Universidade Nova de Lisboa
Faculdade de Ciências e Tecnologia
Departamento de Informática

Dissertação de Mestrado

Mestrado em Engenharia Informática

Extração de Informação de Padrões Pessoais de Tempo e Espaço

Samuel Dário Falcão Gabriel del Bello (28056)

Orientador: Prof. Doutor Nuno Manuel Robalo Correia

Composição do Júri

Presidente: Prof. Doutor José Júlio Alves Alferes
Vogais: Prof. Doutora Ana Paula Pereira Afonso
Prof. Doutor Nuno Manuel Robalo Correia

Lisboa
(Abril de 2011)

Universidade Nova de Lisboa
Faculdade de Ciências e Tecnologia
Departamento de Informática

Dissertação de Mestrado

Extração de Informação de Padrões Pessoais de Tempo e Espaço

Samuel Dário Falcão Gabriel del Bello (28056)

Orientador: Prof. Doutor Nuno Manuel Robalo Correia

Trabalho apresentado no âmbito do Mestrado em Engenharia Informática, como requisito parcial para obtenção do grau de Mestre em Engenharia Informática.

Lisboa
(Abril de 2011)

Extração de Informação de Padrões Pessoais de Tempo e Espaço

À minha família.

Agradecimentos

Em primeiro lugar quero agradecer ao meu orientador, Professor Doutor Nuno Correia, pelo seu papel imprescindível na orientação, nos momentos de discussão, na leitura pormenorizada e crítica, nos contributos que, de diversas formas, permitiram a construção desta dissertação, como também pela disponibilidade e o apoio sempre revelados.

Os meus agradecimentos dirigem-se também para:

- O projeto de investigação onde se insere esta dissertação, intitulado "Máquina do Tempo - Time Machine", com a referência PTDC/EAT-AVP/105384/2008, financiado pela fundação para a Ciência e Tecnologia.
- O CITI e DI/FCT pela oportunidade de trabalhar neste projeto com bolsa de investigação.
- Todos os elementos do projeto Time Machine, nomeadamente para a Sofia Oliveira e o Jared Hawkey pelos testes realizados, para o Olivier Perriquet pelos esclarecimentos e apoio na investigação, para o Tiago Amorim e o Cristiano Lopes pela disponibilidade e ajuda sempre prestados.
- A Professora Armanda Rodrigues e o Professor João Magalhães pelos esclarecimentos de técnicas e opiniões críticas no desenvolvimento da dissertação.
- Todos aqueles que se disponibilizaram a ajudar nos testes realizados e ainda todos aqueles que tiverem um papel mais próximo de suporte ao longo do desenvolvimento desta dissertação, entre os quais destaco (sem qualquer tipo de ordem): Diogo Bernardino, Carlos Júlio, João Mamede, Filipe Gonçalves, Paulo Gabriel, Mari Armesto, João Martins e Rui Esteves.
- Toda a minha família e amigos pelo suporte incondicionável, em especial à minha mãe, Maria de Fátima Gabriel, por tudo o que sempre fez por mim.

A todos,
Muito Obrigado!

Resumo

A partir de informação registada pelo GPS é possível construir uma representação rica, uma cartografia pessoal de deslocações passadas e futuras. Um passo importante neste trabalho, é analisar os dados capturados e transformá-los em variáveis e padrões capazes de suportar a visualização. Estes dados não filtrados têm de ser convertidos em variáveis fundamentais, como a frequência ou duração em cada local, para que padrões de comportamento possam ser obtidos. É importante encontrar a relevância da informação e do local dependendo do contexto, padrões de regularidade e irregularidade e prever futuros comportamentos. Para tornar isto possível, é necessário encontrar técnicas para trabalhar estas variáveis e produzir resultados relevantes. É também preciso ter presente que o software vai correr em dispositivos móveis, razão pela qual os algoritmos escolhidos têm que considerar este ambiente de computação.

No âmbito desta dissertação, diferentes técnicas serão estudadas e analisadas para, combinadas entre si, extraírem variáveis e padrões que suportem a visualização.

Esta investigação foi desenvolvido no âmbito do projeto Time Machine, o qual pretende representar a forma pessoal de experimentar o tempo, através de diferentes propostas de visualização e interação com dispositivos móveis.

Palavras-chave: Inteligência Ambiente, Dados de Localização, Deteção de Padrões, Previsão.

Abstract

From information registered in GPS logs, it is possible to build a rich representation, our individual personal cartography of past and future moves. An important step in this work is to take this raw data and parse it into variables and patterns to support the visualization. These logs have to be parsed into fundamental variables as frequency and time spent in each local, so that the behavior patterns can be obtained from it. It is important to find information and local relevance depending on the context, regularity and irregularity patterns, and future behaviors have to be predicted. To make that happen it is needed to find tecninques, which using our data can produce relevant information.

In the scope of the dissertation several algorithms are experimented and evaluated, that when combined together extract patterns and variables to support the visuatization.

This research was developed in the scope of the Time Machine project, which has the goal to represent the personal way of experiencing time, through different proposed visualizations and interactions with mobile devices.

Keywords: Ambient Intelligence, Location Data, Pattern Detection, Prediction.

Conteúdo

Lista de Figuras

Lista de Tabelas

Listagens

Lista de Algoritmos

1

Introdução

A importância dos dispositivos de computação na nossa sociedade tem vindo a crescer e como resultado do constante desenvolvimento da tecnologia, as capacidades dos dispositivos móveis desenvolvem-se cada vez mais rapidamente, tanto a nível computacional como de interatividade. Novos componentes e sensores para captar diferentes tipos de dados, estão constantemente a ser introduzidos nos dispositivos que utilizamos regularmente. Como exemplo, temos o Sistema de Posicionamento Global [DH97], vulgarmente conhecido pela sigla GPS, que começa a fazer parte destes dispositivos que transportamos no nosso quotidiano - os telemóveis. Enquanto os transportamos, enormes quantidades de dados sobre os estilos de vida de cada um de nós podem ser capturadas. A velocidade com que aumenta a capacidade de armazenamento dos dispositivos em geral, ajuda também a que dados que anteriormente teriam sido desconsiderados, sejam agora facilmente guardados. Estes fatores tornam possível o armazenamento de progressivas quantidades de dados sobre as nossas vidas.

1.1 Motivação

A computação ubíqua, também conhecida por Ubicomp, tem vindo a emergir nos últimos anos, sendo cada vez mais presente no nosso quotidiano. Contudo, esta é uma área complexa com muitos tópicos que requerem atenção, nomeadamente a expectativa das pessoas, centradas na aceitação da automatização. Até agora, as teorias culturais da vida quotidiana têm estado relativamente silenciosas nas discussões sobre o *design* da computação ubíqua e os estudos da vida quotidiana precisarão cada vez mais de integrar as tecnologias ubíquas [PA10].

O projeto Time Machine tem interesse em desvendar os padrões ocultos na vida quotidiana

de cada um de nós. O objetivo é confrontar cada utilizador individual com uma visão global da sua forma de experienciar o tempo, identificar suas rotinas e eventos extraordinários, fornecendo assim, meios para uma reflexão sobre hábitos e estilo de vida.
Embora o projeto tenha preocupações na investigação sobre padrões da vida quotidiana, não se pretende gerar perfis coletivos de utilização do espaço em utilizadores de dispositivos móveis. O foco é o indivíduo e os padrões aqui são entendidos como algo que poderá ser interessante e talvez útil para a reflexão pessoal [PA10].

Esta dissertação foi desenvolvida no âmbito do projeto Time Machine [PA10], financiado pela Fundação para a Ciência e a Tecnologia, com a referência PTDC/EAT-AVP/105384/2008, e desenvolvido numa parceria entre o Centro de Informática e Tecnologias de Informação (CITI) do Departamento de Informática da Faculdade de Ciências e Tecnologias da Universidade Nova de Lisboa e o CADA[i], um coletivo artístico baseado em Lisboa.

1.2 Descrição do problema e contexto

O objetivo principal do projeto Time Machine é construir uma máquina do tempo. Um "relógio" de bolso que faz o *tracking* dos movimentos do utilizador no espaço, utilizando a tecnologia GPS. Sendo o *output*, uma interface de visualização de informação que existe no dispositivo móvel, desenhado de modo a permitir *inputs* individuais e leituras subjetivas, com o objetivo de gerar uma cartografia pessoal. Tudo isto, tem de acontecer no contexto de um dispositivo móvel. Um dispositivo com limitações conhecidas, principalmente a nível da bateria e capacidade de processamento.

O projeto tem como componente fundamental a construção de interfaces com capacidades avançadas de visualização. Contudo, para que estas sejam possíveis de construir, é essencial uma camada de processamento de dados capaz de extrair informação útil e com significado das coordenadas não filtradas do GPS. Sendo nesta informação que se irá basear toda a estrutura de visualização.
Pretende-se, com esta dissertação, conduzir uma investigação no sentido de achar a melhor forma para encontrar os padrões ocultos na vida quotidiana do individuo e usá-la de forma a aprender padrões de regularidade e irregularidade sobre o estilo de vida de cada um. Padrões estes que representam os costumes e hábitos do experienciador no espaço e no tempo. O objetivo é ainda usar esta informação de modo a construir um modelo preditivo capaz de prever movimentos futuros com base no passado.

Os dados existentes são essencialmente registos de posições do GPS, que devido à sua natureza contêm bastante ruído. Estas posições são captadas regularmente, sendo cada uma representada por coordenadas que identificam um ponto na superfície terrestre. Cada registo contém também uma posição no tempo, permitindo definir a posição do utilizador ao longo do

[i] http://www.cada1.net/

tempo. Contudo, falhas no receptor implicam períodos sem dados. É sobre esta vasta quantidade de dados com imprecisões e lacunas que se trabalhou, o que dificultou todo o processo. Sendo o objetivo desta dissertação o suporte de múltiplas visualizações aliadas à ambiguidade dos dados de partida, o desenvolvimento teve um carácter muito experimental. De início, não eram conhecidas quais as técnicas que, quando aplicadas aos dados capturados, resultariam em informação útil. Foi necessária uma constante pesquisa, desenvolvimento, e avaliação de diferentes técnicas que combinadas permitem a extração da informação pretendida.

1.3 Solução apresentada e contribuições previstas

Com base no estudo do trabalho previamente elaborado no âmbito do projeto Time Machine e na pesquisa bibliográfica efetuada, são apresentadas um conjunto de técnicas que foram desenvolvidas, experimentadas e avaliadas nesta dissertação. Estas técnicas foram escolhidas de acordo com as necessidades da interface e visualização. Essas necessidades foram identificadas como um conjunto de questões que o sistema deverá ser capaz de responder [dBOH+11]. Estas podem ser classificadas em quatro grupos diferentes:

- É este um sítio novo? É a casa/trabalho/outro local relevante? Estou num local/conjunto de locais completamente diferente? Quais são os locais onde passo mais tempo? Viajei muito hoje? Quais são os locais onde vou mais vezes?
- É este o meu padrão (usual) para esta altura do dia? É este o meu padrão (usual) para este sítio? É esta a minha sequência habitual de locais?
- É este dia diferente do ordinário? Foi este um dia calmo? (manhã/tarde)? Foi um dia longo?
- Qual o próximo local? Onde irei amanhã/na próxima semana/próximo mês?

Esta divisão, conduz a diferentes conjuntos de técnicas para processamento da informação. Os dados espacio-temporais são processados para identificar locais relevantes e, posteriormente, extrair dados estatísticos que permitam descobrir padrões e prever comportamentos futuros.
Uma das principais preocupações no projeto é mostrar o diferente ou extraordinário relativamente a locais e/ou ao seu uso no tempo. As rotinas extraídas em termos de diferentes variáveis são especificadas para responder às questões apresentadas. Diferentes usos de tempo, velocidade e locais importantes, mostram também diferentes padrões e exceções a estes.

Para filtrar o excesso de dados é necessário um processamento preliminar para encontrar os locais relevantes, através dos registos obtidos da captura do GPS. A informação sobre a ordem dos acontecimentos é preservada, o que permite criar um historial do utilizador definido por locais e trajetórias entre estes. Um processamento simples deste historial, considerando as diferentes dimensões, permite a extração de dados estatísticos que dão resultados sobre a relevância destas variáveis. Estas permitem dar suporte direto a algumas das perguntas mencionadas acima, mas são também *input* para *clustering* e classificação dos dados, que permitem,

por sua vez, obter mais respostas. É também criado um modelo preditivo, baseado nos modelos de Markov, construído com base no historial de locais, que permite prever movimentos futuros do utilizador.

Contribuições previstas.

Com a solução apresentada prevê-se uma contribuição relevante, em particular no projeto Time Machine. Contudo, os estudos feitos nesta dissertação podem contribuir em diversas áreas. Correspondem às seguintes contribuições:

- Conjunto de técnicas para definir locais importantes e o historial rigoroso de movimentos do utilizador a partir de registos GPS;
- Análise do significado e aplicação de diferentes variáveis extraídas de padrões pessoais;
- Análise de diferentes representações de dias e locais para aprendizagem;
- Proposta de modelo preditivo para previsão de movimentos futuros e extração de sequências de locais relevantes da rotina.

É ainda de referir, que a solução apresentada levou à elaboração do artigo "Processing Location Data for Ambient Intelligence Applications"[dBOH+11], o qual foi publicado na conferência *Ambi-sys 2011*[i], The International ICTS Conference on Ambient Media and Systems, Porto, Portugal, 24-25 de Março 2011.

1.4 Organização do documento

O presenten documento encontra-se estruturado em seis capítulos, descritos de seguida:

Capítulo 1 - Trabalho relacionado: Apresenta uma visão geral da dissertação, no que diz respeito à motivação, à descrição do problema e contexto, à solução apresentada e às principais contribuições previstas.

Capítulo 2 - Trabalho relacionado: Estabelece uma relação entre os objetivos desta dissertação e o trabalho relacionado. Começa-se por introduzir as noções básicas inerentes ao projeto, como o sistema de coordenadas geográficas e o sistema de posicionamento global. Em seguida faz-se uma apreciação do trabalho já desenvolvido no âmbito do projeto Time Machine. Depois, é elaborado um resumo geral de várias aproximações que já tiveram lugar e que, de alguma forma, contribuíram para o desenvolvimento desta dissertação. O restante capítulo é dividido em três secções, uma respeitante à deteção e definição de locais importantes, outra à deteção de padrões pessoais de movimento e a última à previsão de movimentos pessoais.

[i] http://ambi-sys.org/2011/

Capítulo 3 - Componentes do sistema desenvolvido: Apresenta as várias componentes que participam no sistema construído para a extração de informação relevante. Este sofreu alterações constantes ao longo do desenvolvimento da dissertação, sendo aqui apresentado na sua versão final.

Capítulo 4 - Resultados experimentais: Apresenta uma análise dos resultados obtidos. São discutidos os vários resultados experimentais que foram sendo obtidos para as várias componentes do sistema. Estes resultados foram condicionando as técnicas escolhidas durante o desenvolvimento das várias componentes.

Capítulo 5 - Avaliação: Apresenta a avaliação das respostas dos utilizadores ao inquérito realizado. São aqui avaliadas a extração dos locais, a relevância destes, bem como as sequências de locais mais relevantes. São revelados e analisados os resultados no contexto da dissertação e do projeto no qual está inserida.

Capítulo 6 - Conclusões e trabalho futuro: Apresenta uma apreciação geral do trabalho desenvolvido no âmbito desta dissertação, assim como as sugestões para alguns caminhos a seguir em relação à evolução do mesmo.

2

Trabalho relacionado

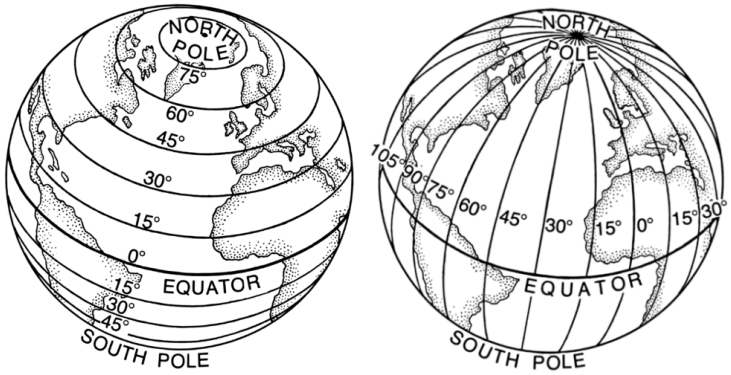
Este capítulo é dedicado à descrição do estado de arte correspondente à bibliografia consultada nas diferentes áreas que esta dissertação está inserida. Assim sendo, na secção 2.1 são explicados os conceitos necessários relativos à captura da localização. Na secção 2.2, são abordados e discutidos os avanços previamente feitos no âmbito do projeto Time Machine. Na secção 2.3, é apresentado um resumo de diferentes aproximações que já tiveram lugar e que de alguma forma contribuíram para o desenvolvimento desta dissertação. Na secção 2.4, a bibliografia consultada para a definição de locais do utilizador é explicada e discutida. Na secção 2.5, são abordadas técnicas para a deteção de padrões em dados. Por último, na secção 2.6.3, são explicadas as técnicas utilizadas para criar um modelo preditivo capaz de prever movimentos futuros.

2.1 Captura da localização

A captura da localização física é feita com base na localização espacial do objeto, expressa pelo sistema de coordenadas geográficas. Esta localização, garante a precisão necessária pois representa um único ponto no espaço euclideano. Em concreto, no projeto Time Machine, o Sistema de Posicionamento Global (GPS) é utilizado para a captura da posição espacial e temporal.
Nas secções abaixo, 2.1.1, 2.1.2 e 2.1.3 , é primeiro apresentado o sistema de coordenadas utilizado, sendo de seguida explicada a forma de calcular a distância entre dois pontos à superfície terrestre e por último abordada a forma como o Sistema de Posicionamento Global funciona.

2.1.1 Sistema de coordenadas geográficas

Um sistema de coordenadas é usado para referenciar uma localização através de um vector de números, aos quais se dá o nome de coordenadas. As coordenadas de um objeto, referem-se à sua posição medida pela distância, ou por um ângulo considerando dois ou três eixos do sistema de coordenadas, dependendo se a posição é para ser determinada num plano ou no espaço tridimensional [Küp05].
O Sistema de Coordenadas Geográficas é o sistema de coordenadas que permite a qualquer localização na Terra ser especificada por um conjunto de coordenadas. No contexto do projeto Time Machine, este sistema é utilizado com as coordenadas, latitude e longitude, na forma de graus decimais segundo um sistema de coordenadas esférico [Amo10].



(a) Latitude.i (b) Longitude.ii

Figura 2.1: Desenho da Terra mostrando os paralelos e meridianos que representam as latitudes e longitudes, respetivamente, em graus.

A latitude geográfica de um ponto na superfície da Terra, figura 2.1(a), equivale ao ângulo entre o plano equatorial e uma linha que passa por esse ponto e é normal à superfície de referência que aproxima a forma da Terra. A latitude mede-se para norte e para sul do Equador, entre -90º no polo sul e +90º no polo norte. A longitude, figura 2.1(b), descreve a localização de um lugar medido em graus, de 0º a -180º para Oeste ou a 180º para leste, a partir do Meridiano de Greenwich. Portanto, se se combinar estes dois ângulos, latitude e longitude, poderá ser indicada qualquer localização na superfície terrestre. Por exemplo, Lisboa tem uma latitude de +38,42º e uma longitude de -9,11º. Por isso, se se traçar um vector desde o centro da Terra até um ponto a 38,42º acima de Equador e 9,11º a oeste de Greenwich, irá passar por Lisboa.
As linhas traçadas de Oeste a Este têm valor constante de latitude e são chamadas de paralelos, enquanto os *meridianos* são as linhas que vão de norte a sul. Os paralelos e meridianos ficam dispostos pela superfície do planeta Terra tal como se pode ver nas figuras 2.1 e 2.2.

Os valores de latitude e longitude são aqui expressos em graus decimais. Neste sistema, usado em bastantes Sistemas de Informação Geográfica (GIS), as coordenadas geográficas latitude e longitude são expressas como frações decimais. Esta é uma alternativa a utilizar graus,

[i] http://es.wikipedia.org/wiki/Archivo:Latitude_(PSF).png
[ii] http://pt.wikipedia.org/wiki/Ficheiro:Longitude_(PSF).png

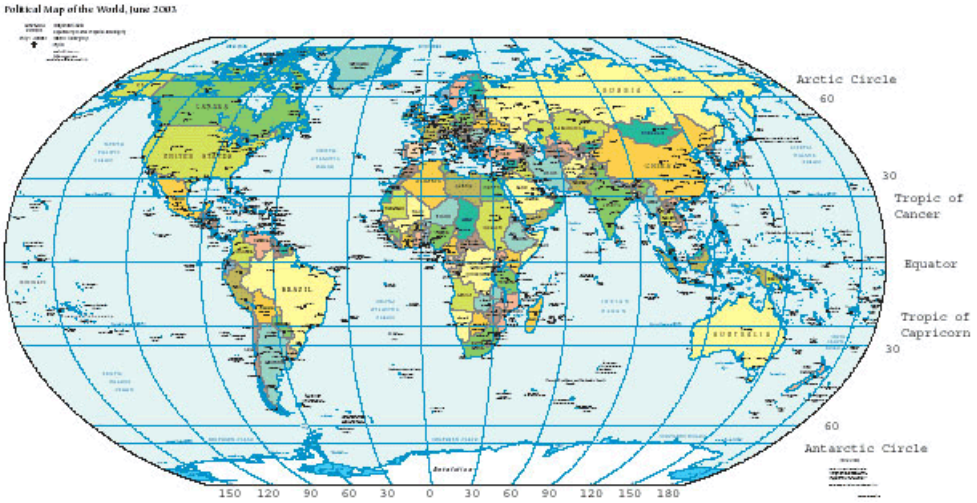


Figura 2.2: Mapa da Terra mostrando as linhas de latitude (horizontalmente) e longitude (verticalmente).i

minutos e segundos (DMS).

Embora longitude e latitude possam identificar posições precisas na superfície do globo, estas não são unidades uniformes de medida. Apenas ao longo da linha do Equador, a distância representada por um grau de longitude, se aproxima à representada por um grau de latitude. Isto deve-se ao facto do Equador ser o único paralelo tão largo quanto os meridianos. Estes círculos, com o mesmo raio que o raio da Terra, chamam-se grandes círculos, do inglês *Great Circle*. O Equador e todos os meridianos são grandes círculos.
Acima e abaixo do Equador, os círculos que definem os paralelos de latitude vão ficando gradualmente mais pequenos, até se tornarem um único ponto nos polos Norte e Sul, onde os meridianos convergem (este fenómeno é visível na figura 2.1). À medida que os meridianos convergem para os polos, a distância representada por um grau de longitude diminui até zero. Com base na esferóide Clark 1866ii, um grau de longitude ao nível do Equador é igual a 111.321 km, enquanto a 60º de latitude são apenas 55.802 km. Devido a este facto, dos graus de latitude e longitude não terem um tamanho uniforme, a distância entre pontos não pode ser medida com precisão utilizando unidades de medida angulares [Sny87].

O Sistema de Coordenadas Geográficas pode ser defendido tanto por uma esfera como por uma esferóideiii que aproxime a forma da Terra. Devido à Terra não ser perfeitamente redonda, uma esferóide pode ajudar a manter a precisão para um mapa, dependendo da localização na Terra [IBM11].

[i] http://pt.wikipedia.org/wiki/Ficheiro:WorldMapLongLat-eq-circles-tropics-non.png
[ii] Clark 1866 foi a esferóide de referência adotada pela costa dos EUA e Geodetic Survey em 1880 para fazer o mapa do Norte da América.
[iii] Uma esferóide é uma elipsóide baseada numa elipse, enquanto a esfera é baseada num círculo.

2.1.1.1 Precisão da resolução

Os graus decimais expressam a latitude e longitude como frações decimais. Quanto maior resolução de casas decimais estes tiverem, maior será a precisão dos dados. Um grau de longitude ao nível do Equador representam aproximadamente 111 km. Sabe-se assim que o número de casas decimais necessário para uma precisão em particular ao nível do Equador está de acordo com a tabela 2.1.

decimal places	degrees	distance
0	1.0	111 km
1	0.1	11.1 km
2	0.01	11.1 km
3	0.001	111 m
4	0.0001	11.1 m
5	0.00001	1.11 m
6	0.000001	111 cm
7	0.0000001	1.11 cm

Tabela 2.1: Tabela com precisões das várias resoluções de graus decimais ao nível do Equador.

Conforme se muda a posição em direção aos polos, muda também a precisão dos graus de longitude. Quanto mais perto dos polos, maior a precisão. Por exemplo, para 60º de latitude a precisão de um grau de longitude é cerca de duas vezes superior, 55.8 km. De notar que a precisão é maior quanto mais baixa for a distância do valor que correspondente.
Isto sucede com os graus de longitude devido aos paralelos não serem todos do mesmo tamanho, como explicado anteriormente. Os meridianos têm todos a mesma dimensão, idêntica também à dimensão da linha do Equador. A precisão dos graus de latitude está de acordo com a tabela 2.1 para todos os pontos da superfície terrestre, assumindo a Terra como uma esfera.

2.1.2 Cálculo da distância

Como já mencionado anteriormente, devido ao facto da latitude e longitude não terem um tamanho uniforme, a distância entre dois pontos não pode ser medida com precisão utilizando unidades de medida angulares.
A fórmula de Vincenty [Vin75], baseada no sistema de Coordenadas Geográficas definidas por uma esferóide, é a fórmula que mais precisão garante para se calcular a distância entre dois pontos à superfície terrestre, garantindo uma precisão de 0.5mm. Contudo, esta tem um peso computacional considerável, superior ao da fórmula de Haversine baseada na ortodromia[i], que assume a Terra como esférica. A fórmula de Haversine garantindo uma precisão de 0.3% é bastante menos complexa e computacionalmente menos dispendiosa [Cha11].
Implementações para ambas as funções são encontradas por exemplo na biblioteca *geosphere*[ii]. De forma a melhor servir os interesses do projeto Time Machine, para o qual um erro de 0.3%

[i] Ortodromia é a linha que une dois pontos à superfície da Terra, à qual corresponde o caminho mais curto entre eles.

[ii] http://cran.r-project.org/web/packages/geosphere/geosphere.pdf

em distâncias é quase insignificante e em que a poupança de recursos é uma das prioridades, a fórmula de Haversine foi a escolhida para calcular distâncias entre dois pontos.

Fórmula de Haversine [Hav84].

Sejam ϕ_i, λ_i; ϕ_f, λ_f as coordenadas, latitude e longitude, dos pontos inicial e final respetivamente. Sejam $\Delta\phi$, $\Delta\lambda$ as diferenças em, latitude e longitude, entre os dois pontos. A distância entre eles é calculada em termos de $\Delta\hat{\sigma}$, que representa a distância angular entre os pontos na esfera. A distância em quilómetros entre estes é dada por $r * \Delta\hat{\sigma}$, onde r é o raio da esfera em kilómetros.
Tem-se que o raio da Terra[i] é aproximadamente 6378.137 km, portanto $r = 6378.137$. É então apresentada a fórmula de Haversine, definida pela função,

$$\Delta\hat{\sigma} = 2\arcsin\left(\sqrt{\sin^2\left(\frac{\Delta\phi}{2}\right) + \cos\phi_i\cos\phi_f\sin^2\left(\frac{\Delta\lambda}{2}\right)}\right). \tag{2.1}$$

2.1.3 Sistema de posicionamento global

O sistema de navegação por satélite fornece a um dispositivo receptor móvel a posição do mesmo, assim como informação horária, sob quaisquer condições atmosféricas, a qualquer momento e em qualquer lugar na Terra. Isto desde que o receptor se encontre no campo de receção de quatro satélites GPS. A Captura da posição através do GPS é feita em três passos, sendo eles: a identificação dos satélites, a medição da distância e o cálculo da posição. São estes três passos que são descritos de seguida [Küp05].

Identificação dos satélites. Como primeiro passo, o receptor tem de identificar os satélites a serem usados para calcular distâncias. Normalmente, o número de satélites ao alcance do receptor está na ordem dos 5 a 10. Se o receptor não tiver qualquer informação sobre a última posição nem sobre o Almanaque[ii], tenta receber a informação dos satélites mais próximos. A este processo chama-se arranque frio. Se tiver informação da última localização e do almanaque, pode estimar a posição tentando aceder aos últimos satélites conhecidos. A este processo chama-se arranque morno. Por último, se o receptor guarda uma efeméride[iii] válida, calcula as posições dos satélites de uma forma muito mais rápida e precisa. Este processo chama-se arranque quente. Depois de identificados os satélites, o receptor escolhe um conjunto de pelo menos quatro satélites. Esta seleção depende da geometria entre cada satélite e o receptor. Tipicamente, as durações de receptores de baixa qualidade são na ordem dos 40 a 60 segundos para arranque frio, 30 a 40 para arranque morno e de 5 a 15 para arranque quente.

[i] http://scienceworld.wolfram.com/astronomy/EarthRadius.html

[ii] Os dados de Almanaque informam ao receptor GPS onde cada satélite deveria estar em qualquer hora ao longo do dia.

[iii] Os dados de efeméride contêm informações importantes sobre a situação de cada satélite (elementos de Kepler e parâmetros associados para compensar forças perturbadoras) como a data e a hora atuais.

Cálculo de distâncias. Os satélites são referenciados no espaço, já que com as informações de navegação transmitidas é possível calcular as suas coordenadas no espaço num dado instante. Assim, pelo processo de trilateração, tendo as coordenadas de pontos e distâncias a um ponto, pode obter-se as coordenadas deste último. O cálculo da distância é feito da seguinte forma: mede-se o tempo que o sinal demora a chegar do satélite até ao receptor, dividindo-se posteriormente pela velocidade de propagação do sinal. O cálculo das distâncias tem de ser feito usando pelo menos quatro satélites, três para calcular as posições em três dimensões e o quarto para sincronizar o tempo entre os satélites e o receptor. Os receptores de GPS estão equipados com relógios de quartzo, estes menos precisos que os relógios atómicos dos satélites.

Cálculo da posição. As distâncias calculadas são na realidade pseudo distâncias, que diferem das distâncias devido a um erro determinado. Estes erros devem-se à reflexão ionosférica, que é responsável por abrandar os sinais vindos dos satélites e originar distâncias falsas. Para compensar estes erros, os GPS transmitem os coeficientes de uma fórmula de correção, que o receptor aplica nas distâncias calculadas para conseguir resultados mais rigorosos. No passo seguinte, o receptor deve calcular as coordenadas de cada satélite considerado durante o cálculo. Para cada satélite é extraída a efeméride da mensagem e calculada a posição do satélite, bem como tempo de transmissão. As coordenadas são dadas no sistema de coordenadas geográficas elipsoidal que utiliza o datum WGS84[i].

Hoje em dia, cada vez mais, os dispositivos móveis estão adaptados para receber dados capturados por GPS. A precisão dos dados recebidos vem evoluindo, adaptando-se à realidade e consequentemente contribuindo para a credibilidade desta informação [DM07].

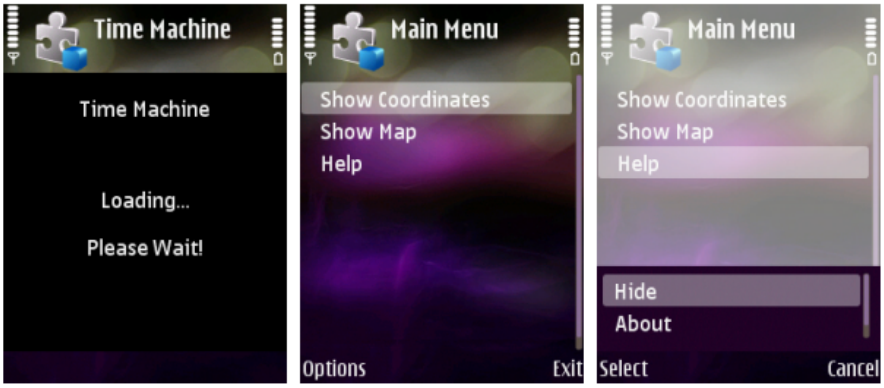
2.1.3.1 Precisão da posição

A precisão do GPS pode ser afetada por variados fatores [gps11], tais como a posição dos satélites, o ruído no sinal de rádio, as condições atmosféricas e os objetos sólidos entre os satélites e o receptor. O ruído pode criar um erro entre 1 a 10 metros, enquanto objetos como árvores, montanhas e grandes edifícios, podem induzir em erros até aos 30 metros. A maior precisão é conseguida quando os satélites e o receptor têm uma visão clara entre eles, sem interferência de outros objetos. É frequente a perda do sinal GPS quando o receptor perde o contacto com os satélites. Isto pode acontecer por erros do GPS, mas principalmente por obstrução do sinal. Em grandes cidades, sítios com muitos edifícios e grandes estruturas, é onde este problema mais se faz notar. É também nestas áreas onde o sinal tem mais ruído e consequentemente uma pior precisão.
A problemática da precisão do recetor GPS pode ser complexa [Gil11], mas geralmente obtem-se uma posição dentro de um raio de 15 metros da verdadeira posição.

[i]Datum refere-se ao modelo matemático teórico da representação da superfície da Terra ao nível do mar, o datum WGS84 é o utilizado pelo GPS.

2.2 Time Machine

No âmbito do projeto Time Machine já foi elaborada uma dissertação [Amo10] que explora a análise e visualização dos padrões pessoais de movimento. Neste âmbito foi desenvolvida uma aplicação que recolhe os dados do GPS, analisando-os e apresentando uma cartografia dos padrões encontrados. Esta aplicação foi desenvolvida para telemóveis e funciona sobre a plataforma Java2ME. A figura 2.3 é ilustrativa do menu inicial da aplicação.



(a) Ecrã de *Loading* da aplicação (b) Ecrã principal da aplicação (c) Menu de Opções do ecrã princi-

Figura 2.3: Ecrã inicial da aplicação Time Machine [Amo10].

