi introduzido um modo de inferência, funcionando próximo ao tempo real. Este modo, sendo baseado no treinamento baseado nos dados históricos que são gerados pela método apresentado no trabalho anterior [37]. Possibilitando assim inferir os modos de transporte ainda em andamento.

O mecanismo proposto por ZHENG, CHEN, et al. nestes três trabalhos apresentados [37] [38] [40], analisa todas as movimentações para identificar agrupamentos de pontos

e definir pontos e mudança, dessa forma os deslocamentos podem ser indexados sequencialmente. Além disto, os trabalhos apresentam um mecanismo de detecção de modo de transporte que se utiliza de limiares de distância e tempo para classificar uma movimentação, em tipo andando e tipo não andando; se a movimentação não ultrapassar limiares de velocidade e aceleração é classificada como andando, caso contrário é classificada como não andando. E numa fase posterior do método proposto, aplica-se um mecanismo para ajustar erros de classificação de acordo com a classificação dos segmentos vizinhos ao atual. Nos três trabalhos foram utilizados logs de GPS, portanto, sem o uso de *sensor fusion* - o que pode gerar registros menos precisos pela falta de inferência por redes WiFi e de celular. Além disso, a utilização do mecanismo de identificação de pontos de mudança, caracteriza um sistema de informações geográficas [4](*Geographic Information System* ou GIS), onde os próprios usuários são usados para "popular" a base de dados. É uma abordagem útil para quando não houver um sistema de informações geográficas, de onde seja possível obter informações sobre os possíveis pontos de mudança. Assim, o uso dos pontos de mudança, por si só, caracteriza a utilização de uma abordagem composta de duas etapas, onde são empregadas técnicas para melhorar os resultados da classificação obtida inicialmente.

O trabalho apresentado por REDDY, ESTRIN et al. [24], trata-se de uma proposta para identificação de modo de transporte através de smartphones com uso de GPS e acelerômetro. O trabalho utilizou 6 usuários para a coleta de dados. O foco adotado foi a análise da variação de velocidade, pelo GPS. Com o uso do acelerômetro de forma complementar, permitiu identificar com um alto grau de precisão os deslocamentos (ente os modos: parado, andando, motorizado e bicicleta). Sendo que combinação dos sensores mostrou um aumento de 10% na precisão, em contraste com a utilização somente do GPS. A taxa de amostragem utilizada foi 1 Hz. O tempo para inferência inicial do meio de transporte foi de apenas 2 segundos transcorrido da coleta de dados. Os resultados obtidos foram bastantes positivos; embora o pequeno número de usuários, o tipo de coleta de dados utilizado e o tempo de coleta possam ter influenciado positivamente os resultados. Os autores puderam detectar uma variação mínima no grau de precisão da leitura de acordo com o local onde o usuário transporta seu telefone

[4] Sistemas de Informação Geográfica: (*Geographic Information Systems* - GIS). São sistemas que utilizam informações geo-espaciais.

(como por exemplo: na bolsa, no bolso, na mão ou no braço). Esta variação é mais relacionada ao uso do acelerômetro do que ao GPS em si. Os autores também analisaram diversos classificadores: *k-Nearest Neighbor*, *Decision-tree*, *Naive Bayes*, *Support Vector Machines*, *Continous Hidden Markov Models* (CH-MM) e um classificador em dois estágios compostos de *Decision Tree* associado a *Discrete Hidden Markov Models* - onde o segundo componente é aplicado como um filtro para eliminar os "ruídos" relativos a fronteiras de trocas de modo de transporte. Através deste último classificador (DT+DHM), os resultados da avaliação executada pelos autores apontaram para uma precisão de 99,8%. A análise e escolha do classificador foi feita com o uso da ferramenta Weka e posteriormente a implementação na linguagem Python com celulares N95 da Nokia. Os celulares usados neste estudo não utilizam fusão de sensores de localização, portanto os dados de velocidade utilizados são provenientes apenas do GPS e do acelerômetro, sendo que as leituras de GPS para velocidade tendem apresentam maior precisão do que as computadas pela distância e tempo entre os pontos.

No trabalho posterior de REDDY, MUN, et al. [25], o processo utilizado foi detalhado e número de usuários foi expandido de 6 para 16. Neste trabalho foi apresentada justificativa para escolha do GPS e do acelerômetro ao invés do uso de WiFi e GSM.

No trabalho de STENNETH, WOLFSON, et al. [29], os autores apresentam uma abordagem para detecção de meios de transporte utilizando informações de posicionamento de smartphones e informações geográficas diversas, como: localização dos pontos de ônibus, localização de linhas ferroviárias e informação do posicionamento dos ônibus em tempo real. Nessa proposta os autores exploraram as informações que estão disponíveis na rede e que é provida pelo sistema de transportes da cidade de Chicago, para assim viabilizar a entrega da informação de modo de transporte em tempo de utilização com melhor precisão.

Diferente dos demais estudos, neste estudo [29], a taxa de amostragem para o posicionamento utilizada foi de 30 segundos. Foram utilizados três tipos de dispositivos: iPhone 3, Android e HP IPAQ. Foram coletados dados de movimentação de seis usuários durante três semanas dentro da cidade de Chicago, Illinois, EUA. Para a detecção do modo de transporte próximo ao tempo real, os autores utilizaram uma janela móvel de 1 minuto (equivalente a dois registros de posicionamento), onde utilizam os seguintes atributos: velocidade média, aceleração média, média de mudança

de direção. Para caracterização de modo de transporte por trem, foi levado em consideração que a disponibilidade de sinal do GPS seria impactado pela ausência de sinal, gerando leituras com a variável de acurácia vertical [5]alta.

Quanto a redução do consumo de energia com espaçamento de trinta segundos entre as leituras, os autores alegaram economia significativa de bateria para justificar a redução da frequência de amostragem. Embora esta conclusão contraste com a sugerida por THIAGARANJAN [3].

O uso de uma janela de 1 minuto, com apenas dois registros pode não ser representativo o suficiente para caracterizar o modo de transporte quando condições de transito e deslocamentos não motorizados forem os mais variados, valores médios com duas medidas pode ser tendenciosos e podem apresentar uma variância alta entre leituras do mesmo modo de transporte. Esta linha de raciocínio, se deriva das conclusões apresentadas por ZENG e LIU [37], onde os autores afirmam que quanto maior o segmento mais rico suas características.

O número de usuários e tempo de coleta foram mais curtos que os utilizados por ZHENG ao longos dos trabalhos aqui referenciados [37] [38] [39] [40].

A disponibilidade de informação acerca das posições de ônibus pode não estar disponível para a maioria das cidades, seja por que o sistema é operado por uma empresa privada ou por falta de interesse do setor público de disponibilizar esta informação.

A utilização de informação de sistemas GIS apontou para um aumento de 9% na precisão da detecção. Os autores analisaram diversos classificadores: Redes Bayesianas, árvores de decisão, *random forest, naive bayes* e *multilayer perceptron*. Sendo que o melhor resultado foi obtido com o uso de *random forest*, 75.4 sem GIS e 93.7 com uso das informações geográficas.

A Tabela 2.1 apresenta um resumo dos resultados para inferência de modos de transporte dos trabalhos relacionados. Cabe ressaltar que estes resultados são bastante resumidos, e não são adequados para comparar o resultado de cada um. Existem

[5] Especificamente para o caso dos dispositivos iPhone da Apple, um alto valor para o atributo *horizontal ou vertical accuracy*, significa menor acurácia, enquanto que o valor -1 denota que a leitura não pode ser obtida.

diferenças importantes na metodologia, nos dados e até no número de modos classificados.

Tabela 2.1 - Resumo dos resultados para detecção de modos de transporte dos trabalhos relacionados.

Trabalho	Classes	Dados		Dados coletados	Precisão	Classificadores
		# Usuários	*Duração Coleta*			
[38]	andar, bicicleta, carro, ônibus.	65 pessoas	10 meses	Projeto GeoLife. GPS	72,8%	Decsion Tree
[38]	andar, bicicleta, carro, ônibus.	65 pessoas	10 meses	Projeto GeoLife. GPS	76,2%	Decsion Tree + pós-processamento
[24]	andar, correr, carro,	6 pessoas	20 horas	Projeto PEIR GPS, Acelerômetro	98,8%	DT + Discrete Hidden Markov Models
[29] com GIS	andar, bicicleta, carro, trem	6 pessoas	3 semanas	GPS, WiFi, 3G, GIS, posição de ônibus.	93,5%	Random Forest
[29] sem GIS	andar, bicicleta, carro, trem	6 pessoas	3 semanas	GPS, WiFi, 3G,	75.4%	Random Forest

CAPÍTULO III - Conceitos, Métodos e Técnicas Aplicados

Para alcançar os objetivos deste trabalho, foi necessário atuar em etapas, que incluíram: aquisição do conhecimento do domínio; planejamento da execução das etapas de pesquisa; coleta de dados; análise dos dados e por análise dos resultados. Para cada uma destas etapas, foram aplicados métodos específicos.

Este Capítulo tem o objetivo de apresentar os principais conceitos, métodos e técnicas empregados ao longo deste trabalho. Desta forma, a Seção 3.1 apresenta os principais conceitos relacionados, começando com os conceitos relacionados a sensoriamento por smartphones, passando pelo processamento de trajetórias e descoberta de conhecimento em banco de dados. A Seção 3.2 apresenta os principais métodos e técnicas aplicados.

3.1. Conceitos relacionados

Esta seção tem o objetivo de apresentar os principais conceitos relacionados com o trabalho aqui apresentado, visando assim subsistir o entendimento deste trabalho e do campo de pesquisa no qual este se insere.

3.1.1. Sensoriamento por smartphones

Um sensor é um dispositivo que transforma um evento físico em sinais elétricos, funcionando como uma interface entre o mundo real e o mundo dos dispositivos eletrônicos, conforme apresentado por KENNY em [34], página 2. Entre as diversas características intrínsecas aos sensores, algumas são de particular importância para este trabalho, são elas:

(i) sensibilidade: é a taxa de mudança reportada no sinal elétrico de saída de um sensor em resposta a uma mudança no sinal físico;

(ii) resolução: é a mínima flutuação de sinal que é detectada pelo sensor;

(iii) acurácia: é dada pelo maior erro em relação ao sinal de saída de um sensor, relativo a uma determinada flutuação de sinal;

(iv) ruído: todo sensor retorna um sinal de ruído junto com o sinal de resultado esperado, estes podem ter diversas origens relacionadas e geralmente são distribuídos de forma homogênea ao longo do espectro.

Dentre os diversos sensores presentes em um smartphone, muitos podem ser utilizados para determinar a localização do dispositivo: os sensores de redes, o GPS, a câmera de vídeo e até mesmo o microfone. Quando utilizado sozinho, segundo REDDY, MUN, et al. [25], o GPS é o sensor de localização de melhor resultado. Este se utiliza da diferença de tempo da chegada do sinal de múltiplos satélites geo-estacionários para calcular a posição do smartphone, possuindo uma precisão de alguns poucos metros na condição de disponibilidade de sinal de vários satélites (para determinar a localização correta com o melhor grau de precisão possível, é necessário o sinal de 5 satélites). Já os sensores de redes WiFi e celular, que utilizam estações base (no caso das redes WiFi, é o *access point*); a posição é determinada através do acesso a uma base de dados remota com informações das redes e dos níveis de sinais associadas a uma determinada localização (caracterizando um *fingerprint* em uma localização). Além destes, há os sensores inerciais MEMS (*micro-electro-mechanical sensors*), como: os giroscópios, bússolas magnéticas e acelerômetros, que também podem ser usados para localização, desde que se tenha uma localização original como referência. Assim, os sensores MEMS registram sinais de movimentação, a mudança na posição e na orientação e podem ser usados para estimar a localização. Os principais problemas com a utilização desses sensores é que não existe um sensor que funcione em todos os locais com um bom grau de precisão e com um baixo consumo de energia, conforme apresentado por THIAGARANJAN [3].

A técnica de fusão de sensores (*Sensor Fusion*), segundo ELMENREICH [10], consiste na combinação de dados de múltiplos sensores, de forma que a leitura resultante seja melhor que a leitura oferecida por apenas um dos sensores utilizado de forma isolada. Esta técnica permite a utilização de uma combinação de sensores heterogêneos, múltiplos sensores homogêneos e até mesmo o uso de informação históricas de um único sensor. É um importante recurso para sistemas cientes de contexto, pois permite

não só a melhoria de dados de sensores, mas como a criação de sensores virtuais que fornecem informações a partir de uma gama de sensores combinados.

Dentre inúmeras novas aplicações para smartphones, uma categoria tem recebido bastante atenção dos pesquisadores, são os sistemas que fazem sensoriamento baseado no ser-humano (ou human-centric sensing), conforme apresentado no trabalho de CAMPBELL, LANE, et al. [44]. Em contraste com as técnicas de sensoriamento remoto, que se utilizam de imagens de satélite para obter informações de locais distantes sem a necessidade de lá estarem. As técnicas de sensoriamento baseadas no ser humano se valem tanto dos sensores embarcados nos celulares, quanto da conectividade e do próprio usuário para coletarem preciosas informações sob a ótica do próprio ser humano. Sendo este o responsável por coletar a informação no local e enviá-la para posterior processamento.

O sensoriamento baseado no ser-humano, pode ser classificado em duas subcategorias, conforme apresentado por LANE, MILUZZO, et al. [45]: o sensoriamento participativo onde o usuário tem um papel ativo na coleta da informação, isto é cabe a ele ir ao local e coletar a informação de forma adequada aos procedimentos previamente definidos e o sensoriamento oportunista, que se vale do smartphone para coletar as informações sem que uma participação do usuário seja necessária. Assim a coleta do sensoriamento oportunista se dá de forma constante ou em resposta a um determinado evento, como por exemplo, quanto o usuário passar por determinada localidade ou num determinado horário ou outras condições obtidas pela leitura dos diversos sensores do equipamento. Já no sensoriamento participativo, o participante é o foco central, pois este é parte ativa do processo de coleta de dados e pode influenciar nos resultados de coleta tanto de forma positiva quanto negativa.

3.1.2. Sistemas baseados em localização

Sistemas baseados em localização (*location based systems*), são aqueles sistemas que utilizam a informação de localização do usuário, apresentado por AHSON e ILYAS [43]. Nesta se categoria, podem se enquadrar: os sistemas de buscas, os sistemas de navegação, os sistemas que usam realidade aumentada, entre outros. Tais sistemas podem ser considerados, mesmo que de forma limitada, sistemas cientes de contexto, uma vez que a localização também faz parte do contexto, como apresentado no trabalho de BAULDALF, DUSTDAR, et al. [4].

3.1.3. Framework Core Location

Trata-se de um *framework* (um componente de software) do sistema operacional iOS da Apple, conforme apresentado na documentação para desenvolvedores iOS [2]. Este componente permite a criação de sistemas baseados em localização pela utilização dos sensores do smartphone. A classe *CLLocationManager*, que é parte do *framework Core Location*, é a classe que faz a interface para o acesso aos sensores do dispositivo. O seu atributo *desiredAccuracy* define o grau de acurácia desejada para a medição (existem cinco possibilidades). Este atributo é diretamente relacionado ao consumo de energia do dispositivo, uma vez que influencia nos número de sensores e na frequência com que são lidos. A classe *CLLocationManager* deve ser acessada através de uma classe delegada, que poderá obter as notificações de mudança de posicionamento pelo método: *didUpdateToLocation: newlocation fromLocation: oldlocation*. Este método gera um registro de posição [6] (ou registro de posicionamento) que poderá ser usado pela aplicação.

3.1.4. Processamento de trajetórias

Quando se trabalha com informações geográficas, deve-se ter em mente que existem algumas particularidades a elas associadas, como por exemplo, o fato de que os dados são relacionados à áreas, pontos e linhas em um espaço geodésico. Já, quando se trabalha especificamente com trajetórias, existem preocupações adicionais relacionadas a dimensão do tempo. Assim, uma trajetória é frequentemente representada como um conjunto de pontos espaciais, ordenados de acordo com o tempo, onde fatores como, granularidade e interpolação precisam ser levados em conta.

Conforme apresentado por ALMEIDA, PIRES et al. [1], uma trajetória consiste em uma sequência de deslocamentos e paradas, onde é possível associar esta sequência a um determinado objetivo. Sendo que este, caracteriza-se por um ponto de origem e um ponto de destino específicos. Os pontos de parada, intermediários, podem ou não ser

[6] Registro de posição: um registro de posição equivale a uma upla (linha) de uma tabela, definindo uma leitura de um posicionamento em um determinado instante. No caso deste estudo, apresenta os seguintes atributos (colunas): identificação do dispositivo, *timestamp*, latitude, longitude, altitude, acurácia, horizontal, acurácia vertical, velocidade e curso.

considerados pontos de interesse. Pontos de interesse ocorrem de acordo com o objetivo do estudo e podem caracterizar-se por uma localidade visitada de forma recorrente pelo objeto móvel (ou usuário).

De acordo com o trabalho de ZHENG, CHEN, et al. [40], cada deslocamento da trajetória pode ser associado a um modo de transporte, sendo que entre cada deslocamento onde existe uma mudança de meio de transporte, existe um deslocamento onde o meio de transporte é “andando a pé”, mesmo que ele seja muito discreto. Para melhor compreensão, a Figura 3.1, apresenta a representação de uma trajetória hipotética de um possível indivíduo a caminho de seu trabalho usando múltiplos meios de transporte.

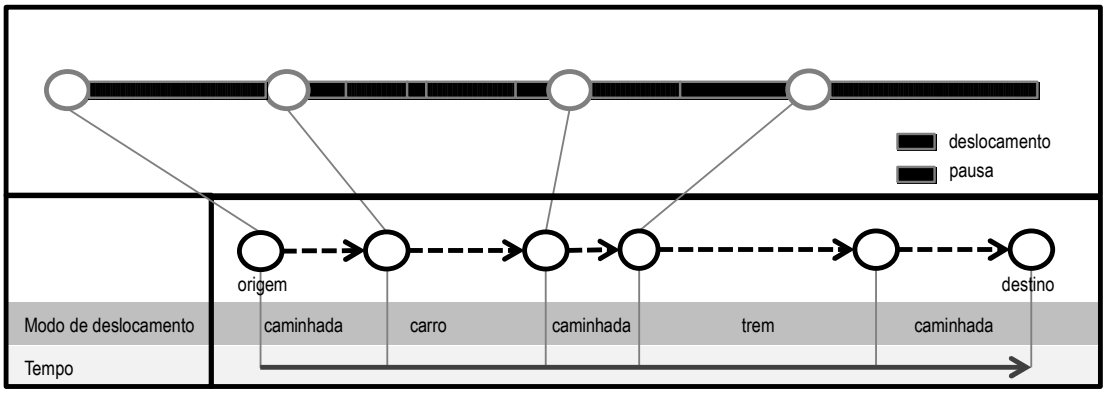


Figura 3.1 - Deslocamento hipotético de uma pessoa da sua casa até o seu trabalho.

A representação de uma trajetória pode se dar de diferentes formas. Assim, a definição da forma a ser utilizada, deve levar em conta quais aplicações deverão usar esta representação de trajetória. Entre as formas mais utilizadas pode-se destacar: (i) uma sequência temporal de pontos, onde nesse caso, para se obter a trajetória é necessário uma interpolação, para que os pontos se transformem em linhas (Figura 3.2), (ii) como uma sequência de vetores; (iii) como um conjunto de diretivas semânticas; sendo também possível combinar diferentes formas a fim de enriquecer a representação.

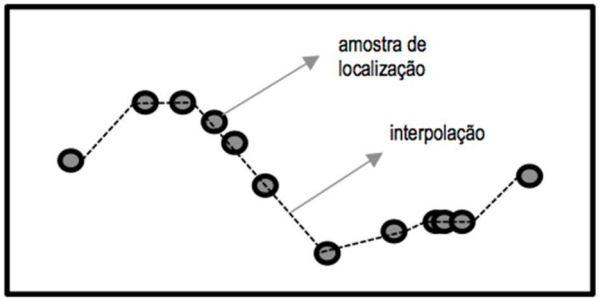


Figura 3.2 - Exemplo de interpolação baseada em amostras de localização.

O processo de estimar uma trajetória através de smartphones, conforme apresentado por THIAGARANJAN [3], consiste em um problema difícil, pois como citado anteriormente, não há um sensor embarcado que seja preciso em todas as localidades.

O GPS é a tecnologia mais utilizada para localização, mas possui duas grandes limitações: gera um gasto considerável de energia e não funciona em ambientes fechados ou onde haja barreira para o sinal do satélite. Entre as formas mais comuns para contornar o problema do gasto de energia do GPS, está a redução do número de amostragens, o uso de tecnologias de localização alternativas ao GPS que apresentem um gasto menor de energia (como por exemplo: estimar a localização pelo uso de *fingerprints* de redes WiFi ou da rede de telefonia celular; técnicas geométricas de localização e uso de sensores de inércia – como o acelerômetro ou o giroscópio). Os sensores alternativos (WiFi e os MEMS) geralmente funcionam onde o GPS não funciona bem, em ambientes fechados. Portanto, o desafio consiste em transformar leituras de localização, infrequentes e imprecisas, em uma de trajetória precisa.

3.1.5. Descoberta de Conhecimento de Banco de Dados

Conforme apresentado por MAIMON e ROKASH [42], este termo define o processo organizado cujo o objetivo é descobrir conhecimento útil em grandes e complexas massas de dados. Assim, segundo FAYYAD, PIATETSKY-SHAPIRO, et al. [12], este processo se caracteriza pelas seguintes etapas: seleção, tratamento, mineração de dados, descoberta de conhecimento e aplicação do conhecimento. Junto com a apresentação deste conceito, cabe tanto a das etapas do processo de descoberta de conhecimento em banco de dados, quanto de técnicas frequentemente empregadas.

Função de agregação

Segundo VEGA LÓPES, SNODGRASS, et al. [31], uma função de agregação é aquela que recebe um conjunto de enuplas como entrada e retorna uma upla, que resume os valores do conjunto. Neste trabalho, consideramos como uma função de agregação, uma função que é capaz de resumir um conjunto de tuplas através de uma única tupla. Conforme apresentado na Figura 3.3.

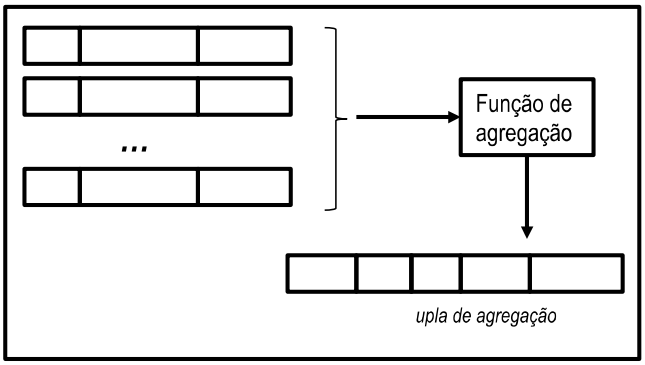


Figura 3.3 - Esquema representando uma função de agregação.

Dimensionalidade de dados

Segundo TAN, STEINBACH, et al. [30], um conjunto de dados pode apresentar um grande número de características (também chamados de atributos ou dimensões). Existem benefícios na redução da dimensionalidade dos dados, entre eles: diminuição do ruído, obtenção de um modelo mais compreensível e de um modelo de melhor performance. Isso pode ser conseguido combinando vários atributos em um novo atributo. No caso específico das técnicas de classificação, pode não haver objetos de dados suficientes para permitir a criação de um modelo que atribua de forma confiável rótulos a todos os dados possíveis. No caso de técnicas de agrupamento (*clustering*), as definições de densidade e distância entre pontos tornam-se menos significativas. Com isso, a performance desses algoritmos fica reduzida.

