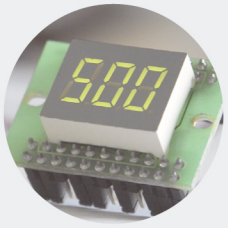
INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES






Joint User Mobility and Traffic Characterization in Temporary Crowded Events

ADRIANO FILIPE BORGES VALADAR
novembro de 2018

POLITÉCNICO
DO PORTO

Instituto Superior de Engenharia do Porto
Departamento de Engenharia Eletrotécnica
Rua Dr. António Bernardino de Almeida 431, 4200-072 Porto

Joint User Mobility and Traffic Characterization in Temporary Crowded Events

Tese/Dissertação do Mestrado em Engenharia Eletrotécnica e de Computadores - Área de Especialização de Telecomunicações elaborada por:

Adriano Filipe Borges Valadar

Orientador: Prof. Jorge Botelho da Costa Mamede

Ano Letivo: 2017/2018

Este relatório satisfaz, parcialmente, os requisitos que constam da Ficha de Unidade Curricular de Tese/Dissertação, do 2º ano, do Mestrado em Engenharia Eletrotécnica e de Computadores no ramo de Telecomunicações

Candidato: Adriano Filipe Borges Valadar
Número: 1110461
1110461@isep.ipp.pt

Orientação Científica: Prof. Jorge Botelho da Costa Mamede
jbm@isep.ipp.pt

Empresa: INESCTEC
Supervisão: Eduardo Nuno Moreira Soares de Almeida
eduardo.n.almeida@inesctec.pt

Instituto Superior de Engenharia do Porto

Departamento de Engenharia Eletrotécnica
Instituto Superior de Engenharia do Porto

Agradecimentos

Agradeço ao meu orientador do Instituto Superior de Engenharia do Porto (ISEP), o Professor Jorge Mamede, pela forma como me auxiliou e orientou ao longo do desenvolvimento da presente Tese.

Ao meu supervisor do Instituto de Engenharia de Sistemas e Computadores, Tecnologia e Ciência (INESC TEC), Eduardo Nuno Almeida, devido a ter acreditado em mim para a realização deste trabalho e por todo o conhecimento e conselhos transmitidos.

A nível pessoal, agradeço à minha namorada pelo apoio incansável e ajuda que me deu. A todos os meus amigos e familiares, em particular aos meus pais, por toda a ajuda que me prestaram. Agradeço ainda ao Prof Dr Tiago Henriques Coelho e ao Dr Miguel Oliveira Soares por toda a dedicação, carinho e profissionalismo.

Resumo

Em eventos temporários com elevada densidade de pessoas (Temporary Crowded Events – TCEs), por exemplo, festivais de música, os utilizadores deparam-se com problemas no acesso a ligações à Internet. TCEs são eventos de duração limitada com elevada concentração de pessoas que se movimentam dentro de um recinto, enquanto acedem à Internet. Contrariamente a outro tipo de eventos onde as localizações dos utilizadores são constantes e conhecidas à partida (ex. estádios), a geração de tráfego e a movimentação dos utilizadores em TCEs é variável e influenciada pela dinâmica do evento. A movimentação dos utilizadores pode levar a sobrecargas em APs (Access Point) no caso de estes serem fixos. De modo a minimizar este fenómeno, têm vindo a ser exploradas novas técnicas que recorrem ao posicionamento ajustável de APs integrados em UAVs (Unmanned Aerial Vehicles). Nestes cenários, a dinâmica da localização dos APs requer que ferramentas de previsão do movimento dos utilizadores e, por sua vez, das fontes de tráfego ganhem particular expressão ao serem relacionadas com os algoritmos de posicionamento dos referidos APs.

De modo a permitir o desenvolvimento e análise de novas soluções de planeamento de rede para TCEs, é necessário recriar estes cenários em simulação, o que, por sua vez, exige uma caracterização detalhada deste tipo de eventos. Esta Dissertação tem como objetivo caracterizar e modelizar conjuntamente a mobilidade e o tráfego gerado pelos utilizadores em TCEs. Esta caracterização possibilitará o desenvolvimento de novos modelos estatísticos de geração de tráfego e mobilidade dos utilizadores em TCEs.

Abstract

At Temporary Crowded Events (TCEs), for example, music festivals, users are faced with problems accessing the Internet. TCEs are limited time events with a high concentration of people moving within the event enclosure while accessing the Internet. Unlike other events where the user locations are constant and known at the start (e.g. stadiums), the traffic generation and the user movement in TCEs is variable and influenced by the dynamics of the event. The movement of users can lead to overloads in APs (Access Point) in case they are fixed. In order to minimize this phenomenon, new techniques have been explored that resort to the adjustable positioning of APs integrated into UAVs (Unmanned Aerial Vehicles). In these scenarios, the dynamic of the location of the APs requires that tools of prediction of the movement of the users and, in turn, of the sources of traffic gain particular expression when being related to the algorithms of positioning of the referred APs.