Esta aplicação captura, armazena, e faz uma análise dos dados para depois poder apresentar várias visualizações diferentes. Nas próximas secções são apresentados e discutidos os avanços já feitos no âmbito do projeto. Serão apresentados todos os aspectos de alguma forma considerados relevantes para esta dissertação.
Na secção 2.2.1 é descrita a forma como os dados são armazenados, enquanto na secção 2.2.2 a análise que é feita sobre estes. Na secção 2.2.3, são apresentadas algumas formas de visualização e por último na secção 2.2.4 é feita uma apreciação global e discussão sobre o trabalho já desenvolvido no âmbito do projeto.

2.2.1 Dados armazenados

A aplicação desenvolvida captura periodicamente dados espacio-temporais do utilizador. O intervalo de captura é configurável, mas por omissão está configurado para recolher dados de 30 em 30 segundos. Os dados são todos guardados num ficheiro de texto, sendo criado um ficheiro para cada dia. Em cada ficheiro de texto criado pela aplicação são guardados o ano, mês, dia, hora, minuto, latitude e longitude no seguinte formato:

inteiro,mês/dia/ano,hora:minuto,latitude,longitude,resto

Cada ficheiro de texto acumula a informação das várias recolhas de dados ao longo do dia. Assim, o utilizador pode manter um registo de todas as atividades ao longo de vários dias/meses. Neste formato o primeiro *inteiro* é um valor que não é utilizado e apenas serve

para identificar a entrada. Os dados temporais guardados (*ano*, *mês*, *dia*, *horas* e *minutos*) são os do sistema. As coordenadas, *latitude* e *longitude*, são as correspondentes à data registada anteriormente e vêm em graus decimais, segundo o datum WGS84. A precisão das coordenadas é configurável, contudo a usada até ao momento é de três casas decimais. Os dados de cada captura ficam divididos por linhas.
A escrita no ficheiro é feita automaticamente, permitindo assim ao utilizador exportar o ficheiro sem ser necessário encerrar a aplicação e continuar a registar todos os movimentos efetuados.

2.2.2 Análise feita aos dados

A aplicação cria uma estrutura baseada em locais, a partir dos dados que capta. Um local é um sítio no espaço representado pelas suas coordenadas: latitude e longitude. Por várias vezes nos registos GPS aparecem entradas que contêm a mesma posição física, sítios como casa, escritório, universidade ou ginásio. Portanto, tendo um local definido como um conjunto único das coordenadas, latitude e longitude, podem começar a ser analisados os dados para cada local e o respetivo comportamento do indivíduo em relação a estes. Cada local tem duas variáveis associadas, sendo elas: duração e frequência. A duração representa o tempo total, em minutos, que o indivíduo passou nesse local. A frequência é o número de vezes que o local foi visitado.

Veja-se o exemplo em que um indivíduo vai três vezes ao restaurante 'Raposa', no qual permaneceu 20, 60 e 5 minutos respetivamente, de cada uma das vezes. Este local deverá ter uma duração com o valor de 85 minutos e frequência de valor 3. Relembra-se que a frequência é o número de vezes que o local é visitado em todo o período de recolha de dados.

2.2.3 Visualização dos dados

A visualização dos dados está fora do âmbito desta dissertação, que se destina a encontrar padrões pessoais nos dados para suportar a visualização. Contudo, para melhor apresentar o que já foi explorado ao nível do projeto, é aqui ilustrado um tipo de visualização que se pode construir com estes dados.

Nesta visualização o utilizador pode escolher o intervalo de dias que deseja visualizar. No caso da fig. 2.4, a visualização corresponde a um período de três dias de captura de dados. Cada círculo na figura representa um local. A posição de cada local na figura 2.4 está diretamente relacionada com a localização física do ponto. Assim, por exemplo, um local mais a Sul está representado mais abaixo na janela. O tamanho de cada local ou círculo, significa a respetiva duração (quanto mais tempo maior o círculo). Sendo a frequência ilustrada pela cor dos círculos que varia entre branco, para a frequência miníma, e preto para a frequência máxima.
Os locais parecem estar quase juntos, mas esta aproximação deve-se ao tamanho reduzido do ecrã e à consequente aplicação do zoom automático. Assim, se algum local estiver fora do campo de visão, a aplicação diminui a escala da visualização para que todos se tornem visíveis no final.

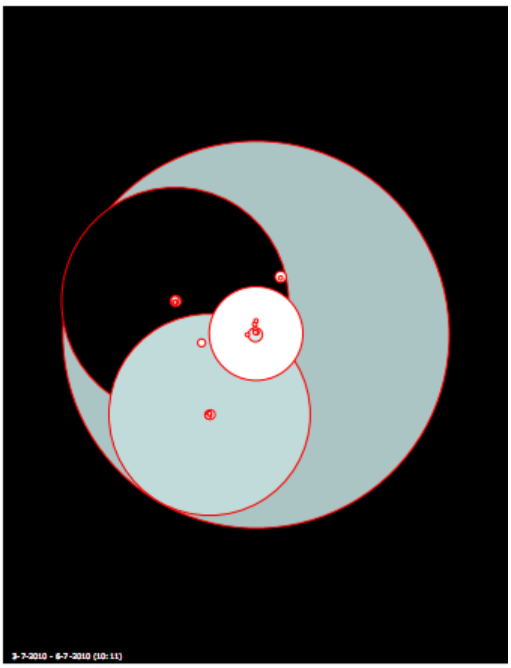


Figura 2.4: Visualização para vários dias da aplicação Time Machine desenvolvida previamente [Amo10].

Foi objetivo desta aplicação experimentar formas diferentes de representar a informação sobre os padrões de movimento do indivíduo. A aplicação foi desenvolvida para permitir diferentes visualizações. Os três tipos de visualização criados permitem ao utilizador analisar os seus próprios movimentos. Possibilitam analizar o seu percurso e comportamento durante um período por este definido, bem como gerar a análise com base em frequência. Sendo que a última dá a oportunidade ao utilizador de analisar o dia presente com base nos dias anteriores. Estas análises e visualizações realizam-se de forma instantânea, permitindo ao utilizador uma rápida perceção dos seus dados.

2.2.4 Discussão

Os desenvolvimentos já realizados no projeto foram mais focados na visualização e não tanto na análise dos dados. Contudo, o trabalho desenvolvido em certos aspectos como a definição de locais e suas variáveis (duração e frequência), são bastante úteis para retirar conclusões que vieram ajudar no desenvolvimento desta dissertação.

O problema mais relevante, que pode ter sérias implicações, é que um local nem sempre é definido pelo mesmo par de coordenadas. Isto provoca que alguns locais fiquem divididos em várias entradas ou que locais próximos fiquem juntos. Como tal, importa considerar dois fatores: os locais abrangerem mais que um par de coordenadas ou ao erro do GPS.
Aconteceu frequentemente existirem dois ou três locais na proximidade do local importante. No mesmo local é possível obter duas a três coordenadas diferentes, num curto espaço de

tempo. O erro do GPS, explicado na secção 2.1.3.1, faz com que mesmo que o recetor esteja parado, este continue a receber diferentes coordenadas referentes às imediações. Ou seja, a mesma posição física do recetor não recebe obrigatoriamente a mesma longitude e latitude.
O outro problema não se deve à imprecisões do GPS, mas sim, a erros de precisão, sendo que os locais não têm todos as mesmas dimensões. Recorde-se que a resolução escolhida para as coordenadas foi de três casas decimais, o que segundo a tabela 2.1 nos leva a uma precisão de aproximadamente $111m$. Ficou então assumido que os locais teriam todos uma dimensão na ordem dos $111m^2$ (isto ao nível do Equador). Contudo, isto não é verdade. Ao assumir um tamanho fixo na precisão fica-se sujeito a dois tipos de falhas: sub-dividir locais e acumular locais diferentes no mesmo.
Os problemas mencionados dão origem a valores incorretos para as variáveis de cada local. Não é possível estudar o comportamento de um indivíduo em relação aos seus locais importantes, se estes não estiverem bem definidos.

Torna-se então óbvio que para se conseguir achar padrões com base na interação que o utilizador tem com os locais, estes têm de estar bem claros. Passa assim, a ser foco desta dissertação encontrar forma de conseguir definir cada local, para depois se poder trabalhar e estudar sobre estes resultados. Uma vez esse problema resolvido, será também importante encontrar forma de tentar evitar erros de GPS que possam afetar os dados, usando as suas medidas de precisão.

2.3 Extração de informação relevante

Embora os humanos tenham padrões relativamente arbitrários de movimento, existem rotinas que são facilmente identificáveis na vida das pessoas. Estas podem ser encontradas dentro de várias escalas de tempo. Nesta secção são apresentados os resultados mais relevantes da pesquisa bibliográfica efetuada. O material aqui referido corresponde a estudos anteriores, que de alguma forma suportaram e fundamentaram o rumo escolhido para o desenvolvimento desta dissertação.

Eagle *et al.* em [ESP06], introduzem um sistema para deteção de sistemas sociais complexos, com dados recolhidos a partir de 100 telemóveis de estudantes durante 6 meses. Estes dados contêm informação sobre a localização temporal, localização física (resultante das antenas telefónicas) e contexto social (dada pela informação de proximidade derivada do Bluetooth). Este tipo de localização física foi ainda classificado em três tipos de locais: casa, trabalho ou outro.
Utilizam estes dados para desenvolver estudos no sentido de reconhecer padrões pessoais na vida diária dos utilizadores, identificar locais socialmente importantes e modelar ritmos organizacionais. Em particular, desenvolveram um Hidden Markov Model simples que depois de treinado com um mês de dados de alguns utilizadores, foi capaz de fazer uma muito boa separação, com mais de 95% de precisão, entre os grupos $\{\{casa\},\{trabalho\},\{outro\}\}$. Este modelo é condicionado pela hora do dia e por ser dia-de-semana ou não.

Utilizando o mesmo conjunto de dados que [ESP06], Farrahi *et al.* em [FGP07], apresentam uma estrutura para classificar as rotinas diárias das pessoas. Tentam classificar dias como sendo dias de semana ou fins-de-semana e ainda dias pertencentes à vida de estudantes de engenharia ou de negócios. A conjugação dos dados de localização física e de proximidade em diferentes intervalos de tempo e a utilização de mais de 87 000 horas de dados, permitiu-lhes uma percentagem de resultados corretos acima dos 80%, utilizando o algoritmo de classificação Support Vector Machine.

Estes dois estudos acima abordados não são individuais. O objetivo é estudar grandes quantidades de utilizadores, ao invés do indivíduo como ser único. Têm assim a vantagem de dispor de um grande conjunto de dados para treinar os seus modelos, baseando-se nesse principio para conseguir tal tipo de resultados. Têm ainda um conjunto de dados diferente do aqui utilizado. Não beneficiam de uma localização física tão precisa como a do GPS, mas dispõem também de informação de proximidade através do *Bluetooth*.
No contexto do projeto Time Machine, é preciso considerar que os conjuntos de dados a serem estudados são reduzidos. O estudo tem por objetivo ser feito no dispositivo móvel de cada indivíduo. Serão usadas normalmente algumas semanas de dados dos movimentos do utilizador.

Já em [ZBST07], são também usadas técnicas de classificação, mas aqui para classificar locais. São extraídas características dos locais (já pré-processados através de dados GPS) para os definir e classificar a sua importância.
Zhou *et al.* compararam dois classificadores (K-Nearest neighbor (KNN) e C4.5) com conjunções diferentes de atributos, num mês de dados de 28 utilizadores. Conseguiram muito bons resultados com o classificador KNN e o conjunto de atributos $\{Readings, ReadingDays, Visits, VisitDays\}$, onde *readings* é o número de leituras do local feita dos ficheiros, *ReadingDays* o número de dias diferentes a que equivalem essas leituras, *Visits* o número de visitas ao local e *VisitDays* o número de dias diferentes em que se visitou o mesmo.

Na arquitetura desenvolvida em [CMR07] é feita numa primeira fase a identificação semântica dos locais através de *reverse geocoding*[i]. Posteriormente, são utilizadas redes Bayesianas para tentar descobrir a rotina das pessoas para cada local, classificando cada um com um estereótipo. Foram aqui considerados os tipos casa, trabalho, restaurante, bar e discoteca.
Contudo, este processo precisa de interação com os utilizadores e de definir estereótipos de utilizadores e locais à partida. As experiências feitas por estes revelaram resultados muito pobres para a parte de classificação de rotinas. Apenas cerca de 64% dos locais são classificados corretamente, de acordo com os tipos definidos. A ideia de extrair informação semântica dos locais através de *reverse geocoding*, pode ser útil e já está a ser desenvolvida em paralelo no projeto, por outro elemento do grupo de trabalho.

Em [AS03], um modelo preditivo dos movimentos do utilizador é elaborado. Os modelos

[i]Reverse geocoding é o processo de transformar um ponto (latitude,longitude) num endereço ou nome de local.

de Markov foram utilizados para representar as transições entre os locais, sendo o movimento futuro depois previsto com base na transição com maior probabilidade do local corrente. Através deste modelo tornam-se também visíveis sequências de locais que ocorrem frequentemente. Revelando-se assim alguns padrões de movimento relevantes na vida dos utilizadores.

Hariharan e Toyama em [HT04], no âmbito do projeto Lachesis, propõem uma definição rigorosa para o historial de locais. Criam sobre este historial modelos probabilísticos para extrair dados relevantes. Sendo possível, com bases nestes, averiguar semelhanças entre períodos de tempo e gerar modelos estatísticos sobre os locais. No entanto, têm de definir intervalos de 30 minutos, onde o utilizador só pode fazer uma transição.

Liao *et al.* em [LFK07], utilizaram uma aproximação completamente diferente. Fizeram uso de Hierarchical Conditional Random Fields para gerar um modelo consistente das atividades e locais de um utilizador, onde obtiveram resultados muito bons.
Contudo, esta aproximação excede o âmbito do projeto Time Machine. São necessários registos GPS com dados a cada segundo e um grande poder computacional para processar estes dados. Como já explicado, o projeto tem limites em termos de poder computacional e de bateria devido a ser pensado para um dispositivo móvel, o que não permite pensar em tal tipo de aproximação.

2.4 Identificação e definição de locais

Os sistemas que usem o GPS para detetar locais têm de incluir um método para diferenciar os locais que são importantes daqueles que podem ser ignorados. Varias aproximações têm sido exploradas para tentar definir os locais pessoais relevantes. Nesta secção, é explorado o trabalho já desenvolvido nesta área.

No projeto Time Machine o interesse é um sistema capaz de definir os locais importantes para o utilizador automaticamente, sem o incómodo de ter de o questionar. Pretende-se que os locais sejam considerados importantes com base nas duas variáveis, frequência e duração. Ou seja, um local onde se vai várias vezes e se permanece pouco tempo é importante, tal como um sítio onde se foi poucas vezes mas se dispendeu muito tempo.
Este problema pode ser dividido em duas partes. Primeiro é necessário diferenciar nos dados do GPS quando se esteve parado num local. Ou seja, encontrar os pontos onde o utilizador esteve parado por um certo tempo e distinguir isso de quando está em movimento. Estes pontos chamam-se pontos de estadia, para os quais será usada a designação inglesa *stay points*. Depois, é preciso descobrir quais desses pontos pertencem aos mesmos locais, definindo assim, cada local pelos *stay points* que o constituem e delimitam.
A restante secção será dividida em três subsecções referentes a cada parte do problema e uma última para discutir as soluções propostas. Na secção 2.4.1 é visto como detetar *stay points* a partir das posições GPS e de seguida na secção 2.4.2 serão vistos diferentes maneiras de determinar os locais importantes. Por último, na secção 2.4.3 são discutidas as diferentes técnicas.

2.4.1 Deteção de locais

A maior parte dos locais habituais são dentro de algum edifício e, dependendo da estrutura do edifício, acontece muitas vezes que o sinal de GPS seja perdido. No seu trabalho para o sistema comMotion [MS00], Marcmasse e Schmandt usaram a perda de sinal para detetar pontos de estadia. Quando o sinal de GPS era perdido e mais tarde adquirido, dentro de um certo limite de distância, é assumido que esteve num edifício. Este processo evita a falsa deteção de edifícios quando da passagem por túneis ou falhas de hardware, como perda de bateria. Quando tal acontece, os utilizadores são confrontados com uma asserção com o objetivo de confirmar se aquele é mesmo um local ou se deve ser ignorado.
Sendo que a pista para identificar locais é a perda de sinal GPS, apenas locais como grandes edifícios podem ser detetados através deste sistema. Muitos locais, com relevância na vida dos utilizadores são assim desprezados, tais como parques, esplanadas ou até mesmo pequenos edifícios onde o sinal GPS continua a ser capturado. Mesmo dentro de edifícios, locais perto da janela ou no último andar, o GPS continua muitas vezes a receber sinal.

Kang *et al.* [KWSB04] utilizaram o endereço MAC dos pontos de acesso de uma rede Wi-fi para capturar a posição física no campus. Desenvolveram um algoritmo de agrupamento baseado no tempo para extrair os locais. Um local novo é descoberto quando a distância para o anterior ultrapassa um limite d e essa nova estadia atinge um limite significativo de tempo t. Este algoritmo é simples e trabalha de forma incremental em dispositivos móveis. Esta aproximação é semelhante à apresentada na secção 2.4.1.1, abaixo.

2.4.1.1 Deteção de pontos de estadia (*stay points*)

Mais recentemente, Li *et al.* [LZX+08], definem um algoritmo para a deteção de *stay points*, a partir dos ficheiros com os dados recolhidos do GPS. Um *stay point* é caraterizado por uma certa região geográfica dentro da qual o utilizador permanece por um certo tempo. Na figura 2.5, em particular, podem-se ver dois *stay points*.

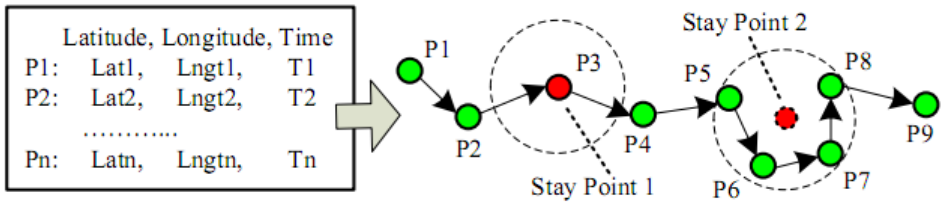


Figura 2.5: Ficheiro GPS e *stay points*. [LZX+08]

Estes dois *stay points* correspondem a duas situações distintas. Numa das situações (*Stay Point 1*) a estadia ocorre em $P3$, onde o utilizador permanece por um período que excede um limite de tempo. Na maior parte dos casos, esta situação acontece quando o utilizador entra num edifício e perde o sinal do satélite até voltar a sair. Na outra situação (*Stay Point 2*) o

utilizador anda dentro de uma certa região por um período de tempo. Neste caso, alguns pontos GPS ($P5$, $P6$, $P7$ e $P8$) são envolvidos nesta região. Consequentemente, a média das coordenadas da região deve ser calculada. Esta situação costuma acontecer quando se anda na rua e se é atraído por locais de interesse em redor.
A extração de *stay points* dos pontos GPS é dependente de dois fatores de escala: um limite de distância *distTresh* e um limite de tempo *timeTresh*. O *disTresh* representa a distância máxima, a que as coordenadas podem estar desde o ponto inicial da região, para pertencerem a este. *timeTresh* representa o mínimo de tempo que o utilizador tem de permanecer na região para que o conjunto de pontos dentro desta sejam considerados um *stay point*. Estes parâmetros devem ser regulados conforme as necessidades.

Algoritmo 1 StayPointParser [LZX+08]

Input: A GPS log P, a distance threshold *distThresh* and time span threshold *timeThresh*.
Output: A set of *stay points* SP={S}.

1: $i = 0$,$pointNum = |p|$;//the number of GPS points in a GPS log
2: **while** $i < pointNum$ **do**
3: $j = i + 1$;
4: **while** $j < pointNum$ **do**
5: $dist = Distance(p_i, p_j)$; //calculate the distance between two points
6: **if** $dist > distThresh$ **then**
7: $\Delta T = p_j.T - p_i.T$; //calculate the time span between two points
8: **if** $\Delta T > timetThresh$ **then**
9: $S.coord = ComputMeanCoord(\{p_k | i <= k <= j\})$;
10: $S.arvT = p_i.T; S.levT = p_j.T$;
11: $SP.insert(S)$;
12: **end if**
13: $i = j$;**break**;
14: **end if**
15: $j = j + 1$;
16: **end while**
17: **end while**
18: **return** SP;

Um *stay point* é caraterizado pela coordenada média ($S.coord$), um tempo de chegada ($S.arvT$) e um tempo de partida ($S.levT$). O algoritmo, que devolve um conjunto de *stay points* dado o registo GPS com o conjunto de pontos, é apresentado no algoritmo 1. A função $Distance(p_i, p_j)$ devolve a distância entre os dois pontos e a função $computMeanCoord(\{p_k | i <= k <= j\})$ devolve a coordenada média das coordenadas que pertencem ao *stay point*.
O algoritmo é iterativo e os *stay points* são detetados diretamente, dados os pontos captados pelo GPS, procurando regiões onde o utilizador tenha passado mais que um período de tempo. A complexidade computacional do algoritmo é $O(n)$, em que n é o numero de pontos GPS.

2.4.2 Determinação de locais

Determinar locais através de um conjunto de pontos é uma tarefa de *clustering*[i]. Aqui apresentam-se duas maneiras diferentes que são as mais utilizadas para se determinar locais. Na secção 2.4.2.1, apresenta-se um método por partição e na secção 2.4.2.2 um método baseado em densidade.

2.4.2.1 Por partição

Ashbrook e Starner, em [AS03], usaram o conhecido algoritmo de *clustering* K-means para identificar os locais importantes de um utilizador através do seu historial de locais. O K-means é um algoritmo iterativo de *clustering* bastante eficiente. Minimiza um termo de erro que consiste na soma dos quadrados das distâncias de cada ponto ao centro do seu grupo. Contudo, o algoritmo tem alguns aspectos negativos para agrupar os vários locais. Primeiro, o número de sítios tem de ser um parâmetro dado para o agrupamento, o que significa que este número tem de ser conhecido à partida. Segundo, todos os pontos são incluídos no resultado final dos grupos, o que torna este resultado muito sensível ao ruído. Um simples local que não seja importante longe dos outros pode puxar o centro de um grupo mais para perto dele do que aquilo que deveria, pois o quadrado da distância do erro pesa bastante em pontos mais afastados. Terceiro e por último, este algoritmo não é determinístico. Ou seja, o resultado final depende da escolha inicial aleatória dos centros dos grupos.
O algoritmo X-means é uma evolução do K-means em que o número de grupos é descoberto de forma automática. No entanto, este continua a partilhar das restantes limitações do K-means.
Detalhes sobre a implementação dos algoritmos K-means e X-means podem ser encontrados nas secções 2.5.2.1 e 2.5.2.2, respetivamente. Foram omitidas aqui as explicações para evitar repetições, uma vez que estes serão descritos mais à frente num contexto mais apropriado.

2.4.2.2 Baseada em densidade

A potencialidade dos algoritmos de *clustering* baseados em densidade torna-os bons candidatos para encontrar locais a partir dos pontos visitados. Zhou *et al.* [ZFL+04] desenharam o DJ-Cluster para descobrir locais pessoais importantes, um algoritmo baseado em densidade que resolve muitas das limitações do K-means.

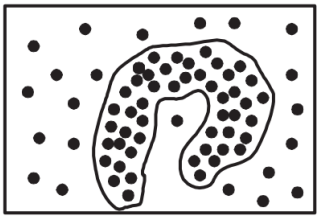


Figura 2.6: Grupo formado por algoritmo baseado em densidade. [ZFL+04]

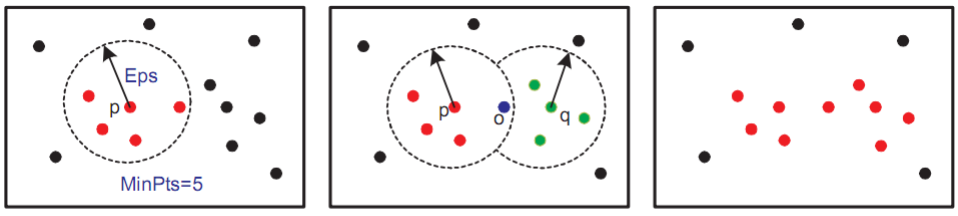
[i] *Clustering* é uma técnica de Data Mining para fazer agrupamentos automáticos de dados segundo o seu grau de semelhança.

Primeiro, este permite descobrir grupos com formas arbitrarias, como demonstrado na figura 2.6. Esta é uma melhoria significativa, uma vez que com o K-means todos os grupos são delimitados por um círculo, com um centro e um raio. Segundo, é menos sensível ao ruído, pontos muito afastados ou apenas pontos esporádicos têm menos probabilidade de participar no resultado final ou de afetar o restante resultado. Um utilizador pode, por exemplo, parar numa bomba de gasolina à qual nunca retorna ou parar em semáforos. Apropriadamente, estes acontecimentos geram poucos pontos que são descartados pois não preenchem o requisito de densidade. Terceiro, embora algoritmos baseados em densidade necessitem de alguns parâmetros (Eps e $MinPts$) estes são bastante menos robustos que o número de locais. Por último, tem resultados determinísticos. Recebendo os mesmos dados de entrada e os mesmo parâmetros produzem sempre os mesmos grupos, ao contrário do K-means que depende da escolha inicial arbitrária dos centros dos grupos.

DBSCAN [EpKSX96, SEKX98] é representativo de um algoritmo baseado em densidade. Contudo, há evidência de que o DBSCAN é demasiado sensível aos parâmetros e que não oferece nenhuma estratégia para trabalhar eficientemente com conjuntos de dados que não caibam na memória [ZFhLW02]. Devido a esses problema, o algoritmo DJ-Cluster foi desenvolvido [ZFL+04].

Algoritmo DJ-Cluster [ZBST07].

O algoritmo recebe um conjunto de pontos como entrada e para cada um calcula a sua vizinhança. A vizinhança de um ponto consiste nos pontos dentro de uma distância máxima representada por Eps, com a condição de que o número de pontos tem de ser no total, pelo menos, $MinPts$. Caso tal vizinhança não seja encontrada o ponto é marcado como ruído. Pelo contrário se for encontrado, os pontos fazem parte de um novo grupo. Se algum dos pontos já pertencer a outra localidade, estas são unificadas.



(a) *Density* − $N(p)$ (b) *Density* − *Joinable* (c) *DJ* − *Cluster*

Figura 2.7: Agrupamento com base em densidade. (a) ilustra uma vizinhança N do ponto p; (b) ilustra que $N(p)$ e $N(q)$ são acopláveis por densidade; (c) ilustra o grupo final em cor vermelha. [ZBST07]

As definições que se seguem definem a vizinhança baseada em densidade de um ponto e a relação de junção por densidade.

Vizinhança baseada na densidade de um ponto. A vizinhança N de um ponto p, denotada por $N(p)$, é definida pela equação (2.2),

$$N(p) = \{q \in S | dist(p,q) <= Eps\}, \tag{2.2}$$

onde S é o conjunto de todos os pontos, q é qualquer ponto da amostra e Eps é o raio do círculo à volta de p que define a densidade. $N(p)$ também necessita de satisfazer condição (2.3),

$$|N(p)| >= MinPts, \tag{2.3}$$

onde $MinPts$ é o número mínimo de pontos necessários numa vizinhança.

Junção por densidade. $N(p)$ é densamente agrupável a $N(q)$, denotado por $J(N(p),N(q))$, com respeito a Eps e $MinPts$, se houver um ponto tal que ambos, $N(p)$ e $N(q)$, o contenham. A relação de junção por densidade é ilustrada na figura 2.7.

Algoritmo 2 DJ-Cluster [ZBST07]

```
1: while there is at least one unprocessed point p in sample S do
2:    Compute the density − based neighborhood N(p) with Eps and MinPts.
3:    if N(p) is null then
4:       Label p as noise.
5:    else
6:       if N(p) is density − joinable to an existing cluster then
7:          Merge N(p) and all its density − joinable clusters.
8:       else
9:          Create a new cluster C based on N(p).
10:      end if
11:   end if
12: end while
```

O algoritmo DJ-Cluster é apresentado no Algoritmo 2. De notar que o algoritmo tem as seguintes propriedades:

- Todos os pontos fazem parte de algum grupo, ou então são ignorados e marcados como ruído;
- Existe sempre, pelo menos, um ponto em cada grupo;
- O algoritmo divide os dados de entrada em grupos não-hierárquicos;
- Não existe sobreposição entre os grupos.

A complexidade computacional do algoritmo é de $O(n^2)$, mas pode ser melhorada usando a indexação por R-Tree[Arg] para $O(nlogn)$, sendo n o número de pontos contidos no conjunto de pontos dado como entrada do algoritmo.

Análises feitas com os parâmetros mostram que Eps deve ser um valor aproximado ao grau de precisão da posição a ser usada, sendo este cerca de 30 metros para dados do GPS, tal como explicado na secção 2.1.3.1. Diferentes escolhas para o parâmetro $MinPts$ levam a resultados consideravelmente diferentes quanto ao número de locais importantes detetados [ZFL+04].

2.4.3 Discussão

Os algoritmos para a deteção de locais apresentados na secção 2.4.1 não consideram a repetição dos mesmo locais. Cada vez que descobrem um local, assumem que é um novo local e não têm como o comparar com os já conhecidos.
Por outro lado, os algoritmos de *clustering*, abordados na secção 2.4.2, aplicados diretamente a todos os pontos dos ficheiros GPS, originam a perda de sítios importantes tais como a casa ou centros comerciais e em vez disso, algumas regiões como passadeiras em que um utilizador passa bastantes vezes, mas não têm significado, são consideradas importantes (tal facto é ilustrado na figura 2.8).

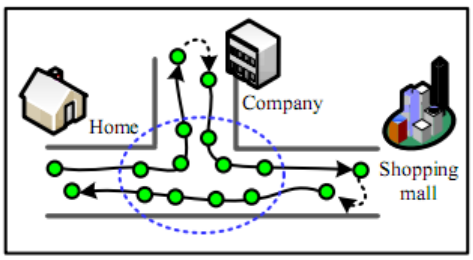


Figura 2.8: Deteção de locais baseada em *clustering*. [LZX+08]

Assim um pré-processamento dos dados, como o de deteção de *stay points* explicado na secção 2.4.1.1, é importante para extrair alguns locais importantes dos dados e ignorar os que não interessam. Depois é utilizado um algoritmo de *clustering* por forma a agrupar os pontos pertencentes aos mesmos locais e assim definir os locais importantes. A computação do *clustering* é também mais pesada se não for previamente feita esta seleção dos dados, pois seriam um número consideravelmente maior de pontos para processar.

Foi escolhido o algoritmo de *clustering* baseado em densidade DJ-Cluster, para a tarefa de determinar os locais pelas vantagens já mencionadas. Este permite definir locais com forma arbitraria, é menos vulnerável ao ruído, não necessita do número de locais como parâmetro e é determinístico. Contudo, tem também ele um problema que é preciso ter em conta. O algoritmo depende demasiado da densidade, entenda-se frequência, e não dá importância ao tempo dispendido nos locais, duração. Por exemplo, se um utilizador ficar bastante tempo num local e nunca mais lá voltar, este não será considerado. Perdem-se assim locais que podem, também eles, ser importantes.
No caso do projeto Time Machine pretende-se que os locais sejam identificados como importantes segundo estas duas varáveis: duração e frequência. Portanto a aproximação a este algoritmo terá de ter em conta que um local também pode ser importante mesmo que não tenha

uma grande frequência, mas sim elevada duração. O agrupamento de locais é feito da mesma forma com base na densidade, decidindo depois com base na duração e na frequência se o local é importante ou não. Esta aproximação ao algoritmo será abordada mais à frente na secção 3.2.1.

2.5 Deteção de padrões pessoais de movimento

Uma componente importante desta dissertação é a deteção de padrões pessoais de movimento. Desde há muito que se procuram encontrar padrões em diferente tipos de dados. Com o aparecimento dos computadores, houve interesse em encontrar mecanismos que tornem esta descoberta de padrões automática. *Data mining* é a área que estuda este processo de extração automática de padrões, a partir de dados guardados de forma eletrónica. Uma definição clássica de *data mining* foi dada por William Frawly, que a descreveu como *"The nontrivial extraction of implicit, previously unknown, and potentially useful information from data." [FJS+92].* Para tal efeito, são utilizados conhecimentos de domínios como a estatística, aprendizagem automática, reconhecimento de padrões, inteligência artificial e visualização de dados. Esta secção apresenta técnicas de aprendizagem automática para encontrar e descrever padrões estruturais nos dados. Para se entender o que significa aprendizagem automática, primeiro tem de se entender o que significa aprender.

Um programa diz-se que aprende de uma experiência E com respeito a alguma classe de tarefas T e medição de desempenho P, se o seu desempenho nas tarefas T, medida por P, melhorar com a experiência E [Mit97].

O processo de aprendizagem automática enfatiza o desenvolvimento de algoritmos e técnicas que implementem vários tipos de aprendizagem, mecanismos capazes de induzir conhecimento através de exemplos de dados. É o estudo de algoritmos computacionais que melhoram automaticamente com a experiência. A aprendizagem automática tem aplicação num vasto leque de domínios como processamento de linguagem natural, mecanismos de pesquisa, diagnósticos médicos e locomoção de robots [WF09].

Existem dois tipo principais de aprendizagem, designados por supervisionada e não supervisionada [WF09]. A aprendizagem supervisionada gera uma função que transforma os dados de entrada em dados de saída desejados. Esta corresponde aos problemas de classificação apresentados na secção 2.5.1, onde os dados de treino equivalem a conjuntos de dados de entrada e de saída. Os dados de entrada são normalmente um vector de valores, enquanto a saída é uma etiqueta que identifica a classe do objeto. O objetivo neste tipo de aprendizagem é construir uma função que aceite um vetor de dados válidos e consiga prever a sua classe. Para conseguir isto, o algoritmo tem de generalizar dos dados de treino para situações não vistas, duma forma razoável. Por sua vez, na aprendizagem não supervisionada os dados são apenas de entrada e o algoritmo tem de modelar o conjunto de dados de saída. Esta corresponde ao domínio das

técnicas de *clustering*, discutidas na secção 2.5.2. As técnicas de aprendizagem não supervisionada aplicam-se quando não existe classe para se prever e as instâncias têm de ser divididas em grupos naturais. Os dados de saída são uma representação que mostre como as instâncias são divididas pelos vários grupos, sendo a forma mais simples de o fazer associar o número do grupo a cada instância.