O processo conhecido como redução da dimensionalidade, visa, entre outros objetivos, evitar o problema da maldição da multidimensionalidade[7], conforme descrito por TAN,

[7] Maldição da multidimensionalidade: refere-se a um fenômeno onde a análise de dados torna-se mais difícil quando a dimensionalidade dos dados aumenta. Isto ocorre por que os dados ficam mais dispersos no espaço, prejudicando a aplicação de técnicas de classificação.

STEINBACH, et al. [30]. Nesse processo, busca-se manter os atributos mais relevantes e evitar a dispersão dos dados que pode ocorrer quando o número de atributos é grande.

Seleção de atributos (ou características)

Trata-se do processo no qual busca-se remover os atributos pouco relevantes para a análise, seja para reduzir a dimensionalidade ou para torná-la mais eficiente [30]. Isto pode ser feito por diferentes técnicas, desde a análise visual a utilização de algoritmos capazes de elencar os atributos mais relevantes. A seleção de atributos, baseia-se na ideia de que em meio ao dado existem muitas características que são redundantes para o processo de classificação. Características redundantes são aquelas que não produzem mais informação para a classificação, do que a característica já selecionada.

Mineração de dados (*data mining*)

De acordo com a definição apresentada por WITTEN, EIBE, et al. [35]:

"Data Mining é o processo utilizado para descobrir padrões em dados. O processos deve ser automático (ou mais usualmente semiautomático). Os padrões descobertos devem possuir sentido para que se traduzam em alguma vantagem, que geralmente é econômica. Para isso, os dados precisam estar disponíveis em uma quantidade significativa."

Com isto, consiste na aplicação de técnicas de processamento para a descoberta de padrões úteis, que de outra forma poderiam permanecer ignorados e obter a capacidade de previsão de resultados de uma observação futura. A mineração de dados (também conhecida como análise preditiva) é parte da área de descoberta de conhecimento em banco de dados. Esta por sua vez, caracteriza-se pelo processo de conversão de dados brutos em informações úteis e, por consequência, é composto de uma série de passos que tipicamente também incluem: pré-processamento, mineração de dados e pós processamento.

Aprendizado de máquina

Trata-se de uma área da inteligência artificial cujo objetivo é o estudo de algoritmos capazes de "aprender", conforme apresentado por MITCHELL [41]. Em seu cerne, lida com técnicas que são aplicadas para descrever padrões estruturais em dados; com os dados assumindo a forma de um conjunto de exemplos (ou instâncias). Trata-se de uma

ferramenta que pode ser utilizada tanto para entender os dados, quanto para fazer predições sobre eles.

De acordo com TAN, STEINBACH, et al. [30], conforme apresentado na Figura 3.4, um classificador é um sistema que recebe de entrada um vetor de dados discretos ou contínuos e entrega um único valor discreto como saída.

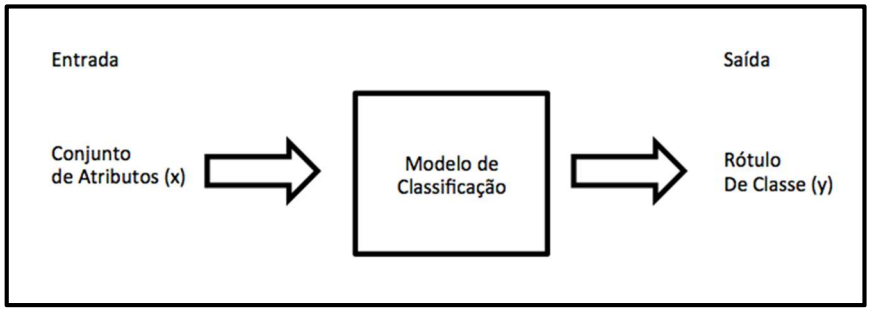


Figura 3.4 - Diagrama representando um classificador genérico.

É uma técnica de aprendizado de máquina, que tipicamente lida com a generalização, onde o objetivo é encontrar uma função aproximada (também chamada de função *proxy*) de uma função real desconhecida. Esta função aproximada é definida pelos dados de entrada, utilizados na etapa de treinamento.

Este trabalho, utiliza de um conjunto de técnicas de aprendizado de máquina, que foram selecionados tanto levando em conta trabalhos anteriores quanto de uma lista dos mais utilizados [36]. Assim os algoritmos selecionados neste estudo são:

Perceptron Multicamadas: o conceito de *perceptron* vem da área de neuro-computação, conforme apresentado por RUSSEL e NORVIG [19]. O *perceptron*, se baseia no funcionamento de um neurônio, propondo uma estrutura de processamento que a partir de um conjunto de entradas, onde são aplicados pesos, “disparam” um resultado quando um determinado limiar é alcançado. Um conjunto de *perceptrons* (ou unidades) podem ser combinados em uma rede neural. Assim, quando se utiliza múltiplas camadas de *perceptrons*, temos uma rede multicamadas. As redes neurais estão entre os algoritmos mais populares [19] e, entre suas características mais importantes estão a alta tolerância a ruídos para os dados de entrada e capacidade de computação distribuída.

Árvore de Decisão: Árvores de decisão, apesar de serem técnicas simples, também fazem parte do conjunto de técnicas mais bem sucedidas de aprendizado de máquina [19]. Trata-se de uma técnica de inteligência artificial usada para predição, que

funciona pela criação de uma árvore que mapeia as observações a uma determinada classe, de acordo com os valores apresentados pelos seus atributos. As folhas da árvore são as classes possíveis e os galhos são desvios condicionais de acordo com os atributos apresentados.

Redes Bayesianas: (bayes-net) Consiste em uma técnica de classificação que se baseia na possibilidade de construção de grafos, estes que representam as relações probabilísticas de um conjunto de variáveis aleatórias e suas interdependências. Este,

é baseado no teorema de Bayes, onde cada nodo do grafo representa uma variável aleatória, cujas as probabilidades de ocorrência são calculadas com base nas frequências apresentadas. Esta tecnologia é apresentada em mais detalhes por RUSSEL e NORVIG [19]. Ainda quando se refere a redes bayesianas, o classificador *nayve bayes*, trabalha de forma similar, embora considerando uma forte independência de premissas. Este, também está entre um dos mais populares, segundo por RUSSEL e NORVIG [19].

SVM: Trata-se de uma técnica que utiliza um ou mais de um hiperplanos para separar os dados [15]. É um dos modelos mais populares, especialmente usado quando não se tem muito conhecimento do domínio dos dados classificados [19]. Trabalha com a classificação de dados representados como pontos no espaço, de forma que os exemplos de cada categoria sejam separados por um espaço tão amplo quanto possível. Quando isto não é possível, exemplos podem ser mapeados no mesmo espaço, mas a categoria é definida de acordo com o lado do espaço onde o exemplo se apresenta.

K-NN: (*K-nearest neighbors algorithm*) Trata-se de um algoritmo para reconhecimento de padrões não paramétrico. É um dos algoritmos de aprendizado de máquina mais simples e se baseia na avaliação da distância dos atributos em relação aos exemplos de treinamento.

Existem diferentes métodos e técnicas que podem ser utilizadas para melhorar a performance dos classificadores. Na impossibilidade de abordar todas, neste trabalho, abordamos o método *ensemble* [30] para combinar classificadores. Este pode ser usado tanto para melhorar a robustez quanto a performance da classificação. Funciona pela estratégia de “dividir para conquistar” e tem a característica de ser mais resistente a ruído e apresentar melhor precisão. Dentro deste método, três técnicas distintas podem ser aplicadas: (i) bagging; (ii) boosting; (ii) stacking.

O termo *bagging* é derivado de *bootstrap aggregation*, consiste na geração de dados adicionais através da repetição instâncias dos próprios dados originais de entrada. Assim, procura-se melhorar o resultado das classificações instáveis.

Já quanto ao *boosting*, definindo de forma simplificada, é uma técnica que pode melhorar a performance de classificadores fracos (quando não se consegue uma precisão muita alta para a classificação, embora melhor que a classificação aleatória). Neste caso, a classificação utiliza dois (ou mais) algoritmos de classificação, sendo que na primeira classificação atribui-se pesos diferentes aos exemplos que foram corretamente classificados e aos que foram classificados de forma errada. Na segunda classificação, usando um classificador forte, os pesos são redistribuídos, também de acordo com os exemplos que foram corretamente classificados.

A utilização de métodos de combinação de algoritmos deve ser usada com cautela, da mesma forma que existem benefícios em sua utilização, muitas vezes pode ocorrer de que as fragilidades sejam acumuladas, onde a utilização de dois algoritmos piorem a classificação ou causem *overfitting*[8].

Os seguintes algoritmos fazem parte do hall de algoritmos que utilizam técnicas *ensemble*:

Random Forest: Trata-se de um algoritmo baseado *ensemble learning*. Usando a técnica de *bagging*, opera pela construção de várias árvores de decisão ainda em tempo de treinamento. Maiores detalhes podem ser obtidos em [6].

Random Tree: Opera pela construção de árvores com K atributos escolhidos de forma aleatória para cada nó [35].

AdaBoost: O *adaboost* é um algoritmo muito popular para aplicar a técnica de *boosting*. Neste caso, ocorre a combinação de diferentes algoritmos para criar um novo conjunto de dados. No *adaboot*, o primeiro algoritmo é chamado *base learner*, já o segundo é chamado *stacking model learner*. É suscetível a ruídos, mas não é muito suscetível a *overfitting*.

---

[8] *Overfitting*: refere-se a um algoritmo de aprendizado de máquina que é ajustado ao extremo para um determinado conjunto de dados, de forma que ele apresente uma boa performance para um conjunto de dados específico e tenha baixa performance com dados de exemplos externos ao conjunto de treinamento.

Tratamentos para amostras não balanceadas

Durante a etapa de mineração de dados, especialmente para o caso de classificadores, com frequência, é possível se deparar com amostras não balanceadas, ou seja o número de ocorrências de instâncias de cada classe varia fortemente. Caso este seja o caso para os dados em estudo, é possível que possa haver problemas na utilização de aprendizado de máquina, uma vez que a classe de ocorrência majoritária pode ter excesso de influência sobre o modelo gerado. Em certos casos, não é interessante modificar esta ocorrência, pois pode descaracterizar o modelo, mas de forma geral existem várias formas de lidar com este problema. Segundo DAL POZOLLO [8]: (i) a técnica de *undersampling*, consiste em reduzir artificialmente a ocorrência das amostras de classe majoritária; (ii) a técnica de *oversampling*, consiste em repetir as amostras das classes minoritárias de forma aleatória (neste caso, aumentando o risco da ocorrência de *overfitting*), pode também ser feito gerando instâncias sintéticas.

SMOTE (*Synthetic Minority Over-sampling Technique*) é uma técnica de *oversampling* onde se criam amostras sintéticas, que são interpoladas nas proximidades (*neighborhood*) das ocorrências dos exemplos das classes minoritárias, o que gera um agrupamento próximo a ocorrência das classes minoritárias.

3.2. Principais métodos e técnicas aplicados

Esta Seção tem o objetivo de apresentar os principais métodos e técnicas aplicados ao longo deste trabalho.

3.2.1. Método aplicado para segmentação de trajetórias

A associação de registros de posicionamento à trajetórias é uma prática comum em banco de dados de trajetórias, como pode ser observado nos diversos trabalhos referenciados ao longo do texto [31] [17] [1] [39] [40]. Dado que uma trajetória consiste no deslocamento de um objeto móvel com um objetivo definido por sua origem e destino. Uma trajetória, pode ser composta de períodos de deslocamento e de pausas intercalados até que o destino seja atingido. Baseado no trabalho de ZHENG, CHEN, et al. [40], pode se considerar que entre cada mudança de modo de transporte, possivelmente existe um período de pausa entre eles, mesmo que discreto. Além disso, ZHENG, CHEN, et al. [40], também demonstram que quando ocorre uma mudança de

modo no transporte, existe uma alternância de modos de transporte, entre o deslocamento andando e o deslocamento não-andando (*walking* e *non-walking*).

No presente trabalho, também foi utilizada uma etapa de processamento aplicada após a coleta de dados para o agrupamento dos diversos registros de posicionamento em trajetórias. Tal etapa visa identificar e associar um grupo de registros de posicionamento à uma trajetória específica (com origem e destino definidos). Então, após esta etapa cada trajetória passa ter seus períodos de deslocamentos e pausas identificados, com seus respectivos registros de posicionamento associados a um único determinado deslocamento ou uma única pausa. Esta etapa está fortemente baseada no trabalho de IDRISOV e NASCIMENTO [17], onde os autores apresentam um framework para agrupamento de registros de posicionamento. Este framework por usa vez, se caracteriza por três etapas: (i) detecção de paradas; (ii) interpolação de segmentos faltantes; (iii) remoção de registros de baixa acurácia.

3.2.2. Métodos aplicados para descoberta de conhecimento em banco de dados

Para a etapa de descoberta de conhecimento em banco de dados, utilizou-se um abordagem baseada no processo KDD (*knowledge discovery in databases*), conforme apresentado por FAYYAD, PIATETSKY-SHAPIRO, et al. [12]. Apesar deste modelo não ser o mais utilizado no mercado, ele é mais simples e voltado diretamente ao processo de mineração de dados em si. Em contrapartida, o mais utilizado no mercado, é o CRISP-DM [11]; cujo o foco é voltado ao processo de gerenciamento das múltiplas etapas subsequentes para atingir o objetivo de aquisição e aplicação do conhecimento e por consequência passa a ser mais complexo que o proposto por FAYYAD, PIATETSKY-SHAPIRO, et al.

O método apresentado por FAYYAD, PIATETSKY-SHAPIRO, et al. [12] propõe uma série de atividades, que se caracterizam como um processo. Estas atividades são aplicadas de forma serial, mas com o uso de iterações sucessivas até que o processo seja completado. A seguir apresentamos uma descrição de cada uma das etapas que compõem o método proposto e quais os métodos empregados em cada uma delas, sendo que Figura 3.5 apresenta o encadeamento das atividades que compõem o processo KDD.

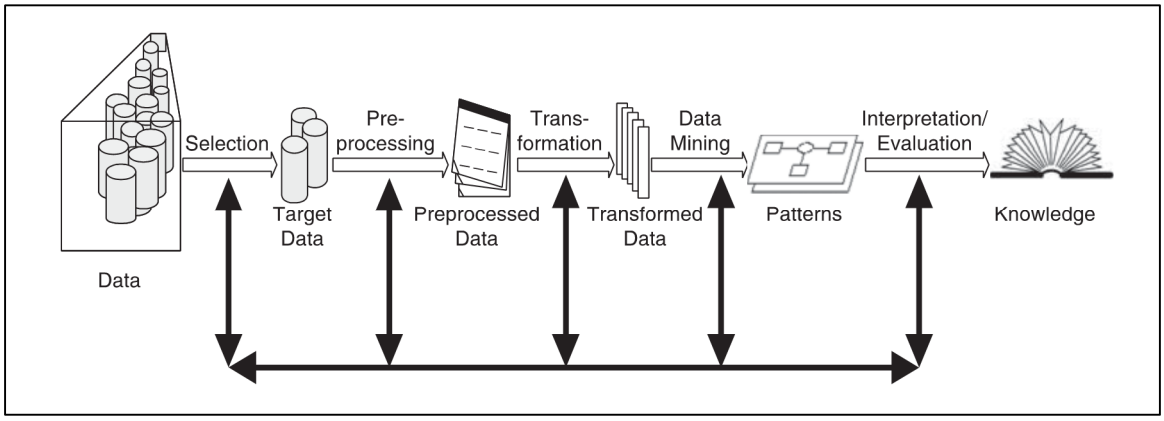


Figura 3.5 - Conjunto de atividade em alto nível do processo KDD (fonte: [11])

Pesquisas de mineração de dados têm característica de serem multidisciplinares, com frequência englobam áreas como matemática, estatística, física, computação e ciências sociais [11]. Assim, a primeira etapa consiste na aquisição do conhecimento do domínio. Este trabalho não foi diferente, precisou de uma longa etapa inicial de pesquisa e revisão bibliográfica (apresentada no Capítulo II) para obtenção de conhecimento do domínio de aplicação. Este domínio inclui: técnicas de localização em dispositivos móveis, padrões de mobilidade humana, técnicas de representação de trajetórias e sistemas cientes de contexto e sistemas de informação geográfica.

3.2.3. Técnicas aplicadas para avaliação de classificadores

Entre as técnicas de avaliação de classificadores, este estudo utilizou-se da técnica de validação cruzada (*cross-validation*), conforme apresentada em [30] e [35], mais especificamente foi utilizado o método *k-fold* (com k=10). Esta técnica foi selecionada por ser um método bem conhecido e aceito, por ser usado nos trabalhos de ZHENG e STENNETH, WOLFSON, et al. e por ser compatível com a ferramenta Weka [9](utilizada para esta etapa do estudo). Sua utilização se deu de forma análoga ao que foi apresentado nos referidos trabalhos [37] [29]. A técnica de *cross-validation* 10-fold, consiste na divisão dos dados em 10 subconjuntos aleatórios, sem repetição e de

[9] Weka: (*Waikato Environment for Knowledge Analysis*) trata-se de um conjunto de ferramentas e algoritmos para análise de dados e predição, preparação do dado, incluindo a seleção de atributos. Desenvolvido através de código aberto pela Universidade de Waikato na Nova Zelândia. Esta ferramenta está disponível em: "http://www.cs.waikato.ac.nz/ml/weka/"

tamanhos iguais. O classificador, então é avaliado com base na média dos resultados de 10 iterações; onde na primeira iteração, os dados dos subconjuntos 1 a 9 são usados para gerar o classificador e o subconjunto 10 para avaliá-lo; na segunda iteração, os subconjuntos de 1 até 8 e o sub-conjunto10 são usados para construir o classificador e o 9 para testá-lo; assim por diante até completar as 10 iterações.

3.2.4. Métricas utilizadas para a avaliação do desempenho de classificação

Os termos *precision accuracy*[10] (acurácia de precisão) e *recall accuracy* (acurácia de sensibilidade), apresentados por SOKOLOVA, JAPKOWICZ, et al. [27] e por TAN, STEINBACH, et al. [30], referem-se a métricas para medição de desempenho do classificadores onde existe ocorrência de classes não balanceadas, o que é muito comum em dados da vida real.

Como exemplo, é possível apresentar o caso de um classificador para uma classe A que seja de rara ocorrência, caso o classificador classifique todos as ocorrências como a classe A, terá acertado com uma precisão de 100%. Assim *precision accuracy* determina a fração de dos registros que realmente acabaram sendo positivos no grupo que o classificador declarou como positivo.

Já, *recall accuracy* mede a fração do total de exemplos positivos corretamente classificados pelo classificador. Um *recall accuracy* alto, reflete que de todas as ocorrências para uma determinada classe, poucas foram as que deixaram de ser classificadas [19].

Construir um modelo que tenha *recall e precision accuracy* elevados é a chave para uma boa classificação. *Precision* e *recall* podem ser resumidos pela F-measure (ou métrica f1), esta representa a média harmônica entre *recall accuracy* e *precision accuracy*. A seguir as fórmulas para calcular cada um deles é apresentada:

$$Precision = \frac{TP}{TP + FN}$$

[10] Foi uma opção do autor utilizar o termo em inglês, uma vez que existem diferentes traduções para o português, e nenhuma delas mostrou-se o padrão de fato. Assim utilizando o termo conforme referências originais, evitou-se a possibilidade de agregar ambiguidade ou mesmo confusão para o texto.

$$Recall = \frac{TP}{TP + FP}$$

$$F1 \ = 2 \times \frac{precision \ \times \ recall}{precision \ + \ recall}$$

Onde:

TP = positivos verdadeiros (*true positives)*
FP = falso positivos (*false positives)*
FN= falso negativos (*false negatives)*
TN = negativos verdadeiros (*true negatives*)

3.2.5. Técnicas aplicadas para segurança da informação

Quanto ao gerenciamento da segurança da informação, cabe ressaltar que informações de movimentação de usuários, são de caráter privado, de acordo com os requisitos legais, precisam ser protegidas de forma adequada. Para isso, na perspectiva de infraestrutura, utilizamos a abordagem de segurança em camadas, com os servidores utilizando firewall de host e autenticação para acesso aos dados.

Os usuários participaram do processo foram voluntários e concordaram em participar da pesquisa. Cada um deles, participou de um breve treinamento, que ensinou-os a utilizar corretamente a aplicação CityTracks. Com isso, o momento de coleta de informações aconteceu por iniciativa dos próprios usuários (para envio das informações, é necessário que o sistema fosse ativado pelo próprio usuário; sendo que o mesmo tem o poder de interromper a coleta com o encerramento da aplicação.). A fim de aumentar a segurança, em caso de roubo dos dispositivos, os dados são mantidos no smartphone apenas até que sejam transmitidos para o servidor.

Todo o processo de coleta aconteceu de forma anônima, o banco de dados a ser usado em pesquisas futuras, não possui identificação de um usuário específico, nem de um dispositivo específico.

CAPÍTULO IV -Proposta de arquitetura para coleta de dados

Existem diversos tipos de traces a disposição de pesquisadores, especialmente quando consideramos a existência de valiosos repositórios como o CRAWDAD[11] (*Community Resource for Archiving Wireless Data At Dartmouth*), que permite que os traces de pesquisas já realizadas sejam compartilhados por diversos pesquisadores ao redor do mundo. Assim, diversos tipos de traces de mobilidade estão à disposição para utilização em pesquisa, cada um deles com características distintas, mas que são diretamente relacionadas aos requisitos técnicos e funcionais do estudo onde foram originalmente aplicados. Do reconhecimento da necessidade de coletar novos tipos dados: sejam de locais diferentes, com dispositivos e com processos diferentes de coleta, surge a proposta de uma arquitetura para coleta de dados de smartphones, que foi utilizada por este trabalho. Durante a etapa de planejamento do estudo, foi levada em consideração tanto a possibilidade de se utilizar alguns dos traces existentes no CRAWDAD, quanto a de efetuar uma coleta de traces específicos para esta e outras pesquisas subsequentes. Sendo que a última alternativa torna-se mais atraente, pois traz maiores facilidades e oportunidades tanto para esta pesquisa, quanto para outras pesquisas que venham a utilizá-los. Mesmo estando os traces utilizados por ZHENG, CHEN, et al. [40], como parte do projeto GeoLife, disponíveis no CRAWDAD, essa possibilidade acabou por ser descartada, dado que estes, originalmente, não foram coletados com smartphones modernos e sim com uso de GPSs [37] (sem uso da tecnologia de fusão de sensores), dado que este trabalho investiga a utilização de smartphones mais modernos que suportam fusão de sensores.

Levando em consideração, tanto o objetivo quanto as técnicas empregadas neste trabalho, a coleta de dados localizada (em vizinhanças delimitas das cidades do Rio de

[11] O acesso ao repositório CRAWDAD está disponível pelo sítio: “http://crawdad.org/”.

Janeiro e de Niterói), embutiu o benefício do conhecimento da localidade (que se traduziu em facilidades para as etapas que precisam da interpretação de dados) e novas oportunidades de pesquisas baseadas em localização. Pela coleta dos próprios traces, foi possível obter as seguintes vantagens: (i) capacidade de ajustar os padrões de coleta, como frequência de amostragem para melhor atender as necessidades desta pesquisa; (ii) controlar os dispositivos utilizados (deve-se considerar que cada dispositivo tem um conjunto de características próprias aos seus hardwares, mais especificamente os sensores suas características, como: precisão e sensibilidade); (iii) o conhecimento das condições do local de coleta e o perfil dos usuários que participaram do processo, aumenta o nível do controle sobre os aspectos de qualidade e condições no momento da coleta (este conhecimento foi aplicado na etapa de mineração de dados) e, por fim , os dados ficam fortemente associados às localidades onde foram obtidos.

Nas seções que se seguem, apresentamos a proposta de arquitetura para coleta de dados de smartphones, sendo a mesmo utilizada neste estudo. Assim, inicia-se pela apresentação dos fatores que levaram a utilização do iPhone da Apple, seguida para apresentação da arquitetura e, futuramente, como a arquitetura foi aplicada a este trabalho.

4.1. Escolha da plataforma de smartphone

Devido a limitações de tempo, esta pesquisa ser limitou a abordar apenas uma plataforma de smartphones. Para selecionar a plataforma a ser utilizada, primeiramente foi necessário analisar a representatividade de mercado de cada plataforma, que foi feita pela análise da pesquisa [9]. Logo em seguida, uma análise detalhada dos aspectos tecnológicos, incluindo: os recursos disponíveis em cada uma delas - tanto os de hardware quanto de software; além dos recursos disponíveis nos respectivos ambientes de desenvolvimento e de distribuição de software. Foram avaliadas as plataforma Android, Windows Phone 7 e Apple iPhone. Sendo assim, escolhida a plataforma que a plataforma iOS no iPhone da Apple, a que melhor atendeu no conjunto dos critérios: penetração de mercado; facilidade de desenvolvimento de software; disponibilidade de sensores e baixa fragmentação de plataforma de hardware.

4.2. Arquitetura S3A para coleta de dados

Visando a execução desta pesquisa, assim como futuras pesquisas subsequentes, uma arquitetura para coleta de dados foi proposta e implementada. Os principais benefícios para esta arquitetura são flexibilidade, segurança e escalabilidade. Dado que pesquisas de sensoriamento por smartphones podem atingir grandes escalas, no que tange a número de usuários, espaço geográfico e quantidade de informação coletada. Este tipo de pesquisa pode ser considerada como pesquisa de grande massa de dados (Big Data), quando atinge um grande número usuários ou que as informações coletadas são em grande volume. Assim, a arquitetura aqui proposta foi denominada S3A (*Scalable Smartphone Sensing Architecture*).

4.2.1. Características gerais da arquitetura

A arquitetura explora a característica sequencial de workflows de pesquisa e permite que um conjunto de computadores hospedeiros seja compartilhado para múltiplos grupos de pesquisa com o mínimo de impacto.

Levando em consideração um ciclo de pesquisa genérico, como o apresentado na Figura 4.1, baseado no modelo utilizado pelo ICPSR (*Inter-university Consortium for Political and Social Research*) [16]; fica fácil constatar a oportunidade de compartilhamento de recursos, pois as atividades de pesquisa tendem a ser sequenciais.

Figura 4.1 – Um workflow de pesquisa baseado no apresentado pelo ICPSR

Nesta arquitetura, utiliza-se o conceito de silo, como uma estrutura abstrata que agrupa um determinado conjunto de ativos virtuais e suas configurações, de forma que cada silo esteja associado uma única etapa do workflow de pesquisa. Com isso é possível obter agilidade e economia de recursos para executar cada etapa da pesquisa. Dessa forma, os componentes envolvidos em cada uma das três etapas de uma pesquisa são virtualizados e associados a um silo. Com isto, quando existe a necessidade de uma determinada etapa de pesquisa, seu respectivo ambiente de pesquisa pode ser facilmente ser ativado nos equipamentos hospedeiros compartilhados. Para isto, bastando

identificar os ativos associados ao silo da etapa e pesquisa correspondente. Então, mais de uma pesquisa pode estar acontecendo de forma paralela, desde que estejam em estágios diferentes, e a capacidade da infraestrutura de servidores comporte os ativos dos silos associados as etapas de pesquisa que estão em andamento. Ainda assim, caso haja a necessidade de comportar múltiplos estágios de pesquisa ao mesmo tempo e de forma temporária, o ambiente pode crescer utilizando serviços na nuvem do tipo IaaS (*infrastructure as a service* – infraestrutura como serviço). Com isso, acrescenta-se características de elasticidade a arquitetura.

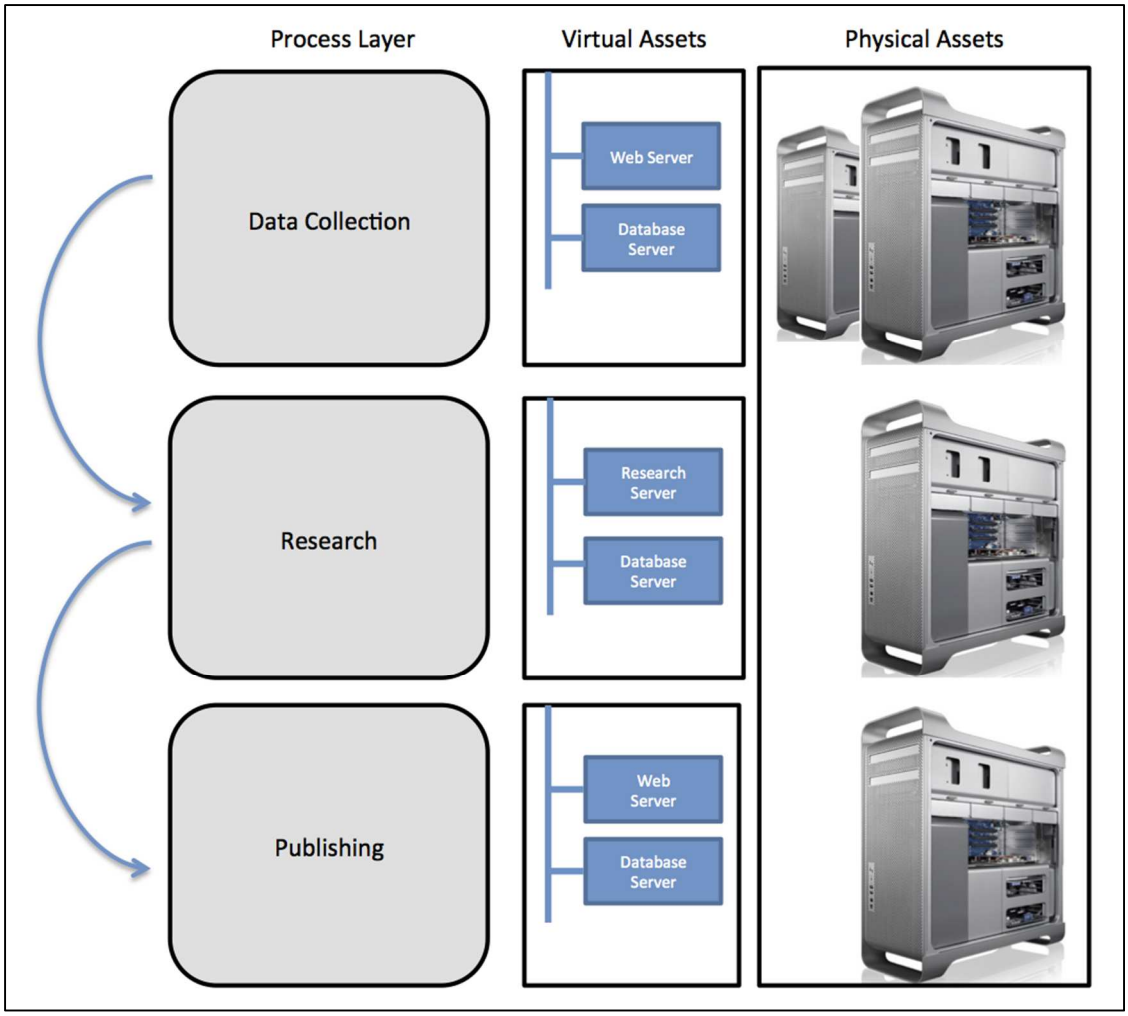


Figura 4.2 - Arquitetura S3A visão.

Em casos, nos quais haja necessidade de compartilhar os dados coletados, pode-se utilizar replicação de dados para disponibilizar as informações para múltiplos grupos de pesquisa em redes diferentes. Já. Com o uso de replicação de ambientes, pode-se acrescentar a característica de escalabilidade. Esta técnica pode ser aplicada conjunto de componentes específicos com a possibilidade de uso de múltiplos bancos de dados. É

também possível usar técnicas de federação, seja distribuindo o banco de dados por equipes de coleta, por localidades geográfica ou por características da informação coletada (por exemplo, vídeo e fotografias podem ser direcionadas para bancos de dados distintos), entre outras possibilidades. A Figura 4.2 apresenta, em camadas as estruturas da camada de processos alocadas à servidores virtuais que por vez são instanciadas em ativos físicos.

Detalhamento dos componentes da arquitetura

Pela utilização do conceito de silos é possível já ter os modelos (ou *templates*) dos ativos virtuais necessários e suas respectivas configurações, para cada uma das etapas de pesquisa; assim os silos podem ser criados, copiados, ativados ou desativados rapidamente. Baseado no modelo do ICPSR [16], são propostos três tipos de silos para suportar uma pesquisa: (i) silo de coleta de dados; (ii) silo de pesquisa e (iii) silo de publicação.

O silo de coleta de dados inclui os mecanismos para distribuição de aplicações e tarefas de coleta (esta última função é muito útil para pesquisas de sensoriamento participativo) e recebimento de dados. Tipicamente é composta de múltiplos servidores web e possivelmente de mais de um banco de dados, dependendo da escala da pesquisa.

Como a arquitetura é flexível, não existem restrições para os tipos de bancos de dados utilizados; em muitos casos, bancos de dados NoSQL têm se mostrados mais flexíveis para a coleta de dados, uma vez que não é necessário um modelo de dados definido para armazenamento da informação coletada. Além disso é possível o uso de múltiplos servidores web para balanceamento de carga. Para grandes massas de dados, existe a possibilidade é utilizar uma infraestrutura de clusters tipo Hadoop para recebimento e armazenamento de dados suportando uma escala da pesquisa que teoricamente poderia chegar a suportar pesquisas de escala global.

O silo de pesquisa é onde ocorre a preparação do dado, mineração, desenvolvimento de software e análise de resultados. Pode-se utilizar diversos tipos de componentes de acordo com o objetivo de pesquisa.

Já o silo de publicação, pela sua função, é um recurso que tende a ser compartilhado para publicação de múltiplas pesquisas. Composto por múltiplos servidores web e bancos de dados, além de infraestrutura para lidar com a carga de acesso e prover os

recursos de segurança necessários, como: balanceamento de carga, proxy reverso, firewall, entre outros.

4.3. Aplicação da arquitetura

Dos três tipos de silos definidos na arquitetura, dois deles foram utilizados neste trabalho: um silo de coleta de dados, composto por dois servidores Linux virtualizados, sendo um servidor web e um servidor de banco de dados MySQL; e um silo de pesquisa, composto de um servidor MySQL (com uma réplica do banco de coleta). Além disso, foi utilizada uma estação de trabalho Linux com o R[12] e o R Studio[13], assim como um laptop rodando MAC OS X Mountain Lion com Weka e R. Os ambientes virtualizados foram implementados sobre o hypervisor Oracle VirtualBox. Este hypervisor foi escolhido por seus atributos de performance e estabilidade, sendo suficientes para atender os requisitos do ambiente.

A aplicação móvel (cliente no smartphone) de coleta de dados tem características de aplicações em duas camadas. Esta aplicação utiliza as tecnologias de redes WiFi e 3G para transferir dados para o servidor web na qual, a transferência utiliza o protocolo HTTP (*hypertext transfer protocol*) e requer autenticação de usuário por senha para a gravação de dados, a frequência de transmissão pode ser configurada na aplicação, inclusive a possibilidade de usar somente rede WiFi para transmissão.

A seguir, cada um dos componentes da arquitetura é apresentado em maiores detalhes (Figura 4.3).

4.3.1. Módulo cliente

O módulo cliente consiste em uma aplicação desenvolvida para iPhone 4 ou superior, desde que rodando IOS 5 ou 6. Essa aplicação foi desenvolvida usando a linguagem *Objective-C* e foi denominada CityTracks. Através do uso do CityTracks, é possível coletar diversos tipos de dados, que incluem: notas, fotos e vídeos geo-localizados; além de dados de movimentação dos usuários. Com isso é possível utilizar o CityTracks para diversos tipos de pesquisa de coleta de dados com smartphones, seja a coleta

[12] R é uma linguagem de programação interpretada de código aberto, que voltada para aplicações estatísticas. Está disponível em: “http://www.r-project.org/”
[13] O R-Studio é um ambiente de desenvolvimento para línguagem R, multiplataforma e de código aberto. Está disponível em: “http://www.rstudio.com/”

participativa ou oportunista [45]. A Figura 4.4 apresenta a interface da aplicação, onde existe uma tela principal que permite o acesso as ferramentas de coleta ao mesmo tempo que o usuário pode ver sua localização estimada e a qualidade da estimativa de localização.

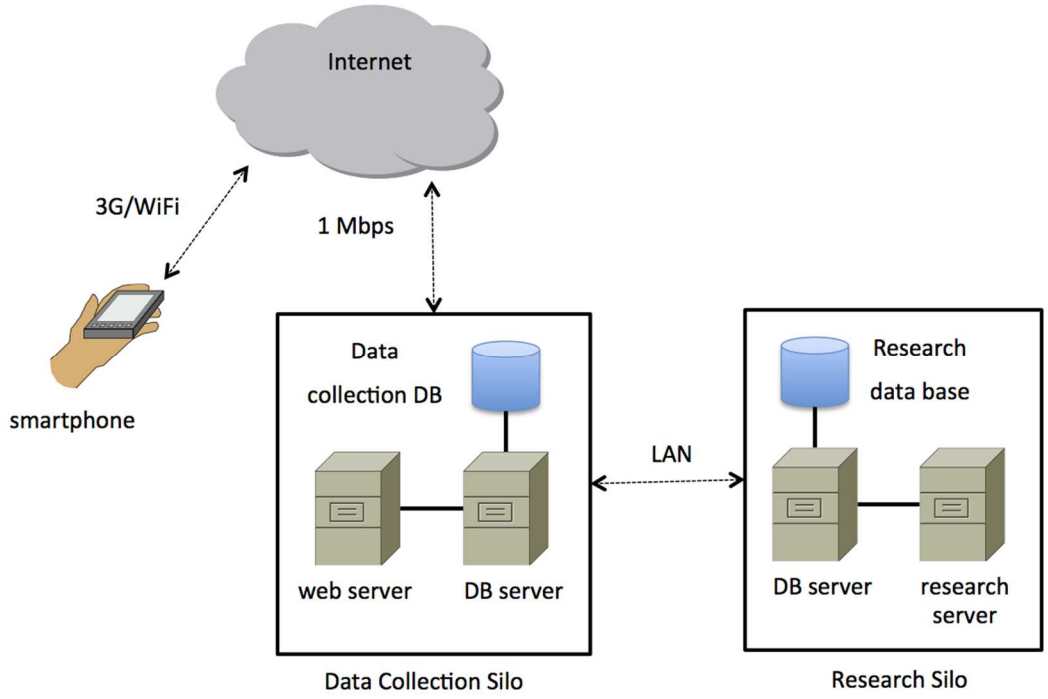


Figura 4.3 – Componentes da implementação da arquitetura S3A utilizados neste trabalho.

Através de um componente de interface o usuário pode informar suas mudanças de modo de transporte, esta funcionalidade foi criada para atender especificamente as necessidades deste trabalho. Esta informação é enviada no mesmo formato de uma nota geo-localizada, com informação do local e hora do seu registro. Da mesma forma, para atender as necessidade deste trabalho, o botão para acesso a câmera foi desativado, uma vez que fotos e vídeos aumentariam a necessidade de armazenamento na infraestrutura de banco de dados de forma desnecessária.

Através do botão de configuração é possível acessar um manual de utilização do CityTracks, assim como configurar a sensibilidade do sensor de localização, esta última configuração foi desabilitada para que todos os smartphones trabalhassem de forma análoga

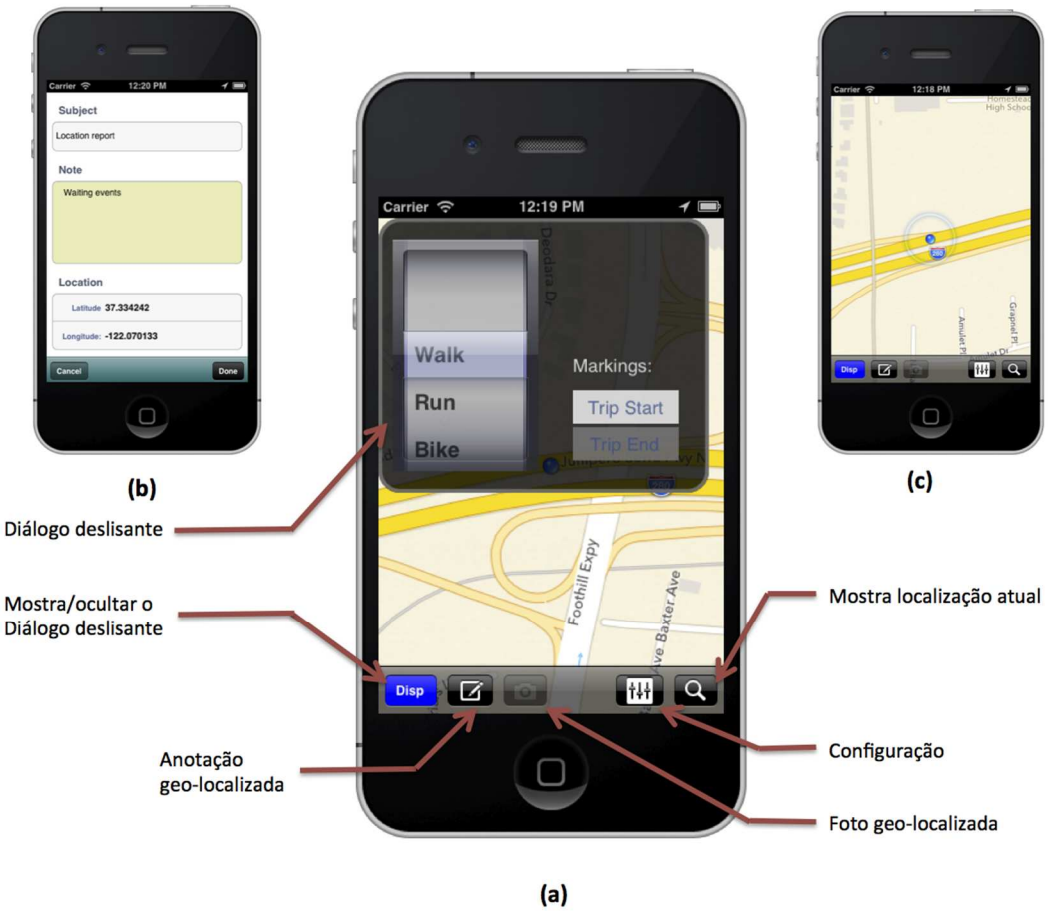


Figura 4.4 - Interface e funcionalidades da aplicação *CityTracks*.

4.3.2. Servidor web

O servidor web é o componente responsável por receber os dados enviados pelos usuários da aplicação *CityTracks*. Foi utilizado um servidor Apache 2.2 com aplicação em PHP5 que faz a autenticação da origem, validação sintática dos dados, registrando a comunicação e inserindo no banco de dados. Foi utilizado um servidor virtual Ubuntu Server Linux 10.2 com 2Gb de RAM e 20Gb de disco. Tanto o servidor web, quanto a plataforma de servidor Linux foram escolhidos por serem padrão de mercado e de código livre. O envio de informações, conforme descrito anteriormente, utiliza métodos POST do HTTP para recebimento dos dados, sendo um tipo de chamada para cada tipo de dados (foto, anotação, trace).

4.3.3. Banco de dados

Os dados reportados pelo sensor de localização possuem o seguinte formato [2]: *<latitude, longitude, timestamp, horizontal_accuracy, altitude, vertical_accuracy,*

speed, course>, onde: *course* se refere a direção do deslocamento, *speed* a velocidade instantânea do dispositivo, *vertical_accuracy equivale a* acurácia vertical e *horizontal_accuracy,* por sua vez, equivale à acurácia horizontal, estes dois últimos refletem o grau de incerteza quanto ao posicionamento expresso em metros. Quando a *horizontal_accuracy*, *vertical_accuracy*, *course* e *speed* assumem o valor -1, significa que a medida não é confiável.

O sistema gerenciador de banco de dados utilizado neste trabalho foi o MySQL, que foi escolhido levando em consideração os seguintes fatores: ser um sistema aberto; com ampla documentação e grande facilidade de uso. Apesar do MySQL ser um banco relacional, não houve necessidade para este estudo de fazer uso de tais características, assim como não foi implementada nenhuma restrição de integridade no de banco de dados.

No banco de dados de coleta, foram utilizadas três tabelas para armazenar os dados, sendo: uma para registrar as conexões para envio de informação, uma para os registros de movimentação e uma para anotações geo-localizadas, conformes apresentadas na Tabela 4.2.

Tabela 4.1 – Lista de tabelas do banco de dados que foram utilizadas neste trabalho.

Tabela	***Campos***
Location_traces	*device_id, time_stamp, horizontal_accuracy, latitude, longitude, vertical_accuray, altitude, speed, course*
Note	*device_id, timestamp, horizontal_accuracy, latitude, longitude, vertical_accuray, altitude, speed, course, note_title, note_subject*
Connection	*device_id, time_stamp, source_IP, records*

4.3.4. Plataforma de tratamento de dados e pesquisa

Para tratamento de dados foi utilizado um laptop MacBook Pro com processador Core i7, 16Gb de RAM e 750Gb de disco. Um servidor de banco de dados foi usado, rodando em uma máquina virtual Linux no próprio laptop.

4.3.5. Críticas sobre a utilização da arquitetura S3A neste trabalho

Apesar da arquitetura ter se mostrado adequada para a coleta de dados, existem alguns fatores que precisam ser levados em conta, quando esta for utilizada em estudos futuros, são eles: houve perda de precisão nos registros de *timestamp* quando estes foram levados do smartphone para o banco de dados; O consumo de bateria do smartphone foi elevado; não foi aplicado criptografia na transferência de dados entre o smartphone e o servidor. Quanto ao primeiro fator, houve uma perda de precisão no registros dos *timestamps* gerados pelo smartphone, quando armazenados no banco de dados MySQL. O *timestamp* no smartphone possui registro de 1/10 de segundos, enquanto no banco de dados o formato de *timestamp* registra somente o segundo inteiro. Apesar disso, não houve impacto pois não usamos restrições de integridade para inserções nas tabelas, todos os registros apareceram mas perderam a casa de centésimos de segundo. Com isso, apareceram leituras duplicadas para um determinado segundo, que foi tratada de forma adequada em etapa posterior do trabalho. Quanto ao consumo de bateria, os usuários foram instruídos a somente iniciarem a aplicação quando houvesse deslocamentos externos (*outdoor*), dado que este é o tipo de movimentação que estamos interessados. Quanto a falta de encriptação, este pode ser resolvido pela utilização do protocolo TLS ou pela encriptação do pacote de traces antes da transmissão, mas não foi possível aplicar para esta pesquisa por restrições de recursos de programação.

CAPÍTULO V - Aplicação de descoberta de conhecimento em banco de dados

Este capítulo tem o objetivo de apresentar as etapas de execução que incluem tanto as atividades de coleta de dados, quanto as atividades relacionadas ao processo de descoberta de conhecimento em banco de dados. Assim, este capítulo está estruturado da seguinte forma: na Seção 6.1, é apresentada a etapa de coleta de dados e da Seção 6.2 em diante são apresentadas as etapas de execução do processo KDD, onde: a Seção 6.2 apresenta a etapa de pré-processamento; a Seção 6.3 a etapa de transformação do dado; a Seção 6.4 a definição da função de mineração de dados; a Seção 6.5 os procedimentos utilizados para avaliação dos classificadores e na Seção 6.6 conclui-se com a análise crítica da aplicação dos processos relacionados a descoberta de conhecimento em banco de dados.