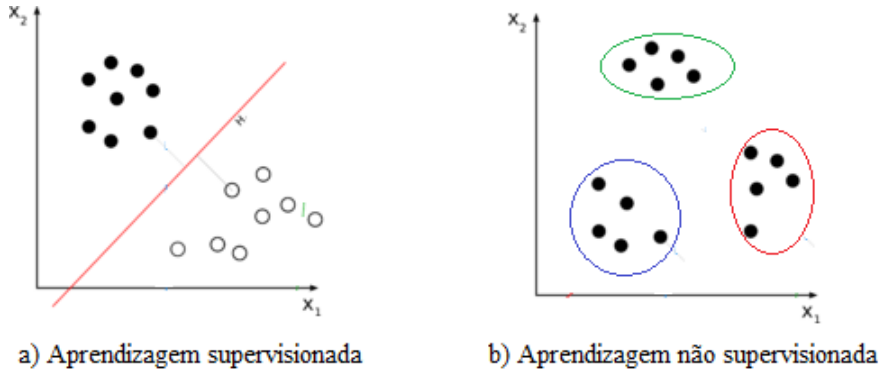


Figura 2.9: Aprendizagem supervisionada vs. aprendizagem não supervisionada

A figura 2.9 ilustra esta diferença. Neste exemplo, os dados de entrada são as coordenadas cartesianas x e y. Na aprendizagem supervisionada o resultado é demonstrado por duas marcas diferentes, conforme um ponto branco ou preto é dado. O classificador, divide o domínio dos dados de entrada em duas partes. Novos exemplos serão classificados com a marca correspondente à parte do domínio a que pertenceram. Por sua vez, na aprendizagem não supervisionada os dados de entrada são divididos em grupos. Neste caso a divisão foi feita em três grupos.

Ambos os tipos de aprendizagem serão relevantes para o desenvolvimento desta dissertação. Como exemplo, pretende-se fazer aprendizagem supervisionada de dias com o intuito de os classificar como dia de semana ou fim-de-semana. Por sua vez, a aprendizagem não supervisionada será útil em casos como a determinação de locais, visto na secção 2.4.2 ou posteriormente para fazer agrupamentos naturais destes, com base nas suas características. Em concreto, aspetos importantes e algoritmos relevantes de classificação e *clustering* serão apresentados de seguida.

2.5.1 Classificação

Em aprendizagem automática supervisionada os algoritmos a partir de instâncias fornecidas externamente produzem hipóteses gerais, usadas posteriormente para prever instâncias futuras. Por outras palavras, o objetivo da aprendizagem supervisionada é construir um modelo consciente da distribuição das classes em termos das características dos objetos. O classificador resultante é depois utilizado para atribuir os identificadores das classes às instâncias de testes, onde os valores dos atributos são conhecidos, mas a sua classe desconhecida [Kot07].

Instances	Feauture 1	Feauture 2	...	Feauture n	Class
1	xxx	x	...	xx	red
2	xxx	x	...	xx	green
3	xxx	x	...	xx	red
...					...

Tabela 2.2: Instâncias e seus atributos com as etiquetas conhecidas da classe correspondente.

Em problemas de classificação, os dados de treino correspondem a tuplos com os atributos desses dados e a etiqueta da respetiva classe. Tal como se pode observar na tabela 2.2, tem-se as várias instâncias, em que cada uma é definida pelo vetor dos atributos que a definem e pertence a uma classe. Os vetores correspondem aos dados de entrada, enquanto as etiquetas das classes são os resultados do classificador. A finalidade é construir uma função que aceite qualquer vetor válido como entrada e consiga descobrir a sua classe. Por forma a atingir este objetivo, é preciso generalizar dos dados com que é treinado para qualquer situação dentro do domínio, de uma forma ponderada.

É preciso ter em consideração que, no contexto do projeto Time Machine, os algoritmos utilizados são para ser implementados em dispositivos móveis. Estes, como já discutido, têm limitações a nível computacional e de autonomia. Logo, um equilíbrio entre o desempenho e o custo computacional dos algoritmos, tem de ser conseguido por forma a satisfazer as necessidades do projeto.
Vários estudos sobre os desempenhos de diferentes algoritmos já foram realizados em [HC93, HLL03, DP97]. Estes estudos utilizam vários conjuntos de dados de exemplos do mundo real para testarem os algoritmos. É concluído que os algoritmos mais básicos, nomeadamente o Naive Bayes, são quase tão eficazes quanto os mais complexos, como o Support Vector Machine (SVM). Os resultados entre os diferentes algoritmos têm pouca divergência, sendo que a precisão destes se revela similar variando em poucos pontos percentuais.
Uma vez que são necessários algoritmos com pouca complexidade computacional e uma diferença na precisão de poucos pontos percentuais não é significativa para os objetivos do projeto, foram escolhidos dois algoritmos simples. Um muito simples, 1R, descrito na secção 2.5.1.1, que serve de introdução ao Naive Bayes, apresentado na secção 2.5.1.2.

2.5.1.1 1R

Este é um algoritmo simples, o que torna fácil a sua compreensão. É a melhor forma de introdução aos algoritmos de classificação e serve também de ponte para o *Naive Bayes*. O algoritmo consiste numa forma clara de encontrar regras de classificação simples a partir de um conjunto de instâncias. Tem o nome de 1R, abreviado de 1-Regra, pois este toma a decisão baseando-se apenas no atributo mais importante dos dados de entrada. Tem baixo custo computacional e muitas vezes permite obter resultados muito satisfatórios, mais do que aquilo que seria de esperar. Talvez tal se deva ao facto de que a estrutura subjacente a muitos conjuntos de dados do mundo real seja rudimentar e apenas um atributo seja suficiente para determinar a classe de

um objeto com uma boa precisão [WF09].

Algoritmo 3 1R [WF09]

```
1: for all Attribute do
2:    for all Value of the attribute do
3:       Count how often each class appears.
4:       Find the most frequent class.
5:       Make the rule assign that class to this attribute-value.
6:    end for
7:    Calculate the error rate of the rules.
8: end for
9: Choose the rules with the smalest error rate.
```

O princípio geral é criar regras que testem apenas um atributo e ramifiquem em conformidade. Cada ramificação corresponde a um valor diferente do atributo e a classificação a dar a cada ramo é a classe que ocorrer mais vezes de acordo com os dados de treino. Posteriormente, é facilmente determinada a taxa de erro das regras. É apenas necessário contar o número de erros que acontecem nos dados de treino, correspondendo estes ao número de instâncias que não pertencem à classe predominante. Cada atributo gera um conjunto diferente de regras, uma regra para cada valor dos atributos. A taxa de erro para o conjunto de regras dos vários atributos é avaliada e a melhor escolhida. O algoritmo 3 mostra o seu pseudo-código.

Foi descrita a forma como o algoritmo 1R funciona, contudo o algoritmo mencionado está apenas preparado para atributos nominais[i]. Atributos numéricos podem ser convertidos em nominais utilizando um método simples de discretização. Este trata todos os valores numéricos como contínuos e usa um método direto para dividir o intervalo de valores em vários intervalos disjuntos [WF09].

Um estudo compreensivo sobre o desempenho do algoritmo de classificação 1R foi elaborado com base em 16 conjuntos de dados diferentes [HC93]. Os dados escolhidos foram frequentemente usados por investigadores de aprendizagem automática para avaliar os seus algoritmos. Surpreendentemente, apesar da sua simplicidade, o 1R obteve um bom desempenho em comparação com outros métodos de aprendizagem. As regras por este produzidas revelaram ser apenas alguns pontos percentuais menos precisas, em quase todos os conjuntos de dados, que abordagens mais complexas como algoritmos baseados em arvores de decisão.

2.5.1.2 Naive Bayes

Como se acaba de ver, o 1R utiliza apenas o atributo que entende ter maior peso para fundamentar a sua classificação. Outra técnica simples, é utilizar todos os atributos, de forma a permitir que todos tenham uma contribuição em partes iguais e independentes para a decisão final. Todavia, esta aproximação não é muito realista. O que torna os conjuntos de dados da

[i] Atributos nominais podem receber apenas uma quantidade finita de valores. Cada um desses valores pertence a uma classe. A estes dá-se também o nome de atributos categóricos.

vida real interessantes, é que os seus atributos não têm todos a mesma importância, nem são independentes. Porém, este algoritmo conduz a bons resultados através de uma aproximação simples. O classificador Naive Bayes é baseado em probabilidades condicionais. Este usa o teorema de Bayes: $P(A|B) := \frac{P(B|A)P(A)}{P(B)}$ [i], uma fórmula que calcula a probabilidade contando a frequência e combinação dos valores no historial dos dados [WF09].

Os classificadores probabilísticos operam nos conjuntos de dados onde cada amostra x consiste nos valores dos atributos $< a_1, a_2 ... a_i >$ e a função alvo aceita qualquer valor de um conjunto pré-definido e finito $V = (v_1, v_2 ... v_j)$. Classificar exemplos nunca vistos envolve o cálculo do valor alvo mais provável V_{max} que é definido por,

$$v_{max} = \max_{v_j \in V} P(v_j | a_1, a_2 ... a_i), \tag{2.4}$$

aplicando o teorema de Bayes, v_{max} pode ser definido como,

$$v_{max} = \max_{v_j \in V} P(a_1, a_2 ... a_i | v_j) P(v_j). \tag{2.5}$$

A regra de Bayes é usada para estimar a probabilidade condicional da etiqueta da classe y. Depois, são feitos pressupostos no modelo para decompor essa probabilidade no produto das probabilidades condicionais, sendo que os valores dos atributos são considerados condicionalmente independentes dado o valor alvo da classe. A fórmula utilizada pelo classificador de Bayes simples é:

$$v = \max_{v_j \in V} P(v_j) \prod_i P(a_i | v_j), \tag{2.6}$$

onde v é o valor de saída alvo do classificador é $P(a_i|v_j)$ e $P(v_i)$ pode ser calculado com base na sua frequência, a partir dos dados de treino.
A aplicação desta fórmula é direta para os atributos categóricos. Para atributos numéricos, pode-se modelar a componente de distribuição marginal de várias formas diferentes. A maneira mais simples é adotar uma forma paramétrica, por norma a escolhida é a distribuição normal[ii].

Existem muitos conjuntos de dados onde este classificador não tem um desempenho muito bom. Este facto ocorre principalmente por se considerar que os atributos são todos independentes e têm todos o mesmo peso. Contudo, existem ainda melhoramentos que podem ser feitos nesta técnica. A seleção de um sub-conjunto de atributos e a mudança no tratamento dos

[i] http://mathworld.wolfram.com/BayesTheorem.html
[ii] http://mathworld.wolfram.com/NormalDistribution.html

atributos numéricos são dois dos melhores sucedidos. Estas duas técnicas e ainda uma outra apelidada de *boosting*, podem ser encontrados em [SP04]. Onde o classificador Naive Bayes foi melhorado combinando o uso da discretização, escolha de atributos e procedimentos de *boosting*. Este classificador de Bayes melhorado foi testado em 26 conjuntos de dados padrão e conseguiu melhor precisão na maioria dos resultados, utilizando menos tempo de treino.

Naive Bayes fornece uma aproximação simples, com semântica clara, para representar, usar e aprender conhecimento probabilístico. Podem ser alcançados resultados muito bons. Tem sido mostrado frequentemente que este classificador é capaz de competir, e muitas vezes superar, rivais mais sofisticados e complexos. Pedro Domingos e Michael Pazzani concluem em [DP97] que o classificador Bayesiano se comporta bem na prática, mesmo quando estão presentes fortes dependências entre os atributos estão presentes, o que também é comprovado em [Ris05]. Foi ainda comprovado que esta técnica obtém frequentemente melhores resultados que outras mais poderosas quando o conjunto de amostra é pequeno. Isto mesmo em domínios onde o modelo de aprendizagem não é o mais apropriado. Contudo, estas experiências foram feitas por meio de domínios artificiais. O classificador simples de Bayes tem mais aplicabilidade do que anteriormente se pensara. Para além da sua precisão, traz vantagens em termos de simplicidade, velocidade de aprendizagem, velocidade de classificação e memória requerida.

2.5.2 *Clustering*

Classes ou grupos conceptualmente significativos de objetos, que partilhem características comuns, têm um papel importante na forma como o mundo é analisado e descrito. As técnicas de *clustering* são aplicadas quando não existe nenhuma classe para ser prevista mas, pelo contrário, as instâncias têm de ser divididas em grupos naturais. Estes grupos, presumivelmente, refletem algum mecanismo que trabalha no seu domínio, um mecanismo que leva algumas instâncias a terem uma semelhança mais forte entre as do próprio grupo, do que com as restantes. Quão maior a semelhança dentro de um grupo e maior a diferença entre estes, melhor e mais distintos são os resultados.
Por outras palavras, *clustering* é um processo de descoberta que agrupa conjuntos de objetos de dados, por forma a maximizar a semelhança entre os objetos dentro do mesmo grupo e a minimizar a semelhança entre objetos de grupos diferentes [KR90].

Em muitos casos, a noção de grupo não está bem definida. A figura 2.10 ilustra este problema. O conjunto de dados é formado por vinte pontos, que se podem dividir por grupos de três formas diferentes aceitáveis. Nas imagens (b) e (d) o conjunto é dividido em duas e seis partes, respetivamente. Porém, a aparente divisão dos dois grupos maiores em três sub-grupos pode apenas ser um artefato do sistema visual humano. Também é razoável dizer que os pontos formam quatro grupos, tal como se vê na imagem (c). Como se constata, a definição de grupo não é exacta, sendo dependente da natureza dos dados e dos resultados pretendidos.

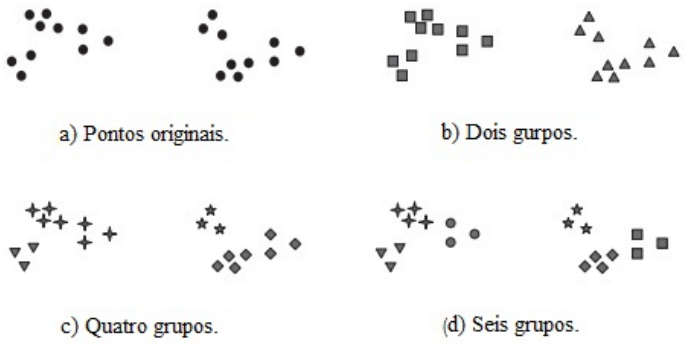


Figura 2.10: Formas diferentes de agrupar o mesmo conjunto de pontos [KR90].

Um dos algoritmos de *clustering* mais conhecido e utilizado é o K-means, que será explorado em seguida na secção 2.5.2.1. Não obstante, este tem alguns problemas conhecidos. O algoritmo X-means, explicado na secção 2.5.2.2, propõe uma solução para a resolução de alguns dos problemas do K-means.

2.5.2.1 K-means

O método K-means é uma técnica iterativa baseada na distância entre instâncias. É uma técnica de *clustering*, com várias aplicações, que procura minimizar a média do quadrado das distâncias entres os pontos no mesmo cluster. K-means é provavelmente o algoritmo mais usado de *clustering*, pois a sua simplicidade e velocidade são bastante apelativas [WF09].

A principal ideia do algoritmo é definir k centroides, um para cada grupo. Estes centroides devem ser atribuídos de uma maneira cuidada, pois diferentes valores atribuídos a estes, causam diferentes resultados. A melhor forma é escolhê-los o mais afastados uns dos outros que for possível. O próximo passo é pegar em cada instância do conjunto de dados e associá-la ao centroide mais próximo. Quando já não existir nenhum ponto pendente o primeiro passo está completo e os grupos iniciais formados. Depois recalculam-se k novos centroides, como centros provisórios para os grupos definidos anteriormente. Com estes novos centroides, uma nova distribuição dos pontos pelos centroide mais próximo tem de ser feita e aqui é gerado um ciclo. Como resultado deste ciclo, os centroides vão mudando a cada iteração, até não existirem mais mudanças ou alternativamente, até os pontos não mudarem mais de grupos. Finalmente, este algoritmo tem por objetivo minimizar uma função, neste caso, a função do quadrado do erro. A função objetivo,

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} ||x_i^{(j)} - c_j||^2, \tag{2.7}$$

onde $||x_i^{(j)} - c_j||^2$ é uma distância escolhida entre um ponto nos dados, $x_i^{(j)}$ é o centro do grupo

e c_j é um indicador da distância dos n pontos de dados aos seus respetivos centros de grupo. De notar que várias medidas de distância podem ser escolhidas.

Algoritmo 4 K-means [WF09]

1: Select k points as the initial *centroids*.
2: **repeat**
3: Form k *clusters* by assigning all *points* to the closest *centroid*.
4: Recompute the *centroid* of each *cluster*.
5: **until** The *centroids* doesn't change.

A simplicidade do algoritmo acabado de descrever pode ser comprovada pelo seu pseudo-código, no algoritmo 4. Existem outras variações, mas este algoritmo manteve-se popular pois converge extremamente rápido na prática. É de notar que se tem observado que o número de iterações necessárias é, tipicamente, bem menor que o número de pontos, o que leva o algoritmo a ter uma complexidade de $O(n)$, onde n é o número de instâncias.

Embora possa ser provado que o procedimento termina sempre, o algoritmo não garante um ótimo global. A qualidade da solução final depende largamente na escolha inicial do centro dos grupos e pode, na prática, vir a ser muito pior que o ótimo global. Uma vez que o algoritmo é extremamente rápido, um método comum é executá-lo algumas vezes e retornar o melhor agrupamento encontrado.

Ainda assim, o problema maior do K-means é que valor escolher para k. Normalmente nada é sabido sobre o número de grupos pretendidos e um dos objetivos principais é descobri-lo. Uma das formas de resolver este problema é tentar executar o algoritmo com vários valores para k e escolher o que tiver melhor valor final da função objetivo. Uma escolha inapropriada de k pode levar a resultados fracos. Este é um ponto que condiciona bastante a aplicabilidade deste algoritmo, em concreto, no caso do projeto Time Machine. Outros algoritmos existem onde o número de grupos não é exigido a-priori. O X-means, que será apresentado em seguida, é um desses algoritmos dinâmicos de *clustering*.

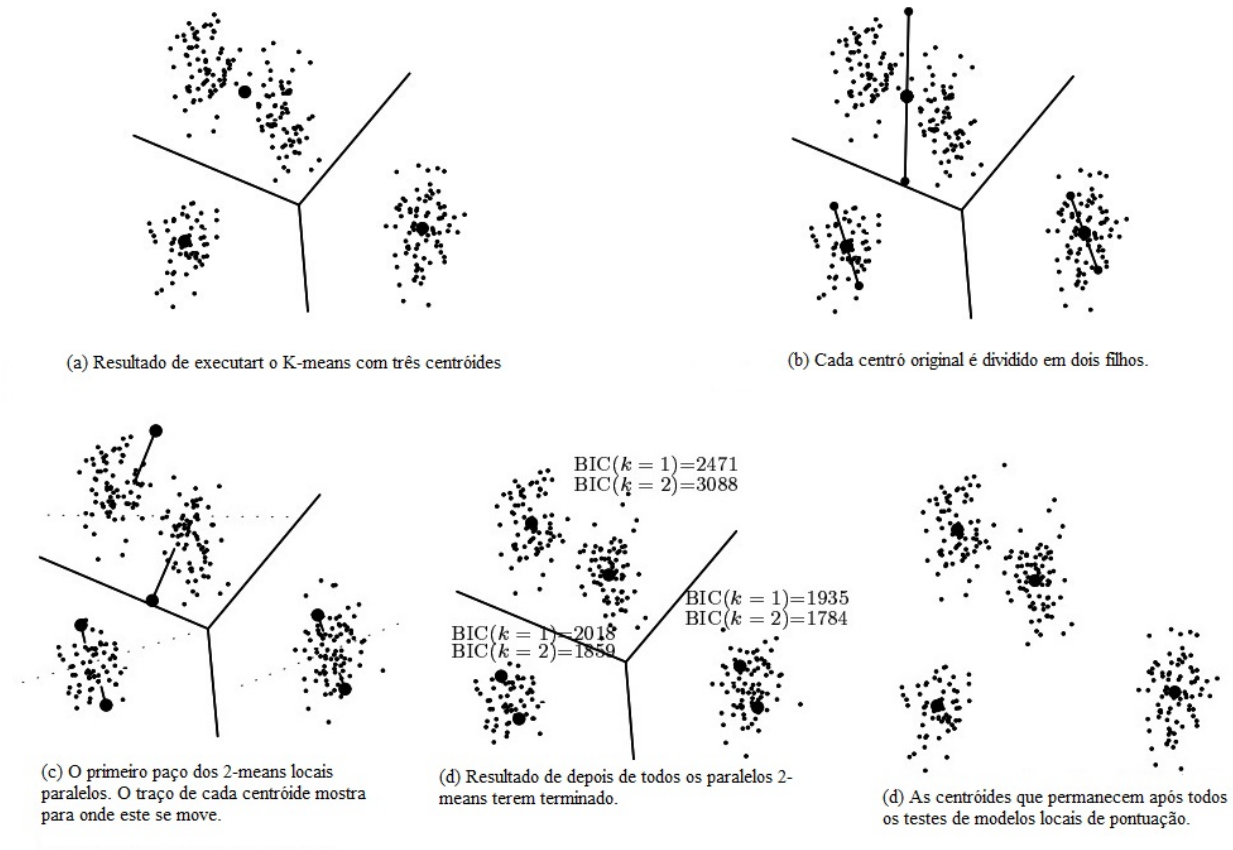
2.5.2.2 X-means

Como já mencionado, apesar da sua popularidade, o K-means sofre de alguns problemas. Tem problemas de escala, o número de locais k tem de ser dado e a busca é propicia a criar mínimos locais. O algoritmo de *clustering* X-means propõe soluções para os primeiros dois problemas e uma solução parcial para o terceiro. A arquitetura muda neste algoritmo e k em vez de ser um parâmetro de entrada passa a ser um resultado. Apenas é necessário definir um limite superior e outro inferior para k. O k escolhido é aquele que obtiver melhor resultado, segundo um modelo de seleção com critério, tal como o Bayesian Information criterion (BIC)[Bie06]. A velocidade do algoritmo é ainda melhorada, através da multi resolução em kD-tree e guardando informações estatísticas suficientes nos seus nós.

Algoritmo 5 X-means [PM00]

1: $k = K_{min}$.
2: **repeat**
3: $Improve - Params$
4: $Improve - Structure$
5: **until** $k > K_{max}$

O algoritmo começa com k a tomar o valor do limite inferior do intervalo dado e continua a adicionar centroides onde elas forem necessárias, até se atingir o limite superior. Durante este processo, o conjunto de centroides que alcança o melhor resultado é gravado e corresponderá ao resultado final. O algoritmo pode ser representado pelo pseudo-código no algoritmo 5.
A operação Improve-Params é simples, consiste em executar o K-means para uma convergência. A função Improve-Structure é responsável por encontrar onde a nova centroide deve aparecer.



(a) Resultado de executart o K-means com três centróides

(b) Cada centró original é dividido em dois filhos.

(c) O primeiro paço dos 2-means locais paralelos. O traço de cada centróide mostra para onde este se move.

(d) Resultado de depois de todos os paralelos 2-means terem terminado.

(d) As centróides que permanecem após todos os testes de modelos locais de pontuação.

Figura 2.11: Passos da função Improve-Structure, correspondente ao algoritmo X-means [PM00].

A estratégia para escolher a melhor centroide a dividir é explicada com a ajuda da figura 2.11. A figura está dividida em 5 imagens, em que cada uma representa um passo diferente da estratégia.

A imagem (a) mostra uma solução estável do K-means, com 3 centroides. As margens das regiões pertencentes a cada centroide são também apresentadas. A operação structure-improvement começa por separar cada centroide em dois filhos (figura 2.11(b)). Estes, estão a uma distância proporcional ao tamanho da região, em direções opostas, consoante um vetor arbitrário. De seguida, em cada região pertencente aos pais, um K-means local (com $k = 2$) é lançado para cada par de filhos. Os filhos, apenas dentro da região dos pais, lutam pelos pontos que lhes pertencem. A imagem (c) mostra o primeiro passo de todos os três 2-means locais executados. Já na imagem (d), vê-se onde os filhos eventualmente ficariam a seguir à computação de todos os 2-means locais.
Por esta esta altura, um teste modelo de seleção é feito em todos os pares de filhos. Uma pontuação é atribuída a cada centroide paterna e a cada par de filhos, utilizando BIC [PM00], como ainda se pode ver na imagem (d). De acordo com os resultados, os pais ou os filhos são mortos. A imagem (e) demonstra o que acontece depois do teste ter sido aplicado aos três pares de filhos.

Este processo também melhora o problema dos mínimos locais. Pelleg e Moore, criadores do algoritmo X-means, concluíram que execuções regionais com apenas 2 centroides têm tendência a ser menos sensíveis a mínimos locais [PM00].
Experiências feitas, mostram que esta técnica revela o número verdadeiro de classes nos conjuntos de dados e que é muito mais rápido que o simples uso do K-means com diferentes valores para k. Este, utiliza um critério baseado em estatística para fazer decisões locais que maximizam as probabilidades posteriores do modelo. Resultados experimentais em conjuntos de dados reais, mostram que este tem um desempenho melhor e mais rápido do que o K-means. [PM00]
Contudo, este tipo de algoritmos apenas consegue bons resultado para dados com poucos atributos. Em geral não devem ser considerados mais de 7 atributos para definir os objetos [WF09].

2.5.3 Discussão

A escolha do algoritmo que deve ser usado é sempre um passo crítico. Os desempenhos dos classificadores dependem muito das características dos dados a serem classificados. Não existe um classificador que se possa dizer melhor em todos os casos. Vários testes empíricos tiveram lugar com o objetivo de comparar desempenhos de classificadores e descobrir as características dos dados que determinam estes desempenhos. Contudo, encontrar um classificador apropriado para um dado problema é, ainda, mais uma arte do que ciência. Mais importante ainda que o algoritmo escolhido, é a descrição feita dos objetos. Ou seja, com que conjunto de atributos definir os dados, de forma a que estes consigam ser aprendidos para os fins pretendidos.

Todas estas características dos algoritmos de aprendizagem automática, serão tidas em conta na execução dos testes. Estes terão de ser realizados em vários algoritmos descrevendo os objetos por conjuntos de atributos diferentes. O objetivo é encontrar os melhores resultados,

não esquecendo o contexto em que estes algoritmos terão que executar.

2.5.4 Waikato Environment for Knowledge Analysis (Weka)

O projeto Weka Machine Learning está a ser desenvolvido na Universidade de Waikato, na Nova Zelândia, com o objetivo de construir uma ferramenta que corresponda ao estado de arte relativo a técnicas de aprendizagem automática. Nesta secção é explicado como esta funciona e porque foi escolhida. O objetivo, por enquanto, é perceber até que ponto estes algoritmos cumprem os objetivos do projeto Time Machine. Por outro lado, os algoritmos descritos e que são testados já existem há algum tempo, dispondo todos eles de várias implementações. Como tal, não seria compensatório nesta fase, estar a gastar tempo em mais uma implementação.

Foi elaborada uma pesquisa no sentido de encontrar implementações sólidas dos algoritmos a testar. Algumas implementações e bibliotecas foram ponderadas, por exemplo, Java Machine Learning Library (Java-ML)[i] e Bayesian Network Classifier Toolbox (jBNC tollkit)[ii]. Cada uma destas continha alguns dos algoritmos que pretendidos, contudo a sua utilização não foi simples por falta de manuais explicativos, nomeadamente relativos a processos de instalação e utilização.
Pelo contrário, a ferramenta Weka contém uma coleção que engloba todos os algoritmos necessários. Conta ainda com vários extras, como um manual bem estruturado, ferramentas auxiliares de processamento e visualização de dados, incluindo uma boa interface gráfica.
Todos os algoritmos carregam os seus conjuntos de dados de ficheiros Attribute-Relation File Format (ARFF). Este é o formato padrão para processar os conjuntos de dados na ferramenta Weka. Os ficheiros descrevem as instâncias independentes e desordenadas, sendo que cada instância contém um conjunto de atributos estático e pré-definido. Alguns algoritmos têm capacidade para trabalhar com dados em falta, enquanto outros não.
Depois dos dados serem carregados, um leque variado de algoritmos pode ser escolhido e as suas opções configuradas. Para além dos resultados, esta ferramenta providencia um módulo comum de avaliação do desempenho dos classificadores.

2.6 Previsão de movimentos pessoais

Outro objetivo desta dissertação e do projeto é a previsão dos movimentos futuros do utilizador. Nesta secção pretende-se encontrar forma de criar um modelo preditivo capaz de descobrir qual a localização futura do utilizador, a partir dos dados que disponíveis sobre o seu passado. Ashbrook e Starner em [AS03] utilizaram, com sucesso, modelos de Markov de forma a criarem dados estatísticos sobre as transições entre locais, prevendo locais futuros com base nas probabilidades das transições passadas. É apresentado de seguida, na secção 2.6.1, os conceitos dos modelos de Markov necessários e discutida esta aproximação na secção 2.6.2. Por fim, na secção 2.6.3, é apresentada a biblioteca com que se trabalhou neste sentido.

[i] http://java-ml.sourceforge.net/
[ii] http://jbnc.sourceforge.net/

2.6.1 Cadeias de Markov

Um modelo de Markov é um modelo estatístico que assume a propriedade de Markov. Geralmente, esta hipótese permite o cálculo com um modelo que, com todas as suas dependências, seria intratável. O que a propriedade de Markov diz, de uma forma simples, é que o próximo estado depende apenas do estado corrente e não do passado. As cadeias de Markov são o modelo de Markov mais simples. Estes são caraterizados por um sistema que transita de estado em estado e a cada transição tem uma probabilidade associada. São aqui estudadas as cadeias de Markov definidas num conjunto de tempo discreto, em que a cada passo se está num estado diferente. As mudanças de estado são definidas pelas transições que contêm, cada uma com a respetiva probabilidade. Os conjuntos de estados e de transições define o modelo.

Definição [Res92].

Assumindo o conjunto finito $S = 1,...,m$ de estados, com m estados diferentes, o modelo de Markov pode ser definido por uma matriz de probabilidades de transição. Cada par $(i, j) \in S^2$, contém a probabilidade de transição do estado i para o estado j, denotada por $p_{i,j}$. Tem que satisfazer as condições 2.8 e 2.9.

$$p_{ij} \geq 0 \forall (i, j) \in S^2; \tag{2.8}$$

$$\sum_{j \in S} p_{ij} = 1 \forall i \in S. \tag{2.9}$$

Tem-se a matriz $\mathbb{P}$,

$$\mathbb{P} = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1m} \\ p_{21} & p_{22} & \cdots & p_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ p_{m1} & p_{m2} & \cdots & p_{mm} \end{pmatrix}$$

A cadeia de Markov em espaço discreto é uma sequência de variáveis aleatórias X_1, X_2, X_3,... com a propriedade de Markov em que, dado o estado atual, os estados futuro e passado são independentes. Formalmente,

$$Pr(X_{n+1} = x | X_1 = x_1, X_2 = x_2, ..., X_n = x_n) = Pr(X_{n+1} = x | X_n = x_n), \tag{2.10}$$

em que os valores possíveis de x_i formam um conjunto finito de estados S.
As cadeias de Markov são geralmente descritas por grafos diretos, onde as arestas têm as respetivas probabilidades de transitar de um estado para outro, sendo que cada nó representa um estado. Esta propriedade define um modelo de Markov de primeira ordem, pois o próximo estado apenas depende do estado atual.

Uma variação deste modelo é a cadeia de Markov de ordem m (ou a cadeia de Markov com memória m), onde m é finito. Este é um processo que satisfaz a equação 2.11 para $n > m$.

$$Pr(X_n|X_{n-1} = x_{n-1}, X_{n-2} = x_{n-2}, ..., X_1 = x_1) = Pr(X_n = x_n|X_{n-1} = x_{n-1}, X_{n-2} = x_{n-2}, ..., X_{n-m} = x_{n-m}) \tag{2.11}$$

Por outras palavras, o estado futuro depende dos m estados passados. É possível construir uma cadeia Y_n, a partir de X_n, que tem a propriedade de Markov. Como se segue:

Seja $Y_n = (X_n, X_{n-1}, \ldots, X_{n-m+1})$ o m-tuplo ordenado de valores X. Então, Y_n é um cadeia de Markov com o espaço de estados S^m e com a propriedade de Markov.

2.6.2 Discussão

As cadeias de Markov são usadas para representar variáveis aleatórias, das quais se sabe informação de probabilidade, podendo para isto ser usadas para prever com base no passado. É assim possível formar um modelo de Markov fazendo uso do historial dos locais e com base neste prever o próximo local dependendo do corrente.

Um modelo Markov de primeira ordem permite responder a perguntas como "Estando em casa qual é o local seguinte mais provável?", enquanto um modelo de segunda ordem pode ser mais preciso por também considerar o local anterior. Por exemplo, se o indivíduo está no trabalho e antes veio do almoço é mais provável que a seguir vá para casa, enquanto que se vier da escola dos filhos e estiver no trabalho é mais provável ir almoçar. Neste caso o estado corrente em vez de ser apenas o local corrente é representado pelo local anterior e o corrente. Esta possibilidade, de criar modelos de diferentes ordens, levanta a questão de qual será a ordem adequada para o problema. Na prática, uma limitação natural é a quantidade de dados para análise, ou seja, o historial. Em [AS03] concluíram que mesmo com dados de 4 meses, as transições de segunda ordem criadas eram relativamente poucas.

2.6.3 Biblioteca Jgram

JGram[i] é uma biblioteca com uma implementação simples das cadeias de Markov, em linguagem Java. Esta biblioteca permite criar modelos de diferentes ordens. Cria o conjunto de estados, as respetivas transições e suas probabilidades com base numa sequência dados. Permite depois, com estes dados, que se gerem sequências aleatórias de tamanho definido. Aqui, não o interesse não gerar caminhos aleatórios, mas em descobrir qual o próximo local com base nos últimos m. Uma vez que esta biblioteca é *open source* permite fazer as pequenas alterações necessárias, de modo a ter o comportamento pretendido. Esta biblioteca é constituída pelas

[i] http://www.java.net/project/jgram-simple-java-markov-chain-n-gram-library

classes listadas na tabela 2.3, onde também é dada uma pequena explicação do papel de cada uma.