5.1. Processo de coleta de dados

A coleta de dados foi executada por voluntários, que utilizando a ferramenta *CityTracks*, coletaram mais de 880.000 registros de posicionamento. Foram utilizados nove voluntários, que atuaram entre setembro de 2012 a maio de 2013. Esta seção apresenta informações relativas ao processo de coleta dos dados, cuja a execução ocorreu nas seguintes etapas: preparação para a coleta de dados; definição da área geográfica; definição dos requisitos de coleta; seleção de voluntários; definição do processo de coleta de dados; execução da coleta de dados, conforme apresentado na Figura 5.1.

Figura 5.1 - Etapas do processo de coleta de dados.

5.1.1. Preparação para a coleta de dados

O processo de preparação para a coleta de dados, consistiu em: disponibilização e teste da infraestrutura de servidores; instalação da aplicação cliente para coleta de dados nos smartphones (a aplicação *CityTracks*) e no planejamento para execução da coleta de dados propriamente dita, esta última incluindo a definição da abrangência da área geográfica para a coleta de dados.

Considerando a existência de evidências que indicam a possibilidade de que características intrínsecas de cada cidade possuem potencial para influenciar nos padrões de mobilidade de sua população, de acordo com o trabalho apresentado por NOULAS, SCELLATO, et al. [20], foi detectada a necessidade de se limitar a área de estudo, para capturar apenas o conjunto de padrões de mobilidade que sejam intrínsecos a área elegida. O alcance desta área foi determinado, visando comportar o número de possíveis voluntários, se possível mantendo-se dentro de um perímetro urbano contínuo. Assim, a área geográfica utilizada neste estudo, incluiu: (i) Zona Sul, Zona Norte e Centro da cidade do Rio de Janeiro; (ii) Zona Sul, Zona Norte, Centro e Região Oceânica da cidade de Niterói. A Figura 5.2 apresenta um mapa com a plotagem de todas as trajetórias registradas, onde é possível verificar a área de alcance geográfico deste estudo.

Figura 5.2 - Mapa apresentando a área de alcance das trajetórias registradas.

Para a seleção de voluntários, os seguintes critérios foram utilizados: (i) local de residência e trabalho dentro da área de estudo; (ii) diversidade de meio de transporte utilizados no seu dia a dia; (iii) variedade de faixa etária e gênero. Foram utilizados nove voluntários, ficando a possibilidade de crescimento do número de usuários para trabalhos futuros.

5.1.2. Caracterização dos dados coletados

Foram coletados 800.000 registros de posicionamento (em tuplas), provenientes de 180 trajetórias. Sendo que estas trajetórias utilizaram os seguintes modos de transporte: carro, motocicleta, ônibus, bicicleta, barcas, a pé e metrô. Não foram feitas distinções de deslocamentos correndo e a pé. Os seguintes dispositivos foram utilizados para a coleta de dados: iPhone 4, iPhone 4S, iPhone 5. Os dados coletados foram armazenados no banco de dados com o formato apresentado na Tabela 5.1. As coletas de trajetórias ocorreram entre 01/09/2012 e 31/05/2013 e deram-se durante a movimentação natural dos usuários e suas rotinas do dia a dia. Cabe notar que a participação dos usuários ocorreu de forma voluntária e alguns usuários participaram por um período maior que outros.

Tabela 5.1-Formato dos registros de posicionamento no banco de dados.

Nome do Atributo	Formato do dado no Banco de Dados	Descrição
device_id	varchar	Identificador do smartphone.
measurement_timestamp	datetime	Data e hora que a leitura de posição foi gerada
course	float(10,6)	Direção em que se deu o deslocamento.
altitude	float(7,2)	Altitude obtida pela leitura
speed	float(10,6)	Velocidade do deslocamento
vertical_accuracy	float(10,6)	Acurácia estimada da leitura vertical
horizontal_accuracy	float(10,6)	Acurácia estimada da leitura horizontal
latitude	float(10,6)	Latitude da leitura
longitude	float(10,6)	Longitude da leitura

5.1.3. Considerações sobre a taxa de amostragem

Devido às características da própria implementação do *CoreLocation framework*, este não permite que seja atribuída uma taxa específica para geração de registros de posicionamento. Por consequência, a taxa de geração de registros de posicionamento se apresenta variável. Pelo modo de funcionamento padrão deste componente de localização do iPhone, um novo registro de posicionamento é gerado, sempre que um limiar de distância entre o último registro e a posição atual for ultrapassada. Como consequência do fato supracitado, a frequência com que novos registros são gerados é diretamente proporcional a "sensibilidade" configurada no componente *LocationManager* do iPhone. Os detalhes de funcionamento são apresentados na documentação da própria APPLE, [2]. Quando utilizado o modo de maior sensibilidade na configuração deste componente, (definida através da constante *bestForNagivation* no código fonte do programa); dependendo do deslocamento do smartphone é possível ter uma taxa com mais de um registro por segundo sendo gerado. Quando feita a comparação com os trabalhos relacionados, é possível notar as taxas de amostragem como: 30 segundos no trabalho [29]; 1 segundo no trabalho [25] e por fim 2 segundos utilizada no trabalho [37].

Para este trabalho buscou-se utilizar a maior sensibilidade possível, visando assim, obter o maior grau de acurácia e sensibilidade para os registros de posicionamento. Dessa forma é possível obter-se um maior grau de flexibilidade para o estudo, uma vez que uma granularidade maior de registros de posicionamento permite simular menores granularidades, apenas pelo descarte de registros selecionados, evitando assim, a necessidade de efetuar nova coleta de dados com diferente configuração de sensibilidade.

5.1.4. Procedimentos aplicados para coleta de dados

Os usuários foram instruídos que ao iniciar suas movimentações, executassem a aplicação *CityTracks* e informassem o início e o fim dos deslocamentos, junto com o meio de transporte utilizado. Foi através da utilização deste procedimento que viabilizou o uso de técnicas de classificação na etapa de mineração de dados. Assim, foi possível atribuir etiquetas (ou *labels*) ao conjunto de traces referentes a um determinado deslocamento para classificação das trajetórias.

5.1.5. Seleção de modos de transporte

Este trabalho buscou obter o maior número possível de meios de transportes em suas amostras, embora não tenha sido possível obter amostras significantes e com qualidade para todos eles. Assim, foi tomada a decisão de não considerar os segmentos de trajetória de barcas e metrô, pelos motivos que seguem:

Quanto ao metrô, houve problema quanto a indisponibilidade de sinal de GPS, WiFi e celular no subterrâneo. Além disto, apenas em poucas estações foi possível obter algum sinal para gerar os registros de localização.

Já quanto as barcas, foi observado que a qualidade do registro de localização pode sofrer considerável variação, principalmente pelos seguintes aspectos: dependendo de onde o usuário se sente dentro da barca, ele receberá ou não sinal de GPS e da rede de telefonia celular. Com o usuário sentado próximo a janela a precisão melhora, pois existe disponibilidade de GPS, caso contrário a única forma de localização são as torres de celulares, quando isso ocorre ele tanto pode se basear em torres de um lado da baía quanto de outro. Outra dificuldade em obter um registro de movimentação de qualidade relacionado a barca, é que o usuário tem a liberdade de se locomover dentro da embarcação, o que pode gerar ruído para os dados tanto pelo deslocamento, quanto pela variação da qualidade de sinal. portanto não precisão suficiente para usar os métodos aqui propostos.

Dados os problemas supracitados, neste trabalho foram utilizados somente os dados de movimentação dos seguintes modos de transporte: ônibus, carro, moto, a pé e de bicicleta.

5.1.6. Crítica ao processo de coleta de dados

Durante e após a execução do processo de coleta de dados, algumas descobertas já mostram-se relevantes - tanto para avaliar os resultados deste trabalho, quanto para orientar estudos futuros. Algumas destas descobertas foram utilizadas para ajustar o processo de coleta de dados, enquanto outras impactaram na seleção dos dados utilizados nas etapas posteriores ou orientaram expansões para trabalhos futuros. Levando isso em consideração e de forma a orientar trabalhos futuros, os problemas e limitações mais importantes do processo de coleta de dados, são aqui apresentados:

(i) a opção de coletar somente as trajetórias, apesar de facilitar o processo de detecção de meio de transporte e permitir a redução do consumo de bateria, limita algumas futuras possibilidades para o estudo, entre elas a possibilidade de detectar pontos de interesse e atribuir semântica a cada um deles, baseado nas características das movimentações;

(ii) alguns registros estacionários foram gerados sem necessidade, pois algumas vezes o aplicativo *CityTracks* era esquecido ativo pelo usuário, e assim apenas passou a registrar sua localização indoor e seus curtos deslocamentos característicos. Futuramente um mecanismo pode ser desenvolvido para lembrar o voluntário a informar término de trajetórias e mudanças de meio de transporte ou resumir o registro de pausas ainda dentro do dispositivo móvel.

(iii) O número de voluntários não permitiu uma representatividade muito significativa, embora outros trabalhos importantes tenham utilizado até um número menor que esse.

5.2. Descoberta de conhecimento em banco de dados

Uma vez que a coleta de dados foi concluída e os dados estando disponíveis para processamento, foi possível iniciar a etapa de descoberta de conhecimento em banco de dados usando o processo KDD (apresentado no Capítulo III). Esta etapa se caracteriza por um conjunto de sub-etapas aplicadas de forma sequencial, mas que permite de forma livre a existência de sucessivas iterações. Iterações estas, podendo iniciar em qualquer etapa do processo KDD, da seleção de dados, passando pela aplicação da mineração de dados e por fim pela aplicação do conhecimento obtido.

Nesta Seção, primeiramente são apresentados os procedimentos aplicados a etapa de pré-processamento, seguido dos procedimentos aplicados à etapa de transformação do dados. Sendo posteriormente abordada a etapa de mineração de dados.

5.2.1. Execução da etapa de pré-processamento

De acordo com o método KDD, já apresentado anteriormente, a sua etapa de pré-processamento inclui as seguintes sub-etapas: limpeza dos dados; remoção de ruídos; remoção de *outliers* e tratamento de dados faltantes. A necessidade destas etapas torna-se mais relevantes quando se leva em conta as consideração de CORTÊS [7]. Segundo ele, dados do mundo real tendem a ser incompletos, fora dos padrões e até

inconsistentes. Assim neste trabalho, esta etapa foi aplicada primariamente para o tratamento dos registros duplicados e logo em seguida os registros imperfeitos.

A existência dos registros duplicados deu-se pelos seguintes fatores diagnosticados: (i) falhas no processo de transferência de registros entre os smartphones e o servidor de banco de dados - com ocorrência de retransmissão (e por consequência a duplicação); (ii) características de funcionamento do próprio componente de localização do iPhone; (iii) perda de precisão do registro *timestamp* no banco de dados. É importante esclarecer, que a existência de registros duplicados não caracteriza um problema, apenas mais uma característica a ser considerada no tratamento dos dados. Assim, o procedimento adotado para remoção de registros duplicados foi a criação de uma nova tabela, onde através de comandos SQL efetuou-se a carga somente das tuplas não duplicadas e de uma das tuplas quando da existência de duplicadas. Sendo que: das duplicadas, foi escolhida a com melhor valor para o campo *horizontal_accuracy* (menor e não zero). Como resultado, foi obtida uma tabela com a mesma estrutura da original, mas com um número de registros menor (pela ausência dos registros duplicados). Este procedimento foi assim definido e aplicado, visando preservar a tabela de dados coletados originalmente, dando um maior grau de liberdade para modificar e transformar os dados, uma vez que a tabela original permaneceu preservada em seu estado bruto.

Quanto aos registros imperfeitos, especificamente nesta etapa, foram tratados aqueles que se caracterizam por possuir baixa acurácia horizontal. Dentre os diversos fatores causais destes registros, é possível elencar os principais: (i) indisponibilidade de sinal de GPS ou WiFi; (ii) e o tempo de convergência para obtenção de um registro de melhor precisão. Quando o componente de localização do iPhone é ativado, os seguintes eventos ocorrem: (i) imediatamente o *LocationManager* retorna a posição do último registro de localização guardado em memória (mesmo que este seja muito antigo), é necessário um tratamento para o descarte do primeiro registro, caso este não seja relevante; (ii) então são ativados os sensores e reportada a posição atual, sendo que com uma acurácia crescente a cada registro, até que todos os sensores retornarem seus respectivos valores. É necessário que haja comunicação com a Internet para que o a localização por *fingerprint* (seja Wifi ou da rede de celular) possa ocorrer. Com isso o único procedimento adotado para limpeza de dados, específico desta etapa, foi a

remoção dos registros onde o atributo *horizontal_accuracy* [14] era maior do que 200 (metros), tendo como objetivo obter somente os registros com melhor precisão. Tal procedimento, atende a uma das etapas do framework apresentado por IDRISSOV e NASCIMENTO [17].

Como não foram detectados registros com informações faltantes, um tratamento para este problema não foi necessário.

5.3. Transformação dos dados

As etapas de transformação e agregação de dados do método KDD, são subsequentes a etapa de pré-processamento e se ocupam da preparação dos dados para a etapa de mineração, de acordo com FAYYAD, PIATETSKY-SHAPIRO, et al. [12]. Elas incluem, mas não se limitam à: adequação dos dados para permitir a descoberta de informações; aplicação de redução de dimensionalidade (esta não foi necessária neste trabalho); além da agregação de dados propriamente dita.

Especialmente para este estudo, que trabalha com dados espaço temporais, uma etapa adicional de agrupamento em trajetórias é de especial importância e foi implementada como parte da etapa de transformação do dado.

A Figura 5.3, apresenta as atividades aplicadas de transformação de dados (através de um diagrama BPMN [15]) que também inclui a atividade da etapa anterior (de pré-processamento), relativo a remoção de registros duplicados e de baixa acurácia.

É importante ressaltar que esta seção apresenta duas etapas, na primeira ocorre a segmentação da trajetória, orientada pelo framework de limpeza de trajetórias apresentado por IDRISSOV e NASCIMENTO [17] e na segunda a transformação do dado em si de acordo com o processo KDD, proposto por FAYYAD, PIATETSKY-SHAPIRO, et al. [12].

[14] Conforme apresentado no Capítulo III, quanto maior o valor do *horizontal_accuracy*, menor a precisão da localização, sendo que um valor -1 indica que não foi possível obter uma localização confiável [2].
[15] *Business Process Modeling Notation*, maiores informações podem ser obtidas em: http://bpmn.org

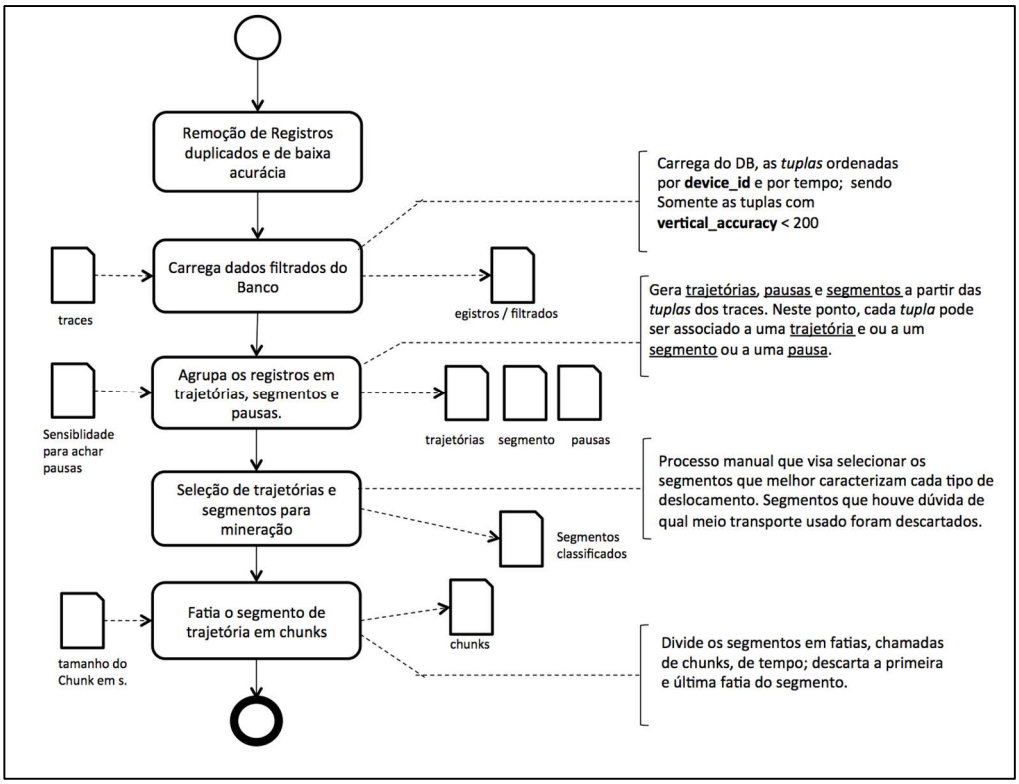


Figura 5.3-Diagrama de fluxo para o processo de pré-processamento.

A aplicação do framework apresentado por IDRISSOV e NASCIMENTO [17], foi feita de forma oportunista, isto é cada um dos seus três passos foram aplicados em etapas afins, tanto do pré-processamento quanto do processo de tratamento de dados (levando em consideração o processo KDD).

Para a detecção de pausas, separamos as trajetórias através da abordagem intuitiva, apresentada por IDRISSOV e NASCIMENTO [17]. Tal abordagem foi uma alternativa, uma vez que o algoritmo CB-SMOT apresentado por PALMA e BORGONY [22], que seria mais adequado para tal fim, não pode ser usado pois este visa a aplicação em trajetórias completas, incompatível com a classificação de trajetórias ainda em andamento (que é o foco deste trabalho). Já a etapa de interpolação de segmentos perdidos do framework proposto por IDRISSOV e NASCIMENTO [17] (ii), foi implementada, posteriormente e junto com a etapa de agregação de dados do processo de KDD (agregação baseada em tempo de um determinado segmento de trajetória).

Durante o processo de agregação dos registros, em sua estrutura de agregação temporal, na falta de alguns registros, foi feita uma interpolação. Esta usou a velocidade do valor anterior, considerando assim, que ocorreu um deslocamento uniforme entre o registro

anterior e o registro corrente. Este tipo de procedimento, produziu efeitos análogos a uma interpolação de pontos convencional (com pontos uniformemente distribuídos).

O tratamento de registros de baixa precisão (terceira etapa do framework de NASCIMENTO e IDRISSOV [12]), foi feito junto da etapa de remoção de registros duplicados e de baixa acurácia; onde, foi aproveitada a existência do registro *horizontal_accuracy* do componente de localização do iPhone para identificar e remover os registros com acurácia pior que 200 metros(ou seja, maior que duzentos ou igual a zero). Este valor (200 metros) foi assumido, buscando-se nem descartar muitos registros, nem aceitar registros com acurácias muito ruins - que possivelmente poderiam prejudicar a segmentação das trajetórias e identificação de pausas. Assim, como resultado desta operação (a remoção dos registros de baixa acurácia), tivemos um efeito colateral, que foi o descarte de registros de posicionamento inferidos apenas com uso do sinal da telefonia celular. Uma vez que não se tem nem sinal de GPS, nem sinal de WiFi, a acurácia cai significativamente, e assim os registros passam a ser descartados pela mesma razão.

5.3.1. Processo aplicado para segmentação de trajetórias

Inicialmente, as trajetórias foram obtidas através da leitura sequencial dos registros de posicionamento. Esta leitura, se deu com os registros ordenados por dispositivo e tempo (com tempo crescente). Um limiar de tempo foi utilizado para agrupar as tuplas referentes a uma mesma trajetória. Para o caso deste limiar ser ultrapassado sem nenhuma ocorrência de registros de posicionamento, a trajetória, então passa a ser considerada encerrada; registros de posicionamento posteriores farão parte de uma nova trajetória. A definição do valor para o limiar de tempo de trajetória, se deu através de um processo de exploração, onde aplicaram-se várias tentativas, até que fosse obtido um valor satisfatório para se agrupar as trajetórias. Como resultado deste procedimento, o limiar para agrupamento ficou definido como vinte minutos.

A detecção de pausa, foi feita através de um limiar de tempo de pausa e de um limiar de distância de pausa. Sendo que o limiar de tempo pausa, deve ser obrigatoriamente menor que o limiar de trajetória. Assim, uma pausa é adicionada a um deslocamento em andamento, sempre que durante um período de tempo maior que o limiar de tempo de pausa, não houver um deslocamento que exceda o limiar de distância de pausa em um segundo. Os limiares utilizados foram 0,4 metros para a distância e 90 segundos para o

limiar de tempo de pausa. Tais valores foram selecionados, levando-se em consideração o tempo de transição dos semáforos de trânsito, a sensibilidade ao deslocamento do framework *Core Location* do iOS e a um grau de identificação de paradas satisfatório.

Processo utilizado para a atribuição de etiquetas para identificação de modo de transporte.

Com os segmentos de trajetória definidos, foi possível analisar as informações de modo de transporte provenientes dos usuários. Para então, atribuir a classificação apresentada a um determinado segmento. Dada esta forma de ação, é possível a utilização de técnicas de classificação, para determinar o modo de transporte usado em uma determinada movimentação (caracterizada por um conjunto de registros de posicionamento). Esta é uma etapa intensivamente manual, isto é, requer a intervenção humana (do especialista), se não, para a classificação dos segmentos de acordo com suas características, ao menos para validar a classificação do usuário e a qualidade do conjunto de registros que compõem o deslocamento. Cada trajetória e seus segmentos foram analisadas e quando estas se mostraram corretas, foi possível aplicar uma etiqueta permitindo identificar o modo de transporte utilizado. A identificação se deu, levando em consideração características da trajetória plotada sobre o mapa utilizando a biblioteca *RGoogleMaps* do R.

A Figura 5.4 apresenta um exemplo de trajetória plotada com o uso desta biblioteca, onde é possível verificar alguns dos tipos de problema que devem ser levados em conta pelo especialista na atribuição das etiquetas dos segmentos da trajetória.

Já a Figura 5.5 apresenta um esquema de diferenças no padrão de registros de posicionamento de uma movimentação, quando observada de forma ampliada, com diferentes velocidades. Cada circulo representa um registro de posicionamento plotado em um espaço geográfico.

Figura 5.4 - Visualização de problemas relativos a perda de sinal em uma trajetória usando um script RGoogleMaps.

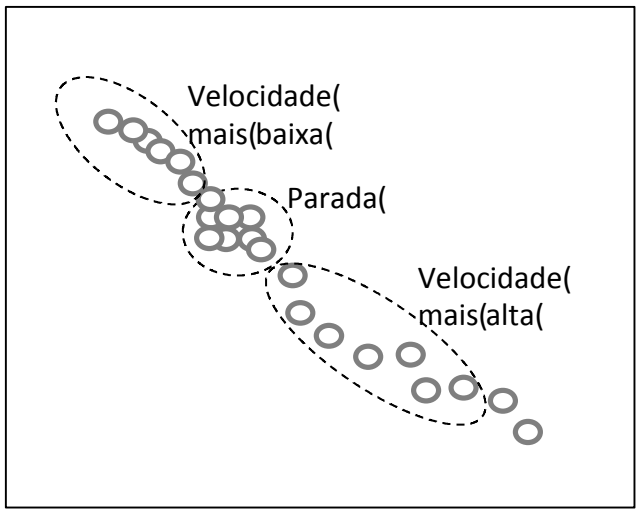


Figura 5.5 - Representação das diferenças de movimentação através de mudanças no padrão de plotagem.