Sumário de classes

Parser	Constrói o modelo. Cria ou edita, *Grams*, *NGrams* e as transições.
Gram	Esta classe representa um estado discreto da sequência. Um local, no caso de uma sequência de locais.
NGram	Esta classe representa um *NGram*, uma unidade composta por *N* número de estados *Grams* em sequência.
Transition	Esta classe representa a transição de um *NGram* para um *Gram*.
AbstractCalculator	A classe base que todas as calculadoras têm de estender. As calculadoras servem para calcular o score de cada transição.
MaxCalculator	Uma calculadora simples que força a seleção da transição com maior probabilidade.
MinCalculator	Uma calculadora simples que força a seleção da transição com menor probabilidade.
SimpleCalculator	Uma calculadora relativamente simples baseada nas probabilidades. É a usada por omissão.
ComplexCalculator	Uma calculadora um pouco mais complicada baseada na *SimpleCalculator*.
Generator	Gera uma sequência aleatória de objetos com base nas sequências passadas, que foram filtradas pelo *parser*.

Tabela 2.3: Sumário de classes da biblioteca Jgram.

Como se pode ver, a classe *Gram* representa os diferentes estados. Depois, consoante a ordem *N* do modelo, a classe *Parser* cria *NGrams* com base nas sequências dadas. Cada *NGram*, é uma unidade composta por *N* número de estados *Grams* em sequência. Para cada *NGram* são também incrementadas e criadas, caso ainda não existam, as várias transições, conforme apareçam na sequência. É desta forma que o modelo é criado. Existem ainda várias calculadoras que, com base num certo *NGram*, calculam pontuações, dentro do intervalo $[0, 1]$ para cada transição deste para um *Gram*. As calculadoras, *MaxCalculator* e *MinCalculator*, atribuem pontuação 1 à *Transition* com mais ou menos ocorrências respetivamente e zero a todas as outras. A *SimpleCalculator*, atribui ao score os valores das probabilidades de cada transição ocorrer dentro do respetivo *NGram*. A *ComplexCalculator* faz um cálculo de pontuação mais complicado, que é o produto de três probabilidades: a probabilidade da *Transition* ocorrer dentro do *NGram* (pontuação da *SimpleCalculator*), a probabilidade de ocorrência do próximo *Gram* numa sequência e a probabilidade de ocorrência do próximo *NGram* também numa sequência. Por fim, a classe *Generator* que serve para gerar sequências aleatórias, com base no modelo.

3

Componentes do sistema desenvolvido

Neste capítulo é apresentado o sistema que foi desenvolvido, o qual integra as várias componentes desenvolvidas consoante os processamentos necessários. O sistema tem como dados de entrada ficheiros com eventos captados pelo GPS, que são processados com o objetivo de extrair informação que revele padrões. O sistema foi desenvolvido de forma modular consoante as necessidades e os resultados obtidos.

Componentes do sistema.

Na figura 3.1 encontra-se o diagrama que agrupa as várias componentes do sistema desenvolvido. Tal como se pode ver pelo diagrama, é a captura de dados que origina os *logs* GPS, sobre os quais todo o processamento é feito posteriormente. O processo de captura é descrito na secção 3.1. O primeiro processamento feito sobre estes ficheiros é a extração dos locais correspondentes ao utilizador. Estes são definidos no módulo Locations Modeling (apresentado na secção 3.2.1) que filtra os *stay points* a partir dos registos GPS e, posteriormente, define os locais. Tendo os locais bem definidos, o passo seguinte é definir de forma clara o historial do utilizador. O módulo Location History Modeling filtra o ruído do historial, de forma a obter uma sequência de estadias e respetivas trajetórias que representem o comportamento do utilizador a cada instante de tempo. Toda esta informação vai sendo guardada na estrutura de dados desenvolvida, apresentada na secção 3.2.3. Esta estrutura, além de interagir com as componentes já mencionadas, permite também a extração de várias variáveis. As variáveis extraídas e modeladas em Information Extracting and Modeling (explicadas na secção 3.2.4), resultam em diferentes ficheiros de saída. Os ficheiros de saída são produzidos para que os dados possam ser lidos pelos programas auxiliares que permitem a visualização, processamento e análise dos

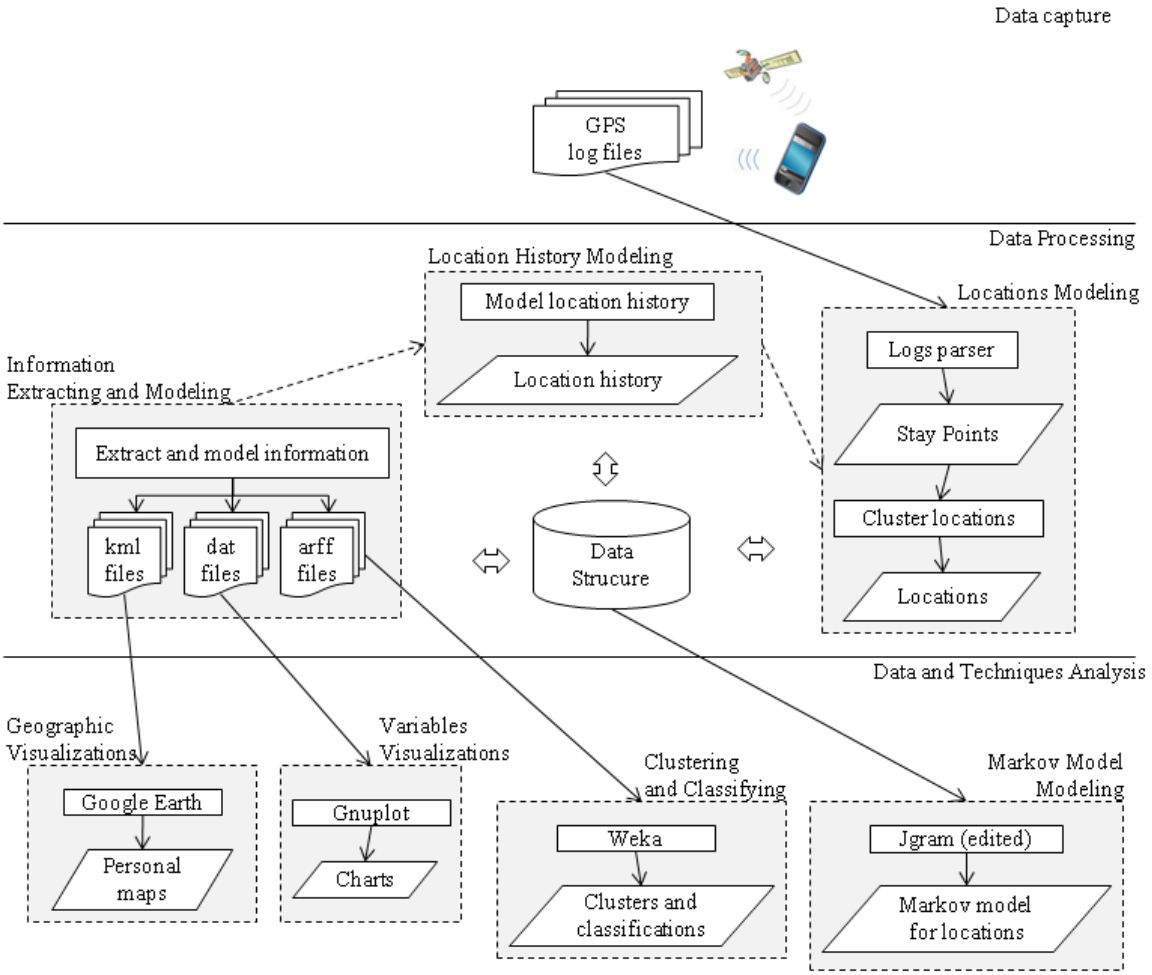


Figura 3.1: Componentes do sistema.

dados. A descrição de como são feitas as análises dos dados é apresentada na secção 3.3. É ainda construído um modelo de Markov que torna possível a previsão de movimentos futuros. A forma como este foi elaborado e funciona é descrita na secção 3.3.1. Por último, na secção 3.4, é feita uma síntese geral do sistema e das suas componentes.

De referenciar ainda que todas as unidades de processamento foram desenvolvidas utilizando a linguagem Java, num ambiente de computador *desktop*. Neste capítulo apenas se apresentam os vários módulos do sistema, deixando os resultados experimentais e as análises para serem discutidos no capítulo 4.

3.1 Captura de dados

Foi desenvolvida uma nova aplicação por outro elemento do projeto, Cristiano Lopes, para a captura de dados. Esta aplicação foi sendo desenvolvida, testada e melhorada em paralelo com o desenvolvimento da dissertação. Foi desenvolvida na plataforma Android (Android SDK),

utilizando a linguagem de programação Java. A aplicação inicia automaticamente e apenas necessita de ter acesso aos satélites GPS para começar a captar dados. Tal como na aplicação anterior descrita na secção 2.2.1, esta escreve os dados por ficheiros. Um ficheiro para cada dia e uma linha para cada evento. Cada evento é caraterizado por uma linha com os seguintes valores,

inteiro,mês/dia/ano,hora:minuto,latitude,longitude,inteiro horaGPS:minutosGPS precisão fornecedor num_{sat} snr_1 snr_2 ... snr_n *[Address].*

A primeira parte é idêntica aos dados que já eram registados. Os dados temporais, horas e minutos, são agora também registados através do sistema GPS (*horaGPS* e *minutosGPS*), a *precisão* do sinal é um valor dado em metros e o $fornecedor$ é por agora sempre o GPS. De seguida incluem-se o número de satélites que participaram no calculo da posição (num_sat) e os respetivos ruídos de cada um destes em percentagem (snr_1, snr_2, até $snr_{num_{sat}}$). É ainda apresentada a informação semântica obtida através de *reverse geocoding* quando o dispositivo tem acesso à Internet no momento da captura, sendo esta representada no campo opcional $address$.

Sobre o parâmetro *precisão* pouca documentação existe, sendo apenas dito que este define a precisão em metros da coordenada dada. Os valores deste são analisados na secção 4.2.2 de forma a perceber se poderá ser útil.

Para além dos novos campos, com novos dados, foram também introduzidas políticas de poupança de bateria. Se antes o registo da posição era periódico, de 30 em 30 segundos, agora o sistema tem um *listener*, com parâmetros configuráveis, que informa quando a posição altera. Este *listener*, apenas notifica a aplicação quando a posição física altera mais de 5 metros e passou mais de 1 minuto desde a última atualização. Foi ainda utilizada a informação do acelerómetro para ligar/desligar a captura de dados. Muito tempo de cada dia está-se parado em certos locais. Contudo, o GPS continua sempre a captar dados e consequentemente a gastar bateria. Com este sensor, consegue-se fazer com que a captura desligue quando o telemóvel esta parado. Esta aplicação está ainda em desenvolvimento, pelo que não podem ainda ser tiradas conclusões finais sobre o melhoramento em termos de bateria que estas alterações provocam. Contudo, é notado que a aplicação gasta muito menos bateria com a captura desligada.

3.2 Processamento de dados

Nesta secção é descrita a camada de Data Prcessing do sistema. Na secção 3.2.1 é descrito o filtro de locais, na secção 3.2.2 apresentada a estrutura utilizada para guardar o historial do utilizador e na secção 3.2.3 a estrutura de dados completa que dá suporte à informação extraída dos dados. Por fim, na secção 3.2.4, é descrito como a extração da informação é feita.

3.2.1 Filtro de locais

Para se definirem os locais do utilizador são utilizadas aproximações aos algoritmos estudados nas secções 2.4.1.1 e 2.4.2.2. É feito um pré-processamento dos dados para extrair os *stay points*. De seguida, com base no conjunto de pontos resultantes, é utilizada uma aproximação do algoritmo de *clustering* baseado em densidade, DJ-Cluster, para determinar os locais importantes e quais os pontos que pertencem a cada local. Nas secções 3.2.1.1 e 3.2.1.2, são apresentadas as alterações introduzidas nos algoritmos.

3.2.1.1 Filtro de pontos GPS

Para implementar o filtro dos registos GPS foi utilizado o algoritmo já estudado em 2.4.1.1, ligeiramente modificado para guardar também as trajetórias entre os *stay points*. Cada *stay point* tem um campo $S.Trajectory$, que contém os pontos que correspondem à trajetória entre os *stay points* de origem e destino. Para guardar este novo campo, o algoritmo 1 sofreu algumas alterações que são introduzidas no algoritmo 6.

Algoritmo 6 LogsParser

Input: A GPS log P, a distance threshold *distThresh*, a time span threshold *timeThresh*, and a minimum precision to close a *stay point precisionMin*.
Output: A set of *stay points* SP={S}.

1: $i = 0$,$pointNum = |p|$;//the number of GPS points in a GPS log
2: $startI = 0$,$endI$;//indexes corresponding to the start and end points of the trajectory
3: **while** $i < pointNum$ **do**
4: $j = i+1$;
5: $endI = i$
6: **while** $j < pointNum$ **do**
7: $dist = Distance(pi, pj)$; //calculate the distance between two points
8: **if** $dist > distThresh \&\& p_j.precision > precisionMin$ **then**
9: $\Delta T = p_j.T - p_i.T$; //calculate the time span between two points
10: **if** $\Delta T > timetThresh$ **then**
11: **if** $SP.isNotEmpty$; **then**
12: $SP.last.Trajectory = \{p_k | startI <= k <= startJ\}$;
13: **end if**
14: $S.coord = ComputMeanCoord(\{p_k | i <= k <= j\})$;
15: $S.arvT = p_i.T; S.levT = p_j.T$;
16: $SP.insert(S)$;
17: $startI = j$;
18: **end if**
19: $i = j$;**break**;
20: **end if**
21: $j = j+1$;
22: **end while**
23: **end while**
24: **return** SP;

Para além da trajetória foi também introduzida uma ligeira alteração na linha 8. Após algumas análises feitas com os novos registos GPS (ver a secção 4.2) chegou-se à conclusão de que a precisão dos pontos captados nem sempre está dentro dos limites de precisão esperados, principalmente quando se está num local com má receção. Os valores de *precisão* contidos nos eventos captados pelo GPS são sempre elevados e não revelam muito significado. Contudo, quando em locais com má receção este valor é consideravelmente pior. Devido a esta análise adiciona-se à condição para encerrar um *stay point*, outra condição que apenas o permite fazer com base num ponto que garanta um valor mínimo de precisão. Alterou-se então a condição do if na linha 6 do algoritmo 1, que agora corresponde à linha 8 no algoritmo 6 para,

$$((dist > distTresh)\&\&(p_j.precision > precisionMin)) \tag{3.1}$$

em que $p_j.precision$ é o novo campo de precisão, correspondente ao ponto j e $precisionMin$ é um parâmetro definido.

3.2.1.2 Definição de locais

Para o *clustering* de locais é utilizado o algoritmo DJ-Cluster, descrito e analisado em 2.4.2.2. Porém, este tem um problema, já enunciado anteriormente, que é o facto de selecionar locais importantes apenas com base na frequência, desprezando a duração. Por este motivo foi alterado o algoritmo para satisfazer as necessidades do projeto. O agrupamento de locais é feito da mesma forma com base na densidade, mas depois o algoritmo decide se a vizinhança deve, ou não, ser considerada um local importante, com base na duração e na frequência.
Recorde-se o algoritmo 2 que define o algoritmo DJ-Cluster. Os pontos são agrupados da mesma forma e a vizinhança calculada de igual modo. Foi apenas substituída a condição de vizinhança, que passa a considerar também a duração. Tem-se então que $N(p)$ continua a ser definido por (2.2), mas muda a condição de satisfação da condição (2.3), para a condição 3.2,

$$(N(p)| >= MinPts \vee N(p).duration >= MinDrt, \tag{3.2}$$

em que $MinDrt$ é um parâmetro dado que representa o mínimo de duração num local e $MinPts$ o mínimo de pontos correspondentes à frequência que o local deve ter. Um local é considerado importante por satisfazer um dos parâmetros. N(p) é uma vizinhança caso tenha um mínimo de frequência $MinPts$ ou um mínimo de duração $MinDrt$.
Com esta condição um local é considerado importante tanto pela sua duração como pela sua frequência.

3.2.2 Location History

Location History é a estrutura utilizada para representar o historial do utilizador. Este historial é representado através de uma sequência de *stay points*. Como cada ponto contém a respetiva

trajetória até ao próximo, o historial pode ser representado ao longo do tempo por uma sequência ordenada de pontos. Cada ponto tem um tempo de chegada, um tempo de partida e uma trajetória até ao próximo.

Definição.

Location History é representada por,

$$LocH = (S_1 \overset{S_1.traj}{\rightarrow}, S_2 \overset{S_2.traj}{\rightarrow}, ..., S_n), \tag{3.3}$$

em que S_i corresponde ao *stay point i* e $S_i.traj$ corresponde à trajetória entre os *stay points* S_i e S_{i+1}. Sendo o espaço no tempo que cada ponto representa definido por,

$$S_i.arvT < S_i.\Delta t \leq S_i.levT; \tag{3.4}$$

$$S_i.levT < S_i.traj.\Delta t \leq S_{i+1}.arvT, \tag{3.5}$$

com $S_i.\Delta t$ e $S_i.traj.\Delta t$ a representarem os períodos de tempo do *stay point i* e da trajetória posterior, respetivamente.

Filtro para sequência de stay points

A estrutura de dados desenvolvida contém a lista ordenada com a sequência de *stay points* que representa o historial do utilizador. Contudo, esta ainda não está na forma desejada. Contém algum ruído como, por exemplo, *stay points* consecutivos que representam a mesma estadia no mesmo local ou *stay points* que não pertencem a nenhum local. De relembrar que ao fazer o *clustering* de locais pode haver *stay points* que não pertençam a nenhum local no caso de não terem os requisitos mínimos.
Foi criado um algoritmo que filtra esta sequência e nos dá uma sequência com menos ruído, tal como desejado. Este algoritmo junta *stay points* seguidos na linha de tempo que representem a mesma estadia e ignora estadias que não pertençam a nenhum local, juntando este tempo à respetiva trajetória.

No algoritmo 7 é apresentado o algoritmo desenvolvido. Neste caso, *si.owner* representa o local ao qual o *stay point i* pertence. A função $mergeble(S_i, S_j)$ retorna um valor de verdadeiro ou falso, consoante os *stay points* devam, ou não, ser unidos. Esta função é definida pela condição,

$$(S_i.owner == S_j.owner)\ \&\&\ (S_i.traj.mergebleSP()), \tag{3.6}$$

onde $S_i.traj.mergebleSP()$ é a função que analisa se os *stay points* dos extremos da trajetória

Algoritmo 7 LocationHistoryParser

Input: A *stay points* sequence SP and a time threshold *timeThresh*.
Output: A set of parsed *stay points* SP=S, meaning the Location History.

```
1: i = 0,SPNum = |SP|;//the number of stay points in the sequence SP
2: while i < SPNum && j < SP.Num do
3:    if S_i.owner != null then
4:       newSP.insert(S_i);
5:       j = i + 1;
6:       while j < SPNum do
7:          i = j;
8:          if S_j.owner != null then
9:             if S_i.mergeble(S_j) then
10:               S_i.owner.mergeSP(S_i, S_j);
11:            else
12:               break;
13:            end if
14:         else
15:            newSP.last.traj.merge(S_j.traj); //if its owner is null, ignores the stay point and merges its trajectory.
16:         end if
17:         j = j + 1;
18:      end while
19:   else
20:      if newSP.isNotEmpty() then
21:         newSP.last.traj.merge(S_i.traj);
22:      end if
23:      i = i + 1;
24:   end if
25: end while
26: return newSP;
```

devem, ou não, ser unidos e a trajetória ignorada. Isto acontece quando os pontos da trajetória pertencem ao mesmo local que ambos os extremos. A função $S_i.owner.merge(S_j)$ faz a junção do ponto S_j ao ponto S_i, enquanto a função $S_i.traj.merge(S_j.traj)$ junta a trajetória S_j à trajetória S_i.

3.2.3 Estrutura de dados

Para as experiências feitas foi necessário desenvolver uma estrutura que suporte a informação extraída dos dados. Esta, foi construída de forma incremental à medida que o sistema ia, também ele, sendo desenvolvido e novos requisitos iam sendo apresentados com o decorrer das experiências feitas. Aqui é apresentado o estado final da estrutura.

O diagrama da figura 3.2 representa a estrutura de dados . As classes *StayPoint* e *Trajectory* representam uma estadia e uma trajetória, respetivamente. A classe *Event* corresponde a um

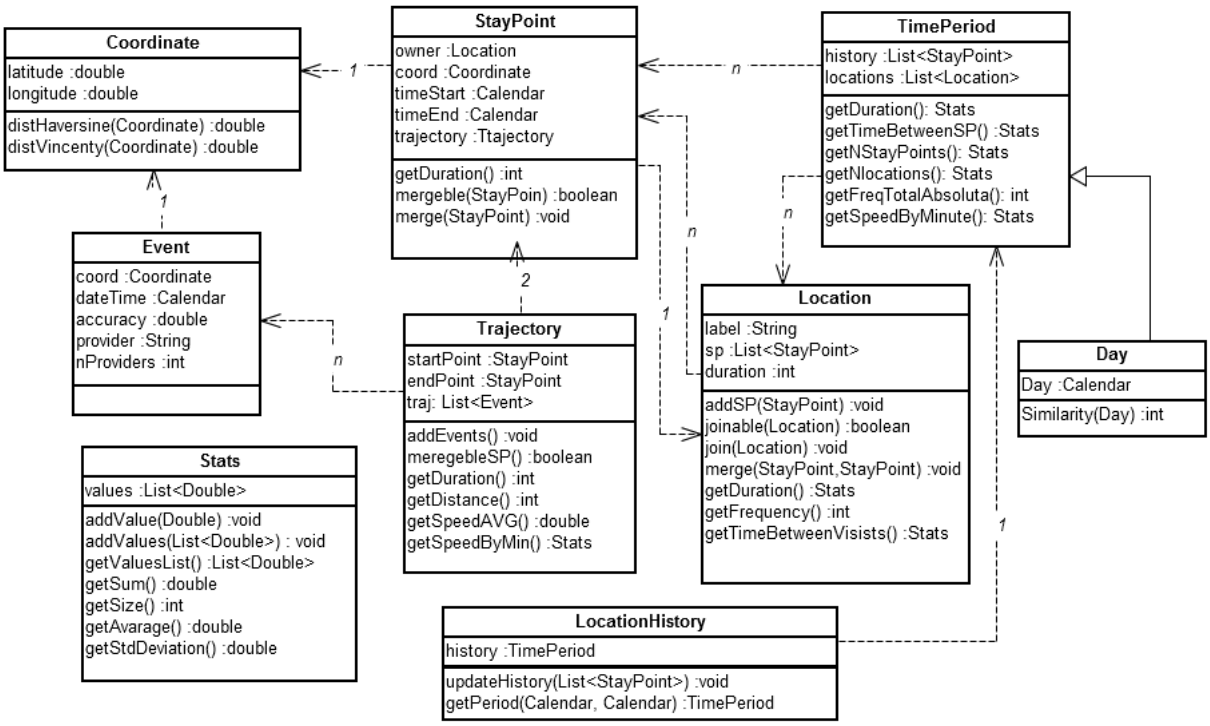


Figura 3.2: Diagrama de classes da estrutura de dados.

evento registado na fase da captura e *Coordinate* uma coordenada no sistema de coordenadas geográficas. Cada *Location* é definida pela sua *label*, que a identifica semanticamente e a sequência ordenada no tempo das respetivas estadias *sp*. Tem-se também a *LocationHistory* que é definida por um período de tempo *TimePeriod*, correspondendo no caso da *LocationHistory* ao historial completo do utilizador. Este *TimePeriod* foi definido para que se possa trabalhar com diferentes intervalos de tempo configuráveis, incluindo extrair informação de dias, semanas ou outros períodos do historial pretendidos. No caso dos dias, como é uma unidade de medida muito utilizada, foi criada uma classe, *Day*, que estende a *TimePeriod*. Tem-se ainda a classe *Stats* criada para trabalhar com os dados estatísticos, como a média e o desvio padrão. O diagrama representa ainda as diversas funções, correspondentes a cada classe, necessárias para interagir com as diferentes partes do sistema. Algumas funções correspondem a funções já apresentadas, que interagem com as componentes do sistema apresentadas, as outras, são necessárias para a extração de informação.

3.2.4 Extração e modelação de informação

Tendo a informação do historial do utilizador configurada é possível extrair os dados pretendidos. Este módulo trata de extrair e modelar as diferentes variáveis.

A extração e modelação da informação é feita com três objetivos principais. Primeiro, para que se possa visualizar em mapas geográficos os pontos sobre os quais se está a trabalhar. Segundo, para formar gráficos relativos a diferentes variáveis, com o objetivo de os estudar e analisar. Por

último, pegando nas diferentes variáveis, dispor a informação de forma a que possa ser aprendida por algoritmos de aprendizagem automática.
Alguma desta informação já está pronta na base de dados e apenas tem de ser apresentada em formatos que sejam lidos pelos programas utilizados para cada efeito. Outras, como dados estatísticos sobre diferentes variáveis, têm de ser primeiro calculados. Para esse efeito foram criadas funções nos diferentes tipos de dados que permitem extrair as variáveis. Foi ainda criada a classe *Stats* com o objetivo de acumular dados estatísticos sobre as variáveis, como a média e o desvio padrão.
As variáveis extraídas são descritas na secção 3.2.4.1 e posteriormente analisadas na secção 4.4 (capítulo de resultados experimentais).

Depois de extraída, a informação é modelada para ser apresentada nas diferentes formas que as aplicações com as quais se trabalha requerem. Os dados extraídos são modelados em três tipos de ficheiros diferentes que são apresentados a seguir. Ficheiros KML para visualização geográfica de pontos, secção 3.2.4.2, ficheiros DAT que contêm os dados que servem de base aos gráficos criados e ficheiros ARFF que são o formato de entrada da ferramenta Weka, a qual nos permite testar e analisar o *clustering* e a classificação.

3.2.4.1 Variáveis extraídas

As variáveis extraídas dizem respeito a locais ou a períodos de tempo, sendo que aqui são abordados períodos de tempo relativos a dias. No entanto, estas variáveis podem ser extraídas para qualquer período dado. As variáveis extraídas são descritas de seguida.

Variáveis extraídas para dias:

Distância. A diferença entre dois pontos corresponde a uma distância calculada através da fórmula de Haversine, apresentada na secção 2.1.2. Para cada trajetória é calculada a distância que esta percorre, somando as distâncias entre cada dois pontos consecutivos da sequência de pontos que a define. A distância percorrida num dia é dada pela soma das distâncias das várias trajetórias do dia.

Velocidade. A velocidade é dada pela fórmula $V = d/t$, onde d é a distância e t o tempo. A velocidade média e respetivo desvio para cada dia são calculados com base nas velocidades médias de cada trajetória.

Duração. Cada stay point contém o tempo que durou a estadia. Utilizando as durações para as diversas estadias ao longo do dia, é obtida a média e desvio padrão para as durações de cada dia.

Número de locais. Corresponde ao número de sítios visitados ao longo do dia. Aqui extraem-se duas variáveis distintas, *nPlaces* e *nLocations*. Na primeira, são contados todos os *stay points*, mesmo os que correspondem ao mesmo local, enquanto na segunda as repetições do mesmo local não são contadas.

Tempo em movimento. Como o próprio nome indica, corresponde à quantidade de tempo do dia que se passa em movimento.

Frequência absoluta total. Aqui, a frequência total do local visitado é a unidade utilizada. Cada *stay point* corresponde a um local e cada local tem uma frequência absoluta que é o número de vezes que o local foi visitado. Para esta medida são somadas as frequências absolutas de todos os locais visitados nesse dia.

Variáveis extraídas para locais:

Duração. Aqui, tal como para os dias, são extraídas as durações correspondentes a cada estadia no local. É calculado a média e respetivo desvio padrão destes valores.

Tempo entre visitas. Tempo entre visitas a cada local. São utilizados os vários períodos de ausência de visitas a cada local, para fazer as suas médias e respetivos desvios. Assim, é representada a periodicidade com que cada local é visitado.

3.2.4.2 Ficheiros KML

Keyhole Markup Language[i] (KML) é uma linguagem baseada em XML e serve para expressar anotações geográficas e visualizar mapas 2D ou 3D. KML é uma norma internacional mantida pela Open Geospatial Consortium (OGC[ii]). O ficheiro começa pelo cabeçalho necessário para abrir o ficheiro KML e o documento. Neste, é também definido o nome do documento a criar, *Clustered locations.kml* neste caso.

Listagem 3.1: Cabeçalho num ficheiro KML

```
<?xml version=''1.0'' encoding=''UTF-8''?>
<kml xmlns=''http://www.opengis.net/kml/2.2'' xmlns:gx=''http://www.google.com/kml/
    ext/2.2''xmlns:kml=''http://www.opengis.net/kml/2.2''xmlns:atom=''http://www.w3.
    org/2005/Atom''>
  <Document>
    <name>Clustered locations.kml</name>
```

De seguida, criam-se as pastas que se pretende com os respetivos marcadores para cada uma. Uma pasta, *Folder*, é definida pelo nome e marcadores. Cada marcador, *Placemark*, tem um nome, uma descrição e as suas coordenadas geográficas. O exemplo seguinte ilustra uma pasta, que corresponde a um local, com os respetivos *stay points* como marcadores.

Listagem 3.2: Pastas e marcadores num ficheiro KML

```
<Folder>
  <name>Miradouro</name>
  <Placemark>
```

[i] http://code.google.com/intl/pt-PT/apis/kml/
[ii] http://www.opengeospatial.org/

```
    <name>StayPoint 1</name>
    <description>Start Time:12/15/2010,15:9; End Time:12/15/2010,16:31</description>
    <Point><coordinates>-9.14567,38.712478</coordinates></Point>
  </Placemark>
  <Placemark>
    <name>StayPoint 2</name>
    <description>Start Time:12/24/2010,21:28; End Time:12/24/2010,21:34</description
        >
    <Point><coordinates>-9.145531,38.712569</coordinates></Point>
  </Placemark>
</Folder>
```

Após a inserção de todos os locais desejados no ficheiro, são inseridas as etiquetas necessárias para fechar o documento e o ficheiro KML.

Listagem 3.3: Fecho de etiquetas num ficheiro KML

```
  </Document>
</kml>
```

3.2.4.3 Ficheiros DAT

DAT é o tipo de ficheiros que utilizado para guardar os dados. Estes dados são posteriormente carregados pelo Gnuplot - o programa utilizado para desenhar gráficos. Num ficheiro de dados as colunas de dados são separadas por espaços em branco ou tabulações. Se uma linha começar por #, esta é ignorada por representar um comentário.

Listagem 3.4: Ficheiro DAT

```
#1Day 2DurationAVG  3DurationStdDev 4SpeedAVG 5SpeedStdDev  6Distance 7nLocations
2010-12-15  120.0 150.26  4.97  3.77  0.74  5
2010-12-16  282.4 488.97  37.28 26.01 12.1  5
2010-12-17  268.6 299.8 23.79 3.3 36.85 5
2010-12-18  197.14  314.24  1.36  1.51  4.18  7
2010-12-19  461.33  556.45  16.96 0.96  16.11 3
2010-12-20  466.33  639.72  21.43 5.15  13.6  3
2010-12-21  469.0 451.74  3.88  1.08  1.8 3
2010-12-22  170.62  319.13  14.39 13.3  30.68 8
2010-12-23  183.71  247.13  16.57 8.59  50.35 7
2010-12-24  111.17  205.39  4.07  5.17  19.48 12
```

No exemplo acima pode-se ver um ficheiro de dados com 7 colunas. Cada coluna representa uma variável diferente. Os valores podem ser numéricos ou datas e o seu valor é posteriormente definido durante a execução do programa. As variáveis desejadas na visualização são também selecionadas posteriormente.

3.2.4.4 Ficheiros ARFF

Attribute-Relation File Format[i] (ARFF) é um tipo de ficheiro que descreve uma lista de instâncias, as quais partilham um conjunto de atributos. Este é o tipo de ficheiros que a ferramenta Weka aceita como entrada de dados. Este tipo de ficheiro foi, também ele, desenvolvido pelos criadores da ferramenta no âmbito do projeto de desenvolvido na universidade de Waikato. Este tipo de ficheiros podem ser divididos em duas partes. A primeira contém o cabeçalho com o nome da relação e a definição dos atributos. A segunda contém os dados repartidos pelas várias instâncias.

Listagem 3.5: Cabeçalho de um ficheiro ARFF

```
@RELATION locations

@ATTRIBUTE locationLabel  STRING
@ATTRIBUTE longitude  NUMERIC
@ATTRIBUTE latitude NUMERIC
@ATTRIBUTE durationAVG  NUMERIC
@ATTRIBUTE frequency  NUMERIC
```

Tem-se o exemplo do cabeçalho, listagem 3.5, que contém dois tipos de declarações: *@RELATION* para o nome da relação e *@ATTRIBUTE* para os vários atributos. Os atributos podem ser numéricos, nominais, datas ou um conjunto de caracteres (*STRING*).

Listagem 3.6: Dados de um ficheiro ARFF

```
@DATA
Home,38.711752,-9.145617,631.85,28
FCT-DI,38.661118,-9.203184,117.0,9
FutebolField,38.755782,-9.168955,74,2
Lux,38.716678,-9.118047,180.6,2
Teater,38.709281,-9.142128,165.0,2
Diogoshouse,38.612871,-9.186265,790.0,1
ClubKubik,38.711733,-9.126608,175.01
gym,38.624479,-9.202327,114.0,16
Lust,38.707929,-9.153035,154.0,1
```

Os dados são precedidos da declaração *@DATA*. As várias instâncias correspondem às várias linhas e os atributos de cada são separados por vírgulas. Os atributos têm de estar ordenados e corresponder ao tipo declarado.

3.3 Análise de dados

As análises dos dados são feitas com ferramentas auxiliares que lêem os dados já modelados apropriadamente para cada uma.