Como resultado final da etapa de agrupamento e segmentação de trajetórias, cada registro de posicionamento, pode então ser associado ou a um deslocamento ou a uma pausa. Estas por sua vez são associadas a uma determinada trajetória.

Baseado nas informações providas pelos usuários através da ferramenta CityTracks e da análise visual das trajetórias; foi então possível atribuir um meio de transporte para um determinado segmento de trajetória.

Caso o objetivo desse trabalho fosse apenas classificar os segmentos de trajetórias após seu término, de forma similar ao apresentado no trabalho de ZHENG, LIU, et al. [37], a partir desta etapa, isso já seria possível; efetuando-se as sumarizações e entrando na etapa de mineração de dados propriamente dita.

Considerando-se que o objetivo deste trabalho é a classificação dos deslocamentos em andamento (que ainda estão ocorrendo), uma etapa adicional se fez necessária. Tal etapa, visa sumarizar os dados de frações de um segmento de trajetória, como parte de um deslocamento ainda incompleto. Assim que a primeira fração de um segmento esteja disponível, será possível classificá-lo de acordo com suas caraterísticas. Tal conceito é a seguir apresentado.

Agregação dos registros de posicionamento de um dado segmento de trajetória:

Para agregar registros de posicionamento, etapa necessária para utilização na etapa de mineração de dados, as seguintes técnicas de agregação foram levadas em conta: (i) agregação por distância; (ii) agregação por pontos de amostra; (iii) agregação por janela deslizante; (iv) agregação por tempo; (v) agregações híbridas. Cada uma delas possui vantagens e desvantagens inerentes:

Agregação por distância: Consiste em agrupar uma série de tuplas até que uma determinada distância de deslocamento seja ultrapassada, gerando a tupla de agregação com informações para este deslocamento. Este tipo de agregação não é muito adequado para sistemas interativos, pois requer que haja um deslocamento para que a agregação seja gerada. Pode acontecer rapidamente ou levar um certo tempo, dependente da taxa de movimentação. Quando consideramos o uso dos sensores de localização do iPhone que geram um registro, somente quando existe movimentação, esse tipo de abordagem pode gerar uma agregação mais homogênea.

Agregação por pontos de amostra: Consiste em gerar uma tupla de agregação quando um número específico de registros for obtido. O que pode gerar tuplas com um longo

período de pausa embutido. Também pode não ser adequado para sistemas interativos, pois pode decorrer longos períodos sem que o número de registros necessários para sumarização seja obtido. Com isso o tempo para início da classificação pode ser longo.

Agregação por janela deslizante: Similar a janela deslizante adotada no protocolo TCP, consiste em gerar uma tupla de agregação com um número de tuplas fixos – pegando as n últimas tuplas, sempre que um limiar específico for atingido, seja de tempo ou espaço. Nesse tipo de agregação pode-se repetir tuplas dentro de tuplas de agregação. Requer um mecanismo mais complexo e pode gerar informação redundante de movimentação.

Agregação por tempo (temporal): Consiste em agrupar uma série de tuplas até que um determinado tempo seja atingido, gerando uma tupla de agregação com informações relativas ao período de tempo. Segundo [31], gerar agregados temporais é um processo complexo, pois cada registro tem associado um *timestamp* que indica o período que a informação da tupla é válida.

Agregações híbridas: Existe ainda a possibilidade de se combinar as técnicas de agregação anteriores em novas formas mais complexas de agregação. Mas, devido a própria complexidade envolvida, estas não foram exploradas neste trabalho, ficando a cargo de trabalhos futuros.

Especificamente para este trabalho, a função de agregação utilizada é a temporal. Esta foi adotada tanto pela sua característica determinística, que é importante para sistemas baseados em tempo, quanto por apresentar outras das características necessárias para sistemas cientes de contexto que foram relacionadas no Capítulo II. Com isto, baseado em um determinado intervalo de tempo, um registro sumarizando os atributos de um determinado segmento de trajetória é gerado. Estes segmentos foram denominados *chunks*, visando facilitar referências posteriores. Assim, de forma a detalhar a abordagem utilizada: uma trajetória passa a ser composta de períodos de segmentos de deslocamento e de períodos de pausa. Onde, por sua vez, cada período de deslocamento é também subdividido. Para esta subdivisão, são usados componentes de tamanho fixo de tempo, os *chunks*. A implementação deste tratamento é feito através de scripts em R. Foram usados quatro scripts, um para cada uma das quatro últimas etapas apresentadas na Figura 5.3 (Diagrama de fluxo para o processo de pré-processamento).

Conforme citado anteriormente, o *chunk*, como uma subdivisão de um segmento de trajetória, tem a função tanto de agregar quanto de sumarizar um conjunto de registros de posicionamento referentes a um determinado período de tempo. Tal período, apresenta um tamanho (tempo) fixo, exceto quando se tratar do último *chunk* de um segmento. As características de um *chunk* podem então ser extraídas, pela sumarização através de atributos que serão utilizados em etapa posterior para a mineração de dados.

A Figura 5.6 apresenta, através de um diagrama de entidades e relacionamento o modelo conceitual para os dados de uma trajetória segmentada.

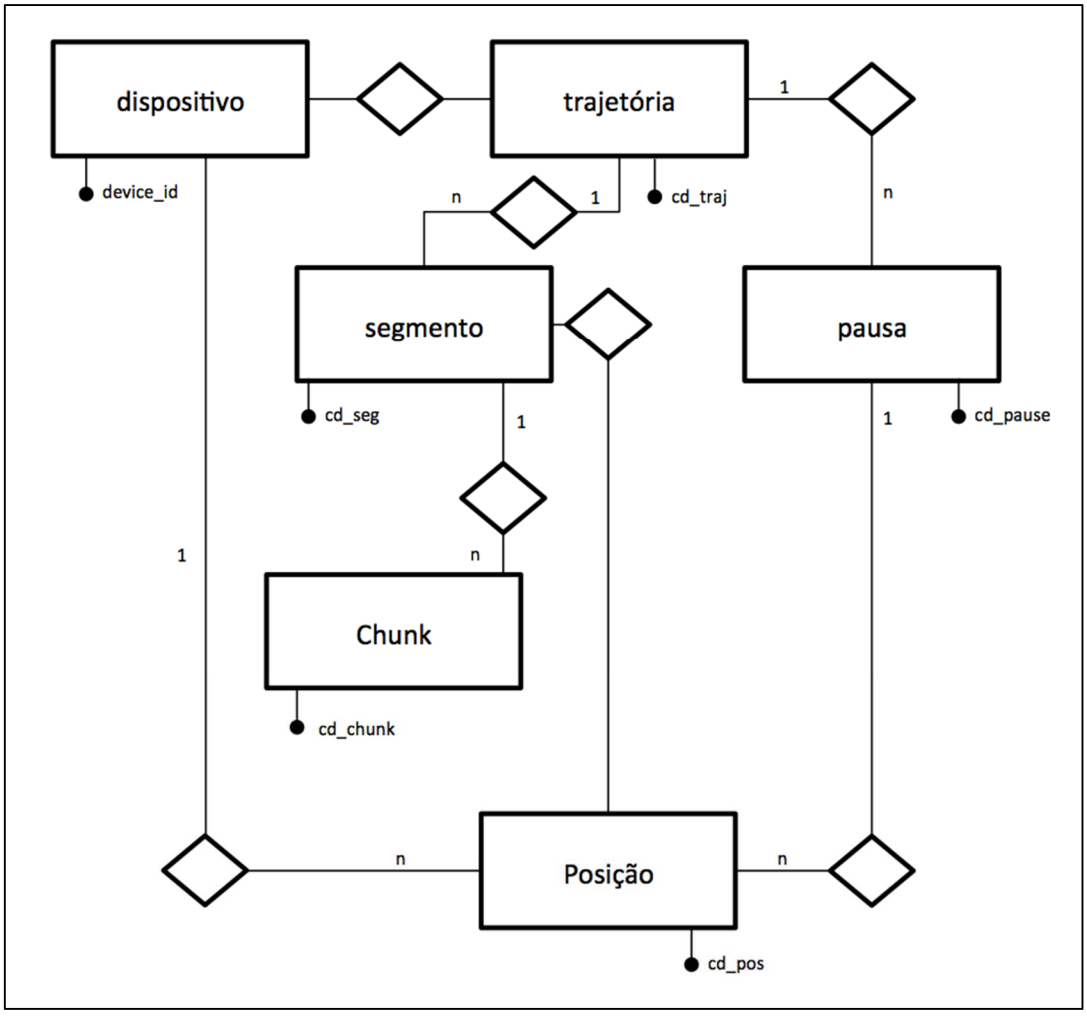


Figura 5.6 - Modelo conceitual dos dados para trajetória segmentada.

Foram utilizados três tamanhos de *chunks* para análise: 60; 90 e 120 segundos. Estes valores foram escolhidos tendo em vista não utilizar um valor pequeno demais, que poderia não incluir características de movimentação suficientemente ricas, ou um valor grande demais, que além de retardar o processo de identificação do modo de transporte,

poderia também reduzir o número de registros disponíveis para etapa de mineração de dados.

5.3.2. Redução / sumarização de dados

O objetivo da redução de dados é obter um conjunto de dados reduzido, que seja capaz de representar os dados originais. Dessa forma, temos um conjunto de dados compacto que mantém a integridade da informação original (mesmo que de forma resumida) e ainda assim permite a mineração de dados com igual ou melhor resultado.

Durante esta etapa, algumas decisões foram tomadas quanto a sumarização e criação de novos atributos. Considerando que a natureza da informação de deslocamento é vetorial, basicamente direção, distância e tempo. Estas por sua vez podem ser traduzidas para grandezas mecânicas, como: velocidade, aceleração, direção do deslocamento, quantidade de paradas e duração das paradas. Outras informações também podem ser extraídas dos próprios dados de movimentação ou pelo cruzamento com outras fontes de dados. Como exemplo de tal possibilidade, é possível citar: o trabalho de ZHENG, LUI, et al. [37], onde os autores utilizaram uma etapa de pós processamento para obter a probabilidade (*likehood*) de que um usuário esteja utilizando um determinado meio de transporte de acordo com o histórico de utilização dos demais usuários e o trabalho STENNETH, WOLFSON, et al. [29], onde os autores se utilizam-se, de forma oportuna, da existência de sistemas de informações geográficas (GIS) do sistema de ônibus e de trens urbanos. Sendo assim, foi possível incluir uma métrica de distância do ponto de ônibus ou estação de trem mais próxima que foi incluída no mecanismo de classificação.

Técnicas utilizadas para sumarização do dados.

Pela utilização dos conceitos de agrupamento foi possível sumarizar dados característicos do deslocamento, como: velocidade média; velocidade máxima; aceleração máxima; número de paradas; número de mudanças de direção e total de tempo parado no cluster. Conforme apresentado na Figura 5.7.

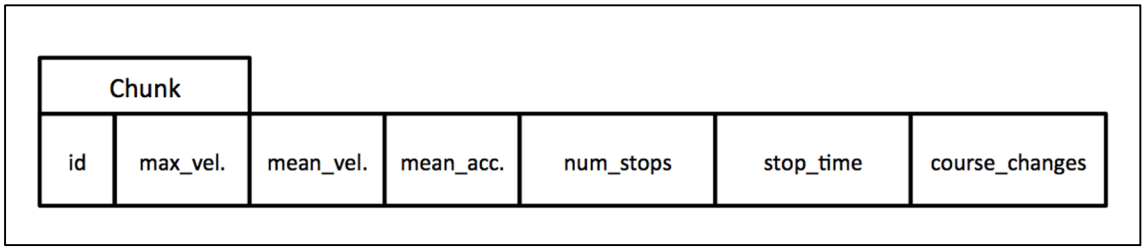


Figura 5.7 -Atributos computados de *chunks*

Na obtenção da informação de velocidade foram avaliadas duas possíveis abordagens: (i) O cálculo da velocidade levando em consideração a distância vetorial entre dois pontos consecutivos e o tempo entre eles; (ii) utilização da leitura de velocidade instantânea do GPS. Na primeira abordagem foi possível observar, que devido a imprecisão do GPS, a localização estimada variava muito em relação a posição real, levando a computação de velocidades e acelerações irreais quando a acurácia das leituras tinham menor qualidade. Este problema também foi identificado no trabalho de STENNETH, WOLFSON, et al. [29]. A opção escolhida foi utilizar a segunda abordagem, que além de apresentar resultados melhores, são de menor custo computacional.

A informação de aceleração foi obtida através da computação da variação de velocidade ponto a ponto: $(v_{p1} - v_{p0})/t$, onde t é o intervalo de tempo e v_{p1} e v_{p2} , são respectivamente, velocidade reportada no ponto imediatamente anterior e velocidade reportada no ponto atual.

Para mudança de curso**,** o curso (que é medido em graus) foi dividido por dez, onde somente a parte inteira da divisão foi considerada visando diminuir a sensibilidade a mudança de direção.

Para tempo de parada e número de paradas, foi aplicado o mesmo método utilizado na agregação de trajetórias. Este leva em consideração a distância do deslocamento ponto a ponto pela fórmula de Haversine para obter as distâncias entre cada um dos pontos de coordenadas. Assim, deslocamentos com velocidade menor que 0,4 m/s foram considerados estacionários.

Para subsidiar o entendimento completo, a Figura 5.8 representa as saídas dos processos de transformação do dado e sumarização. Esta figura deve ser interpretada de baixo para cima, levando em consideração: uma trajetória interpolada dos registros de posicionamento, que é representada pelas linhas pontilhadas; já, as mudanças de modo

de transporte pelos círculos em (a),(b) e (c). Com isto, a Figura 6 (a), representa os dados após a etapa de segmentação, cujo o objetivo é separar os deslocamentos das pausas; e a Figura 6 (b) representa os dados após uma etapa de agregação e sumarização dos dados, que divide os segmentos em *chunks* e calcula os seus atributos baseado em caraterísticas dos registros de posicionamento que a compõem.

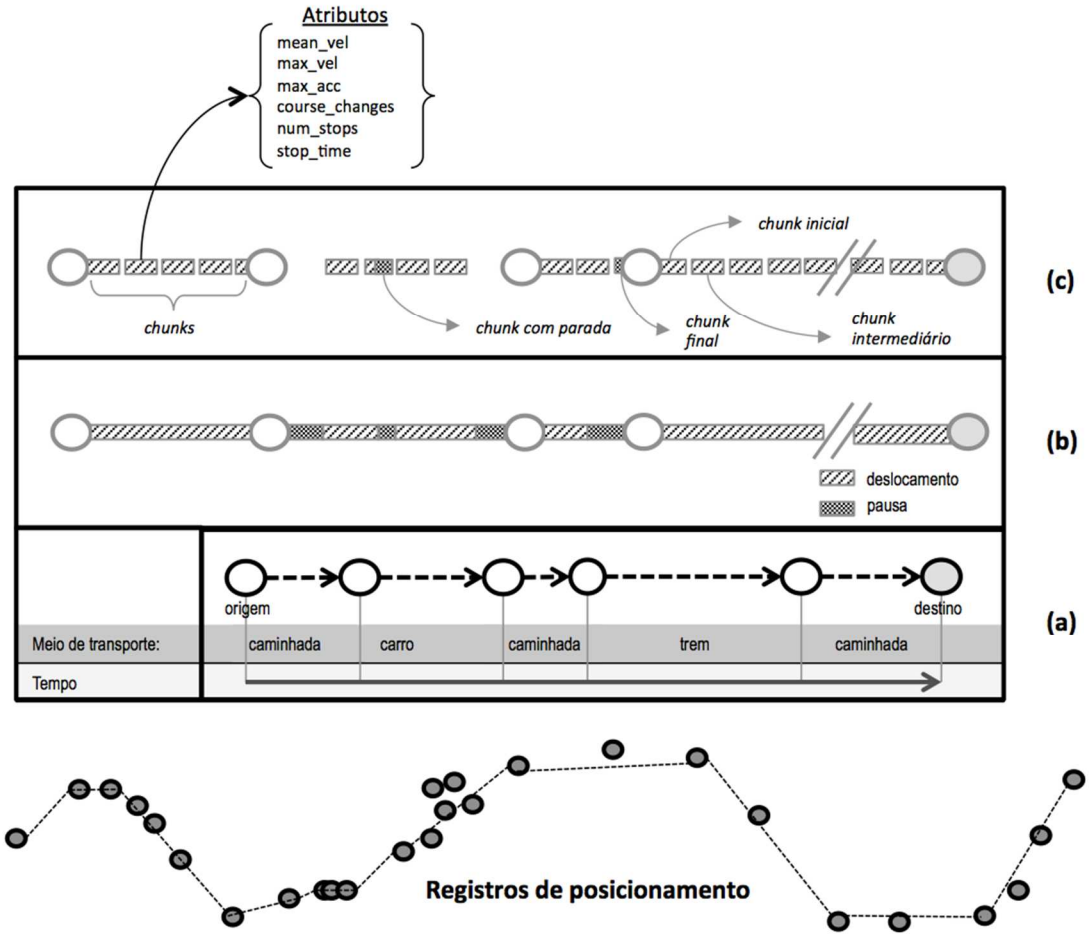


Figura 5.8 - Representação do ganho de informacional passo a passo do processo de transformação e sumarização dos dados.

5.3.3. Redução da dimensionalidade

Especificamente para este trabalho não houve necessidade de aplicar técnicas de redução da dimensionalidade dado. Tal fato se deve aos atributos que se resumem a apenas dois tipos de informações primitivas, dados estes que são relacionados ao tempo e ao espaço. Desta forma, como exemplo é possível elencá-los: velocidade instantânea; velocidade média; velocidade máxima; aceleração; aceleração média; aceleração máxima; mudanças de curso; tempo de parada e número de paradas.

5.4. Seleção de atributos

Para seleção dos atributos a serem utilizados para mineração de dados, dois procedimentos foram adotados: (i) primeiramente uma visualização dos dados coletados, que foi feita através de gráficos nos softwares R e Weka (conforme exemplo apresentado na Figura 5.9); (ii) em seguida utilizamos as funções do próprio Weka, com a parametrização padrão (apresentada na Tabela 5.2), para subsidiar a seleção dos atributos mais relevantes.

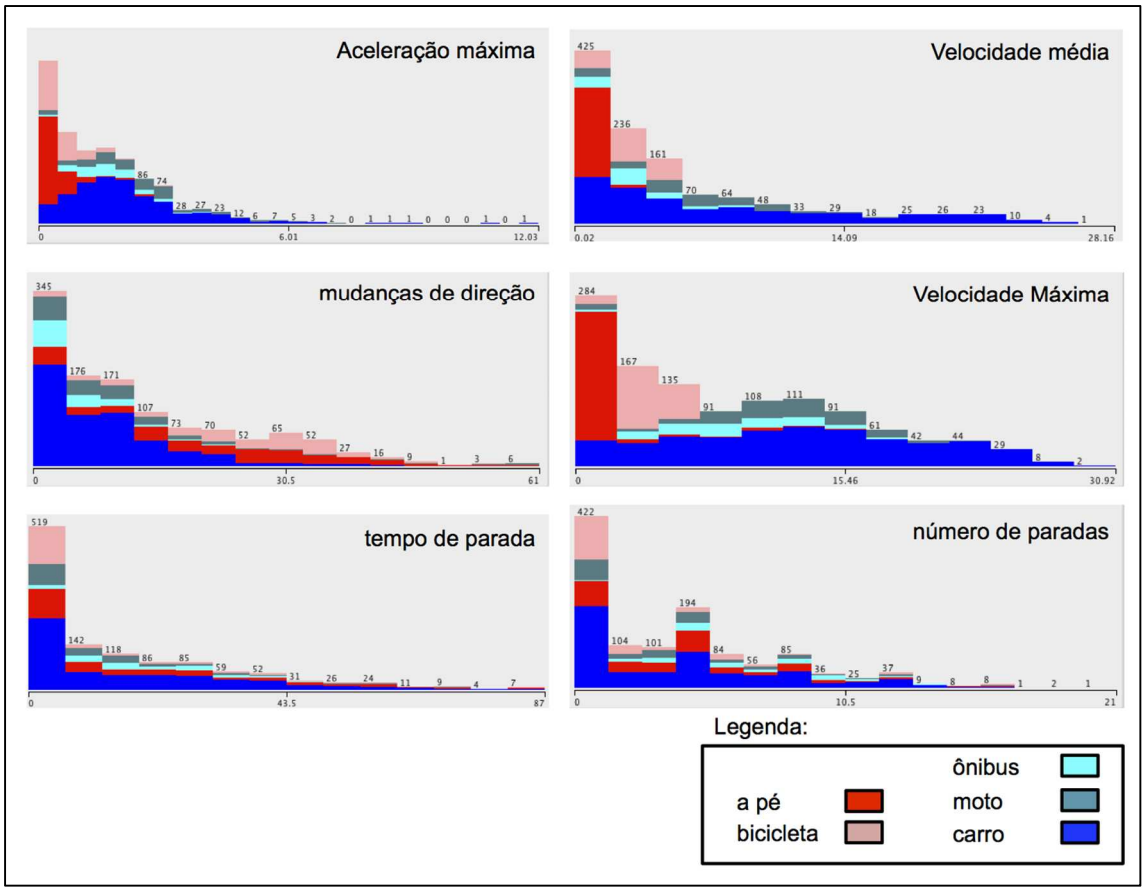


Figura 5.9 - Distribuição de frequência de modos de transporte de chunks (usando 90 segundos)

Logo a seguir uma análise dos dois resultados foi executada, levando em consideração o ganho informacional, a simplicidade e o grau de heterogeneidade entre os atributos; assim culminando na seleção dos atributos propriamente dita. Cabe ressaltar, que a investigação de todas as combinações de atributos, muitas das vezes é muito custosa e tende a ser impossível na maioria dos casos, conforme apresentado por WITTEN e HALL em [35].

A Figura 5.10 apresenta um diagrama com as atividades aplicadas para seleção de atributos. Ainda assim, para decidir na escolha dos atributos foram feitos alguns experimentos através do *Weka Experimenter* para validar a combinação destes atributos. Desta forma, os atributos que foram apresentados na etapa de agregação e sumarização, foram avaliados utilizando o processo apresentado na Figura 5.10. Sendo que na etapa de análise por algoritmos, foram utilizados os algoritmos apresentados na Tabela 5.4. Com isto, os seguintes atributos foram selecionados para utilização na etapa de mineração de dados: velocidade máxima, aceleração máxima, mudanças de direção.