[i] http://www.cs.waikato.ac.nz/ ml/weka/arff.html

Para visualizar dados geográficos é utilizado o Google Earth[i]. Esta ferramenta permite ver e analisar os eventos captados pelo GPS, os *stay points* e os locais assinalados em mapas reais. Sendo este tipo de visualizações é fundamental para se poder tirar conclusões sobre os dados.
Para o desenho de gráficos foi utilizado o Gnuplot[ii], um programa de linha de comandos que desenha gráficos em duas e três dimensões. Este dá liberdade para configurar várias opções de visualização, permitindo uma melhor visualização dos gráficos. Várias variáveis respeitantes a períodos de tempo foram assim analisadas, com o objetivo de entender que tipo de padrões podem revelar.
As experiências de *clustering* e classificação são, como já referido, feitas utilizando a ferramenta Weka, apresentada na secção 2.5.4. Esta, permite carregar os ficheiros com os dados e aplicar diferentes algoritmos estudados em 2.5. Criando modelos de aprendizagem que podem ser visualizado de várias formas. O modelo preditivo é criado com base nos modelos de Markov. Para este fim é utilizada a biblioteca Jgram já apresentada anteriormente. Esta foi modificada e foram-lhe adicionadas funcionalidades. As modificações feitas na biblioteca são explicadas na secção 3.3.1. Nesta secção apenas é explicado o procedimento para análise dos dados. Deixando pormenores sobre as experiências, resultados, análises e conclusões para o capítulo dos resultados experimentais (capítulo 4).

3.3.1 Cadeias de Markov

Tendo as centenas de milhares de coordenadas originais do GPS sido reduzidas a apenas alguns locais importantes, é agora mais simples criar um modelo preditivo como discutido anteriormente. É utilizada a biblioteca Jgram, já apresentada na secção 2.6.3. Esta biblioteca cria um modelo de Markov com base numa sequência. A biblioteca foi modificada para que em vez de criar caminhos aleatórios, permita obter o próximo local mais provável dados os m anteriores. Foi alterada ainda para em vez de gerar o modelo apenas para uma ordem, gerar antes, todos os modelos até à ordem m.

Foi então alterada a biblioteca Jgram para, que com base numa sequência, o *Parser* em vez de criar os *NGrams* só com N estados, crie os *NGrams* para as várias ordens até N. Seja N a ordem do modelo, para cada sequência dada, o parser cria *XGrams* compostos por X estados, para $X = 1,..,N$. Assim, o modelo fica formado para as várias ordens diferentes. De seguida, com base na *SimpleCalculator*, as pontuações são calculadas para as diferentes *Transitions* referentes a cada *NGram*.
Tendo o modelo completo, foi criada uma função $predictNext(Sequence < Location > seq)$ que permite prever o próximo local, com base no modelo preditivo criado, dada a sequência *seq* dos últimos m locais. Esta função vai analisar o *NGram* correspondente à sequência *seq* dada e caso este já tenha ocorrido mais que *minOcorr* vezes escolhe a *Transition* com maior pontuação do *NGram*. Caso contrário, elimina o local mais antigo da sequência e repete o procedimento para o *NGram* correspondente à nova sequência, até que um *NGram* satisfaça o número mínimo

[i] http://www.google.com/earth/index.html
[ii] http://www.gnuplot.info/

de ocorrências ou até que este seja composto pela sequência com apenas um local. Na hipótese do modelo não conter nenhum dos *NGrams*, não devolverá qualquer resultado.
Este método faz com que a previsão seja feita com base no modelo de maior ordem que preencha os mínimos de confiança, um parâmetro que representa o número mínimo de ocorrências necessárias.

Assim, dada a lista de locais visitados ordenados no tempo é criado um modelo em que cada nó *Gram* é um local, cada *NGram* as subsequências de locais até ordem *m*. Testes sobre estes modelos e valores aconselháveis para a ordem *m* do modelo e para *minOcorr* são discutidos no capítulo 4.

3.4 Síntese

O sistema criado é composto por várias componentes e pode ser dividido em três unidades principais: (1) a captura dos dados do utilizador, (2) a transformação desses dados em variáveis possíveis de analisar e (3) interpretação e a extração de informação e padrões úteis dos dados, através dessas variáveis.

Embora a componente da captura não tenha sido desenvolvida pelo autor, este teve um papel preponderante no processo de análise e desenvolvimento. Os vários erros detetados as melhorias inseridas tornaram possível a construção de uma aplicação de captura sólida e coerente. A inovação da utilização do acelerómetro para ligar/desligar a captura do sinal GPS permite combater uma das principais limitações do projeto, o consumo da bateria do dispositivo móvel.

A transformação dos dados num historial baseado na sequência de locais foi conseguida de forma bastante rigorosa através dos algoritmos utilizados. A redução dos dados iniciais do GPS a *stay points* e respetivas trajetórias, permite perceber realmente quando o utilizador está parado ou em movimento. Este processamento, reduz também as inúmeras coordenadas iniciais a um número limitado de pontos, que são os essenciais. Depois, com base neste número reduzido de pontos, o algoritmo de *clustering* é aplicado de forma a decidir quais os pontos que dizem respeito aos mesmos locais. Os locais ficam assim bem definidos, o que torna possível a extração de variáveis que ajudam na interpretação desses dados e tornam possível a busca de padrões nestes. Estes algoritmos para além de terem resultados muito bons, como será apresentado no próximo capítulo, garantem uma baixa complexidade computacional. É importante ainda notar que os algoritmos podem ser migrados para telemóvel e adaptados para uma utilização dinâmica. Ou seja, ao mesmo tempo que os dados estão a ser capturados, podem logo ser definidos os *stay points* e quando estes são fechados serem agregados ao local respetivo.
O algoritmo para filtrar a Location History, foi elaborado para eliminar o ruído da sequência de *stay points*. No entanto, este apenas é necessário pois aqui é realizado um processamento

estático, ou seja, os algoritmos são aplicados sobre dados de vários dias. Quando este processamento for feito imediatamente após a captura deixa de ser necessário, pois os locais já são conhecidos. Nesse caso, um *stay point* é apenas fechado quando as coordenadas saírem fora da área abrangida por esse local. O que agora não é possível pois quando é feito o processamento do filtro dos *stay points*, os locais ainda não são conhecidos.

Tendo o historial do utilizador já bem definido, foi desenvolvida uma estrutura capaz de extrair diferentes variáveis. Sendo a análise destas feita com base em ferramentas já existentes. É assim possível testar várias aproximações diferentes para extrair padrões dos dados e perceber quais as variáveis e ferramentas que conjugadas podem ser úteis. O módulo para previsão de movimentos é criado utilizando a biblioteca Jgram ligeiramente alterada. Desta forma a informação estatística e probabilística dos movimentos passados é utilizada para previsão de deslocações futuras. A estrutura de dados criada é também ela bastante versátil, permitindo a interação das diferentes componentes com os objetos existentes.

4

Resultados experimentais

Este capítulo destina-se a apresentar testes desenvolvidos ao longo da dissertação. São aqui revelados, analisados e discutidos os resultados experimentais obtidos. Na secção 4.1 são descritos os dados utilizados nas experiências e na secção 4.2 é feita uma análise geral dos ficheiros GPS. Depois, na secção 4.3 são analisados os locais criados e na secção 4.4 as diferentes variáveis extraídas. Na secção 4.5 é feita a análise aos testes de classificação e na secção 4.6 ao modelo preditivo. Por fim, na secção 4.7 é feita uma síntese dos resultados obtidos.

4.1 Dados utilizados

Os dados utilizados nos testes experimentais dizem respeito a dois indivíduos diferentes, representando aproximadamente dois meses da vida de cada um. A recolha foi feita entre os meses de Dezembro e Fevereiro, pelo que contêm as semanas do Natal e fim de ano, semanas estas que fogem um pouco ao habitual da rotina de cada um. Um HTC Desire e um LG P500 foram os dispositivos utilizados pelos indivíduos A e B, respetivamente. Ambos fizeram uso da aplicação Time Machine a executar sobre a plataforma Android 2.2.
O indivíduo A embora não tendo horários fixos, tem uma vida familiar relativamente rotineira. Filhos para levar à escola de manhã, um escritório onde costuma passar parte do dia a trabalhar e alguns locais onde vai regularmente. Por outro lado, o indivíduo B não tem horários nem compromissos fixos. Não tem um local de trabalho fixo, nem locais que tenha de visitar obrigatoriamente com alguma periodicidade.
Os nomes para os locais mais importantes de cada indivíduo foram dados pelos próprios. Estes foram confrontados com os seus locais representados num mapa e foi-lhes pedido que os identificassem com nomes descritivos como casa, trabalho ou universidade. Esta informação

permite uma melhor interpretação semântica dos locais que a informação da morada disponibilizada através de *reverse geocoding*.

De relembrar, que o objetivo é estudar o indivíduo como um uno, pelo que os dados dos indivíduos nunca são combinados. Serão utilizados os dados do indivíduo A ou do indivíduo B conforme seja mais adequado para ilustrar o pretendido. Alguns dos testes contêm apenas um subconjunto do total de dados, enquanto outros o conjunto completo.

4.2 Análise dos registos GPS

É gerado um ficheiro KML para cada dia, com todos os eventos GPS capturados, contendo na descrição de cada ponto o instante em da captura. Este ficheiro permite uma visualização de todos os eventos em mapas geográficos, revelando-se bastante útil para a análise a das posições GPS. Ao longo do extenso período de análise foram obtidas conclusões sobre a qualidade destes dados capturados pelo GPS. Em geral, foi notado que a receção é pouco rigorosa. Os problemas constatados são enunciados na secção 4.2.1. Em paralelo, foi também sendo desenvolvida a aplicação de captura e os motivos para a introdução de algumas melhorias são explicados em 4.2.2.

4.2.1 Problemas constatados

Um problema constatado é, muitas vezes, o tempo que o GPS demora a adquirir sinal. Quando se saí de um local sem receção e se inicia logo o movimento, acontece por vezes o receptor demorar vários minutos até obter sinal. Se o destino for outro local sem receção e durante esse a transição o GPS não obtiver sinal, provoca a falta de informação sobre a estadia no novo local.

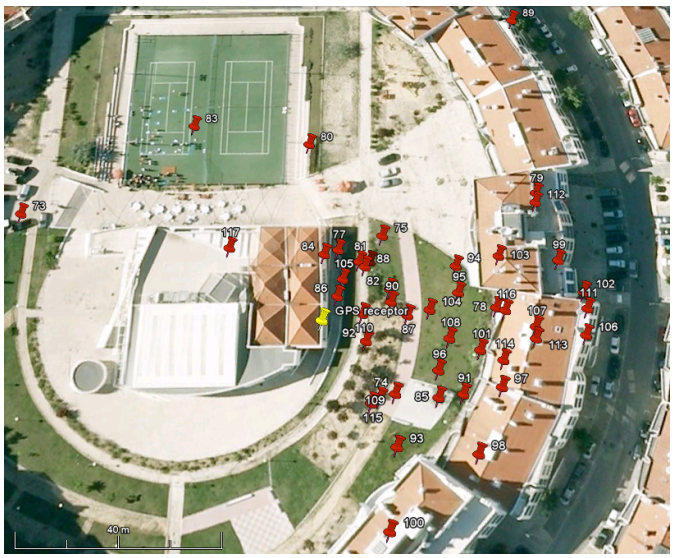


Figura 4.1: Ilustração de má receção do GPS. Pontos captados (marcadores vermelhos 74-116) com o receptor GPS estático (marcador amarelo).

Existem também muitos locais *indoor*, especialmente últimos andares de edifícios ou locais onde o receptor permaneça perto de uma janela que, não perdendo completamente sinal, têm sinal muito reduzido.
Este problema é ilustrado na figura 4.1. O receptor GPS representado pelo marcador amarelo, manteve-se estático dentro do edifício durante aproximadamente duas horas. No entanto, como estava junto à janela, recolheu durante esse período vários pontos GPS, os quais correspondem aos marcadores vermelhos, do 74 ao 116. Como se pode observar, os valores são muito dispersos e bastante distantes em relação à posição real. Alguns chegam a distar aproximadamente 60 metros desta posição.

Outro problema recorrente é a perca do sinal GPS. Acontece com alguma frequência, nomeadamente em viagens, que o receptor perca o sinal e apenas o volte a recuperar passado alguns minutos.
Em geral e como seria de esperar, o sinal é consideravelmente melhor em espaços abertos do que em espaços com muita densidade de edifícios, como centros de cidade.

Uma vez que o objetivo é estudar o indivíduo, deve-se ainda considerar outro tipo de falhas. Aqui são agrupadas todo o tipo de falhas que não resultam da receção do GPS, mas sim do indivíduo. Isto inclui falhas como o telemóvel ser esquecido ou a receção do GPS estar desligada. Esta última pode resultar tanto do facto do telemóvel estar desligado, como por falta de bateria ou simplesmente porque o utilizador desliga a receção GPS. Todas estas falhas originam lacunas nos dados.

Os problemas aqui enunciados da recolha de sinal criam limitações à partida para os estudos que se pretendem efetuar. A existência deste tipo de falhas é tida em conta e tenta-se lidar com elas nos algoritmos de processamento.

4.2.2 Desenvolvimento da aplicação de captura

A aplicação de captura foi desenvolvida em paralelo com a dissertação. As visualizações fora usadas como auxílio à análise das novas versões da aplicação, sendo que o desenvolvimento desta teve também como objetivo melhorar a captura dos dados. Passaram a ser registados os parâmetros disponibilizados pelo receptor sobre a exatidão das coordenadas captadas, sendo estes: a *precisão* do sinal e o *snr* que mede o ruído dos vários satélites que participaram na captura da posição.

Foi analisado o campo *precisão*, apesar de não especificado na documentação. Primeiro notou-se que este campo contém sempre valores potencialmente maus, a variar entre os 20 e os 200 metros. Estes valores não revelaram coerência quando comparadas as suas coordenadas capturadas com as coordenadas reais. Tanto existem pontos com um valor de *precisão* de 40 metros, que distam esses 40 metros do ponto real, como pontos com 200 metros nesse valor em que

as coordenadas se revelavam bastante próximas do valor real. Contudo, foi possível notar que os dados têm maior probabilidade de ser fracos quando o valor da *precisão* é, também ele, fraco. Assim, com base nestas observações, foi introduzido um filtro para *stay points* baseado neste valor (alteração já explicada na secção 3.2.1.1). Para o pârametro de *precisionMin* estabeleceu-se o valor de 90 metros, resultante das análises feitas. Já os valores de ruído do satélites *snr*, não nos levaram a nenhuma conclusão que pudesse ser utilizada no processamento.

Outra alteração introduzida, já explicada na secção 3.1, foi o ligar/desligar da captura GPS consoante as informações do acelerómetro. Para além dos ganhos em termos de bateria, permite também resolver alguns problemas de captura, nomeadamente quando o receptor está parado num sítio com má receção e recebe coordenadas muito fracas. Com esta nova alteração este problema deixa de acontecer pois a captura é desligada.

4.3 Análise da modelação de locais

Recorde-se que a modelação de locais é um procedimento com duas etapas. Primeiro são filtrados todos os pontos GPS em *stay points* e trajetórias. Depois, com esta filtragem já feita, é aplicada um técnica de *clustering* para definir os locais. Na secção 4.3.1 são discutidos os valores para os parâmetros destes algoritmos, enquanto na secção 4.3.2 ilustrados para análise os resultados de cada um.

4.3.1 Parâmetros dos algoritmos

O algoritmo para filtrar os pontos GPS, descrito na secção 3.2.1.1, contém dois parâmetros dos quais dependem os seus resultados: $timeThresh$ e $distThresh$. $timeThresh$ é o parâmetro que limita o mínimo de tempo que o indivíduo deve estar num certo sítio para que se considere uma estadia. No âmbito do projeto Time Machine é essencial que um local não seja desprezado se for visitado por curtos períodos de tempo. Por exemplo, o sítio onde se vai tomar café todos os dias e apenas se permanece 5 minutos ou, a escola onde se levam os filhos e se demoram apenas 3 minutos devem ser captados. Ambos os exemplos são locais com muita relevância no quotidiano que não podem ser ignorados. Como tal e com base em algumas experiências feitas, 3 minutos foi o valor definido para esta variável. Um valor assim baixo, origina a que muitos sítios que não são importantes sejam também filtrados como *stay points*, no entanto estes serão ignorados ou ser-lhes-á dada pouca relevância no decorrer do processamento. O parâmetro $distThreh$ representa a distância máxima que as coordenadas podem distar do ponto médio da estadia, para ainda pertencerem a esta. Aqui, é novamente necessária a precisão máxima, com o intuito de não unir locais diferentes. No entanto, é preciso ter em conta a imprecisão do sinal GPS. Como tal, após os testes realizados, foi definido o valor deste parâmetro em 30 metros, tal como também aconselhado em [HT04].

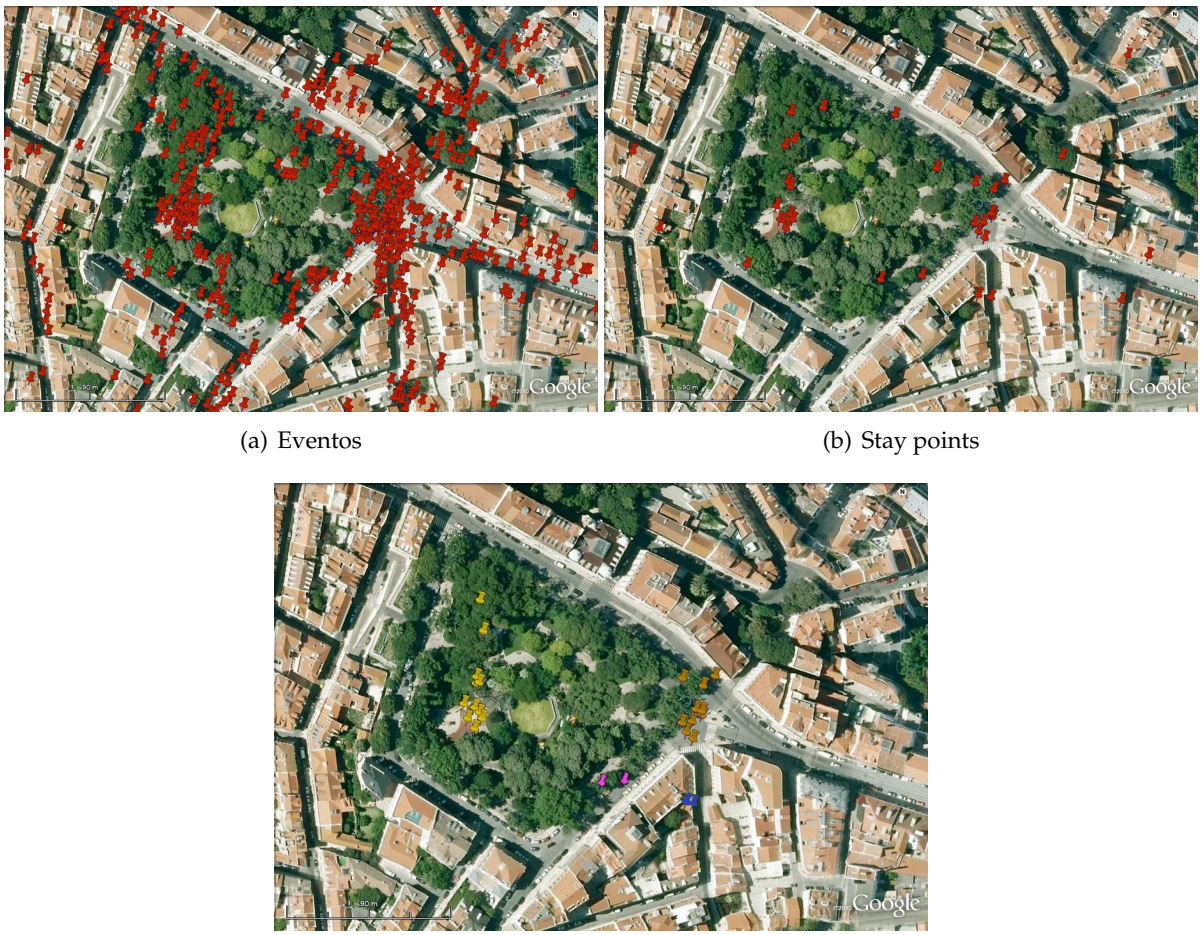
Como já explicado na secção 3.2.1.2, o algoritmo para definir os locais, dados os vários *stay*

points, depende de três parâmetros. Um deles Eps, com base no qual se unem pontos à mesma vizinhança e outros dois, $minPts$ e $minDrt$, com base nos quais se determina se uma vizinhança deve ou não ser considerada um local.
Para o parâmetro Eps é aconselhado, pelos criadores do algoritmo, um valor próximo da precisão do GPS [ZFL+04]. Segundo os testes realizados, o valor de 25 metros revelou-se como o mais adequado. Os parâmetros para aceitar locais são mais sensíveis e vão mais de encontro ao que é pretendido no projeto. Como tal, foi decidido no âmbito do projeto que para ser considerado um local, esse terá de ser visitado no mínimo duas vezes, ou em alternativa, ter uma permanência de 20 minutos. Estes valores correspondem aos parâmetros $minPts$ e $minDrt$, respetivamente.
Esta conjugação de parâmetros origina muitos locais, pois os limites mínimos são muito baixos. Sendo que alguns dos locais originados nem chegam a ser locais onde este permaneceu. Se, por exemplo, o sinal do GPS se perder por pelo menos 20 minutos enquanto o indivíduo estiver em movimento, origina um local na última coordenada adquirida pelo GPS. Estes valores foram assumidos devido à natureza do projeto. Uma variação destes, para outros objetivos, pode levar a que locais quase insignificantes deixem de ser detetados, ficando apenas os que tenham maior duração e frequência.

4.3.2 Ilustração da definição de locais

No caso do indivíduo A, foram capturados 8532 pontos GPS, que foram filtrados em 789 *stay points* e originaram 120 locais. Já para o indivíduo B, de 5274 pontos resultaram 852 *stay points*, que por sua vez deram origem a 58 locais.
No anexo A podem-se ver várias visualizações que ilustram os resultados da execução de cada um dos algoritmos. Primeiro tem-se as imagens relativas aos eventos, *stay points* e respetivos locais do conjunto de dados do indivíduo A, correspondentes às figuras A.1, A.2 e A.3 respetivamente. De seguida apresenta-se uma aproximação ao centro de atividade do indivíduo, para que melhor se percebam as divisões feitas (figuras A.4, A.5 e A.6).

É analisado aqui o conjunto de imagens da figura 4.2, que contém o *close-up* a um jardim visitado com frequência pelo indivíduo A. Esta imagem foi escolhida para ilustrar a formação de quatro locais diferentes. Na figura 4.3(a), podem-se ver todos os eventos registados pelo GPS, durante todo o período de captura, dentro dos intervalos geográficos definidos pela imagem. O resultado do algoritmo que filtra os dados GPS em *stay points* é correspondente à figura 4.3(b). Aqui, é bem visível a grande filtragem de pontos que existiu. A este conjunto de *stay points* é aplicado o algoritmo DJ-Cluster, com as modificações e parâmetros já discutidos. Os locais resultantes deste processamento são ilustrados em 4.3(c). Como se pode ver, resultaram quatro locais, cada um definido pelos seus *stay points*, com cores diferentes para locais diversos. A cor-de-laranja o quiosque do jardim, a azul o moleiro, a amarelo zona de lazer e por último, a cor-de-rosa, outra zona de lazer distante da anterior.



(a) Eventos

(b) Stay points

(c) Locais por cores

Figura 4.2: Indivíduo A: *close up* a jardim.

4.4 Análise de variáveis extraídas

As funcionalidades permitidas pela extração dos locais, das estadias e a formação de um historial de movimentos são ilustradas nas secções seguintes. São apresentadas várias estatísticas de variáveis extraídas destes dados, as quais resultam em informação relevante sobre as vidas dos indivíduos. Nas seguintes secções, 4.4.1 e 4.4.2, serão analisadas as variáveis extraídas de locais e as variáveis extraídas de períodos de tempo, respetivamente.

4.4.1 Variáveis relativas a locais

Dos dados extraídos a partir dos registos GPS resulta um conjunto de locais frequentados pelo utilizador durante o período de captação. Esta informação já é suficiente para responder em tempo real a perguntas como "É este um local novo?". Bastando, para isso, percorrer o conjunto de locais e ver se o novo local faz parte dos já existentes.
Na tabela 4.1 é apresentado o conjunto dos cinco locais frequentados por cada um dos indivíduos. Para cada entrada são apresentadas as informações estatísticas referentes às variáveis

extraídas dos locais.

Subject	Location	Nº Visits	Duration			Time between vistits	
			Total	Average	StdDev	Average	StdDev
A	home	113	68308	604	554	419	754
	work	54	16346	196	291	1172	1662
	kids school	33	1708	47	164	3162	3600
	bica	21	857	37	61	5191	4381
	near bunker	18	429	23	12	5561	3469
B	home, secundary	39	42044	1078	962	1479	2451
	home, primary	34	44736	1315	1249	1605	3603
	gym	22	2623	119	28	4554	2596
	university	13	1338	102	61	7946	3446
	football field	3	165	55	26	10798	9268

Tabela 4.1: Conjunto de cinco locais com maior frequência e respetivas variáveis extraídas, para cada indivíduo.

Esta ordenação permite mostrar quais os locais que o indivíduo mais frequenta ou, em alternativa, se forem ordenados pela duração total quais os locais onde o utilizador dispende mais tempo. Por exemplo, no top de locais do indivíduo B, é visível que o local onde este foi mais vezes não é o mesmo que aquele onde passou mais tempo. Este tipo de informações são úteis para o projeto Time Machine, dando suporte à visualização dos dados.

Outras informações, como "É este o meu padrão usual para este local?", são respondidas utilizando os padrões de uso do local. A média de tempo que o indivíduo costuma passar em cada local e respetivo desvio padrão permitem verificar se uma nova estadia num dado local está dentro do padrão. Outra unidade que define o padrão do local é a frequência e neste caso tem-se em conta a média e o desvio dos tempos entre cada visita a esse local. Assim, quando o utilizador chega a um local é possível identificar se esta visita está ou não dentro do padrão de visitas ao local. Para o mesmo fim, podem-se ainda utilizar as sequências de locais que são analisadas na secção 4.6. Quando um local é visitado pode-se verificar se a visita está dentro da sequência de locais habituais.

A seleção dos locais relevantes é feita pela componente Locations Modeling, explicada na secção 3.2.1. Nesta apenas são considerados locais aqueles que satisfizerem os valores mínimos de duração ou frequência. Neste conjunto de locais frequentados pelo utilizador, existem aqueles que são mais e menos relevantes. Contudo a definição de relevância de um local não é simples. Existem locais relevantes pelo tempo que lá se passa, outros pelo número de vezes que são visitados e outros que são importantes por outros aspetos como ter ocorrido um evento importante nesse local. Aqui, é possível responder à pergunta "É este um sítio relevante?" com base nas unidades de frequência e duração e pode-se ainda definir os mais ou menos importantes com base nestas. Informações semânticas adicionais ou informações de eventos durante

as respetivas estadias, podem ser adquiridas através de *reverse geocoding* com o objetivo de perceber a relevância dos locais (como é feito em [CMR07]).

4.4.2 Variáveis relativas a períodos de tempo

Foram criados diferentes gráficos, com diferentes tipos de variáveis, para analisar o significado e a relevância das variáveis extraídas. Uma listagem com todos os gráficos pode ser encontrada no anexo B. Estes gráficos representam variáveis ao longo dos vários dias correspondentes à amostra de dados. Todos os gráficos têm no seu eixo das abcissas a linha do tempo, onde o intervalo entre cada marcação corresponde exatamente a uma semana. Sendo estas marcações são sempre feitas aos Sábados. O eixo das ordenadas tem diferentes unidades conforme o que a variável representada meça. É feito de seguida a análise dos gráficos. Estes estão ordenados pelas diferentes variáveis, aparecendo para cada uma os gráficos dos indivíduos A e B seguidos. A descrição das variáveis foi previamente definida na secção 3.2.4.1.

O primeiro par de gráficos apresenta o número de locais visitados em cada dia. É de notar bastante mais atividade e movimento no indivíduo A, que no indivíduo B. No indivíduo A, figura B.1, é possível notar que os valores são altos antes do Natal, havendo depois uma quebra considerável entre o Natal e o fim de ano, seguido de uma retoma à normalidade com valores ligeiramente mais baixos que antes do Natal. O gráfico demonstra a atividade na vida do indivíduo que, claramente, teve um pico de atividade nas duas semanas antes do Natal, uma grande quebra entre o Natal e o Ano Novo e depois um retomar da atividade normal. No indivíduo B, figura B.2, a situação é diferente. Nota-se mais atividade desde poucos dias antes do Natal até à altura da passagem de ano, havendo uma grande quebra no dia de Natal. No novo ano é de notar uma certa rotina nas primeiras semanas, existindo repetidos picos e quebras de atividade com alguma regularidade.
Nota-se ainda que o indivíduo B raramente repete locais no mesmo dia, tendo os valores das duas variáveis sempre próximos. Já o indivíduo A, repete locais várias vezes ao dia.

Os gráficos, B.5 e B.6, são ilustrativos do movimento dos utilizadores A e B ao longo do tempo. Nestes gráficos estão representados os impulsos de velocidade no historial do utilizador. O eixo x está dividido em 48 blocos de 30 minutos correspondentes aos vários dias do eixo y. Os impulsos são a média das velocidades durante o período de 30 minutos e estão em km/h de acordo com o eixo z.
No indivíduo A é bem visível o período de atividade ao longo do dia, este é regular ao longo dos vários dias, começa normalmente por volta das 8 da manhã e termina às 20 horas. É muito raro qualquer impulso de movimento fora deste horário. Os impulsos são normalmente de baixa velocidade e esporadicamente de maior velocidade, o que sucede principalmente aos fins de semana. No gráfico do indivíduo B os impulsos já não estão tão concentrados como os do indivíduo A, nem se encontram com com tanta densidade. No entanto, é também possível

notar um período de maior atividade que corresponde ao blocos entre as 12 e as 24 horas. Muitos dos impulsos do indivíduo têm valores altos que correspondem ao uso de algum meio de transporte no seu quotidiano. É de notar a boa ilustração que estes gráficos disponibilizam das diferenças entre os comportamentos dos dois indivíduos.

Os gráficos seguintes têm como variáveis a velocidade média com que o indivíduo se movimentou durante o dia e a distância que percorreu. Nota-se que o indivíduo A, figura B.3, durante os dias de semana percorre normalmente distâncias curtas e com uma velocidade baixa, tendo por vezes dias com picos de velocidade que correspondem à utilização de algum transporte. É de notar que os maiores picos de velocidade são todos ao fim de semana. Mais uma vez, o gráfico do indivíduo B revela o oposto. Quando este percorre distâncias, fá-lo quase sempre por algum meio de transporte, conclusão retirada devido à velocidade. Nota-se uma certa periodicidade na sua atividade, com constante variação entre picos e quebras relativamente periódicas. Também aqui se nota uma maior atividade junto do Natal e Ano Novo.
É de notar que acontece por vezes, principalmente nos gráficos do indivíduo B (figura B.4) que o número de quilómetros percorridos seja bem maior que a velocidade média. Este facto deve-se muitas vezes a um erro do GPS já falado na secção 4.2.1. Se se sair de um sítio de carro e só passado 10 min se apanhar sinal de GPS, é percorrida uma distância enquanto supostamente se estive parado no local anterior, logo não existe velocidade. Este problema leva a concluir que a medida da velocidade nem sempre é de confiar, sendo que, pelo contrário a distância não reflete este tipo de erros.

A média das velocidades ao longo de um dia pode ser uma informação enganadora. Esta é mais completa se for acompanhada do respetivo desvio. É isso que os gráficos seguintes ilustram, figuras B.7 e B.8. O ponto no meio de cada barra é a velocidade média e a respetiva barra o seu desvio nesse dia. Aqui, mais que nos restantes gráficos, é de notar uma grande regularidade nos dias do indivíduo A. A velocidade a que este se desloca é regularmente baixa e com pequenos desvios, o que nos leva a entender que normalmente se desloca a pé. Contudo, tem por vezes uma média um pouco mais elevada com um desvio maior para dias em que utilize por pouco tempo algum transporte. Outros ainda, onde tem uma média bastante mais elevada com maior desvio, que corresponde a quando se desloca principalmente por algum meio de transporte. De notar que isto acontece principalmente aos fins de semana. Pelo contrário, o indivíduo B volta a revelar-se completamente irregular. Mais uma vez é de notar uma grande discrepância entre os dados dos diferentes indivíduos.
Os gráficos seguintes são do mesmo tipo, mas desta vez com a duração como variável. Estes voltam a revelar regularidade nos dados do indivíduo A, figura B.9, tendo normalmente um valor médio baixo com um desvio pouco acentuado. Notam-se algumas fugas à normalidade quando o valor médio é um pouco mais elevado ou o desvio mais acentuado. No indivíduo B, figura B.10, é de notar que existem dias com valores semelhantes, porém não estão distribuídos de forma regular. Volta-se a notar a diferença de atividade entre ambos os indivíduos.

Tem-se de seguida os gráficos com os minutos em andamento para cada dia, figuras B.11 e B.12, e a frequência absoluta dos locais visitados em cada dia, figuras B.13 e B.14. Por sua vez, os gráficos correspondentes aos minutos em andamento não revelam regularidade. É apenas possível notar que para o indivíduo A varia muito de dia para dia entre valores próximos, enquanto para o indivíduo B tem períodos de mais e menos movimento. Os gráficos da frequência absoluta são os que melhor revelam a periodicidade no comportamento dos indivíduos. No indivíduo A, são perfeitamente claros os valores próximos durante a semana e quebras aos fins de semana, tal como também a quebra na semana festiva. Tal facto, deve-se principalmente a este não ter visitado o local de trabalho durante esses dias, local com elevada frequência absoluta.

O gráfico B.15 apresenta as horas que o indivíduo A passou no local de trabalho ao longo do período de captura. Neste, é visível que o indivíduo costuma ir com regularidade ao local de trabalho, por norma frequenta este local apenas durante os cinco dias úteis da semana, com raras exceções. Na última semana do Ano apenas foi um dia ao trabalho e por vezes em certos fins de semana frequenta o trabalho. Porém, as horas que o indivíduo fica neste local variam muito, permanecendo neste entre duas a oito horas na maioria dos dias. Apenas a primeira semana do ano foi mais regular, onde o utilizador permaneceu sempre entre 4 e 6 horas por dia no local de trabalho. Na última semana nota-se que o utilizador teve um aumento considerável do tempo passado no trabalho.
O gráfico B.16 contém a média de horas que o indivíduo costuma frequentar o local de trabalho, para cada dia da semana. Para além das médias, são também representados os desvios. Pode-se constatar através deste padrão de uso do local de trabalho pelo indivíduo A que, entre segunda-feira e quinta-feira são os dias em que normalmente trabalha mais. O pico de produtividade registou-se às quarta-feiras e as sextas são dias de poucas horas no trabalho.

Discussão.

A análise feita das variáveis permite tirar algumas conclusões em relação aos seus significados e aplicações. Em todas as variáveis foram obtidos gráficos consideravelmente diferentes para os dois indivíduos. Em geral nota-se que o indivíduo A tem uma vida muito mais regular e ativa que o indivíduo B. O indivíduo A tem um comportamento com mais locais habituais e movimento entre estes. Contudo, os dias não diferem muito uns dos outros, notando-se variações principalmente ao fim de semana ou na semana de festas correspondente ao Natal e Ano Novo. Esta variação nota-se mais na variável da velocidade que normalmente é baixa e por vezes tem picos. O indivíduo B tem uma vida pouco ativa e desregulada, por vezes é possível encontrar uma certa periodicidade mas apenas por curtos períodos.

Estas variáveis podem ser utilizadas para dar respostas diretas, relativas a períodos de

tempo, a algumas das perguntas relevantes para o projeto Time Machine. A resposta à pergunta "Viajei muito hoje?" pode ser obtida comparando a distância percorrida com o comportamento habitual do indivíduo para esta variável. Para responder à pergunta "Foi este um dia cansativo?" tem-se em conta o número de sítios e o número de minutos que o indivíduo esteve em movimento. Cada indivíduo tem padrões diferentes, pelo que é necessário ter alguma informação sobre o historial deste para se poderem definir médias e desvios que possam ser comparados com os valores do dia corrente. Como foi analisado, variáveis como o número de locais, a velocidade e a duração definem a atividade do indivíduo. Estas, quando comparadas com os seus valores padrão, podem responder a perguntas como: "Foi este um dia calmo?" ou "Foi este um dia ativo?". Os dias podem ainda parecer mais longos ou mais curtos. Para responder à pergunta "Foi este um dia longo?" pode-se utilizar o tempo que o utilizador passou no trabalho e as alturas do dia em que começou e terminou a sua atividade.

4.5 Análise da classificação e do *clustering*

Nesta secção são apresentados os testes relativos à classificação e ao *clustering* sobre dias e locais. Na secção 4.5.1 são ilustradas as diferentes representações de dias e resultados da sua classificação. Na secção 4.5.2 são apresentados os resultados dos agrupamentos de locais realizados. Por fim, na secção 4.5.3, os resultados obtidos são discutidos.

4.5.1 Classificação de rotinas diárias

O objetivo é representar cada dia utilizando a informação do indivíduo de forma a discriminar padrões diários de uso do espaço e do tempo, por parte do indivíduo. Primeiro, utilizam-se diferentes combinações das variáveis extraídas de cada dia (explicadas na secção 3.2.4.1) e depois definem-se os dias pelas durações e frequências dos locais mais relevantes nesse dia. Neste último caso, são tidos em conta os cinco locais com maior duração ou com mais visitas para o dia em questão. Desta forma, apresenta-se de seguida algumas das diferentes formas testadas para descrever os dias. Foram escolhidas as combinações de variáveis que se entendeu terem revelado resultados mais interessantes.

Representações de dias:

D_a = <Número de locais, Duração média, Duração desvio padrão, Velocidade média, Velocidade desvio padrão, Distância, Tempo em movimento, Frequência absoluta>.

D_b = <Distância, Tempo em movimento, Frequência absoluta>.

D_c = <Frequência absoluta>.

D_d = <D_1,D_2,D_3,D_4,D_5>, onde D_i é a duração no i^o local onde o indivíduo esteve mais tempo nesse dia.

D_e = <F_1,F_2,F_3,F_4,F_5>, onde F_i é o número de visitas ao i^o local onde foi mais vezes nesse dia.

D_f = <D_1,F_1,D_2,F_2,D_3,F_3,D_4,F_4,D_5,F_5>, onde D_i e F_i são respetivamente a duração e número de visitas para o i^o local onde o indivíduo esteve mais tempo nesse dia.

Em geral as pessoas têm diferentes tipos de dias quando trabalham ou não trabalham. Para a grande maioria das pessoas estes tipos de dias estão divididos entre os dias de semana e os fins de semana. Desta forma, com base nas representações acima descritas tentou-se classificar os dias como dias de semana ou fins de semana. Os dados fornecidos à ferramenta Weka para aprendizagem e testes do algoritmo de classificação foram descritos pela entrada:

$<D_j$,*Weekend*>,

onde D_j é o conjunto de atributos que correspondem à representação de dia utilizada em cada teste, com $j \in \{a,b,c,d,e,f\}$, e *Weekend* a etiqueta correspondente a uma variável binária que toma o valor de 1 ou 0, consoante o dia seja fim de semana ou dia de semana, respetivamente.

Day Accuracy (%)		
Representation	Subject A	Subject B
D_a	78.3	64.8
D_b	83.4	57.3
D_c	86	59.2
D_d	70.5	64,8
D_e	73.9	63,4
D_f	72.2	57,8

Tabela 4.2: Resultados obtidos nos testes de classificação de dias como dias de semana ou fins de semana.

Na tabela 4.2 são apresentados os resultados obtidos com as diferentes representações de dias, para cada um dos indivíduos. Os testes foram todos feitos utilizando o algoritmo de classificação Naive Bayes estudado na secção 2.5.1.2, com os parâmetros sempre normalizados e utilizando a técnica de validação *Cross-validation*[i] com dez repetições.
Como seria de esperar os resultados do indivíduo A para a classificação de dias foram muito melhores que os do indivíduo B. Já tinha sido comprovado na secção 4.4.2 que o indivíduo A tem uma vida com mais rotina, trabalhando geralmente durante os dias de semana e com os seu dias de folga aos fins de semana. O mesmo não se passa com o indivíduo B, no qual dificilmente se distingue um dia de semana de um fim de semana.
Em geral, para o indivíduo A, todas as representações alcançam resultados satisfatórios, com mais de 70% de dias classificados corretamente. Os melhores resultados foram obtidos apenas com o uso de uma variável, a frequência absoluta, com 86% de instâncias classificadas corretamente. Como já tinha sido analisado, esta variável denota a falta de comparência do

[i]Cross-validation é uma técnica para analisar como se espera que os resultados de uma análise estatística generalizem para um conjunto de dados independente. Uma ronda de Cross-validation envolve a partição da amostra de dados em subconjuntos complementares, onde é feita a análise num subconjunto (o conjunto de treino) e a validação no outro subconjunto (o conjunto de teste). Para reduzir a variabilidade, este método é repetido várias vezes para diferentes partições.

indivíduo no local de trabalho ao longo dos fins de semana, o que explica o sucesso nos resultados obtidos. A distância percorrida e o tempo em movimento permitem também chegar a bons resultados. Não tão eficazes, as durações e frequências nos locais mais relevantes a cada dia, aparentam maior semelhança entre os dias.
É de evidenciar que o objetivo principal nestas experiências não foi tentar classificar o máximo de dias corretamente, mas sim arranjar formas de os descrever com base no padrão de utilização do espaço e do tempo ao longo desse dia. Seria possível, por exemplo, obter melhores resultados apenas comprovando se o indivíduo tinha, ou não, se deslocado ao seu local de trabalho. Contudo, o objetivo do projeto é encontrar dias diferentes com base nos padrões de uso do espaço e tempo.

4.5.2 *Clustering* de locais

Pretende-se criar grupos de locais e distribuir cada local por cada um destes grupos com base em dois critérios distintos: geograficamente e por padrões de uso. Ou seja, deseja-se dividir os locais (1) com base na sua proximidade geográfica e (2) com base nos padrões de uso que estes têm. Em todos os testes apresentados nesta secção de *clustering* foi sempre utilizado o algoritmo X-means, estudado em 2.5.2.2, utilizando a distância euclideana e com os atributos sempre normalizados.

4.5.2.1 Geograficamente

Para que os locais do utilizador sejam agrupados por proximidade foram representados pelas suas coordenadas geográficas. Ou seja, a assinatura de cada local é dada pelo seu par <Latitude, Longitude>. Os testes feitos com os indivíduos A e B mostram que são agrupados os conjuntos de locais mais próximos. Os grupos formados para o indivíduo A estão ilustrados na figura A.7, no anexo A. Contudo, ambos os indivíduos testados têm todos os seu locais muito próximos, o que não revela as potencialidades deste tipo de agrupamento. Devido a esse facto foram criados conjuntos de dados artificiais com locais em diferentes regiões do mesmo país e em diferentes países. Como se pode ver na figura A.8, o algoritmo é capaz de dispor por diferentes grupos os locais de países e regiões diferentes.
A distância utilizada para estes testes foi a distância euclideana. Contudo, em caso de implementação deste *clustering* é aconselhada a distância de Haversine, apresentada na secção 2.1.2, sem normalização dos atributos. Nestes testes não foi utilizada esta distância devido à impossibilidade, imposta pela ferramenta Weka, de utilizar uma distância diferente das distâncias por ela pré-definidas.

4.5.2.2 Por padrões

Para agrupar os locais segundo os seus padrões de uso apenas se pode considerar locais sobre os quais exista um limite minimo de informação. Para o efeito foram considerados apenas locais com mais de três visitas. Após esta filtragem, ficam 40 locais para o indivíduo A e 6 para o indivíduo B que preenchem os requisitos para serem agrupados. Foram testadas diferentes

combinações das variáveis dos locais, apresentadas na secção 3.2.4.1. As três representações testadas são apresentadas de seguida.

Representações de locais:

L_a = <Duração média, Duração desvio padrão>.
L_b = <Tempo entre visitas média, Tempo entre visitas desvio padrão>.
L_c = <Duração média, Duração desvio padrão, Tempo entre visitas média, Tempo entre visitas desvio padrão>.

Em todos os testes realizados os resultados foram sempre a divisão em dois grupos. Para o indivíduo A, a representação L_a cria um grupo com apenas três locais: a casa, o trabalho e a casa de um amigo. A representação L_b cria um grupo com todos os sítios mais frequentados, nomeadamente durante os dias de trabalho, como o sítio onde se vai tomar café, a escola dos filhos, o minimercado, isto para além da casa e do trabalho. A representação L_c obteve exatamente os mesmo grupos que L_b. O segundo grupo criado em cada um dos testes é o grupo complementar com todos os locais restantes que não ficaram no grupo evidenciado. Para o indivíduo B, a representação L_a criou um grupo apenas com as casas primária e secundária, enquanto a representação L_b juntou a estes o ginásio. Para este indivíduo a representação L_c obteve exatamente os mesmos resultados que a representação L_a.
A análise feita aos resultados comprova que estes variam consoante as diferentes representações de locais. Contudo, locais como a casa e o trabalho aparecem sempre no grupo mais evidenciado. A representação L_a evidência um grupo com os locais onde a permanência é normalmente mais prolongada, enquanto a representação L_b evidência os locais visitados com mais frequência. A representação L_c parece evidenciar, dentro das duas anteriores, aqueles que tiverem maior peso. Como o indivíduo B passa mais tempo em casa, L_c evidenciou apenas as suas duas casas. Pelo contrário, o indivíduo A tem uma vida mais repartida por vários locais com repetidas visitas, pelo que L_c evidenciou os vários locais mais frequentados.

4.5.3 Discussão

Embora se tenha conseguido chegar a resultados e conclusões interessantes, estes métodos necessitam de mais dados para se conseguir fazer melhores experiências e chegar a mais conclusões. Estudos mais profundos e sobre uma amostra maior de indivíduos são necessários para chegar a conclusões que levem a uma utilização mais consistente no projeto.

4.6 Análise do modelo preditivo

Um modelo de Markov foi criado para cada instância. Para o indivíduo A, o modelo originou x nós correspondentes aos x locais e y nós para o indivíduo B correspondentes aos seus y locais importantes. Foram imprimidos em forma de tabela as transições criadas por cada um dos

modelos de forma a serem analisados.

A figura 4.3 mostra um modelo parcial de ordem 1 relativo ao indivíduo A. Este contém apenas três locais com respetivas transições entre eles. São considerados apenas a casa, o trabalho e o Chiado, ignorando-se todos os outros nós para que as transições entre estes se tornem mais claras. Os nós representam os locais, enquanto as setas transições entre estes. Cada seta contém uma etiqueta que corresponde à probabilidade relativa de transição entre os nós. Por exemplo, foram feitas 113 viagens de casa para outros locais, das quais 13 foram para o local de trabalho. Das três viagens feitas do Chiado para outros sítios, uma foi para casa e outra para o trabalho.

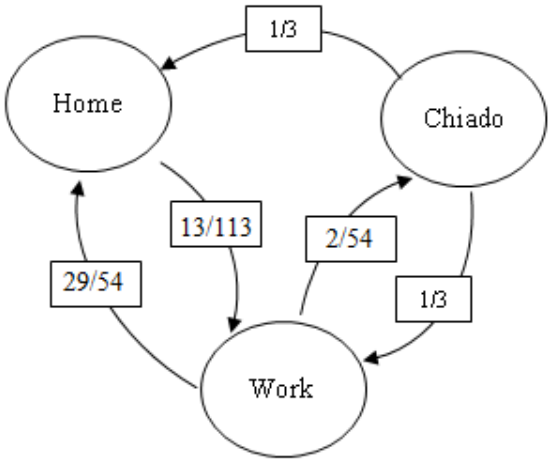


Figura 4.3: Grafo correspondente a modelo de Markov parcial de primeira ordem para o indivíduo A.

É de notar que o número de viagens feitas do Chiado é muito inferior ao dos outros dois locais. Este tanto pode ser um local novo, com um local que apenas é visitado esporadicamente pelo utilizador. Uma vez que o nó tem poucas ocorrências, tem uma componente de confiança menor que os nós com mais ocorrências. Para além disso, este contém uma probabilidade equivalente para todas as transições (1/3), o que faz com que a sua utilização para previsão seja mais difícil.

Na tabela 4.3, é apresentada uma seleção de algumas das subsequências e transições com mais ocorrências para indivíduo B. O modelo foi construído com $m = 3$, sendo nesta tabela já apresentadas as diferentes ordens. Embora o modelo completo contenha muitos mais caminhos, foi feita uma seleção de transições a apresentar dada a vastidão dos dados. Cada entrada da tabela contém uma transição de uma subsequência de locais $subSeq = L_1 : L_2 : ... : L_n$ de ordem n (que correspondem aos *NGrams* formados pela biblioteca Jgram) para um local D, representado por $subSeq -> D$. A frequência relativa é dada pelo número de ocorrências da transição, $ocurr$, sobre o número de ocorrências da sequência, n. Esta fórmula, $ocurr/n$, permite também

obter a probabilidade de ocorrer a transição, dada a prévia ocorrência da subsequência.

Order	Transition	Relative frequency	Probability
1	A->B	3/28	0.107
1	A->C	5/28	0.179
2	A:C->B	1/5	0.200
2	A:C->D	4/5	0.800
3	A:C:D->B	4/4	1.000
1	B->C	7/29	0.241
1	B->D	10/29	0.345
1	B->A	5/29	0.172
2	B:D->B	7/10	0.700
2	B:D->A	2/10	0.200
1	C->B	2/12	0.167
1	C->D	7/12	0.583
2	C:D->B	5/7	0.714
1	D->B	13/19	0.684
1	D->A	4/19	0.211
2	D:B->C	5/13	0.385
2	D:B->D	3/13	0.231
2	D:B->A	3/13	0.231

Tabela 4.3: Probabilidades de transições entre locais em modelos Markov de diferentes ordens para o indivíduo B. Legenda: A="Casa primária"; B="Casa secundária"; C="FCT-DI"; D="Ginásio".

O uso de modelos de ordem superior pode aumentar consideravelmente o poder de predição. Por exemplo, na tabela 4.3 a probabilidade do utilizador viajar de D para B é de 68%. Contudo, se se souber que antes esteve em C, a probabilidade de viajar para B vindo de C:D é de 71%. Neste caso não se tem um aumento muito significativo, mas com a informação de veio antes de A, ou seja, da sequência A:C:D a probabilidade de ir para B sobe para 100%. Algo semelhante sucede em C->D que tem uma probabilidade de 58.3% e A:C->D que tem probabilidade de 80%. Acontece também a previsão ser alterada, como se pode constatar na tabela 4.3. Se o utilizador vier de B o mais provavel é viajar para D (B->D) com uma probabilidade de 34,5%. Porém se for sabido que este veio previamente de D passa a ser mais provavel que se desloque para C (D:B->C) com uma probabilidade de 38,5%.

Foi definido 3 como limite mínimo de ocorrências de uma subsequência para esta fazer parte do modelo e poder ser usada para previsão. A esta variável deu-se o nome de *minOcurr* na secção 3.3.1. Este deve ser um parâmetro configurável consoante os objetivos pretendidos. Escolheu-se este valor pois foi o que se revelou mais adequado no âmbito do projeto, onde mesmo com poucas ocorrências e sem grau de certeza muito elevado, interessa ter uma previsão.
Também nos casos em que haja transições empatadas com pontuação máxima e não haja mais

nenhuma subsequência de diferente ordem, será escolhida aquela transição que levar ao local com mais ocorrências em todo o historial.

		Number of *subSeq*				
		With x occurrences			Total	
Subject	Order	$3 \leq x < 10$	$10 \leq x < 25$	$25 \leq x$		
A	1	32	2	3	37	91
	2	26	6	1	33	
	3	20	1	0	21	
B	1	0	2	2	4	19
	2	7	1	0	8	
	3	7	0	0	7	

Tabela 4.4: Número de subsequências *subSeq* criadas por cada indivíduo, para ordens e números de ocorrências diferentes.

Na tabela 4.4, pode-se observar as estatísticas das *subSeq* geradas para cada um dos indivíduos A e B. Para o indivíduo A, o modelo gerou 91 *subSeq* diferentes, sendo que 37 são de primeira ordem, 31 de segunda e 21 de terceira. Para o indivíduo B gerou apenas 4, 8 e 7 subsequências para os modelos de 1ª, 2ª e 3ª ordem, respetivamente. Esta diferença considerável entre os modelos dos indivíduos deve-se à diferença de estilos de vida. O indivíduo A tem uma vida mais regular em que todos os dias vai a vários sítios e que se repetem frequentemente, o que gera diversas ocorrências das mesmas subsequências de locais. Pelo contrário, o indivíduo B move-se menos e tem apenas um pequeno conjunto de locais que repete frequentemente. Daí, que o número de locais com mais de três ocorrências seja muito pequeno. Contudo, aqueles que ocorrem, ocorrem com muita frequência. Repare-se que nenhum local se repete menos de 10 vezes.
Quantas mais ocorrências tiver cada *subSeq*, mais segura se torna a predição feita por esta. Interessa que o algoritmo tome a decisão com base numa *subSeq* que tenha sido observada o máximo de vezes possível.

Em [AS03] foi concluído que, mesmo com 4 meses de dados, as transições geradas para o modelo de segunda ordem eram poucas. Aqui, é mostrado que depende muito do tipo de utilizador. O indivíduo A gerou modelos consistentes de segunda ordem e ainda alguns de terceira. Como se pode constatar, não existe uma ordem que seja sempre a melhor. Dependendo de diferentes casos, subsequências de diferentes ordens podem ser as mais adequadas para a previsão. Como existe a possibilidade de trabalhar com modelos de diferentes ordens em simultâneo, modelos até ordem 3 demonstram ser os ideais no âmbito do projeto Time Machine que tem um pequeno período de tempo para o estudo do indivíduo. Talvez seja aconselhável aumentar o valor deste parâmetro m no caso de se trabalhar com um conjunto maior de dados.

Este método, para além de ser um modelo preditivo faz, também ele, a extração de padrões dos dados. Através das tabelas, podem-se extrair as subsequências de locais mais vezes

percorridas pelo utilizador. Esta informação pode ser muito útil para extração de padrões de sequências de locais mais evidentes nos movimentos de um utilizador.

4.7 Síntese

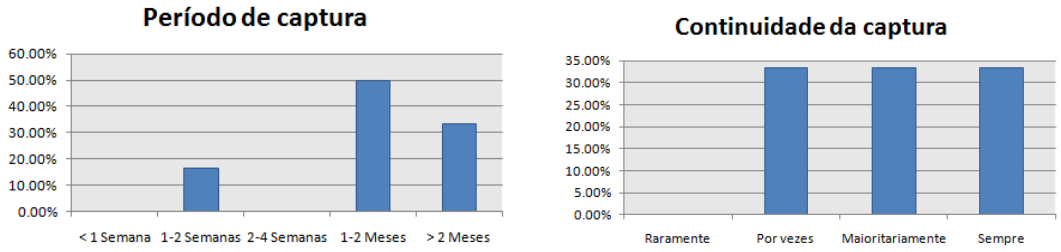
Foram apresentados nesta secção os resultados das diferentes componentes do sistema. Começou-se por explicar os dados utilizados nos testes e foi feita a análise destes. De seguida foi analisada a componente de extração de locais e os significados das diferentes variáveis extraídas. Foram ainda analisados os resultados obtidos pelos testes de classificação e *clustering*, como também, os resultados do modelo preditivo.

5

Avaliação

Neste capítulo apresenta-se o trabalho efetuado para avaliação das componentes desenvolvidas, em particular a extração de locais relevantes a partir dos registos GPS. Foram contactados nove utilizadores que se dispuseram a recolher dados. Cada um utilizou o seu dispositivo móvel com a aplicação de captura desenvolvida. Foram utilizados os seguintes dispositivos móveis (todos eles com o sistema operativo Android): dois LG P500, dois Sapo A5, um HTC Desire HD, um HTC Desire, um Optimus Boston, um HTC Magic e um Vodafone 945. Contudo, os últimos três dispositivos não se revelaram capazes de executar a aplicação de captura em boas condições por incapacidade do telemóvel ou por incompatibilidades de versões do sistema Android. Portanto, para avaliação apenas foi possível utilizar dados de seis, dos nove utilizadores iniciais. Sendo que destes, quatro participam no desenvolvimento do projeto Time Machine.

Após a recolha dos dados capturados pelos utilizadores, utilizando o sistema desenvolvido, foram gerados ficheiros com informação personalizada para cada um dos utilizadores. Esta informação corresponde a três ficheiros diferentes: (1) um ficheiro kml que contém o mapa com os marcadores dos locais extraídos dos dados do utilizador, (2) uma tabela com os locais dos utilizadores ordenados decrescentemente por número de visitas (apenas aparecem locais com mais de três visitas) e (3) as sequências de locais mais relevantes extraídas das tabelas de Markov. A cada utilizador foi enviado este conjunto de ficheiros com informação personalizada e ainda o questionário apresentado no anexo C por correio eletrónico. Pretende-se assim, avaliar o modelo criado para a extração de locais e comprovar que os modelos de Markov servem para extrair rotinas do quotidiano dos utilizadores. Os resultados do inquérito aos utilizadores são apresentados no anexo D.



(a) Período de captura.

(b) Continuidade da captura

Figura 5.1: Informação sobre a captura realizada pelos inquiridos.

Como se pode ver na figura 5.2(a), tem-se que quase todos os inquiridos captaram dados durante um período superior a um mês, sendo que dois deles captaram por mais de dois meses. Contudo, como se pode constatar na figura 5.2(b), dois dos utilizadores apenas por vezes realizaram uma captura contínua. É normal que este fator prejudique um pouco os resultados, pois estes utilizadores não têm dados tão consistentes como os restantes.

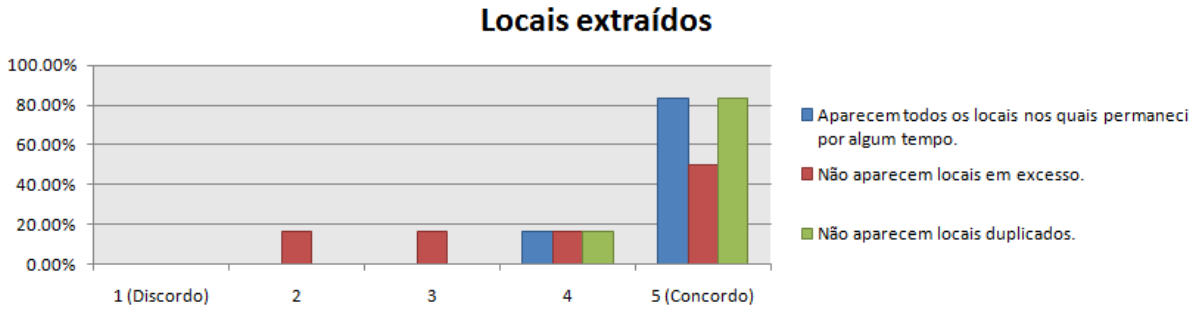


Figura 5.2: Resultados da avaliação dos utilizadores sobre os locais extraídos.

Como se pode constatar pela figura 5.2, a grande maioria dos inquiridos concorda com os seus locais extraídos. Apenas indicam que existem alguns locais em excesso, como é constatado pela avaliação da afirmação "Não aparecem locais em excesso". Contudo, esta avaliação mais baixa já era esperada. Como já referido na secção 4.3.1, os baixos parâmetros utilizados originam alguns locais sem importância. Um dos utilizadores escreveu "Aparece um local numa estrada nacional por onde passei.", o que se deve ao problema já referido de por vezes o sinal GPS ser perdido durante alguns minutos e o utilizador continuar em movimento. Embora estes locais apareçam no mapa, facilmente se depreende a sua pouca relevância pelo número de visitas e duração do local.
Outro utilizador escreveu "Aparecem marcadores em 2 ou 3 pontos perto de minha casa onde nunca permaneci.". Este problema acontece por vezes quando a última coordenada capturada pelo receptor, antes do utilizador entrar em algum sítio sem receção de sinal, ser ainda um

pouco distante desse local. Este problema origina uma estadia num local errado que corresponde a essa última coordenada capturada.

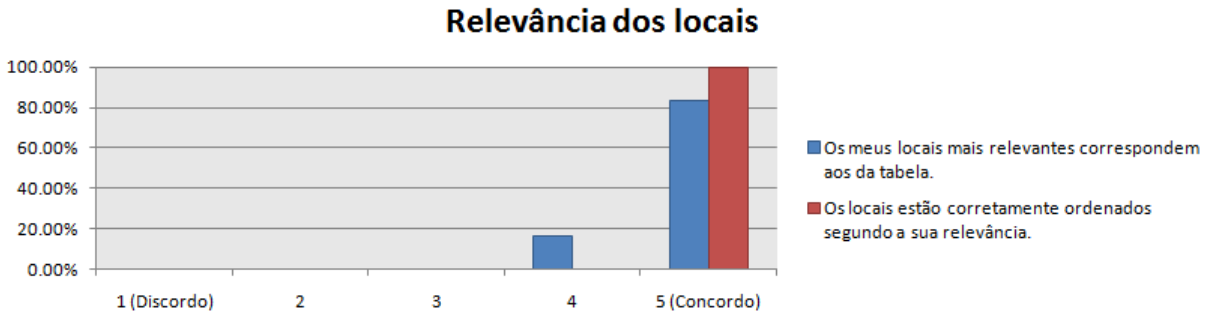


Figura 5.3: Resultados da avaliação dos utilizadores sobre os locais mais relevantes.

Relativamente à relevância dos locais, como se pode ver na figura 5.3, os utilizadores concordam todos com os seus locais relevantes extraídos. Pode-se ainda constatar que a ordenação por número de visitas aparenta ser a ideal. 100% dos inquiridos concordaram com a ordenação dos seus locais relevantes.

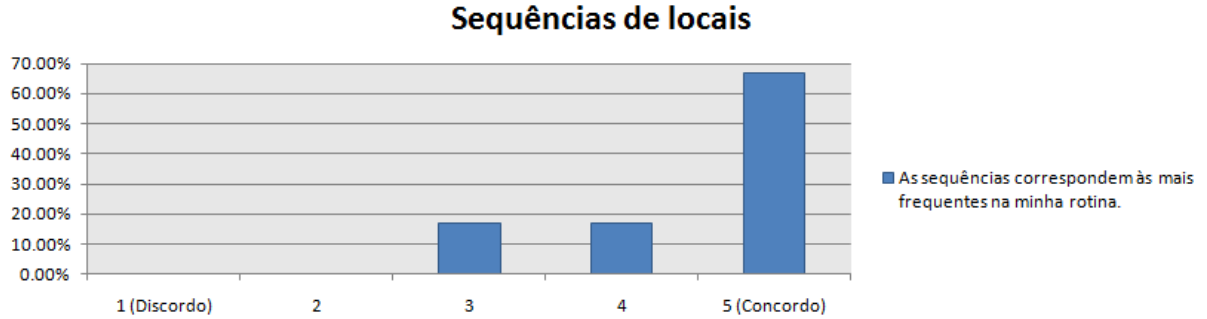


Figura 5.4: Resultados da avaliação dos utilizadores sobre as sequências de locais mais relevantes.

É também possível afirmar que a avaliação prova que os Modelos de Markov são uma aproximação válida para extrair as sequências de locais mais relevantes do utilizador. Como se pode ver na figura 5.4. Contudo, as sequências que se evidenciam não são muitas, surgindo mais consoante o período de recolha de dados aumenta.

Síntese

Os testes foram realizados com um número limitado de utilizadores devido à falta de recursos. Os utilizadores tinham de possuir um telemóvel com sensor GPS que utilizasse o sistema Android. Mesmo assim, como foi visto, nem todos os que preenchiam estes requisitos foram capazes de recolher dados para as experiências necessárias. Os resultados foram em geral

muito positivos. Os utilizadores concordaram com os locais extraídos dos seus dados, tendo apenas sido revelada a existência de locais em excesso. Este resultado já era esperado devido aos parâmetros baixos para deteção de locais. Os inquiridos concordaram por completo com os seus locais mais relevantes ordenados por número de visitas e deram uma boa avaliação às sequências de locais extraídas através das cadeias de Markov.

Para esta avaliação foram utilizados os parâmetros especificados na secção 4.3.1 para os algoritmos de extração de locais. Contudo, melhores resultados podem ser obtidos no caso destes serem ajustados ao estilo de vida de cada utilizador. Se o utilizador tiver locais muito próximos uns dos outros é provável estes convergirem no mesmo. Ao se ajustar os parâmetros ao estilo de vida de cada um melhora o comportamento dos algoritmos mas, mesmo assim, o erro do GPS é sempre um limite.

6

Conclusões e trabalho futuro

Neste capítulo são apresentadas as principais conclusões desta dissertação, bem como o trabalho futuro que fica como sugestão do autor e poderá se útil na continuação do projeto Time Machine. Estes são apresentados nas secções 6.1 e 6.2, respetivamente.

6.1 Conclusões

Esta dissertação apresenta diversos métodos para extração de informação útil a partir de dados de GPS recolhidos pelo dispositivo móvel transportado pelo utilizador, o seu telemóvel. Este dispositivo captura periodicamente a posição espácio-temporal do seu utilizador e cria registos da sua utilização do espaço e do tempo. O objetivo desta dissertação foi estudar métodos que, utilizando os dados contidos nos registos, permitam a extração de padrões de regularidade e irregularidade.
O trabalho da dissertação foi iniciado com a revisão do trabalho já desenvolvido no âmbito do projeto Time Machine e de técnicas de classificação e *clustering* para extração de padrões. Com o desenrolar dos primeiros testes foi constatado que um passo essencial para a extração de padrões seria, primeiro, a extração dos locais do utilizador. Foi então feito um levantamento do estado de arte nessa área que permitiu a construção de um mecanismo para tal efeito.

Foi criado um sistema, composto por diferentes componentes, que permite analisar dados não filtrados do GPS e extrair informação útil, nomeadamente os locais importantes para um utilizador e as diferentes variáveis que os caraterizam. Foi também rigorosamente definido um historial de movimentos do utilizador, do qual se podem extrair diferentes características para diferentes períodos de tempo. Para além disso, foi criado um modelo preditivo capaz de prever

qual o próximo movimento no espaço de um utilizador e que permite ainda extrair os padrões de transição entre locais mais comuns da sua rotina.
Os resultados experimentais comprovaram que o sistema proposto é uma boa solução para atingir os objetivos iniciais. Foram feitos testes às várias componentes do sistema, onde foram obtidos resultados muito bons. Os vários problemas criados pelo erro do GPS e possíveis implicações foram, também eles, diagnosticados.
A aplicação foi avaliada através de testes efetuados com utilizadores onde foram obtidos resultados muito positivos. Realizaram-se testes para avaliar principalmente a extração de locais, mas também a relevância destes e os padrões de sequências de locais mais comuns.

Pode-se então concluir que o principal objetivo da dissertação foi cumprido. O sistema criado é capaz de elaborar um historial rigoroso dos movimentos do utilizador, extrair com precisão os seus locais relevantes e variáveis relacionadas. Foi também realizado um estudo sobre estas variáveis que revela o significado de cada uma. O sistema cria ainda modelos de Markov que permitem prever movimentos futuros e extrair os padrões mais comuns de sequências de locais. Todo este trabalho de pesquisa e desenvolvimento contribuiu para o progresso do projeto Time Machine.

6.2 Trabalho futuro

Esta dissertação está integrada no projeto Time Machine e o trabalho futuro irá avançar em várias direções. Os resultados obtidos nesta dissertação estão já a ser utilizados para a aplicação no telemóvel e está também em desenvolvimento uma base de dados com uma estrutura aproximada à apresentada na secção 3.2.3, para um acesso mais eficaz aos dados.
No âmbito do projeto continuam os estudos para extração de informação dos dados. Em concreto, estão a ser feitos estudos para encontrar os ciclos de dia/noite de cada utilizador individual, para assim se poder utilizar como unidade de tempo os períodos de atividade do utilizador. Serão ainda realizados estudos com o objetivo de encontrar a melhor forma de confrontar o indivíduo com a informação extraída sobre o seu quotidiano.