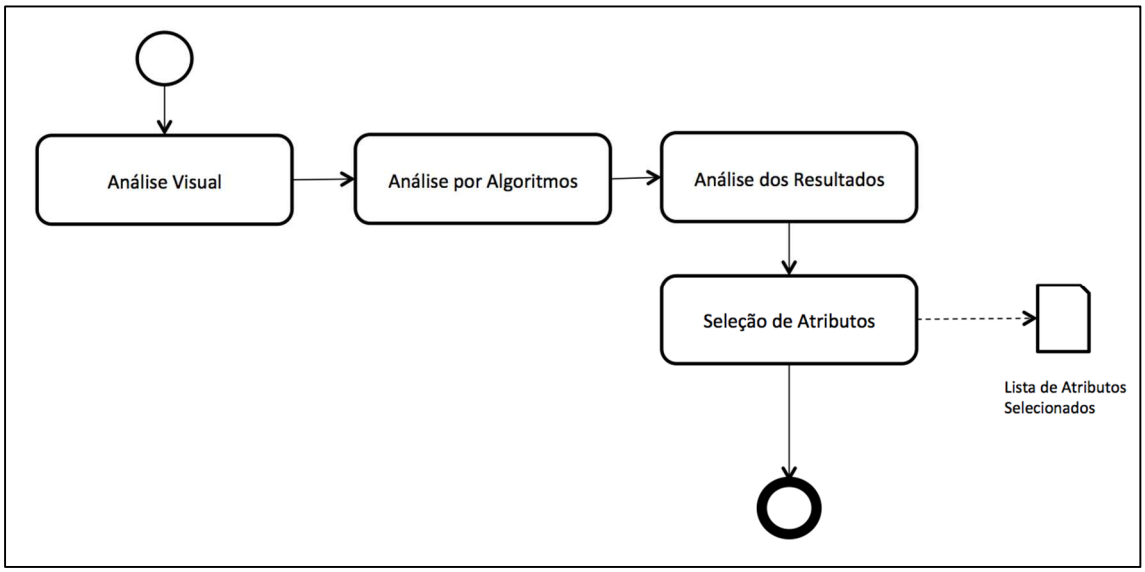


Figura 5.10 - Diagrama do processo aplicado para seleção de atributos

Tabela 5.2 - Algoritmos utilizados para seleção de atributos na ferramenta Weka.

1	InfoGainAttributeEval	Ranker
	Parâmetros: -T -1.7976931348623157E308 -N -1	
2	CfsSubsetEval	BestFirst
	Parâmetros: -D1 –N5	

5.5. Definição da função de mineração e escolha dos algoritmos

Durante a etapa de coleta de dados, os usuários foram solicitados a fornecer marcações para suas movimentações. Estas permitiram que um conjunto de registros de posicionamento pudessem ser associados ao meio de transporte que foi por ele utilizado. Tal informação, pode então ser utilizada para a etapa de mineração de dados, permitindo a utilização da técnica de classificação. Funções de classificação, são as mais comumente empregadas em mineração de dados. Uma vez que temos um conjunto de registros de posicionamento classificados, podemos utilizá-los para alimentar algoritmos de classificação e, assim, classificar os *chunks* de movimentação de acordo com a função que foi aprendida. Depois que a função de mineração de dados foi definida, é possível então selecionar quais serão os algoritmos de mineração que serão aplicados. A escolha de algoritmos, foi feita com base nos trabalhos [40], [29], [25] e [36]. Informações detalhadas sobre cada um dos mecanismos de classificação podem ser obtidas em nos trabalhos, [35], [30] e [36]. O critério adotado para a seleção dos algoritmos foi investigar quais os algoritmos que apresentaram um bom resultado em trabalhos anteriores, assim como testar outros algoritmos que ainda não foram aplicados. Estes algoritmos adicionais, foram selecionados dos mais utilizados, a partir do trabalho de WU, KUMAR, et al. [36]. Como resultado da aplicação destes critérios obtivemos a seguinte lista de algoritmos: (i) Redes Bayesianas, (ii) Naive Bayes, (iii) SVM, (iv) Multilayer Perceptron; (v) Decision Tree; (vi) Random Forest; (vii) Random Trees; (viii) K-Means; (ix) K-NN; (x) Ada boost. Sendo que, CHMM, Conditional Random Fields e K-Means, não foram avaliados por não serem compatíveis com a abordagem aqui utilizada. A Figura 6.12 apresenta o conjunto de algoritmos testados.

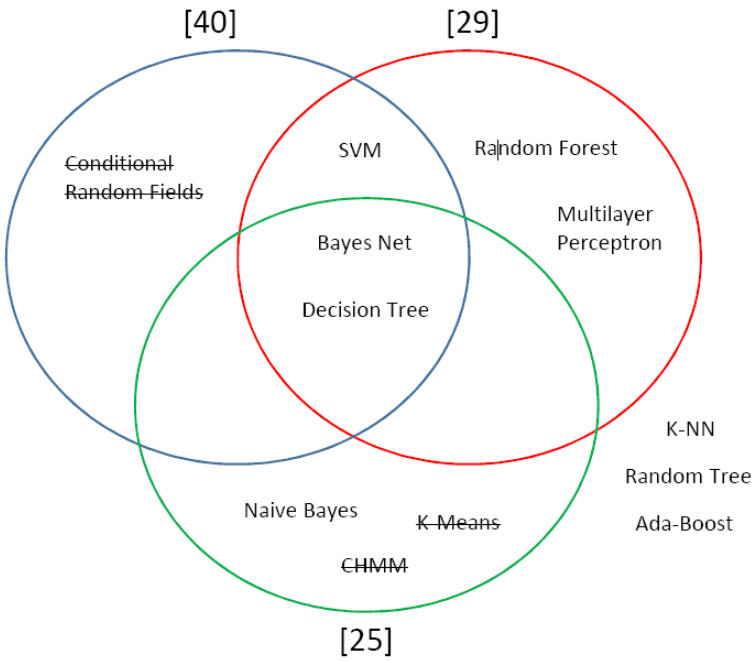


Figura 5.11 - Diagrama de Venn representando os algoritmos testados neste estudo.

5.6. Procedimentos utilizados para avaliação dos classificadores

O processo de avaliação dos classificadores foi feito com uso da ferramenta *Weka*, por meio dos módulos *Experimenter*, *Knowledge Flow* e *Explorer*. Todos os algoritmos utilizaram suas configurações originais, exceto quando foram aplicadas técnicas *ensemble* com múltiplos classificadores. Para estes casos foi usada uma combinação baseada nos classificadores de melhor resultado, levando em consideração a combinação entre as classes de algoritmos (como, por exemplo: classificadores de classes diferentes e com matriz de confusão complementares).

Para a avaliação dos mecanismos de classificação utilizou-se o método de *cross-validation* 10-*fold* (conforme apresentado no Capítulo III).

Para a mineração de dados foi utilizada a abordagem indutiva construtivista, baseada em múltiplas iterações, onde, ao término de cada iteração, dá-se uma etapa de agregação do conhecimento. Com isto, o conhecimento gerado ao final de cada iteração foi aplicado na orientação das perguntas a serem respondidas nas iterações seguintes.

Na primeira iteração, o objetivo foi definir o melhor tamanho para o *chunk*, dentro de três possíveis tamanhos investigados: 60, 90 e 120 segundos. De forma paralela a este

objetivo, também foi analisada a performance de classificação dos algoritmos como um todo e resultado de aplicação de técnicas *ensemble* e tratamento para classes heterogêneas (apresentadas no Capítulo III).

Na segunda iteração buscou-se verificar a existência de melhorias para classificação ao se adotar uma classificação em dois níveis.

Já na terceira iteração se investigou a possibilidade de classificar em duas etapas com as seguintes abordagens: (i) o modelo a-pé/não-a-pé; versus o modelo (ii) motorizado/não-motorizado. O primeiro modelo de classificação consiste em separar primeiramente as movimentações a pé, que apresentou uma melhor precisão de classificação, para depois numa segunda etapa, classificar dentre as outras possíveis modalidades. Já o segundo modelo consiste em separar os modos de transporte motorizados dos não motorizados; estes que em geral possuem perfis de movimentação mais similares entre si. Na segunda etapa deste modelo, aplica-se o algoritmo de melhor performance para cada uma das classificações. A Figura 5.12 apresenta as questões investigadas a cada iteração.

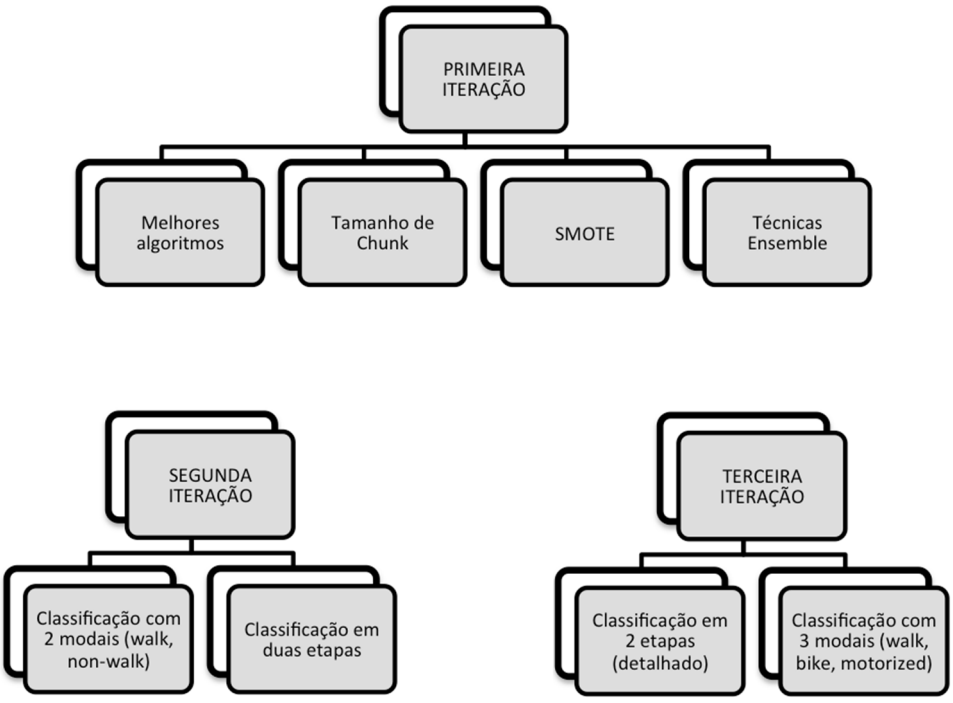


Figura 5.12 – Diagrama hierárquico apresentando os objetivos investigados em cada iteração do processo de mineração de dados

5.6.1. Configuração da primeira iteração

Para a primeira iteração, os seguintes **objetivos** foram estabelecidos: **(**i) descobrir, dentro dos três tamanhos de *chunks* investigados, o que apresenta melhor performance para classificação; (ii) avaliar a performance dos classificadores para classificar entre todos os tipos de dados; (iii) avaliar a performance do *adaboost* com os algoritmos compatíveis de melhor resultado; (iv) avaliar se existe algum impacto ao mecanismo de aprendizado pelo uso de amostra desbalanceada. Para isto, os seguintes **testes** foram definidos e aplicados: teste de classificação dos *chunks* para os tamanhos: (a) 60, (b) 90 e (c) 120 segundos, dentre todos os modos de transportes investigados, utilizando todos os algoritmos; (d) teste com uso de amostra menos desbalanceada com uso de amostras sintéticas (SMOTE); (e) testes aplicando dois classificadores utilizando técnicas ensemble (*boosting*, *stacking*, *bagging*).

5.6.2. Configuração da segunda iteração

Para a segunda iteração, os seguintes objetivos foram estabelecidos: (i) análise da abordagem em uma etapa versus a abordagem em duas etapas (levando em consideração os resultados da saída da primeira iteração); (ii) adicionalmente, buscou-se avaliar a possibilidade de classificação do modelo entre andando a pé versus andando não a pé. Para isto, os seguintes testes foram definidos e aplicados: (i) teste de classificação dos *chunks* de tamanho 90 segundos (melhor tamanho de *chunk*, identificado na primeira iteração) classificando entre andando a pé versus andando não a pé; (ii) teste de classificação dos chunks de tamanho 90 segundos classificando entre deslocamento motorizado e deslocamento não-motorizado.

5.6.3. Configuração da terceira iteração

Para a terceira iteração, o objetivo definido foi identificar a melhor abordagem para classificação dos *chunks*. Para isto, os seguintes testes foram definidos e aplicados: (i) teste de classificação dos *chunks* de tamanho 90 segundos, entre andando a pé e de bicicleta; (ii) teste de classificação dos *chunks* de tamanho 90 segundos, entre carro, motocicleta e ônibus e (iii) teste de classificação dos *chunks* de tamanho 90 segundos, entre carro, motocicleta e ônibus. Também foi necessário incluir nesta etapa: (i) revisão dos testes de classificação dos *chunks* de tamanho 90 segundos, entre deslocamento motorizado e deslocamento não-motorizado; (ii) revisão dos testes de classificação dos *chunks* de tamanho 90 segundos, entre andando a pé versus andando não a pé e (iii) a

classificação de chunks com três modais, andando, bicicleta e motorizado (este inclui, motocicleta, ônibus e carro). A Tabela 5.3 apresenta o número de exemplos de cada classe utilizados nos testes.

Tabela 5.3 - Número de exemplos usados nos testes.

	60 segundos	90 segundos	120 segundos	90s+SMOTE *	90s+SMOTE **
andando	355	230	162	230	230
bicicleta	270	177	131	177	177
ônibus	145	94	69	188	188
motocicleta	217	144	103	144	288
carro	797	528	379	528	528
TOTAL de exemplos	**1784**	**1173**	**844**	**1267**	**1411**
* aumento em 100% para exemplos de ônibus. ** aumento de 100% para exemplos de motocicleta e ônibus.					

5.7 Críticas ao processo de descoberta de conhecimento em banco de dados

Esta seção apresenta uma breve análise crítica dos problemas encontrados na etapa de descoberta de conhecimento em banco de dados.

5.7.1. Crítica ao processo de segmentação de trajetórias

O mecanismo de detecção de pausas, mostrou não ter uma precisão satisfatória. Muitas pausas não foram detectadas (devido a variações da leituras registro a registro). A redução da taxa de amostragem tende a amenizar esse problema, os registros passariam a ter uma maior distância entre eles, minimizando o efeito da variação na leitura. Cabe considerar que o uso de técnicas de *clustering*, aplicadas a uma etapa de pré-classificação, como é o caso do CB-SMOT, mais se adequa para trabalhos onde se analisa trajetórias completas. Quanto ao impacto da imprecisão da detecção de pausas, este materializa-se posteriormente, como a existência de períodos de pausa dentro dos segmento de deslocamento e, por consequência, passam a fazer parte dos *chunks*. Para minimizar tal efeito, foi necessário um tratamento posterior antes de se aplicar a classificação. O objetivo da filtragem foi excluir as pausas que não foram propriamente detectadas dentro dos segmentos, reduzindo-se assim o seu impacto para o treinamento.

5.7.2. Críticas quanto ao processo de agregação de dados

Quanto a agregação dos dados, apenas a agregação temporal foi analisada. Ficou para trabalhos futuros, a possibilidade de explorar outras técnicas de agregação dos registros de movimentação. A agregação temporal foi escolhida baseada nas análise das vantagens e desvantagens, embora isso não significa que não se possa obter bons resultados com a utilização de outras técnicas.

CAPÍTULO VI -Análise dos Resultados

Este capítulo tem o objetivo de apresentar os resultados obtidos na etapa de mineração de dados. Sua organização dá-se da seguinte forma: na Seção 6.1 são apresentados os resultados da etapa de mineração e na Seção 6.2 é apresentada a análise destes resultados.

6.1. Resultados obtidos da etapa de mineração de dados

Os critérios de avaliação utilizados foram baseados nas métricas *precision accuracy* e *recall accuracy*, que são sumarizadas pelo *F-Measure*. É interessante observar as três métricas de forma combinada com a *confusion matrix*[16]. Isto é especialmente relevante, quando se utiliza amostras não balanceadas - ou seja, aquelas amostras que possuem um grau de distribuição não homogêneo. As taxas de *precision* e *recall accuracy* são mais adequadas para avaliar classificações (conforme apresentado no Capítulo III). Além disto, deve-se analisar caso a caso, as ocorrências de cada classe, isto é, as taxas de *precision* e *recall* accuracy, associadas às ocorrências de cada modo de transporte da classificação.

Foram utilizadas três iterações, as quais após a aplicação de cada uma delas, apresentou-se um aumento gradual do conhecimento construído. Conhecimento este, dado por uma etapa de análise preliminar dos resultados, após sua respectiva interação. Na primeira iteração, os objetivos foram: investigar os melhores resultados para classificação de *chunks* com tamanhos: 60 segundos, 90 segundos e 120 segundos; descobrir os

[16] Matriz de confusão ou matriz de classificação: (*confusion matrix* ou *contingency table)* consiste em uma tabela que pela sua organização, permite avaliar a performance de algoritmos de aprendizado de máquina. Este é descrito por TAN, STEINBACH, et al. [30].

algoritmos com melhores desempenhos; investigar a possibilidade de se obter benefícios quando a amostra é melhor balanceada (usando amostras sintéticas); e por último verificar a aplicação das técnicas *ensemble* para combinação de classificadores. Segue a apresentação do processo de análise levando em consideração os critérios aqui apresentados.

6.1.1. Resultados da primeira iteração

A primeira iteração, inicialmente utilizou três testes de classificação, onde foram testados os *chunks* de 60, 90 e 120 segundos, sendo que todos os algoritmos do escopo proposto (Seção 5.4) foram testados. Os resultados são apresentados na Tabela 6.1.

Tabela 6.1 - Resultados sumarizados da classificação de *chunks* de 60, 90 e 120 segundos usando diferentes algoritmos

PRIMEIRA ITERAÇÃO

A 60 segundos

	J.48	Random Forest	Random Tree	Nbayes	BayesNet
Precision	0.619	0.618	0.612	0.628	**0.672**
Recall	0.671	0.651	0.606	0.667	0.684
FMeasure	0.624	0.629	0.609	0.628	0.609

	LibSVM	Perceptron	iBk	Decision Table	SMO
Precision	0.622	0.617	0.606	**0.645**	0.562
Recall	0.688	**0.691**	0.599	0.684	0.68
FMeasure	0.622	0.619	0.602	0.62	0.603

B 90 segundos

	J.48	Random Forest	Random Tree	Nbayes	BayesNet
Precision	0.65	0.631	0.61	**0.659**	0.635
Recall	0.698	0.663	0.615	0.685	**0.711**
FMeasure	0.654	0.643	0.612	**0.65**	0.634

	LibSVM	Perceptron	iBk	Decision Table	SMO
Precision	0.615	0.614	0.604	**0.689**	0.561
Recall	0.7	**0.706**	0.603	0.704	0.677
FMeasure	0.632	0.633	0.603	0.631	0.597

C 120 segundos

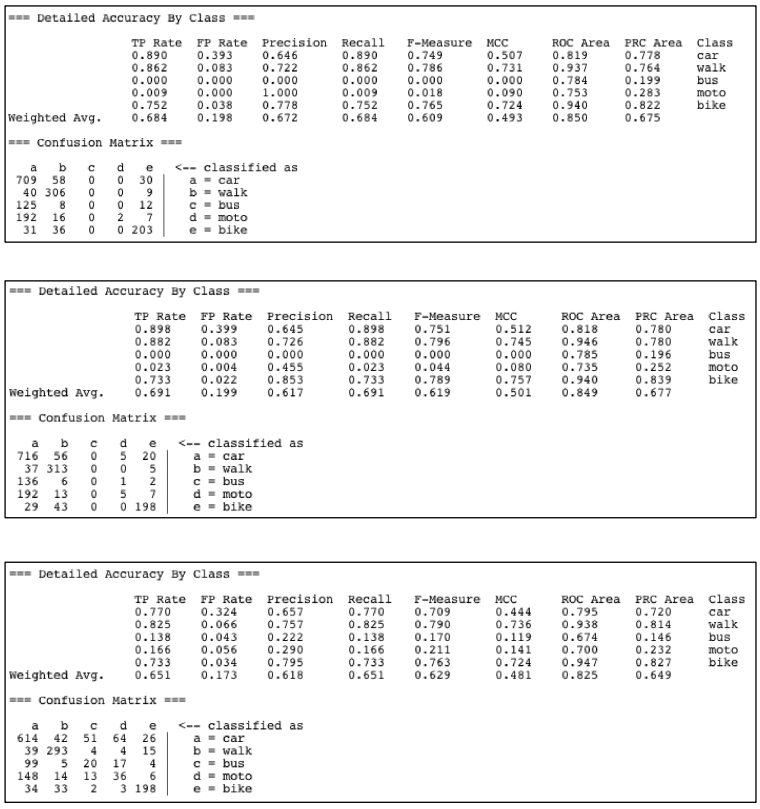
	J.48	Random Forest	Random Tree	Nbayes	BayesNet
Precision	0.62	0.639	0.604	**0.66**	0.615
Recall	0.674	0.673	0.588	0.688	**0.706**
FMeasure	0.628	0.651	0.595	**0.661**	0.631

	LibSVM	Perceptron	iBk	Decision Table	SMO
Precision	0.654	**0.659**	0.627	0.594	0.551
Recall	0.699	**0.71**	0.624	0.701	0.656
FMeasure	**0.65**	0.648	0.626	0.626	0.571

Resultados para a análise de *chunks* com 60 segundos

Para os *chunks* de 60 segundos, o melhor *precision accuracy* (ou melhor, a média ponderada do *precision accuracy* de cada classe) foi apresentada pela aplicação do algoritmo *bayes-net* (redes bayesianas), com 0.672, mas é possível notar pela *confusion matrix*, apresentada na Figura 6.1-a, um baixo grau de eficácia para detectar ônibus e motocicletas, uma vez que este algoritmo classificou muitas das instâncias como classe carro. Já o melhor *recall accuracy* foi apresentado pelo *multilayer perceptron* com 0.691, que também não conseguiu um bom resultado para moto e ônibus (Figura 6.1-b). Quando se leva em consideração o *F-measure*, este aponta para o *random forest* que

parece ser mais interessante, com o valor de 0,629, refletindo o *recall* e o *precision accuracy* como 0,651 e 0,618, respectivamente (Figura 6.1-c). Como conclusão, é possível eleger o algoritmo *random forest* como o mais adequado para classificar os *chunks* de 60 segundos, mesmo este não conseguindo separar satisfatoriamente registros de motocicletas e ônibus. Ainda assim, o resultado de classificação entre todas as classes se apresenta um pouco melhor que os apresentados pelos demais algoritmos.



```
=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.890    0.393    0.646      0.890    0.749      0.507    0.819     0.778     car
                 0.862    0.083    0.722      0.862    0.786      0.731    0.937     0.764     walk
                 0.000    0.000    0.000      0.000    0.000      0.000    0.784     0.199     bus
                 0.009    0.000    1.000      0.009    0.018      0.090    0.753     0.283     moto
                 0.752    0.038    0.778      0.752    0.765      0.724    0.940     0.822     bike
Weighted Avg.    0.684    0.198    0.672      0.684    0.609      0.493    0.850     0.675

=== Confusion Matrix ===

   a   b   c   d   e   <-- classified as
 709  58   0   0  30 |   a = car
  40 306   0   0   9 |   b = walk
 125   8   0   0  12 |   c = bus
 192  16   0   2   7 |   d = moto
  31  36   0   0 203 |   e = bike
```

(a) Bayes-net


```
=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.898    0.399    0.645      0.898    0.751      0.512    0.818     0.780     car
                 0.882    0.083    0.726      0.882    0.796      0.745    0.946     0.780     walk
                 0.000    0.000    0.000      0.000    0.000      0.000    0.785     0.196     bus
                 0.023    0.004    0.455      0.023    0.044      0.080    0.735     0.252     moto
                 0.733    0.022    0.853      0.733    0.789      0.757    0.940     0.839     bike
Weighted Avg.    0.691    0.199    0.617      0.691    0.619      0.501    0.849     0.677

=== Confusion Matrix ===

   a   b   c   d   e   <-- classified as
 716  56   0   5  20 |   a = car
  37 313   0   0   5 |   b = walk
 136   6   0   1   2 |   c = bus
 192  13   0   5   7 |   d = moto
  29  43   0   0 198 |   e = bike
```

(b) Multilayer-percetron

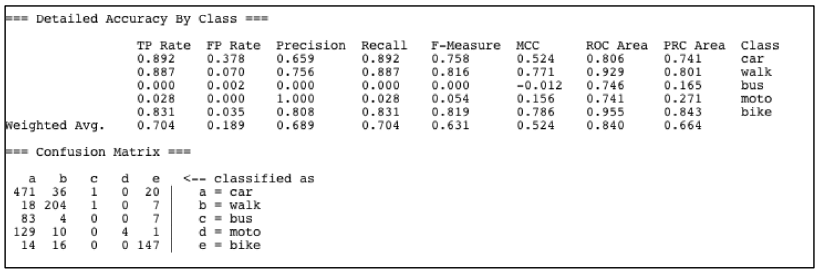

```
=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.770    0.324    0.657      0.770    0.709      0.444    0.795     0.720     car
                 0.825    0.066    0.757      0.825    0.790      0.736    0.938     0.814     walk
                 0.138    0.043    0.222      0.138    0.170      0.119    0.674     0.146     bus
                 0.166    0.056    0.290      0.166    0.211      0.141    0.700     0.232     moto
                 0.733    0.034    0.795      0.733    0.763      0.724    0.947     0.827     bike
Weighted Avg.    0.651    0.173    0.618      0.651    0.629      0.481    0.825     0.649

=== Confusion Matrix ===

   a   b   c   d   e   <-- classified as
 614  42  51  64  26 |   a = car
  39 293   4   4  15 |   b = walk
  99   5  20  17   4 |   c = bus
 148  14  13  36   6 |   d = moto
  34  33   2   3 198 |   e = bike
```

(c) Random forest

Figura 6.1 - Resultados para algoritmos que apresentaram melhor resultado para *chunks* de 60 segundos (a) *bayes-net,* (b) *multilayer percepton*, (c) *random forest*.

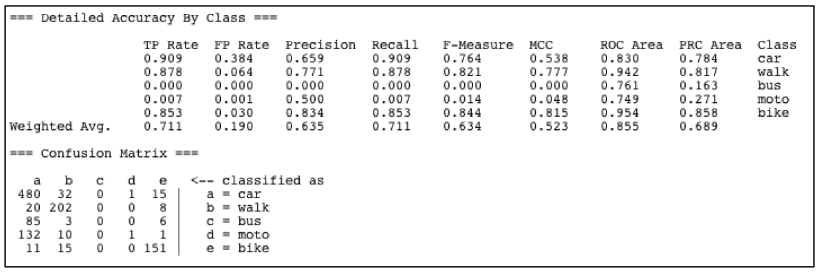
Resultados para a análise de chunks com 90 segundos

Para os chunks de 90 segundos, a melhor precisão foi apresentada pelo algoritmo *decision table* com 0,689 de *precision accuracy*. Pela análise da *confusion matrix* (Figura 6.2-a), pode-se descobrir que este algoritmo não consegue classificar bem ônibus e motocicleta, pois ele tende a confundi-los com carros. Já o melhor *recall acuracy* foi apresentado por *bayes-net*, com 0,711 (Figura 6.2-b). O melhor *F-measure* foi dado pelo J.48 com 0,654, que também não classificou bem ônibus e motocicletas (Figura 6.2-c).



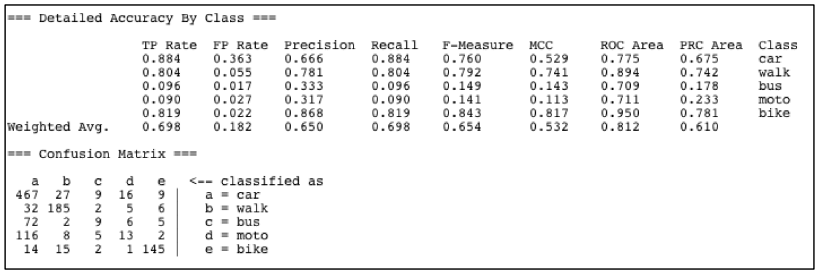
```
=== Detailed Accuracy By Class ===

                TP Rate   FP Rate   Precision   Recall   F-Measure   MCC      ROC Area  PRC Area  Class
                0.892     0.378     0.659       0.892    0.758       0.524    0.806     0.741     car
                0.887     0.070     0.756       0.887    0.816       0.771    0.929     0.801     walk
                0.000     0.002     0.000       0.000    0.000       -0.012   0.746     0.165     bus
                0.028     0.000     1.000       0.028    0.054       0.156    0.741     0.271     moto
                0.831     0.035     0.808       0.831    0.819       0.786    0.955     0.843     bike
Weighted Avg.   0.704     0.189     0.689       0.704    0.631       0.524    0.840     0.664

=== Confusion Matrix ===

   a   b   c   d   e   <-- classified as
 471  36   1   0  20 |   a = car
  18 204   1   0   7 |   b = walk
  83   4   0   0   7 |   c = bus
 129  10   0   4   1 |   d = moto
  14  16   0   0 147 |   e = bike
```

(a)$Decision$table$



```
=== Detailed Accuracy By Class ===

                TP Rate   FP Rate   Precision   Recall   F-Measure   MCC      ROC Area  PRC Area  Class
                0.909     0.384     0.659       0.909    0.764       0.538    0.830     0.784     car
                0.878     0.064     0.771       0.878    0.821       0.777    0.942     0.817     walk
                0.000     0.000     0.000       0.000    0.000       0.000    0.761     0.163     bus
                0.007     0.001     0.500       0.007    0.014       0.048    0.749     0.271     moto
                0.853     0.030     0.834       0.853    0.844       0.815    0.954     0.858     bike
Weighted Avg.   0.711     0.190     0.635       0.711    0.634       0.523    0.855     0.689

=== Confusion Matrix ===

   a   b   c   d   e   <-- classified as
 480  32   0   1  15 |   a = car
  20 202   0   0   8 |   b = walk
  85   3   0   0   6 |   c = bus
 132  10   0   1   1 |   d = moto
  11  15   0   0 151 |   e = bike
```

(b)$Bayes1net$



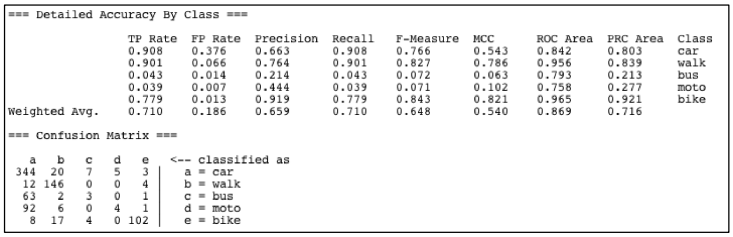
```
=== Detailed Accuracy By Class ===

                TP Rate   FP Rate   Precision   Recall   F-Measure   MCC      ROC Area  PRC Area  Class
                0.884     0.363     0.666       0.884    0.760       0.529    0.775     0.675     car
                0.804     0.055     0.781       0.804    0.792       0.741    0.894     0.742     walk
                0.096     0.017     0.333       0.096    0.149       0.143    0.709     0.178     bus
                0.090     0.027     0.317       0.090    0.141       0.113    0.711     0.233     moto
                0.819     0.022     0.868       0.819    0.843       0.817    0.950     0.781     bike
Weighted Avg.   0.698     0.182     0.650       0.698    0.654       0.532    0.812     0.610

=== Confusion Matrix ===

   a   b   c   d   e   <-- classified as
 467  27   9  16   9 |   a = car
  32 185   2   5   6 |   b = walk
  72   2   9   6   5 |   c = bus
 116   8   5  13   2 |   d = moto
  14  15   2   1 145 |   e = bike
```

(c)$J.48$

Figura 6.2- Resultados para algoritmos que apresentaram melhor resultado para *chunks* de 90 segundos, *decision table* (a), *Bayes-net*(b), J.48 (c).

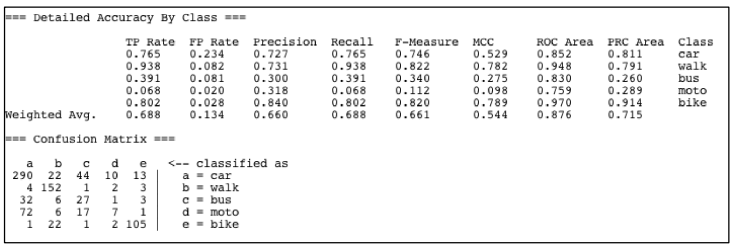
Resultados para a análise de chunks com 120 segundos

O algoritmo *multilayer perceptron* apresentou o melhor resultado para *chunks* de 120 segundos, tanto para *precision accuracy*, quanto para *recall accuracy*, respectivamente 0,659 e 0,71 (Figura 6.3-a). No entanto, a classificação de ônibus e motocicleta foi mais uma vez confundida com a classe carro. Já o melhor *F-measure* foi apresentado pelo algoritmo naive-Bayes: 0,654, onde a classificação de ônibus ficou um pouco superior ao *multi-layer perceptron* (Figura 6.3-b).



```
=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.908    0.376    0.663      0.908    0.766      0.543    0.842     0.803     car
                 0.901    0.066    0.764      0.901    0.827      0.786    0.956     0.839     walk
                 0.043    0.014    0.214      0.043    0.072      0.063    0.793     0.213     bus
                 0.039    0.007    0.444      0.039    0.071      0.102    0.758     0.277     moto
                 0.779    0.013    0.919      0.779    0.843      0.821    0.965     0.921     bike
Weighted Avg.    0.710    0.186    0.659      0.710    0.648      0.540    0.869     0.716

=== Confusion Matrix ===

   a   b   c   d   e   <-- classified as
 344  20   7   5   3 |   a = car
  12 146   0   0   4 |   b = walk
  63   2   3   0   1 |   c = bus
  92   6   0   4   1 |   d = moto
   8  17   4   0 102 |   e = bike
```

(a) Multilayer-percetron



```
=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.765    0.234    0.727      0.765    0.746      0.529    0.852     0.811     car
                 0.938    0.082    0.731      0.938    0.822      0.782    0.948     0.791     walk
                 0.391    0.081    0.300      0.391    0.340      0.275    0.830     0.260     bus
                 0.068    0.020    0.318      0.068    0.112      0.098    0.759     0.289     moto
                 0.802    0.028    0.840      0.802    0.820      0.789    0.970     0.914     bike
Weighted Avg.    0.688    0.134    0.660      0.688    0.661      0.544    0.876     0.715

=== Confusion Matrix ===

   a   b   c   d   e   <-- classified as
 290  22  44  10  13 |   a = car
   4 152   1   2   3 |   b = walk
  32   6  27   1   3 |   c = bus
  72   6  17   7   1 |   d = moto
   1  22   1   2 105 |   e = bike
```

(b) Naive Bayes

Figura 6.3 – Resultado da análise de chunks de 120 segundos para multilayer perceptron (a) e naive Bayes (b).

Foi possível concluir, logo na primeira etapa da primeira iteração, que os algoritmos que apresentaram melhor resultado para classificação foram: *bayes-net, naive-Bayes, multilayer perceptron, decision table, J.48* e *random forest*. Os resultados dos *chunks* de noventa segundos foram muito próximos aos *chunks* de cento e vinte segundos (quando se analisa o *precision e recall accuracy*). Já pela análise do *F-measure*, este aponta para os *chunks* de 120 segundos, mas apresenta apenas um ganho marginal quando comparado ao de 90 segundos. Portanto, com os valores são muito próximos, entre *chunks* 90 e 120, foi decidida a utilização dos *chunks* de 90 segundos, uma vez que permitem que a classificação aconteça mais rapidamente que o *chunk* de 120 segundos, trazendo benefícios adicionais para as possíveis futuras aplicações.

Utilização de Técnicas Ensemble.

A segunda etapa desta iteração se preocupou em analisar o emprego de técnicas *ensemble* de classificação (cujo a síntese dos resultados está apresentada na Tabela 6.2). Esta etapa levou a obtenção dos seguintes resultados: (i) a utilização da técnica de *boosting* com a utilização do algoritmo *adaboost*, combinado[17] com J.48, *decision table* e *decision stump* não apresentaram ganhos significativos quando comparadas a técnicas com somente um algoritmo, considerando amostras de *chunks* de 60, 90 e 120

[17] Estes algoritmos foram selecionados pois apresentaram um bom resultado para classificação com *chunks* de 90 segundos, o J.48 e o *decision table* foram os algoritmos com melhor resultado de classificação com 90 segundos, já o *decision stump* foi usado por ser default ao *adaboost* no Weka.

segundos. (ii) Também não houve benefícios com o uso de *stacking* combinando redes bayesianas com *multi-layer perceptron* nem redes bayesianas com árvores de decisão (J.48). (iii) Quanto a aplicação da técnica de *bagging*, esta também não apresentou resultados significantes para *random forest*, *naive bayes* e redes bayesianas.

Tabela 6.2 - Resultado da aplicação de técnicas ensemble para a classificação de chunks considerando todos os modais.

APLICAÇÃO DE TÉCNICAS ENSEMBLE

BOOSTING

D	*60 segundos*	*90 segundos*	*120 segundos*	*90 segundos*	*90 segundos*
	AdaBoost+DS	AdaBoost+DS	AdaBoost+DS	AdaBoost+J48	AdaBoost+DT
Precision	0.376	0.377	0.38	0.63	0.689
Recall	0.525	0.537	0.551	0.657	0.704
F-Measure	0.424	0.432	0.441	0.64	0.631

STACKING

D	*90 segundos*	*90 segundos*	*90 segundos*
	BayesNet+J.48	BayesNet+Perceptron	Perceptron+J.48
Precision	0.57	0.648	0.633
Recall	0.692	0.617	0.692
F-Measure	0.62	0.613	0.624

BAGGING	*90 segundos*	*90 segundos*	*90 segundos*
D			
	Random Forest	Naive Bayes	BayesNet
Precision	0.629	0.659	0.677
Recall	0.675	0.685	0.714
F-Measure	0.645	0.65	0.641

Redução da heterogeneidade de amostras de classes.

A partir dos dados apresentados na Tabela 6.3, pode-se dizer que não houve ganhos significantivos com a redução da heterogeneidade de classes das amostras pela aplicação de *oversampling* pela técnica de SMOTE. Esta foi utilizada para analisar os *chunks* de 90 segundos com os algoritmos J.48 e *decision table*. A técnica que utiliza a criação de exemplos sintéticos foi utilizada com 100% de aumento para ônibus e em seguida com 100% para motocicleta (estas classes apresentaram um menor número de ocorrências). Portanto, pode-se concluir que o impacto das classes heterogêneas ao

algoritmos testados não foi determinante para aumentar ou reduzir os resultados de forma significante.

Tabela 6.3 - Resultado de aplicação de técnicas SMOTE para *chunks* de 90s.

SMOTE D	*90 segundos*	*90 segundos*	*90 segundos*
	SMOTE + J.48*	SMOTE + DT**	SMOTE + Perceptron
Precision	0.608	0.536	0.536
Recall	0.649	0.574	0.574
F-Measure	0.619	0.532	0.532

* aumento de 100% para exemplos de ônibus.
** aumento de 100% para exemplos de ônibus e de motocicleta.

6.1.2. Resultados da segunda iteração

Para a segunda iteração os mesmos *chunks* de 90 segundos foram utilizados, sendo que a etiqueta de classificação foi alterado para apresentar os valores *walk* e *non-walk*, respectivamente para os modos de deslocamento a pé e não a pé. Separar os *chunks* de movimentações dessa forma, apresentou um resultado bem interessante, indicando a possibilidade de se classificar de forma incremental, isso é separando os modais de transporte em etapas sucessivas de classificação. Conforme pode ser observado na Tabela 6.4, diversos algoritmos apresentaram 100% de *precision* e *recall accuracy*, são eles: *SMO, perceptron, iBk, adaboost, decision table, naive-bayes, bayes-net, random forest* e *J.48*.

Da mesma forma que a alteração das etiquetas foi feita na primeira etapa desta iteração, novamente foi aplicada, mas para a classificação dos *chunks* de 90 segundos entre motorizado e não motorizado. O melhor resultado foi obtido pelo algoritmo SMO, com os valores de *precision* e *recall accuracy* respectivamente: 0,923 e 0,921; e com F-mesure 0,921.

Tabela 6.4 - Resultados sumarizados para a segunda iteração.

SEGUNDA ITERAÇÃO

A 90 segundos *a pé / não apé*

	J.48	Random Forest	Random Tree	Nbayes	BayesNet	LibSVM
Precision	**1**	**1**	**0.999**	0.985	0.996	0.972
Recall	**1**	**1**	**0.999**	0.085	0.996	0.972
F-Measure	**1**	**1**	**0.999**	0.985	0.996	0.972

	Perceptron	iBk	AdaBoost*	Decision Table	SMO
Precision	**1**	**1**	**1**	**1**	**1**
Recall	**1**	**1**	**1**	**1**	**1**
F-Measure	**1**	**1**	**1**	**1**	**1**

* Usando AdaBoot+DecisonStump

B 90 segundos *motorizado / não-motorizado*

	J.48	Random Forest	Random Tree	Nbayes	BayesNet	LibSVM
Precision	0.908	0.903	0.881	**0.916**	0.911	0.912
Recall	0.906	0.904	0.881	0.907	0.907	**0.911**
F-Measure	0.907	0.904	0.881	0.909	0.908	**0.911**

	Perceptron	iBk	AdaBoost*	Decision Table	SMO
Precision	0.917	0.889	0.916	0.908	**0.923**
Recall	0.916	0.888	0.911	0.907	**0.921**
F-Measure	0.916	0.888	0.912	0.907	**0.921**

* Usando AdaBoot+DecisonStump

6.1.3. Resultados da terceira iteração

Os objetivos da terceira iteração foram definidos de acordo com os resultados obtidos pela segunda iteração.

(i) analisar a possibilidade de classificação em duas etapas, separando em classes de modo de transporte e num segundo estágio de classificação, separando cada meio de transporte específico;

(ii) verificar o resultado para classificação entre: andando, bicicleta e motorizado.

A Figura 6.4 apresenta um possível esquema para classificação em duas etapas, onde o primeiro objetivo é classificar o *chunk* entre motorizado e não motorizado. Em seguida, na segunda etapa, o objetivo é a classificação para separar os modos andando e bicicleta. No caso da primeira etapa de classificação corresponder a não motorizados ou uma para separar: motocicleta, carro e ônibus no caso da primeira etapa de classificação como motorizado. Deve-se levar em consideração a possibilidade de acumulação dos erros na duas etapas, isto é caso haja uma classificação errada na primeira etapa gera uma classificação errada ao fim do processo.

Figura 6.4 – Esquema apresentando a classificação de *chunks* em duas etapas.

Conforme pode ser observado na Tabela 6.5, para a classificação de *chunks* de deslocamento apenas entre bicicleta e a pé os melhores resultados foram apontados por *bayes-net* e *decision table*, onde houve um empate técnico. Como ambos apresentaram o mesmo resultado, uma suspeita é que possa ter restado um certo nível de ruído na amostra, mesmo havendo filtragens na etapa de KDD. Ainda assim, como as taxas de falsos positivos e de falsos negativos são baixas (pela análise da *confusion matrix*), leva-nos a crer na possibilidade de ruídos presentes na amostra, com isto é possível crer que existe possibilidade de melhoria do desempenho.

Para a classificação de *chunks* (de noventa segundos) de deslocamento apenas entre carro, motocicleta e ônibus, o melhor resultado foi apresentado pelo algoritmo *decision table*, com *precision accuracy* de 0,64 e *recall accuracy* de 0,696. O melhor *F-measure* foi apresentado pelo *naive- bayes*. Como uma análise da *confusion matrix* podemos ver que o *naive-bayes* tem realmente uma melhor performance geral, pois o *decision table* não foi capaz de classificar adequadamente ônibus.

Tabela 6.5 - Resultado da análise da terceira iteração.

TERCEIRA ITERAÇÃO

A 90 segundos *Classificando entre: a pé e de bicleta*

	J.48	Random Forest	Random Tree	Nbayes	BayesNet	LibSVM
Precision	0.932	0.924	0.909	0.913	**0.936**	0.913
Recall	0.931	0.924	0.909	0.909	**0.936**	0.912
F-Measure	0.931	0.924	0.909	0.908	**0.936**	0.911

	Perceptron	iBk	AdaBoost*	Decision Table	SMO
Precision	0.925	0.875	0.912	**0.936**	0.893
Recall	0.924	0.875	0.912	**0.936**	0.892
F-Measure	0.924	0.874	0.912	**0.936**	0.891

* Usando AdaBoot+DecisonStump

B 90 segundos *Classificando entre: carro, motocicleta e ônibus*

	J.48	Random Forest	Random Tree	Nbayes	BayesNet	LibSVM
Precision	0.615	0.58	0.585	0.608	**0.618**	0.527
Recall	**0.692**	0.617	0.576	0.658	**0.692**	0.678
F-Measure	0.583	0.596	0.58	**0.604**	0.578	0.566

	Perceptron	iBk	AdaBoost*	Decision Table	SMO
Precision	0.555	0.573	0.475	**0.64**	0.475
Recall	0.687	0.574	0.689	**0.696**	0.689
F-Measure	0.577	0.573	0.563	**0.579**	0.563

* Usando AdaBoot+DecisonStump

C *90 segundos* *Classificando entre: bicicleta, carro, motocicleta e ônibus*

	J.48	Random Forest	Random Tree	Nbayes	BayesNet	LibSVM
Precision	0.587	0.597	0.603	0.645	**0.75**	0.585
Recall	0.67	0.628	0.589	0.681	0.685	**0.69**
F-Measure	0.605	**0.611**	0.595	0.634	0.591	0.604

	Perceptron	iBk	AdaBoost*	Decision Table	SMO
Precision	0.578	0.587	0.476	0.614	**0.645**
Recall	0.685	0.585	0.616	0.682	**0.697**
F-Measure	**0.608**	0.586	0.53	0.594	0.607

* Usando AdaBoot+DecisonStump

C *90 segundos* *Classificando entre: andando, bicicleta e motorizado*

	J.48	Random Forest	Random Tree	Nbayes	BayesNet	LibSVM
Precision	**0.904**	0.887	0.869	0.896	0.899	0.896
Recall	**0.901**	0.887	0.869	0.879	0.894	0.894
F-Measure	**0.902**	0.887	0.869	0.882	0.896	0.894

	Perceptron	iBk	AdaBoost*	Decision Table	SMO	Stacking**
Precision	**0.903**	0.859	0.876	0.899	0.869	**0.9**
Recall	**0.899**	0.858	0.876	0.892	0.856	0.893
F-Measure	**0.899**	0.859	0.875	0.894	0.853	0.894

* Usando AdaBoot+multiplayer-perceptron

** J.48+multilayer-perceptron

6.2. Resultados da análise

Baseado nos resultados apresentados na Seção 6.1, foi possível extrair conclusões que caracterizam o objetivo deste estudo. Além disso, foi possível obter indícios relacionados a objetivos secundários, que sugerem investigações a serem executadas em pesquisas futuras. A seguir, apresenta-se a lista de conclusões baseadas na análise de resultados apresentada na seção anterior:

(a) O número de meios de transporte a serem detectados tem impacto direto sobre a probabilidade de sucesso da detecção. Isso pode ser constatado pela exclusão de um dos modos de transporte (uma classe) do processo de classificação. Considerando que o número de meios de transportes disponíveis é dependente da localidade geográfica, parece ser interessante ter um perfil de classificação customizado a uma determinada localidade.

(b) Detectar um usuário andando não é tarefa difícil, pelo método aqui proposto. Diversos algoritmos conseguiram separar com precisão próxima a 100%.

(c) Detectar somente usuários andando e de bicicleta também não é tarefa difícil, os resultados ficam acima de 90%.

(d) Separar as movimentações de usuários entre carro, moto e ônibus, somente com os registros de localização não apresentou bons resultados com nenhum dos algoritmos investigados. Por outro lado considerando os fatores b e c, para separar usuários motorizados dos andando e de bicicleta, o acerto é muito bom (acima de 90%). O que por si só, já pode viabilizar um conjunto de aplicações, inclusive as aplicações de e-*health* (uma vez que estas, em geral, se baseiam no monitoramento dos deslocamentos não motorizados).