O tempo disponível para o desenvolvimento desta dissertação não permitiu estudos profundos nos algoritmos de classificação e *clustering*, pelo que, estes métodos devem ser melhor explorados no futuro. Estudos com mais dados, diferentes utilizadores e também diferentes algoritmos são aconselhados. Foi ainda testada uma medida de semelhança entre dias com base no tempo passado nos mesmos locais. A medida testada atribui um valor a variar dentro do intervalo [0,1] consoante o tempo passado em comum nos mesmos locais. 0 para nenhum tempo em comum e 1 se as 24 horas forem passadas nos mesmos locais com a mesma distribuição de tempos. Contudo, esta foi pouco aprofundada e testada. Mais testes com diferentes formas de calcular a medida são aconselhados para um estudo mais detalhado. Deve-se, por exemplo, ter também em conta a ordem dos locais e não só as durações.

Para completar lacunas que possam existir nos dados recolhidos é aconselhada também a inclusão da captura da posição GSM[i] e *WiFi*, tal como foi feito em [HCL+05]. É também possível fazer um estudo mais elaborado das trajetórias. Estas são agora guardadas na estrutura de dados, mas distâncias e velocidades são a única informação extraída. É possível fazer estudos no sentido de inferir o meio de transporte utilizado pelo utilizador, como estudado em [ZLC+08], ou ainda estudar os diferentes percursos. Contudo, para tal é aconselhada uma captura mais integral dos dados GPS. Ou seja, é preciso reduzir o intervalo de captura de informação, para que as trajetórias fiquem melhor definidas. Com tal informação é também possível inferir as atividades em cada local, tal como fez Liao em [LFK07].

[i]Sistema Global para Comunicações Móveis é uma tecnologia móvel e o padrão mais popular para telemóveis do mundo.

Bibliografia

[Amo10] Tiago Amorim. Visualização de padrões pessoais de movimento. Tese de Mestrado, FCT/UNL, 2010.

[Arg] Lars Arge. Abstract the priority r-tree: A practically efficient and worst-case optimal r-tree.

[AS03] Daniel Ashbrook e Thad Starner. Using GPS to learn significant locations and predict movement across multiple users. *Personal Ubiquitous Comput.*, 7(5):275–286, 2003.

[Bie06] Herman J. Bierens. *Information Criteria and Model Selection*. Pennsylvania State University, March 2006.

[Cha11] Robert G. Chamberlain. Great circle distance between 2 points, January 2011. *http://www.movable-type.co.uk/scripts/gis-faq-5.1.html*.

[CMR07] Gabriella Castelli, Marco Mamei, e Alberto Rosi. The whereabouts diary. In *Proceedings of the 3rd international conference on Location-and context-awareness*, LoCA'07, pp. 175–192, Berlin, Heidelberg, 2007. Springer-Verlag.

[dBOH+11] Samuel del Bello, Sofia Oliveira, Jared Hawkey, Olivier Perriquet, e Nuno Correia. Processing location data for ambient intelligence applications. Porto, Portugal, 2011. Ambi-Sys.

[DH97] Dana e Peter H. Global positioning system (GPS) time dissemination for real-time applications. *Real-Time Syst.*, 12(1):9–40, 1997.

[DM07] Mitch J. Duncan e W. Kerry. Mummery. GIS or GPS? : a comparison of two methods for assessing route taken during active transport. pp. 33(1):51–53, 2007.

[DP97] Pedro Domingos e Michael J. Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, 29(2-3):103–130, 1997.

[EpKSX96] Martin Ester, Hans peter Kriegel, Jörg S, e Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. pp. 226–231. AAAI Press, 1996.

[ESP06] Nathan Eagle e Alex (Sandy) Pentland. Reality mining: sensing complex social systems. *Personal Ubiquitous Comput.*, 10:255–268, March 2006.

[FGP07] Katayoun Farrahi e Daniel Gatica-Perez. Daily routine classification from mobile phone data, 2007. To appear in MLMI'08.

[FJS+92] Frawley, W. J., Shapiro, Piatetsky G., e C. J. Matheus. Knowledge discovery in databases - an overview. *Ai Magazine*, pp. 57–70, 1992.

[Gil11] Chuck Gilbert. How is the accuracy of a gps receiver described, January 2011. *http://www.romdas.com/technical/gps/gps-acc.htm*.

[gps11] GPS accuracy - how accurate is it?, February 2011. *http://www.maps-gps-info.com/gps-accuracy.html*.

[Hav84] *Virtues of the Haversine*, volume 68. Sky and Telescope, 1984.

[HC93] Holte e Robert C. Very simple classification rules perform well on most commonly used datasets. *Mach. Learn.*, 11(1):63–90, 1993.

[HCL+05] Jeffrey Hightower, Sunny Consolvo, Anthony LaMarca, Ian E. Smith, e Jeff Hughes. Learning and recognizing the places we go. In *Ubicomp'05*, pp. 159–176, 2005.

[HLL03] Jin Huang, Jingjing Lu, e Charles X. Ling. Comparing naive bayes, decision trees, and SVM with AUC and accuracy. In *ICDM '03: Proceedings of the Third IEEE International Conference on Data Mining*, pp. 553, Washington, DC, USA, 2003. IEEE Computer Society.

[HT04] Ramaswamy Hariharan e Kentaro Toyama. Project lachesis: parsing and modeling location histories. In *In Geographic Information Science*, pp. 106–124, 2004.

[IBM11] IBM. Geographic coordinate system, January 2011. *http://publib.boulder.ibm.com/infocenter/db2luw/v8/index.jsp?topic=/com.ibm.db2.udb.doc/opt/csb3022a.htm*.

[Kot07] S. B. Kotsiantis. Supervised machine learning: A review of classification techniques. *Informatica*, 31:249–268, 2007.

[Küp05] Axel Küpper. *Location-based Services: Fundamentals and Operation*. Willey, 2005.

[KR90] L. Kaufman e P. J. Rousseeuw. *Finding groups in data: An introduction do Cluster Analysis.* John Wiley & sons, 1990.

[KWSB04] Jong Hee Kang, William Welbourne, Benjamin Stewart, e Gaetano Borriello. Extracting places from traces of locations. In *Proceedings of the 2nd ACM international workshop on Wireless mobile applications and services on WLAN hotspots*, WMASH '04, pp. 110–118, New York, NY, USA, 2004. ACM.

[LFK07] Lin Liao, Dieter Fox, e Henry Kautz. Extracting places and activities from GPS traces using hierarchical conditional random fields. *Int. J. Rob. Res.*, 26(1):119–134, 2007.

[LZX+08] Quannan Li, Yu Zheng, Xing Xie, Yukun Chen, Wenyu Liu, e Wei-Ying Ma. Mining user similarity based on location history. In *Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems*, GIS '08, pp. 34:1–34:10, New York, NY, USA, 2008. ACM.

[Mit97] T. Mitchell. *Machine Learning*. McGraw Hill, 1997.

[MS00] Natalia Marmasse e Chris Schmandt. Location-aware information delivery with commotion. In *Proceedings of the 2nd international symposium on Handheld and Ubiquitous Computing*, HUC '00, pp. 157–171, London, UK, 2000. Springer-Verlag.

[PA10] Projecto Time Machine (PTDC/EAT-AVP/105384/2008). *FCT - Fundação para a Ciência e a Tecnologia*. 2010.

[PM00] Dan Pelleg e Andrew W. Moore. X-means: Extending k-means with efficient estimation of the number of clusters. In *ICML '00: Proceedings of the Seventeenth International Conference on Machine Learning*, pp. 727–734, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[Res92] Sidney I. Resnick. *Adventures in stochastic processes*. Birkhauser Verlag, Basel, Switzerland, Switzerland, 1992.

[Ris05] Irina Rish. An empirical study of the naive bayes classifier. In *IJCAI-01 workshop on Empirical Methods in AI*, October 2005.

[SEKX98] Jörg Sander, Martin Ester, Hans-Peter Kriegel, e Xiaowei Xu. Density-based clustering in spatial databases: The algorithm GDBSCAN and its applications. *Data Min. Knowl. Discov.*, 2:169–194, June 1998.

[Sny87] John P. Snyder. *Map Projections: A Working Manual.* Number 1395 in Professional Paper. Geological Survey (U.S.), 1987.

[SP04] S.Kotsiantis e P. Pintelas. Increasing the classification accuracy of simple bayesian classifier. volume 3192 of *Lecture Notes in Artificial Intelligence*, pp. 198–207. Springer Berlin / Heidelberg, 2004.

[Vin75] *Direct and inverse solutions of geodesics on the ellipsoid with application of nested equations*, volume 22. Survey Review, 1975.

[WF09] Ian H. Witten e Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufman, second edition, 2009.

[ZBST07] Changqing Zhou, Nupur Bhatnagar, Shashi Shekhar, e Loren Terveen. Mining personally important places from GPS tracks. In *ICDEW '07: Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering Workshop*, pp. 517–526, Washington, DC, USA, 2007. IEEE Computer Society.

[ZFhLW02] Osmar R. Zaïane, Andrew Foss, Chi hoon Lee, e Weinan Wang. On data clustering analysis: Scalability, constraints and validation. In *In Proceedings of the 6th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD*, pp. 28–39, 2002.

[ZFL+04] Changqing Zhou, Dan Frankowski, Pamela Ludford, Shashi Shekhar, e Loren Terveen. Discovering personal gazetteers: An interactive clustering approach. In *In Proc. ACMGIS*, pp. 266–273. ACM Press, 2004.

[ZLC+08] Yu Zheng, Quannan Li, Yukun Chen, Xing Xie, e Wei-Ying Ma. Understanding mobility based on GPS data. In *Proceedings of the 10th international conference on Ubiquitous computing*, UbiComp '08, pp. 312–321, New York, NY, USA, 2008. ACM.

Visualizações geográficas

Visualizações geográficas utilizando o Google Earth. Estas visualizações correspondem a mapas com marcadores que representam eventos, *stay points* e locais do utilizador. As imagens foram escolhidas para ilustrar diferentes aspetos apresentados ao longo da dissertação.

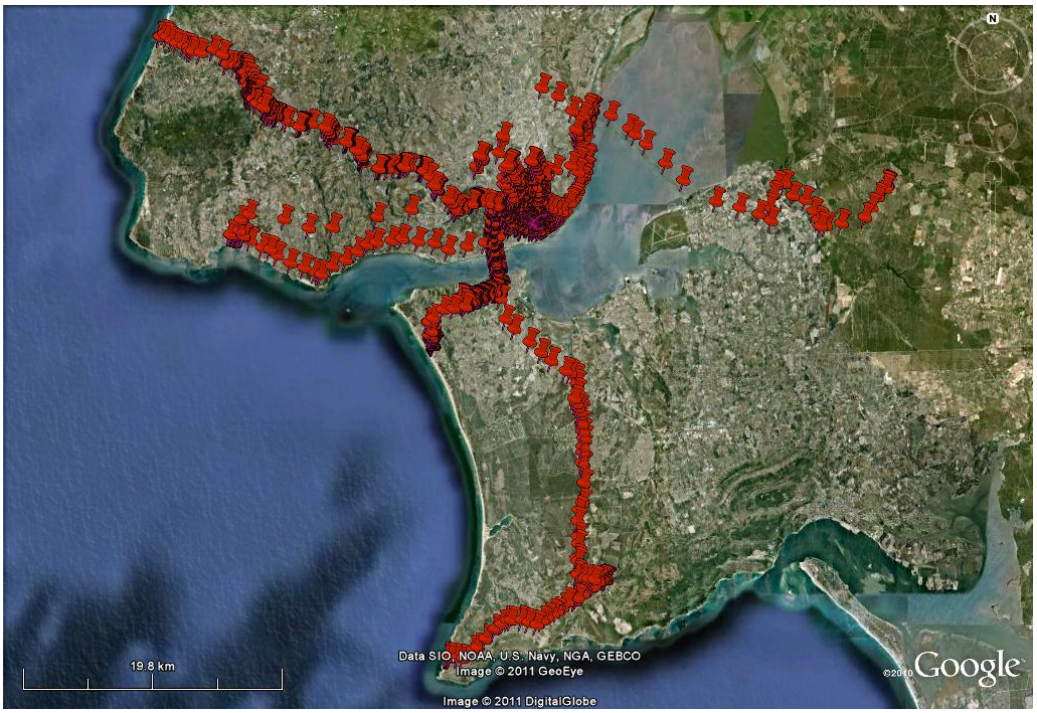


Figura A.1: Indivíduo A: mapa com todos os eventos recolhidos.

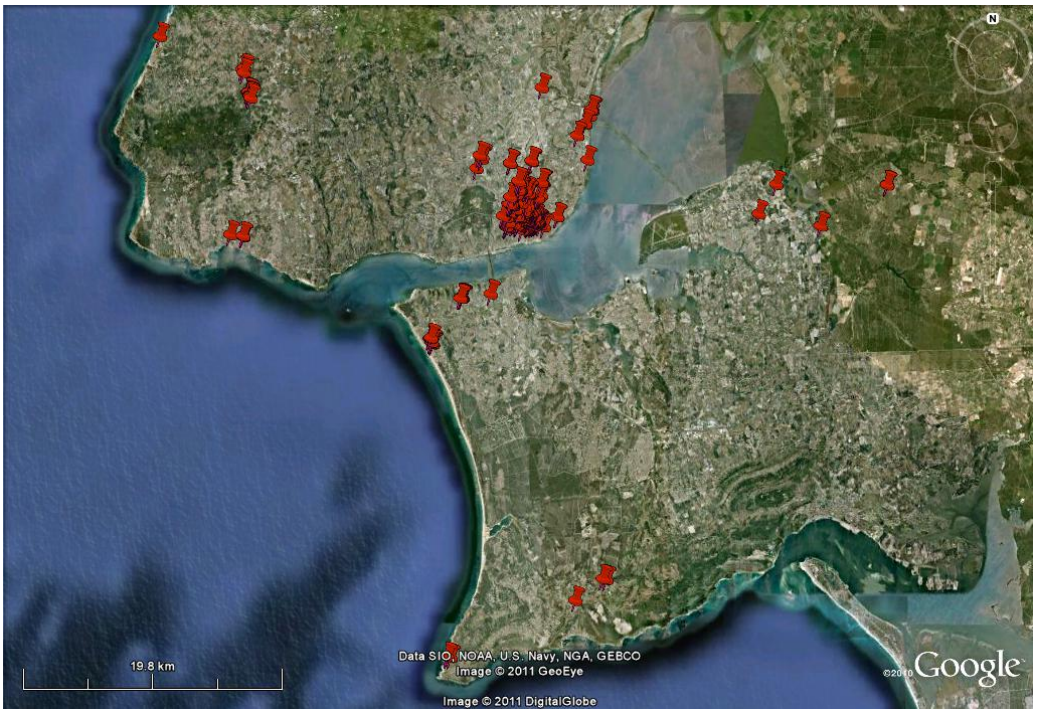


Figura A.2: Indivíduo A: mapa com todos os *stay points* filtrados a partir dos eventos.

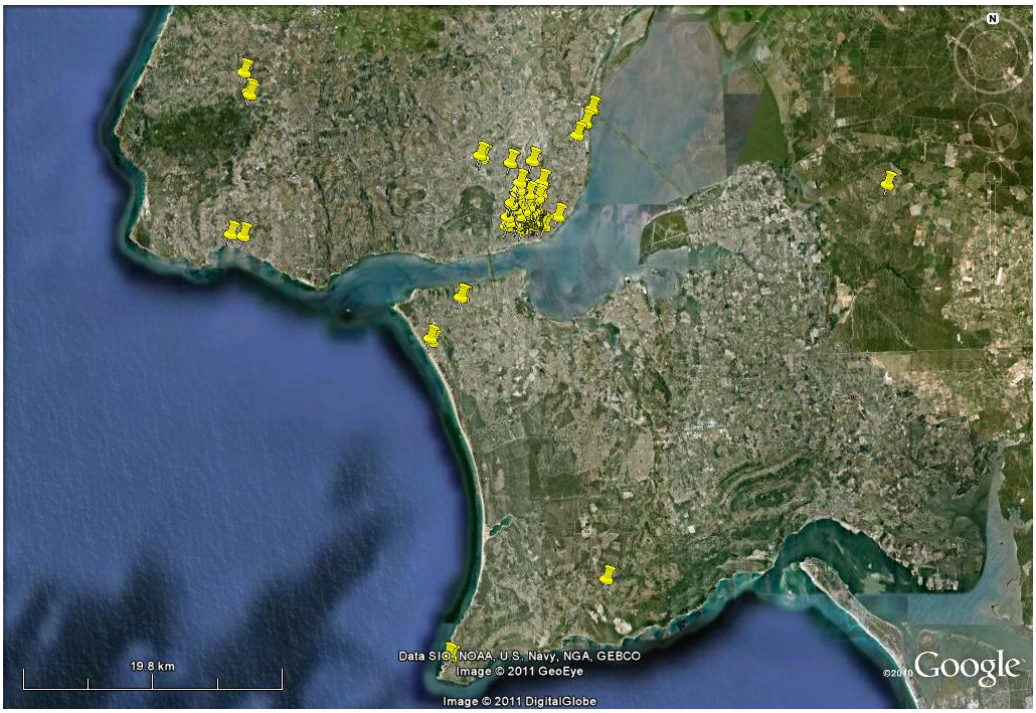


Figura A.3: Indivíduo A: mapa com todos os locais extraídos dos *stay points*.

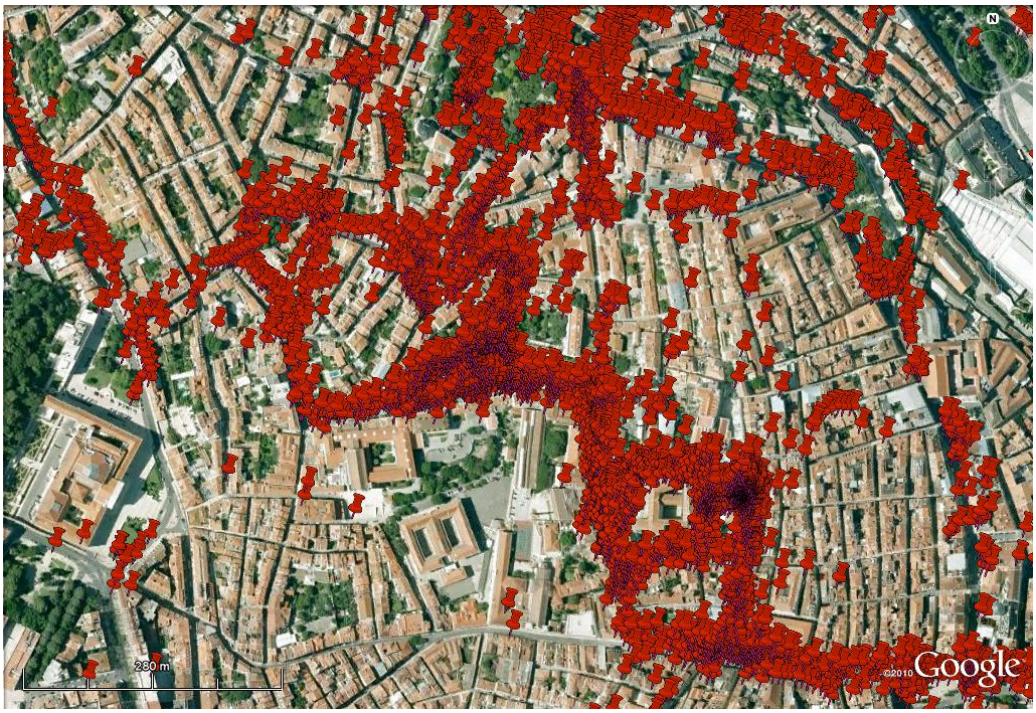


Figura A.4: Indivíduo A: *close-up* ao centro de atividade, com todos os eventos recolhidos.

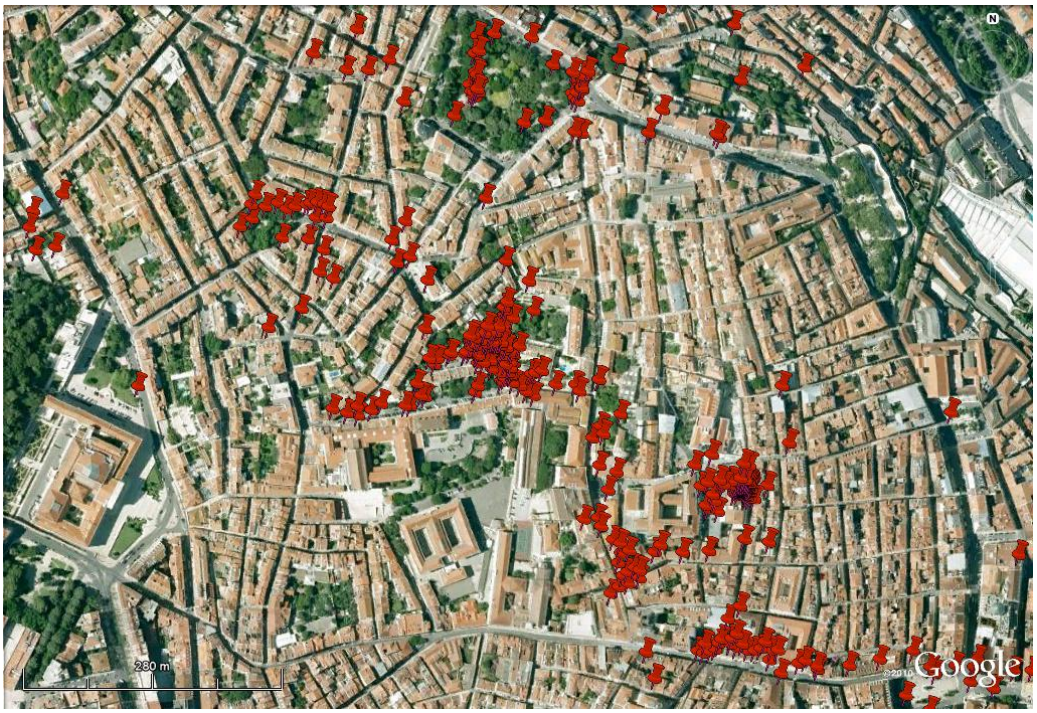


Figura A.5: Indivíduo A: *close-up* ao centro de atividade, com todos os *stay points* filtrados a partir dos eventos.

Figura A.6: Indivíduo A: *close-up* ao centro de actividade, com todos os locais extraídos dos *stay points*.

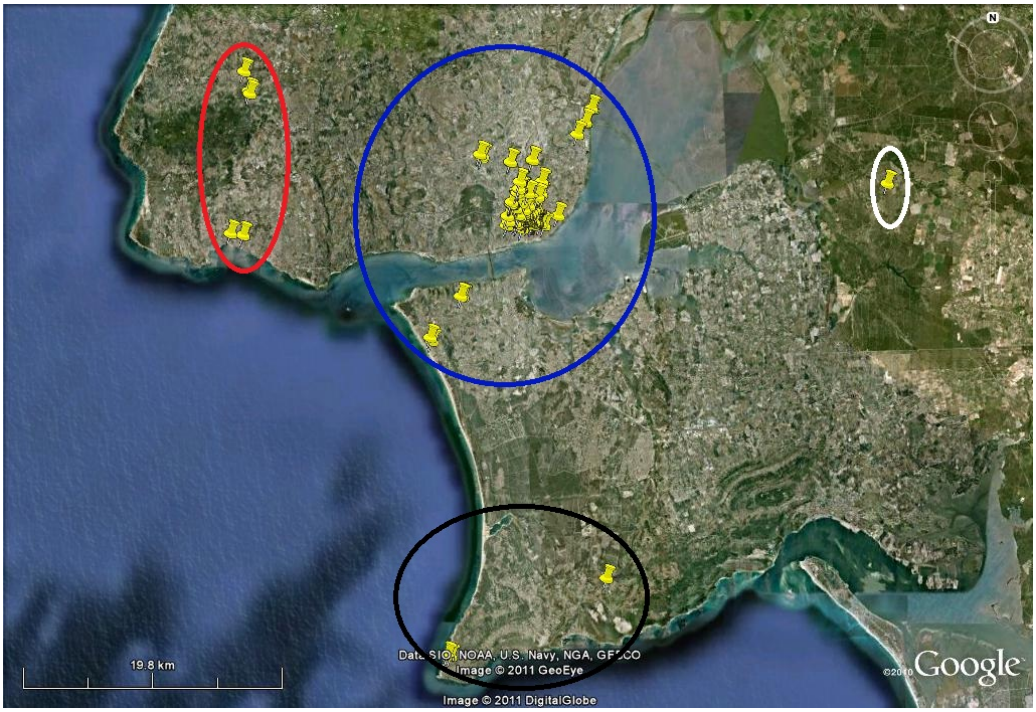


Figura A.7: Indivíduo A: mapa com todos os locais divididos pelos grupos formados pelo *clustering* geográfico.

Figura A.8: Dados artificiais: mapa com todos os locais divididos pelos grupos formados pelo *clustering* geográfico.

B

Visualizações de variáveis extraídas

Visualizações em gráficos de variáveis extraídas dos dados. São apresentados gráficos construídos a partir de informação extraída dos dados de cada utilizador.

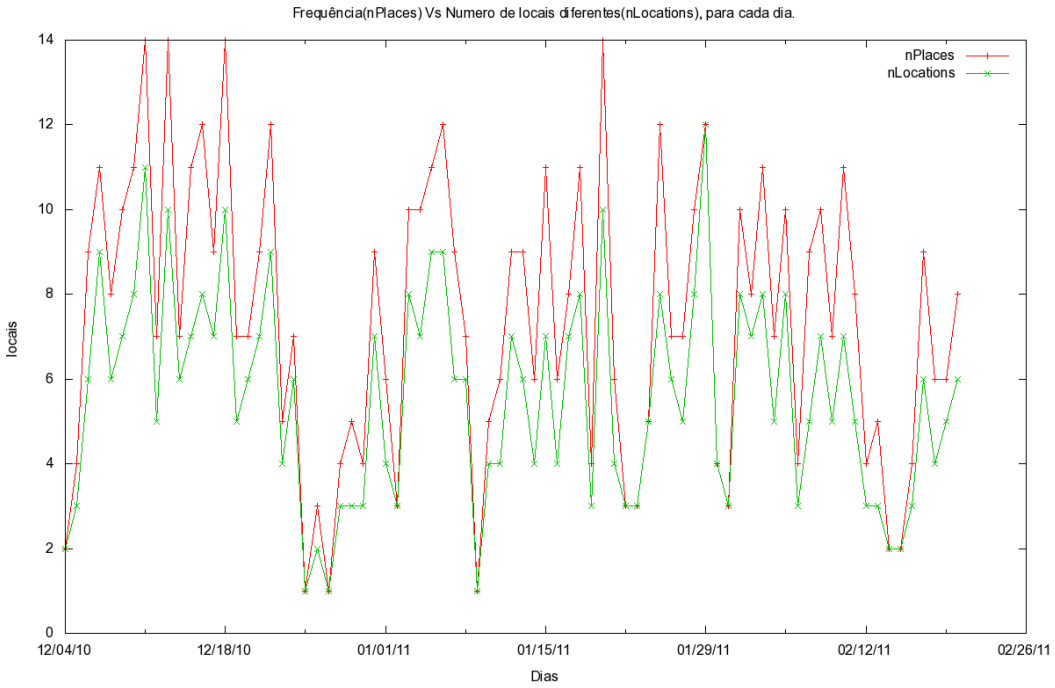


Figura B.1: Indivíduo A: gráfico para vários dias com número de sítios e número de locais visitados.

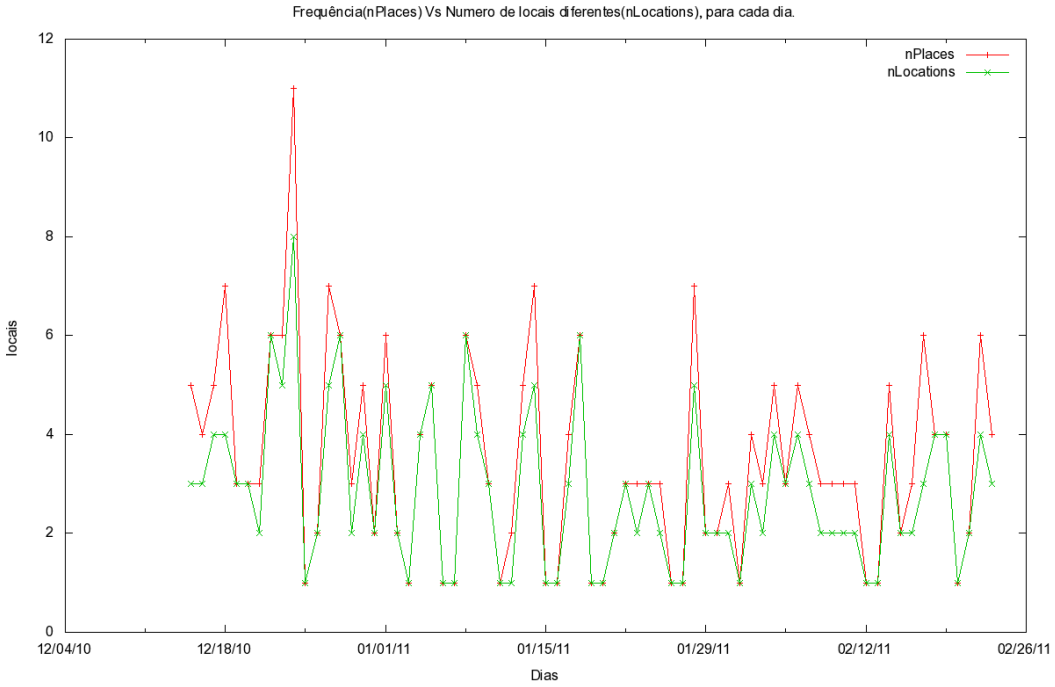


Figura B.2: Indivíduo B: gráfico para vários dias com número de sítios e número de locais visitados.

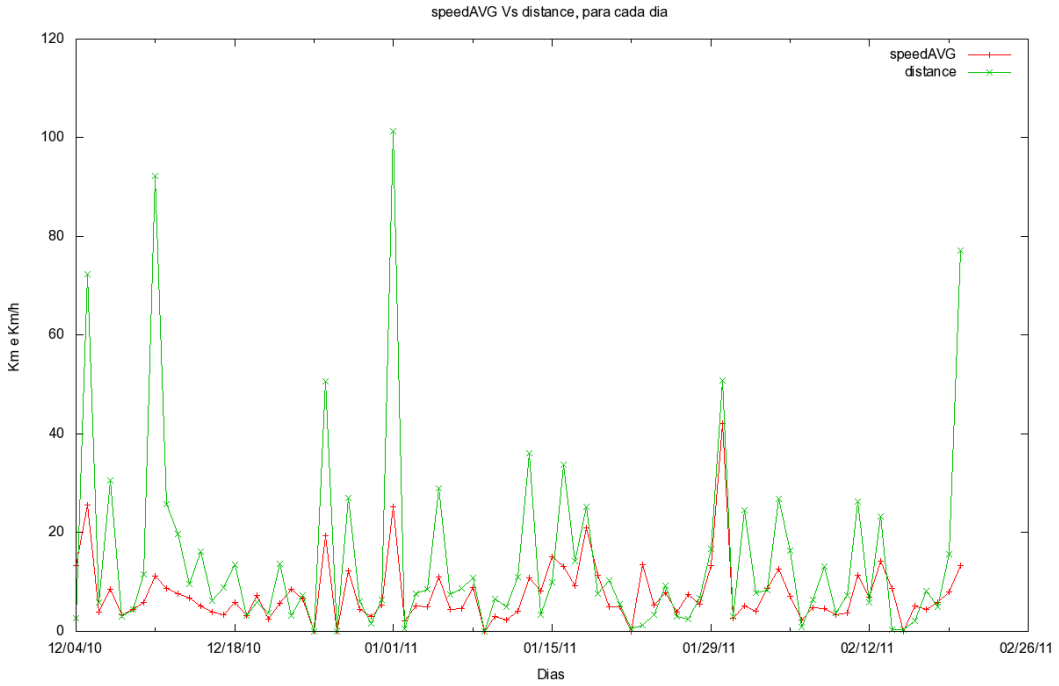


Figura B.3: Indivíduo A: gráfico para vários dias com a média das velocidades e distância percorrida.

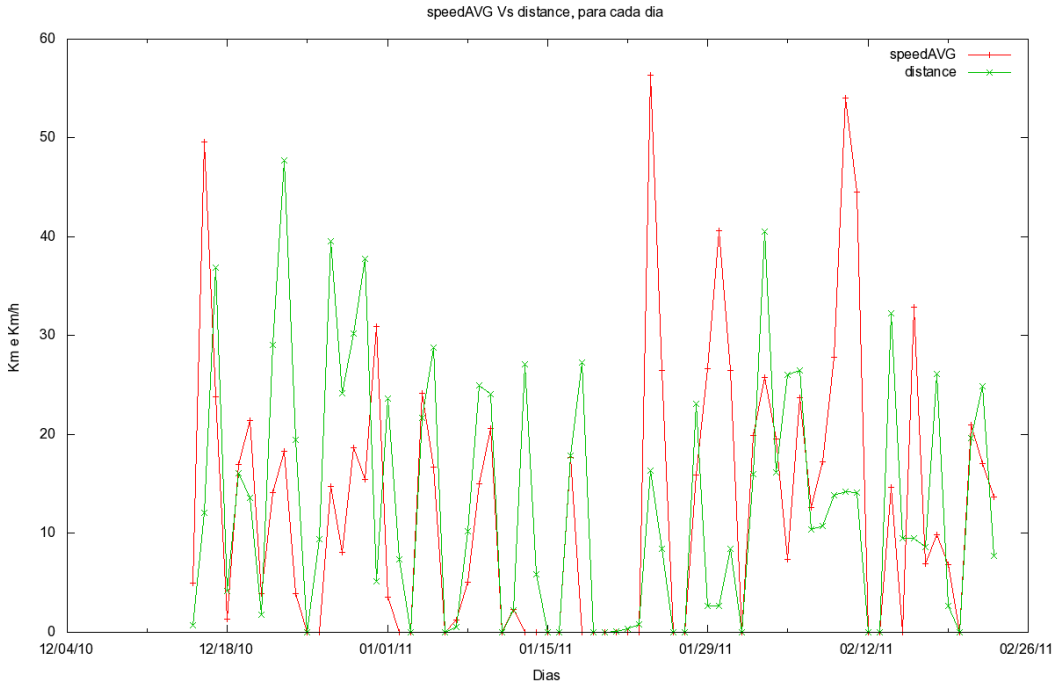


Figura B.4: Indivíduo B: gráfico para vários dias com a média das velocidades e distância percorrida.