(e) Usando o mecanismo apresentado é possível desenvolver uma solução de classificação, apenas pelo expurgos dos períodos de pausa, somente quando estes forem superiores ao limiar de detecção de paradas. Para em seguida computar os atributos para os pontos que fizerem parte do período de 90 segundos (limiar que foi escolhido para segmentar um *chunk*, conforme apresentado no Capítulo V.), mas somente no caso este *chunk* possuir mais de 9 registros de movimentação (ou 10% do tamanho máximo de registros, conforme considerado na etapa de mineração de dados – menos que isso denota um registro imperfeito) . Com isto, a solução proposta permitiria a classificação dos modos de transporte a partir de 90 segundos de iniciado o deslocamento.

(f) Considerando os diversos tipos de aplicações cientes de contexto que podem fazer uso da informação de meio de transporte utilizada pelos usuário de smartphones, algumas destas aplicações somente se interessam por tipos específicos de modos de transporte. Assim uma aplicação para contabilizar o gasto calórico, precisaria somente da informação dos deslocamentos não motorizados, enquanto aplicações para acompanhar a pegada de carbono do usuário poderia somente se interessar pelos deslocamentos motorizados. Com isso, através da utilização dos middlewares de sistemas cientes de contexto, futuras aplicações poderiam especificar o modos de deslocamentos desejados (através da assinatura de notificações do middleware) e receber somente as notificações relativas aos eventos que sejam seu foco de interesse. Com isso o middleware poderia aplicar a classificação em múltiplas etapas utilizando os algoritmos melhores para cada caso, o que potencialmente melhoria o desempenho da classificação.

CAPÍTULO VII - Considerações finais

Após os resultados apresentados no Capítulo VI, o presente Capítulo tem como objetivo, apresentar a conclusão e os trabalhos futuros relacionados a esta pesquisa.

7.1. Conclusão

Este trabalho apresentou um largo processo de pesquisa, onde foram feitas detalhadas análises de diversos estudos relacionados com o tema abordado. Tais estudos incluíram, mas não se limitaram a: principais trabalhos relacionados a inferência de modos de transporte através de registros de movimentação (*traces*) de smartphones; principais características e desafios relacionados aos sistemas cientes de contexto; arquiteturas para coleta de dados de smartphones; técnicas de localização para as aplicações baseadas em localização; aplicações para contexto de modo de transporte; técnicas para representação e segmentação de trajetórias.

Ainda como parte deste trabalho, foi definido e implementado um sistema para coleta de dados de movimentação de usuários através dos seus smartphones, que utilizou nove usuários voluntários em um esforço ao longo de 9 meses, onde foram geradas mais de 250 trajetórias de movimentações. Estas trajetórias foram então, armazenadas em um banco de dados que totalizou quase 1 milhão de registros de posicionamento. Tanto o sistema de coleta de dados com smartphones, quanto o banco de dados de trajetórias poderão ser utilizados em diversas pesquisas subsequentes.

A partir do banco de dados de trajetórias, foi definido e aplicado um mecanismo de segmentação de trajetórias que é baseado em relevantes pesquisas da área processamento de trajetórias.

Uma vez as trajetórias segmentadas, foi feita uma análise baseada em técnicas de descoberta de conhecimento em banco de dados. Esta análise incluiu a aplicação de uma lista de algoritmos de aprendizado de máquina, para identificar aqueles que apresentam os melhores resultados para inferir o modo de transporte utilizado em movimentações que ainda estão em andamento. A partir destes resultados será possível criar

mecanismos para permitir a obtenção do contexto de modo de transporte em aplicações cientes de contexto.

De forma colateral, buscou-se também avaliar a existência de indícios de melhora da detecção, quando comparada a trabalhos já existentes, para a utilização de um mesmo tipo de smartphone, utilizado em uma área geográfica delimitada – subsidiando pesquisas futuras que poderão abordar cada um dos diferentes aspectos relacionados.

Como resultado dos esforços supracitados temos, além da proposta para arquitetura para coleta de dados de smartphones; um processo para segmentação de trajetórias e uma proposta para aplicação de descoberta de conhecimento em banco de dados, as seguintes descobertas: (i) os algoritmos com melhores resultados: *multilayer-perceptrons*, SVM, *bayes-net*, *naive-bayes*, J.48; (ii) Existem vantagens no uso de um agrupamento temporal de 90 segundos se comparado com os de 120 e 60 segundos; (iii) apenas com o uso do sensor de localização, foi obtido um ótimo nível de precisão para separar usuários andando dos demais modais de transporte investigados; (iv) foi obtido um bom nível precisão para classificar usuários entre, andando, de bicicleta e motorizado. Já quando se tenta separar as classes de movimentações motorizadas (carro, ônibus e moto) os resultados não são muito satisfatórios.

Com a análise dos resultados foi possível obter indícios, a serem estudados futuramente, que uma proposta de classificação baseada em dois estágios; separando primeiro as classes de transporte (como motorizado e não motorizado), para depois separar cada modo de transporte, pode apresentar bons resultados. Foram também identificados indícios de melhoria de resultados se modelos diferentes forem aplicados a diferentes regiões geográficas e tipos de aparelhos.

7.2. Trabalhos futuros

O presente trabalho, dado seu escopo, abriu um amplo leque a ser explorado em trabalhos futuros. Além dos aspectos operacionais da detecção de modo de transporte, ainda existem inúmeros desafios a serem explorados. Desafios estes, que se desdobram em diversas áreas, incluindo: a proteção à privacidade, compartilhamento de contexto, redes sem fio, infraestrutura de redes, mobilidade, entre outras áreas.

Esta seção do trabalho se organiza apresentando os principais trabalhos futuros relacionados a pesquisa aqui apresentada, organizada por áreas.

7.2.1. Arquitetura para coleta de dados de *smartphones*

Quanto a arquitetura (S3A) proposta para coleta de dados de smartphones, os seguintes trabalhos futuros foram identificados:

(i) Avaliação de capacidade e carga de dados, apesar de sua funcionalidade já ter sido testada com tráfego verdadeiro, sua utilização foi de dimensões bastante modestas, se comparada com o objetivo de uso. É importante fazer testes não só para fundamentar a proposta de melhorias, mas para atestar o grau de escalabilidade da arquitetura. Estes testes podem ser feitos através de simulação e geração de carga sintética, que podem ser suficientes para atestar benefícios e identificar os gargalos.

(ii) Melhoria dos componentes de banco de banco de dados. Apesar de funcional, a implementação do bando de dados foi bastante modesta, além disso o MySQL, apesar da facilidade de uso, não se mostrou a solução mais adequada para: volumes maiores de dados, dados geo-espaciais e temporais; O uso de banco de dados NoSQL pode trazer benefícios para a solução uma vez que dados de sensores não precisam do mesmo nível de formalidade de bancos de dados de negócios. Além disso, diferentes bancos de dados podem ser utilizados para alavancar as capacidades necessárias a cada etapa do processo.

(iii) Adaptar a arquitetura para operar sobre clusters *Hadoop*. Este tipo de arquitetura tem se mostrado cada vez mais utilizada para lidar com grandes massas de dados, é importante investigar quais os benefícios de sua utilização tanto para o processamento de informações de sensores de telefones celulares quanto para utilização com middlewares de sistemas cientes de contexto.

(iv) Implementação de uma camada de segurança para proteger a privacidade dos dados baseado no trabalho *personal data vaults* do CENS, conforme apresentado por SHILTON em [26].

(v) Integrar middleware de aplicações cientes de contexto na arquitetura S3A para permitir melhor suporte a coleta de dados oportunistas e viabilizar pesquisas onde informações detalhadas de contexto sejam necessárias.

7.2.2. Influência de localidades nos padrões de movimentação

Quando considera-se que as características intrínsecas de uma determinada região geográfica possuem o potencial para influenciam nos meios de transportes utilizados e

nas próprias características das movimentação, como por exemplo: a distribuição de frequência dos meios de transportes utilizados; a velocidade máxima alcançada; as taxas de mudanças de direção; o número de paradas, entre outras características. Por consequência, cada localidade pode ter suas características resumidas através de um perfil. De forma análoga, períodos de tempo e o clima podem influenciam nas características das movimentação. Existe um amplo campo a ser explorado, para verificar o quanto a cada vizinhança e suas características podem influenciar na movimentação de cada modo de transporte, trazendo a possibilidade de investigar:

(i) A influência de determinadas características urbanas e de infraestrutura, nas variáveis de movimentação, como exemplo: a taxa de ocupação, grau de movimento pendular, concentração da cidade, disponibilidade de rodovias, entre outros.

(ii) Uma vez as variáveis do item (i) detectadas e associadas a uma localidade, existe possibilidade de melhorar a predição de modo de transporte, dado que se conhece as variáveis relacionadas a característica da vizinhança onde ocorreu o deslocamento? A própria mudança na distribuição de classe e o número de classes (modos de transporte que podem ser utilizados), já indica uma possibilidade a ser explorada.

(iii) Devido à falta de sinal (de celular, GPS e WiFi), a detecção dos padrões de movimentação no metrô foi muito prejudicada. Com isso, seria interessante investigar quais outros sensores podem ser utilizados para obter o contexto de meio de transporte para o metrô subterrâneo. Seria viável identificar a estação onde um usuário se encontra pelos anúncios sonoros dentro do carro? Usando o acelerômetro seria possível identificar que o usuário está usando o metrô? Seria possível identificar a estação onde um usuário se encontra de acordo com o padrão de movimentação utilizado no trem entre a estação anterior e a atual posição? Quantidade de dispositivos *bluetooth* vizinhos e o tempo de contato entre eles pode abrir outra frente para investigação.

(iv) Considerando que as condições de tráfego tendem a interferir na mobilidade, especialmente nos modos de transporte motorizados. Seria interessante manter diferentes perfis para diferentes faixas de horário de cada vizinhança?

(v) Condições de tráfego podem ser influenciadas pelo clima, um estudo futuro pode mapear as mobilidades em diferentes circunstâncias para investigar como a movimentação dos diversos modos de transporte são afetadas em diferentes condições climáticas em determinada vizinhança.

7.2.3. Mineração de dados

Apesar deste trabalho ter explorado uma ampla gama de algoritmos, ainda existem algoritmos e técnicas a serem exploradas, entre os possíveis trabalhos futuros nesta área, é possível citar:

(i) Implementação do mecanismo de classificação baseado em duas etapas, conforme proposto no Capítulo anterior.

(ii) Explorar os resultados de outros algoritmos e formas de implementação; principalmente os baseados na cadeia de Markov e no teorema de Bayes. Algoritmos de *clustering* não foram investigados, mas têm ampla aplicação principalmente para se utilizar de forma combinada na transformação dos dados e como coadjuvante na classificação.

(iii) Investigar outras formas de agregar o registro de movimentação. Durante este trabalho, somente a agregação temporal foi investigada, dado seu nível determinístico, mas outras formas de agregar, como exemplo as híbridas, também podem ter um bom nível de determinismo.

(iv) Ainda existem variações a serem exploradas quanto aos atributos (de sumarização) de chunks. Tanto variância de velocidade quanto de aceleração não puderam ser exploradas.

(v) Aplicar uma detecção de pausas mais eficiente permite obter um registro de movimentação com menos ruído, provavelmente refletindo na acurácia de detecção.

7.2.4. Melhorias para o módulo cliente *CityTracks*

O *CityTracks* se mostrou ser uma ferramenta bastante útil, não só para a coleta de traces, como também para pesquisas de sensoriamento participativo.

(i) Dados os últimos eventos relacionados a mobilização popular visto em diferentes países, como foi o caso do *occupy Wall Street* e tantos outros, no Brasil e no mundo, implementar completamente a captura de som e vídeo na ferramenta *CityTracks* pode torná-la apta para capturar outros tipos de informação de campo, principalmente para pesquisas sociais.

(ii) Existe a possibilidade de utilizar outros sensores (temperatura, pressão atmosférica, umidade, iluminação, entre outros) para coleta de informações ainda mais ricas, estes

sensores podem ser selecionados de uma nova gama de sensores e implementados através de um microcontrolador *Arduino* e conectados ao *CityTracks* com uso da conexão *bluetooth* do smartphone.

(iii) Implementar melhorias para a segurança da aplicação *CityTracks*, tanto pela utilização de encriptação para proteger sua comunicação, quando implementar a possibilidade de pre-processar os dados, enviando apenas a sumarização de mobilidade, excluindo a localização dos registros, ainda dentro do smartphone. Assim somente as características de movimentação seriam enviadas, dispensando o envio da localização e contribuindo para preservar a privacidade.

(iv) Implementar a coleta de dados de acelerômetro, sinais de WiFi e *bluetooth* podem viabilizar novos campos para pesquisa, como redes DTN, pesquisas de micro mobilidade e mobilidade contínua, de redes WiFi, etc.

(v) Implementação de medidas de economia de bateria e de redes, pelo uso do acelerômetro e da compressão de dados na aplicação.

7.2.5. Processo de coleta de dados

Existem diversas oportunidades com a melhoria e expansão do processo de coleta de dados. Alguns dos problemas relatados no Capítulo V (quanto a coleta de dados) podem ser solucionados ou terem medidas de redução de impacto definidas e implementadas.

Entre as possíveis abordagens para explorar estas tais oportunidades, é possível citar:

(i) A coleta de dados contínua, pode ser possível se implementado um mecanismo para economia de bateria no *CityTracks*. Com isto seria possível viabilizar outras áreas de pesquisas relacionadas a mobilidade, como por exemplo: algoritmos híbridos de mobilidade, estudo da movimentação urbana em larga escala, aplicado para as cidades brasileiras com foco social e geográfico (*webmaps*).

(ii) Coleta de dados em larga escala pode permitir outros tipos de estudo, além de viabilizar testes mais criteriosos para a arquitetura proposta e permitir uma análise de movimentação ainda mais detalhada.

Referências Bibliográficas

[1] ALMEIDA, C. A. S., PIRES, E., et. al., “Um Modelo de Dados para Trajetórias de Objetos Móveis com suporte a Agregação de Movimentos”, iSys - Revista Brasileira de Sistemas de Informação, Volume 4, 2011.

[2] APPLE Inc., *Location Manager Class Reference*, disponível em: “https://developer.apple.com/library/ios/#documentation/CoreLocation/Reference/CLLocationManager_Class/CLLocationManager/CLLocationManager.html#//apple_ref/doc/uid/TP40007125”, acesso em: 14/04/2013

[3] THIAGARANJAN, A. “Probabilistic Models For Mobile Phone Trajectory Estimation”, Tese de doutorado, Massachusets Institute of Technology, Setembro de 2011.

[4] BALDAUF, M., DUSTDAR, S., ROSENBERG, F. “A Survey on context-aware systems”, Int. J. Ad Hoc and Ubiquitous Computing, Vol. 2, No. 4, 2007.

[5] BELLAVISTA, P., CORRADI, A., FANELLI, F., and FOSCHINI, L., “A survey of context data distribution for mobile ubiquitous systems”. ACM Comput. Surv. 44, 4, Article 24, Setembro de 2012.

[6] BREIMAN, L., CUTLER, A., “Random Forest”, disponível em: http://stat-www.berkeley.edu/users/breiman/RandomForests/cc_home.htm, acesso em: 10/08/2013.

[7] CORTÊS, S., PORCARO, R., LIFSHIT, S., *Mineração de Dados – Funcionalidades, Técnicas e Abordagens* – Relatório Técnico, PUC-Rio, 2002,disponível em: “ftp://ftp.inf.puc-rio.br/pub/docs/techreports/02_10_cortes.pdf”, acesso em: 10/07/2012.

[8] DAL POZOLLO, A., CAELEN, O., WATERSHOOT, S., et al, “Racing for unbalanced methods selection”, Brussels Institute for Research and Innovation, 2013.

[9] COMSCORES, “ComScores Reports November 2011- U.S. Mobile Subscriber Market Share” disponível em: “http://www.comscore.com/por/Insights/Press_Releases/2011/12/comScore_Reports_November_2011_U.S._Mobile_Subscriber_Market_Share”, acesso em: 10/12/2012.

[10] ELMENREICH,W., “Sensor Fusion in Time-Triggered Systems”, Technischen Universitat Wien, Fakulta¨t fu¨r Technische Naturwissenschaften und Informatik 2002.

[11] MARISCAL, G., et al. “A survey of data mining and knowledge discovery process models and methodologies”, Knowledge Engineering Review, v. 25, n. 2, p. 137, 2010.

[12] FAYYAD, U., PIATETSKY-SHAPIRO, G., SMYTH, P., “The KDD process for extracting useful knowledge from volumes of data.”, Communications of the ACM, v. 39, n. 11, p. 27-34, 1996.

[13] FRANKLIN, R., "Exploratory Experiments", Philosophy of Science, vol. 72 (2005), pg. 888-899, disponível em: “http://www.experimentalmath.info/papers/franklin-expm.pdf”, acesso em 02/03/2013.

[14] HARVARD, “Research Methods: Some notes to orient you.”, disponível em: “http://isites.harvard.edu/fs/docs/icb.topic851950.files/Research%20Methods_Some%20Notes.pdf”, acesso em 02/03/2013.

[15] HSU, C., CHANG, C., et al.,”A Practical Guide to Suport Vector Classification” - National Taiwan University, disponível em: “http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf”, acesso em: 10/08/2013.

[16] ICPSR, “Guide to Social Data Preparation and Archiving”; disponível em: “http://www.icpsr.umich.edu/files/ICPSR/access/dataprep.pdf”, acesso em: 13/10/2012.

[17] IDRISSOV, A., NASCIMENTO, M., *Trajectory Cleaning Framework*, Mobile Data Challenge (by Nokia) Workshop, 2012.

[18] MEHRA, P., "Context-Aware Computing: Beyond Search and Location-Based Services", IEEE Internet Computing, vol. 16, no. 2, pp. 12-16, March-April, 2012.

[19] RUSSEL, S., NORVIG, P., *Artificial Intelligence: A Modern Approach*, 3rd edition, New Jersey - EUA, Prentice Hall, 2010.

[20] NOULAS, A., SCELLATO, S., LAMBIOTTE, R., et. al., “A tale of many cities: universal patterns in human urban mobility”, PloS one, v. 7, n. 5, p. e37027, 2012.

[21] OLIVEIRA, E., DE ALBUQUERQUE, C., “NECTAR: a DTN routing protocol based on neighborhood contact history”, In: Proceedings of the 2009 ACM symposium on Applied Computing. ACM, 2009, p. 40-46.

[22] PALMA, A. T., BORGONY, V., et al. “A clustering-based approach for discovering interesting places in trajectories”. In: Proceedings of the 2008 ACM symposium on Applied computing. ACM, 2008, p. 863-868.

[23] POSLAD, S., *Ubiquitous computing: smart devices, environments and interactions*. EUA, Wiley, 2011.

[24] REDDY, S., BURKE, J. ESTRIN, D., Et al. “Determining Transportation Mode On Mobile Phones”, In: Proceedings of ISWC, IEEE, page 25-28, 2008.

[25] REDDY, S., MUN, M., BURKE, J., et al., “Using mobile phones to determine transportation modes,” ACM Transactions on Sensor Networks (TOSN), v. 6, n. 2, p. 13, 2010.

[26] SHILTON K., “Four Billion Little Brothers? Privacy mobile phones and ubiqitous data collection.”, Communications of the ACM, Novembro de 2009, vol. 52, no. 11, pages 48-53.

[27] SOKOLOVA, M., JAPKOWICZ , N. SZPAKOWICZ, S., “Beyond Accuracy, F-score and ROC: a Family of Discriminant Measures for Performance Evaluation”, AI 2006: Advances in Artificial Intelligence, p. 1015-1021, Springer, Berlin Heidelberg, 2006,

[28] SOUZA, D., MULLER, D., et. al., “Manual de Orientações para projetos de Pesquisa”, disponível em: “http://www.liberato.com.br/UserFiles/File/noticias/Manual_de_orientacoes_para_projetos_de_pesquisa.pdf”, acesso em: 21/04/2013

[29] STENNETH, L., WOLFSON, O., Yu, P., et al., “Transportation mode detection using mobile phones and GIS information”, In Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, ACM, 2011, p. 54-63.

[30] TAN, P., STEINBACH, M., KUMAR, V., *Introdução ao Data Mining, Mineração de Dados*, Rio de janeiro - Brasil, Editora Ciência Moderna Ltda., 2009

[31] VEGA LÓPEZ, I., SNODGRASS, R., MOON, B., “Spatiotemporal Aggregate Computation: A Survey”, IEEE Transactions on Knowledge and data engineering”, Vol. 17 No. 2, Fevereiro de 2005.

[32] WAZLAVICK, R., *Metodologia de pesquisa para ciências da Computação*, ed , Brasil, Elsevier, 2008

[33] JUNG, C., “Projetos de Pesquisa: Guia Rápido para elaboração”, Página de metodologia, disponível em: http://www.jung.pro.br/moodle/”, acesso em 10/03/2013.

[34] WILSON, J., *Sensor Technology Handbook*, 1ª edição, Massachusetts, EUA, ELSEVIER, 2005.

[35] WITTEN, I., EIBE, F.,HALL, M., *Data Mining : Practical Machine Learning Tools and Techniques*, 3ª edição, Massachusetts-EUA, Elsevier, 2011.

[36] WU, X., KUMAR, V., QUINLAN, J. R., et al., “Top 10 algorithms in data mining”, Knowledge and Information Systems, 14(1), 1-37, 2008.

[37] ZHENG, Y., LIU, L., WANG, L., et al., "Learning transportation mode from raw gps data for geographic applications on the web", *Proceeding of the 17th international conference on World Wide Web - WWW '08*, p. 247, 2008.

[38] ZHENG, Y., LI, Q., CHEN, Y., XIE, X. and MA, W., "Understanding mobility based on GPS data", Proceedings of the 10th international conference on Ubiquitous computing, ACM, p. 312-321, 2008.

[39] ZHENG, Y., ZHANG, L., XIE, X., and MA, W., "Mining interesting locations and travel sequences from GPS trajectories", Proceedings of the 18th international conference on World Wide Web, ACM, p. 791-800, 2009.

[40] ZHENG, Y., CHEN, Q., XIE, X., et al., "Understanding transportation modes based on GPS data for web applications," *ACM Transactions on the Web*, vol. 4, no. 1, pp. 1–36, Jan. 2010.

[41] MITCHELL, M. T., *Machine Learning*, Burr Ridge, IL, EUA, McGraw-Hill, 1997.

[42] MAIMON, O., ROKASH, L., *Data Mining and Knowledge Discovery Handbook*, 2ª edição, EUA, Springer, 2010.

[43] AHSON, S., ILYAS, M., *Location-Based Services Handbook, Applications, Technologies, and Security*, 1ª edição, EUA, CRC Press, 2011.

[44] CAMPBELL, A., LANE, N., MILUZZO, E., PETERSON R., LIU, H., ZHENG, X.,MUSOLESI, M., FODOR, K., EISENMAN, S., AHN, G., "The Rise of People-Centric Sensing", IEEE Internet Computing, vol. 12, no. 4, pp. 12-21, Julho/Agosto, 2008.

[45] LANE, N., MILUZZO, E., LU, H., PEEBLES, D., CHOUDHURY, T., CAMPBELL, A., "A Survey on Mobile Phone Sensing", Communications Magazine, IEEE , vol.48, no.9, pp.140,150, Setembro 2010.

[46] LAFFERTY, J.; MCCALLUM, A.; PEREIRA, F., "Conditional random fields: Probabilistic models for segmenting and labeling sequence data", 2001.