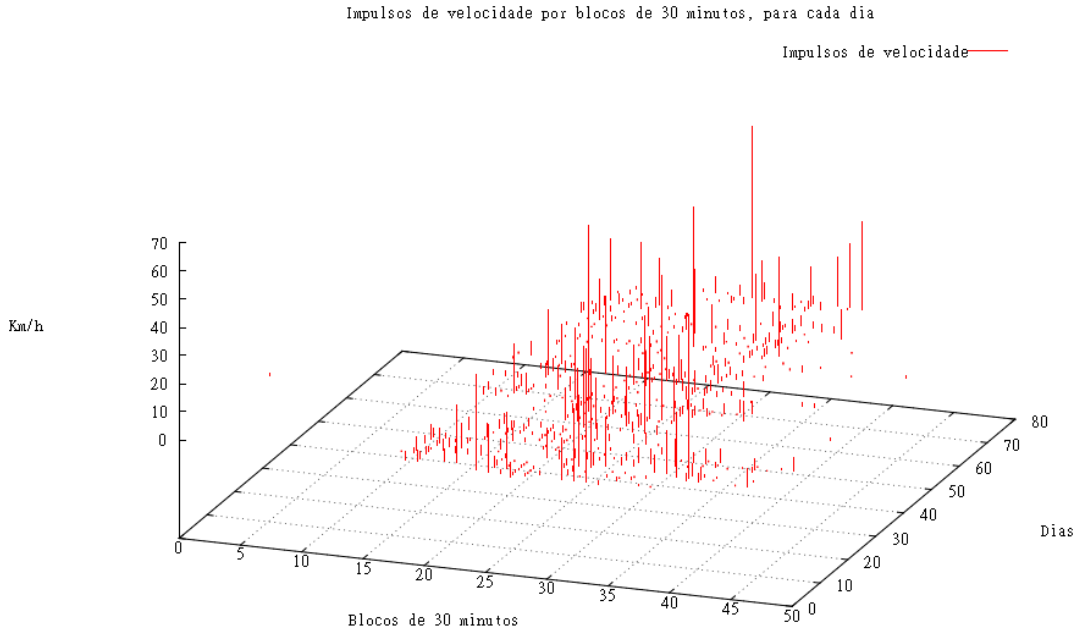


Figura B.5: Indivíduo A: gráfico com os impulsos de velocidade ao longo dos vários dias.

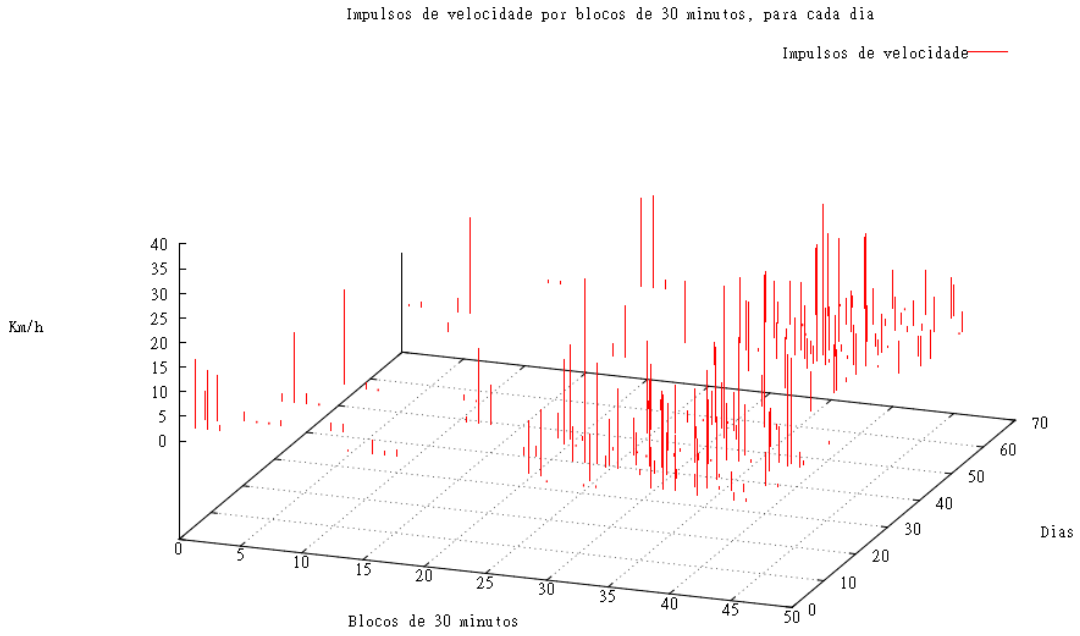


Figura B.6: Indivíduo B: gráfico com os impulsos de velocidade ao longo dos vários dias.

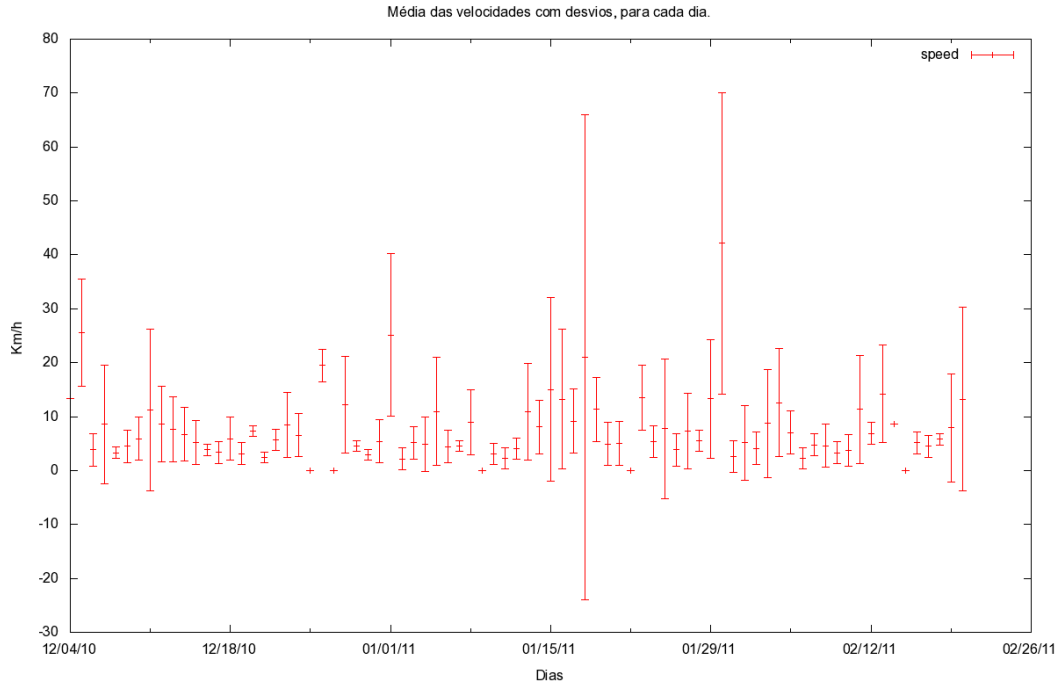


Figura B.7: Indivíduo A: gráfico para vários dias com velocidade média e respetivos desvios.

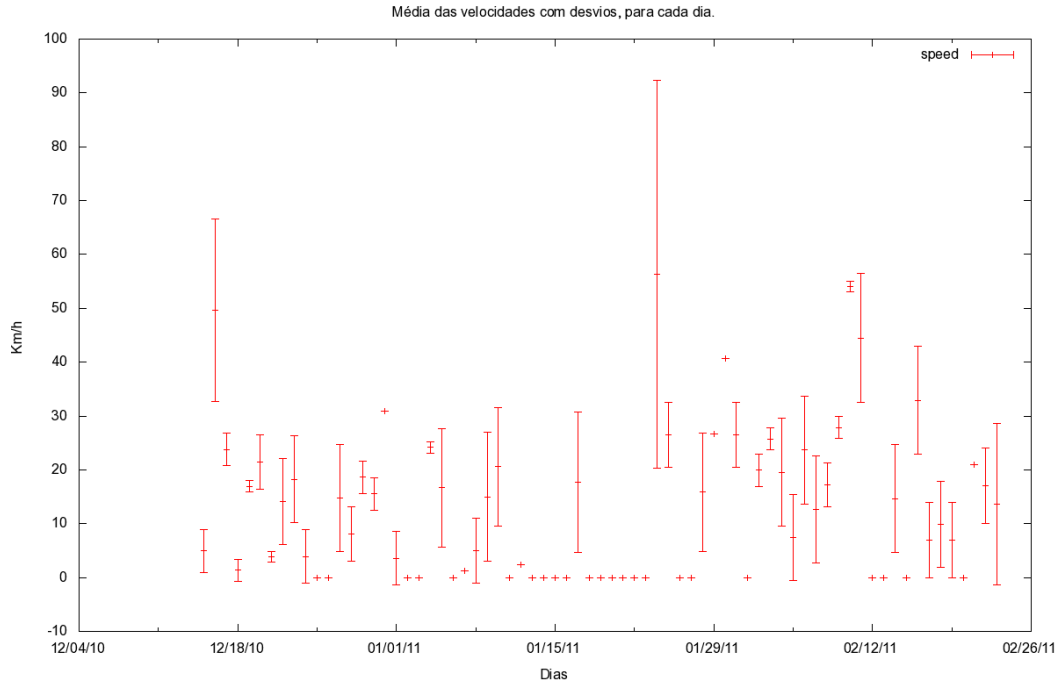


Figura B.8: Indivíduo B: gráfico para vários dias com velocidade média e respetivos desvios.

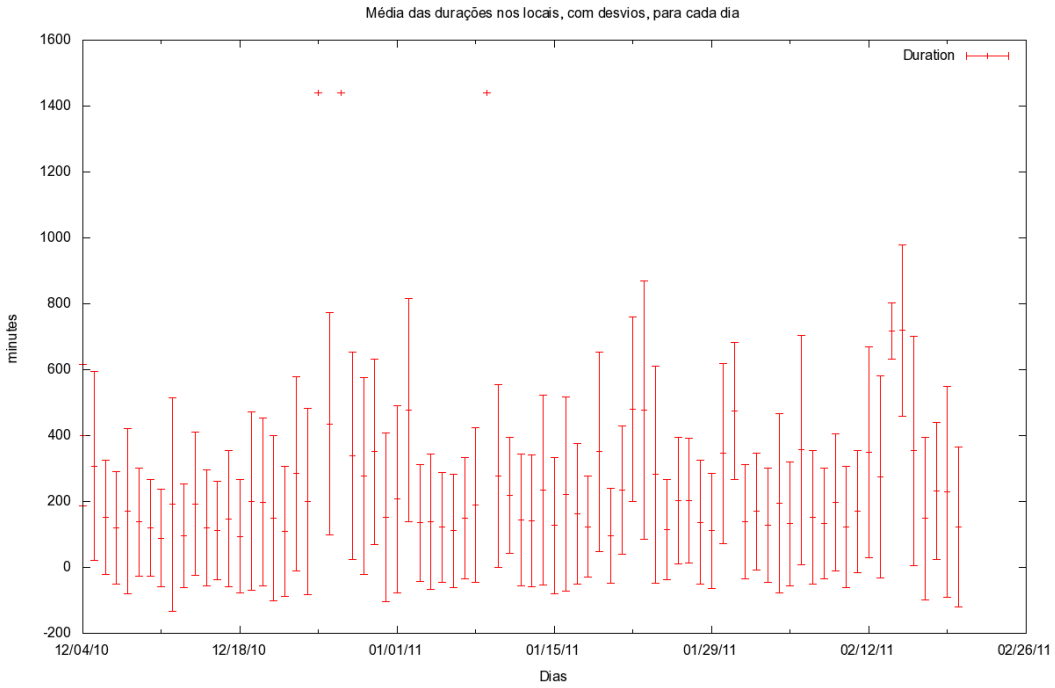


Figura B.9: Indivíduo A: gráfico para vários dias com durações médias nos locais e respetivos desvios.

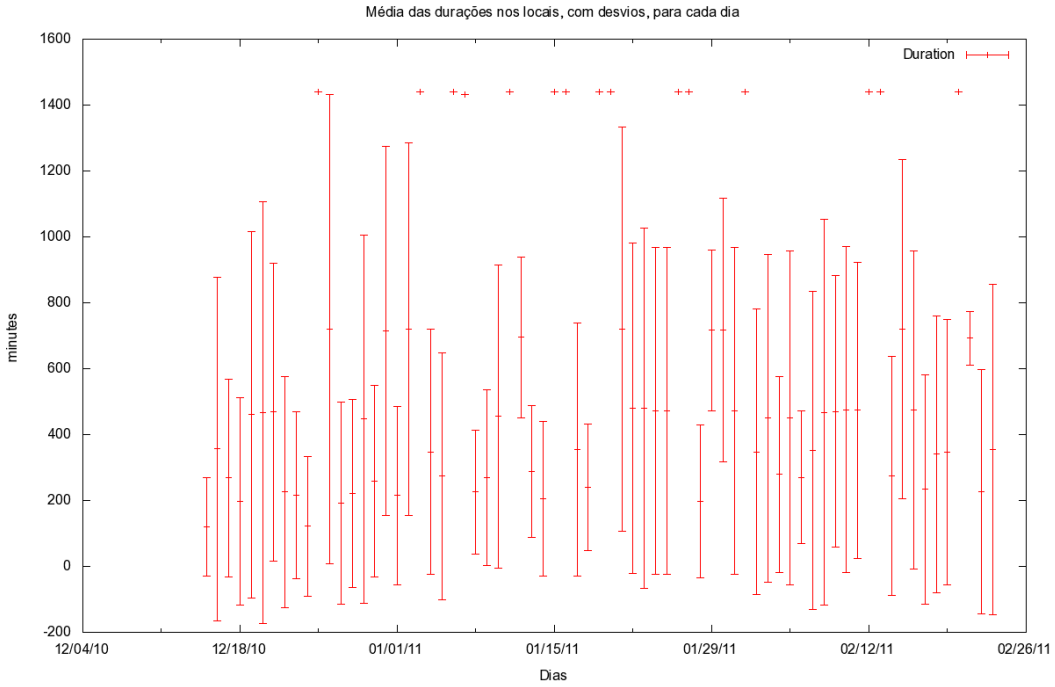


Figura B.10: Indivíduo B: gráfico para vários dias com durações médias nos locais e respetivos desvios.

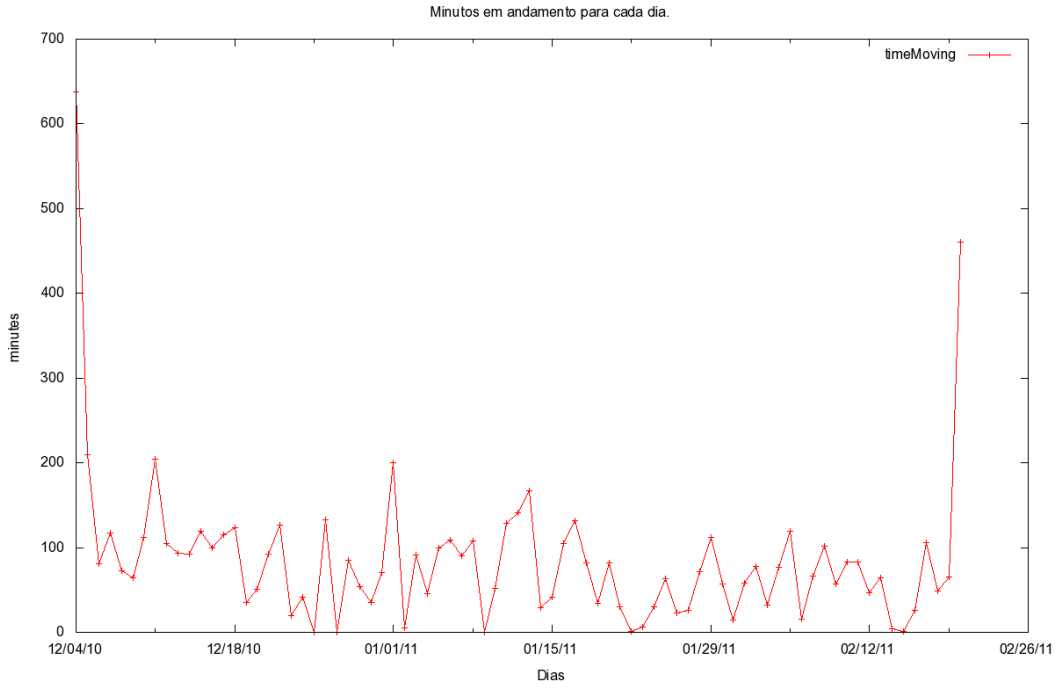


Figura B.11: Indivíduo A: gráfico para vários dias com quantidade de tempo em movimento.

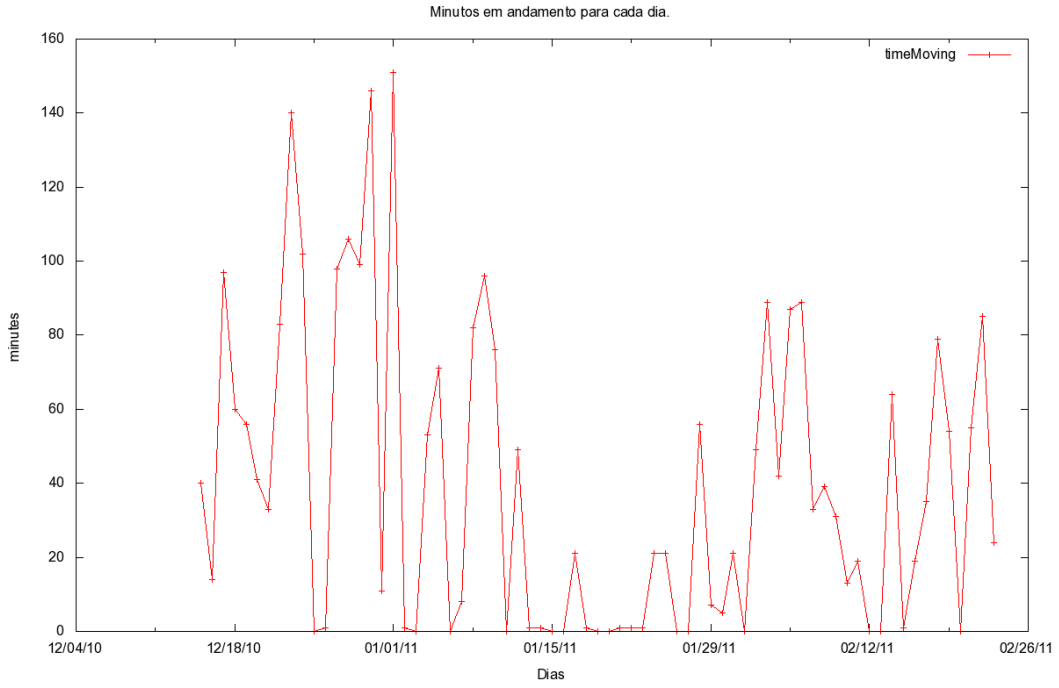


Figura B.12: Indivíduo B: gráfico para vários dias com quantidade de tempo em movimento.

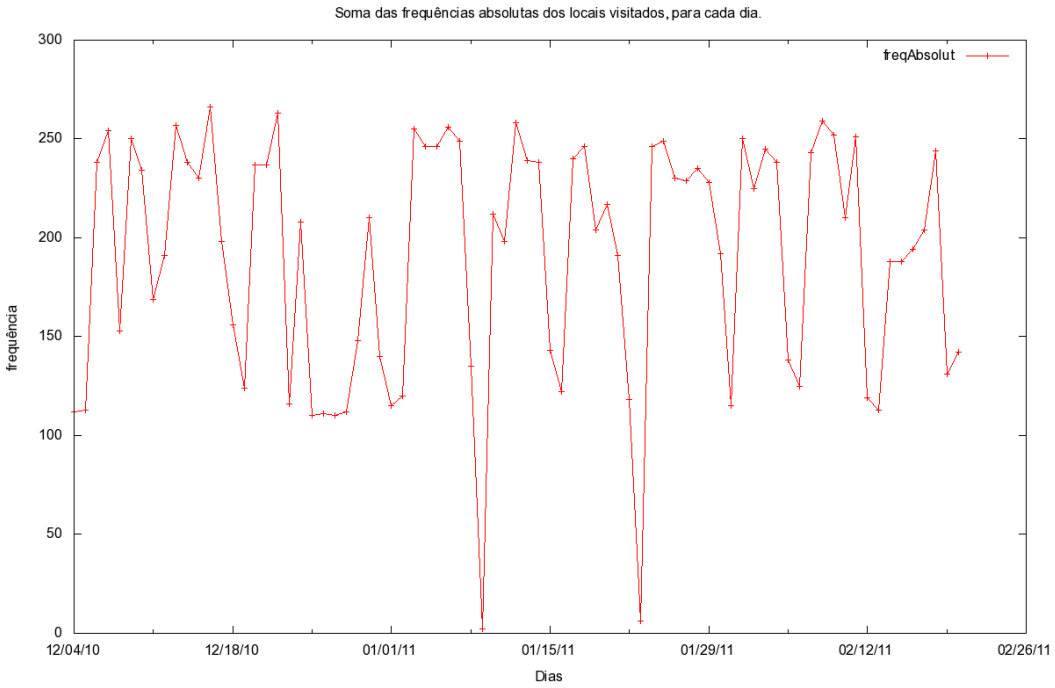


Figura B.13: Indivíduo A: gráfico para vários dias com a soma das frequências totais dos locais visitados.

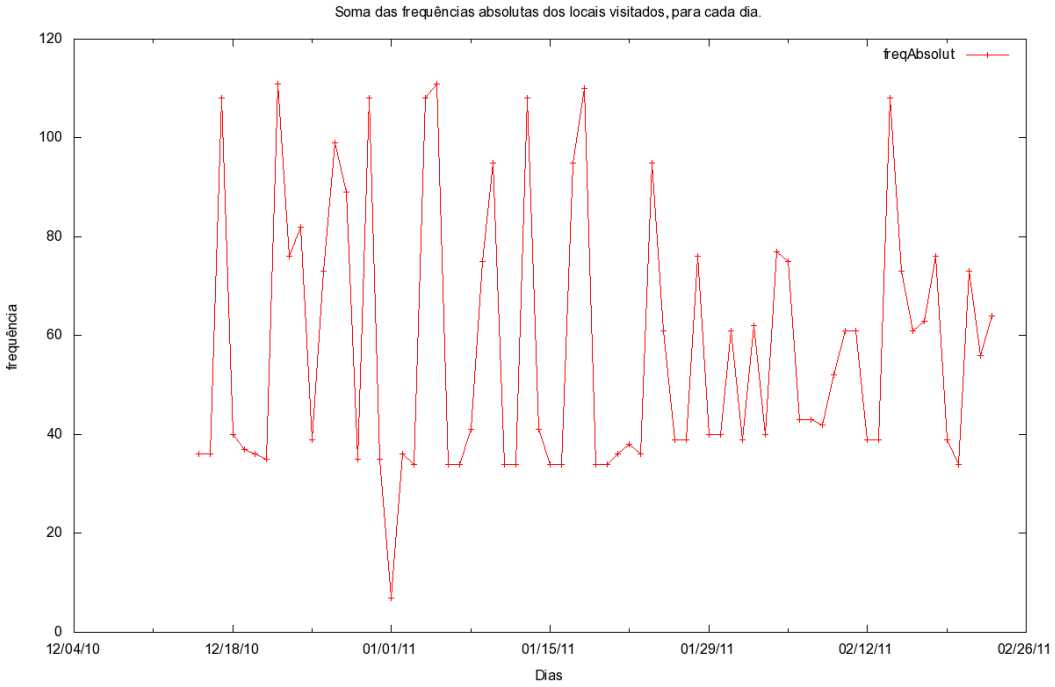


Figura B.14: Indivíduo B: gráfico para vários dias com a soma das frequências totais dos locais visitados.

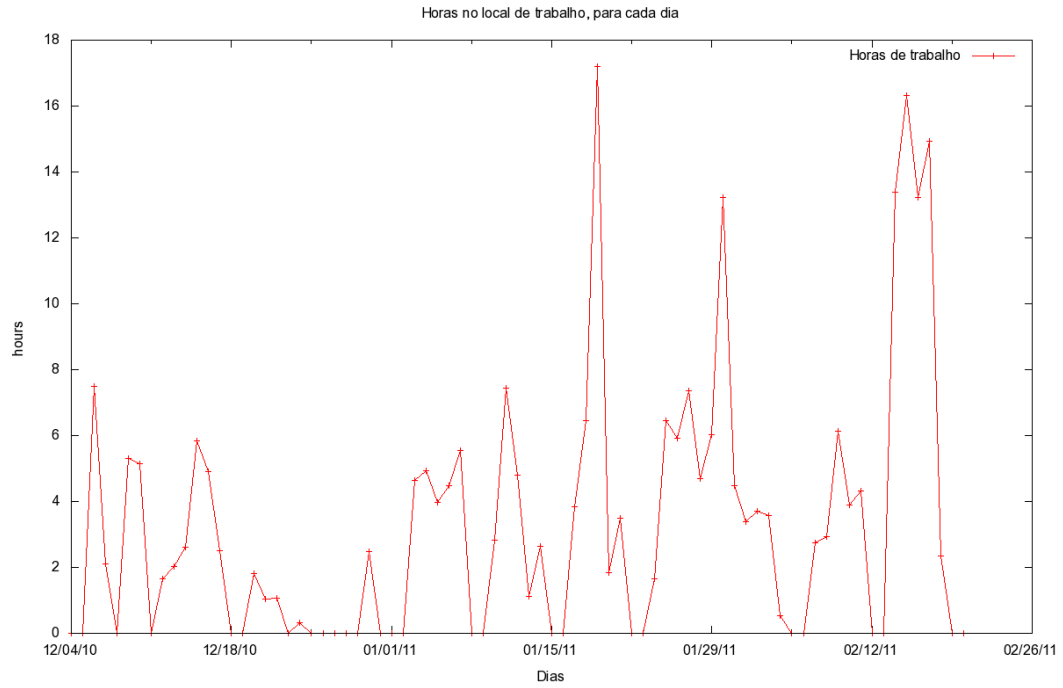


Figura B.15: Indivíduo A: gráfico para os vários dias com as horas passadas no local de trabalho.

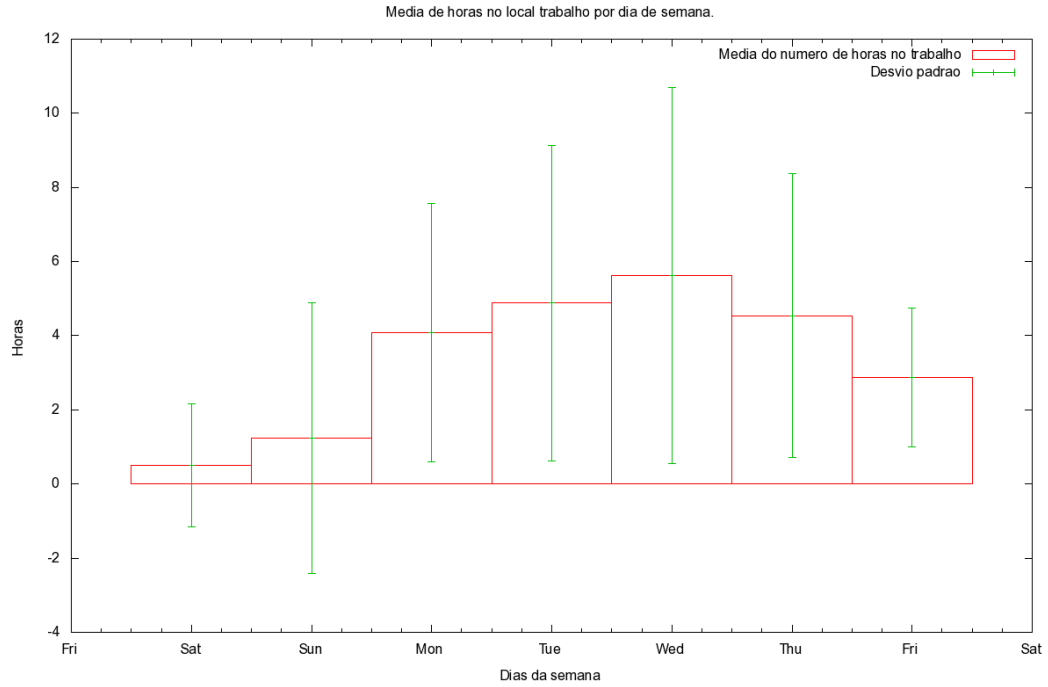


Figura B.16: Indivíduo A: gráfico com as médias de horas passadas no trabalho para cada dia da semana.

Questionário

O modelo do questionário utilizado na avaliação através de inquérito aos utilizadores é apresentado de seguida.

QUESTIONÁRIO

Este questionário tem como objetivo o estudo da informação extraída dos ficheiros GPS captados através de uma aplicação para dispositivos móveis.
Todos os dados recolhidos são confidenciais e não serão utilizados com qualquer outra finalidade.

UTILIZADOR

1. Faixa etária

- ☐ 10-19
- ☐ 20-29
- ☐ 30-39
- ☐ 40-49
- ☐ 50-59
- ☐ 60-69
- ☐ 70-79
- ☐ mais de 80

2. Género

- ☐ Feminino
- ☐ Masculino

3. Participo no desenvolvimento do projeto Time Machine

- ☐ Sim
- ☐ Não

CAPTURA

4. Capturei dados durante um período de:

- ☐ < 1 Semana
- ☐ 1-2 Semanas
- ☐ 2-4 Semanas
- ☐ 1-2 Meses
- ☐ > 2 Meses

5. A captura foi contínua.

Sempre	Maioritariamente	Por vezes	Raramente
❑	❑	❑	❑

INFORMAÇÕES EXTRAÍDAS DA CAPTURA DE DADOS

Tem disponíveis na pasta anexa ao correio electrónico três tipos de informações diferentes: (1) um kml com os locais extraídos, (2) uma tabela com os locais ordenados por relevância e (3) o conjunto de sequências de locais mais relevantes. Cada grupo de alíneas que se segue diz respeito a um destes itens. Pedimos que consulte primeiro o ficheiro em questão e o analise, para depois responder às perguntas feitas.
Durante a execução do questionário tenha presente que a informação tratada diz apenas respeito ao período de recolha dos dados.

6. Locais extraídos (Ficheiro: Locations.kml – abrir com Google Earth)

6.1. Aparecem todos os locais nos quais permaneci por algum tempo.

Discordo				Concordo
1	2	3	4	5

6.1.1. Caso existam, dê exemplos de locais que não aparecem: _____

_____

6.2. Não aparecem locais em excesso.

Discordo				Concordo
1	2	3	4	5

6.2.1. Caso existam, diga a que associa esses locais em excesso: _____

_____

6.3. Não aparecem locais duplicados.

Discordo				Concordo
1	2	3	4	5

6.3.1. Caso existam, dê exemplos de locais duplicados: _____

_____

7. Relevância dos locais (Ficheiro: Locations by relevance.xlsx)

7.1. Os meus locais mais relevantes correspondem aos da tabela.

Discordo Concordo

1 2 3 4 5

7.1.1.Caso existam, diga quais os locais com os quais não concorda: _____

_____

7.1.2.Caso existam, diga quais os locais que pensa faltarem: _____

_____

7.2. Os locais estão corretamente ordenados segundo a sua relevância.

Discordo Concordo

1 2 3 4 5

7.2.1.Caso discorde, diga qual a ordenação que pensa ser a mais correta: _____

_____

8. Sequências de locais (Ficheiro: Most common sequences.txt)

8.1. As sequências correspondem às mais frequentes na minha rotina.

Discordo Concordo

1 2 3 4 5

8.1.1.Caso existam, diga quais as sequências com que não concorda: _____

_____

8.1.2.Caso existam, diga quais as sequências que estão em falta: _____

_____

9. Sugestões e comentários

_____

_____

_____

_____

_____

_____

_____

_____

_____

Muito obrigado pela sua colaboração.

D

Resultados do inquérito aos utilizadores

Os resultados obtidos na avaliação, através de um inquérito por questionário aos utilizadores, são apresentados no conjunto de tabelas que se seguem.

Faixa etária	Utilizadores
10 aos 19	0
20 aos 29	4
30 aos 39	0
40 aos 49	2
50 aos 59	0
60 aos 69	0
70 aos 79	0
mais de 80	0

Género	Utilizadores
Masculino	5
Feminino	1

Participo no desenvolvimento do projeto Time Machine	Utilizadores
Sim	4
Não	2

Capturei dados durante um período de	Utilizadores
< 1 Semana	0
1-2 Semanas	1
2-4 Semanas	0
1-2 Meses	3
> 2 Meses	2

A captura foi contínua	Utilizadores
Raramente	0
Por vezes	2
Maioritariamente	2
Sempre	2

Locais extraídos	1	2	3	4	5
Aparecem todos os locais nos quais permaneci por algum tempo.	0	0	0	1	5
Não aparecem locais em excesso.	0	1	1	1	3
Não aparecem locais duplicados.	0	0	0	1	5

Relevância dos locais	1	2	3	4	5
Os meus locais mais relevantes correspondem aos da tabela.	0	0	0	1	5
Os locais estão corretamente ordenados segundo a sua relevância.	0	0	0	0	6

Sequências de locais	1	2	3	4	5
As sequências correspondem às mais frequentes na minha rotina.	0	0	1	1	4