In order to allow the development and analysis of new network planning solutions for TCEs, it is necessary to recreate these scenarios in simulation, which, in turn, requires a detailed characterization of this type of events. This dissertation aims to characterize and model the mobility and traffic generated by users in TCEs. This characterization will enable the development of new statistical models of traffic generation and user mobility in TCEs.

Conteúdo

Lista de Figuras

Glossário

Abreviatura	Descrição
ISEP	Instituto Superior de Engenharia do Porto
INESC TEC	Instituto de Engenharia de Sistemas e Computadores, Tecnologia e Ciência
TCE	Temporary Crowded Event
e.g	exempli gratia
AP	Access Point
UAV	Unmanned Aerial Vehicles
LTE	Long Term Evolution
WSS	Within Sum of Square
SMS	Short Message Service
TCP	Transmission Control Protocol
WC	Water Closet

Capítulo 1

Introdução

Neste capítulo é feita uma introdução que contextualiza esta tese, passando por expor os problemas que levaram aos objetivos da mesma, terminando com a calendarização e estrutura do relatório.

1.1 Problema

Os TCE, como por exemplo os festivais de música, são eventos caracterizados por possuírem um recinto que é subdividido em áreas recreativas, tais como o palco de atuações, a zona das bebidas, a zona de oferta de brindes, entre outras. São ainda caracterizados pelo universo de pessoas (utilizadores) que participa no festival pois cada utilizador pode mover-se ao longo do decorrer do evento de forma espontânea e imprevisível dependendo da atividade que pretender realizar. Existe ainda outra variável que condiciona este evento, que é o quanto um utilizador utiliza a Internet, ou seja, o tráfego gerado por cada pessoa através do seu *smartphone*. Devido ao comportamento de cada utilizador ser imprevisível, estes deparam-se com problemas no acesso à Internet [1]. Estes problemas devem-se não só ao tipo de evento, que são caracterizados por terem a geração de tráfego e a movimentação dos utilizadores variável e dependente da dinâmica do evento, mas também ao planeamento da rede que é efetuado. O planeamento que é efetuado normalmente é feito tendo em conta que os dispositivos de rede se encontram fixos. Existem novas técnicas que fazem com que esses dispositivos se possam mover, tornando assim interessante e vantajosa a relação entre o movimento desses dispositivos e o movimento dos utilizadores.

Estes eventos são uma realidade à escala mundial, sendo que a variedade e a lotação tem aumentado exponencialmente. Há vários casos em que os ingressos para os mesmos esgotam num curto prazo de tempo e, em situações em que a

entrada é gratuita, existem também casos de sobrelotação na zona do evento. No caso de Portugal, no ano de 2017 houve 272 festivais de música [2], havendo um crescimento a nível do número de festivais na ordem dos 9% quando comparado a 2016.

A par do crescimento de festivais de música, está também o mercado de *smartphones* e a utilização dos mesmos em festivais para os mais diversos objetivos, que pode passar de uma simples chamada a um *stream* de um vídeo. No caso do festival MEO Sudoeste, a Altice revelou os dados da edição de 2018 deste festival através de um comunicado de imprensa [3]. Este comunicado revela que houve um reforço a nível de rede, através da adição de 138 km de fibra e de 70 novas células 2G, 3G e 4G, de forma a garantir a total operacionalidade das várias redes MEO no recinto. De um universo de 147 mil festivaleiros, os números deste evento são os seguintes:

- 11 terabytes de dados WiFi;
- 2,6 milhões de sessões únicas WiFi;
- 9089 Gigabytes de dados móveis;
- 428 mil chamadas telefónicas;
- 265 mil SMS (*Short Message Service*).

Os elevados valores de Internet utilizada neste exemplo refletem o nível de planeamento exigido às operadoras para estes eventos.

1.2 Objetivo

A necessidade de caracterizar, modelizar e prever conjuntamente a mobilidade e o tráfego gerado pelos utilizadores em TCEs motivou a elaboração desta Tese. Através desses resultados será possível que os dispositivos de rede se movam consoante a previsão efetuada.

Como tal, será desenvolvida uma ferramenta que caracteriza e modeliza conjuntamente a mobilidade e o tráfego gerado pelos utilizadores que tendo como ponto de partida o posicionamento, o movimento e o tráfego dos utilizadores consiga estimar a sua posição em instantes seguintes, bem como quantificar o número de utilizadores que se encontra em cada zona do recinto e ainda a quantidade de tráfego gerado em cada uma dessas áreas. Esta ferramenta poderá ser, posteriormente, utilizada para simular o movimento e geração de tráfego dos utilizadores em TCEs, o que, por sua vez, irá permitir avaliar novas soluções de planeamento de redes para este tipo de eventos.

Na medida do conhecimento obtido durante a pesquisa das técnicas utilizadas neste tipo de eventos, é possível afirmar que é a primeira vez que é desenvolvida uma aplicação que considere a mobilidade e o tráfego conjuntamente.

1.3 Calendarização

Este trabalho foi dividido em 4 fases:

- Pesquisa, estado de arte e planeamento;
- Implementação;
- Testes e análise de resultados;
- Escrita do relatório.

A calendarização deste trabalho que engloba todas estas fases encontra-se na figura 1.1.

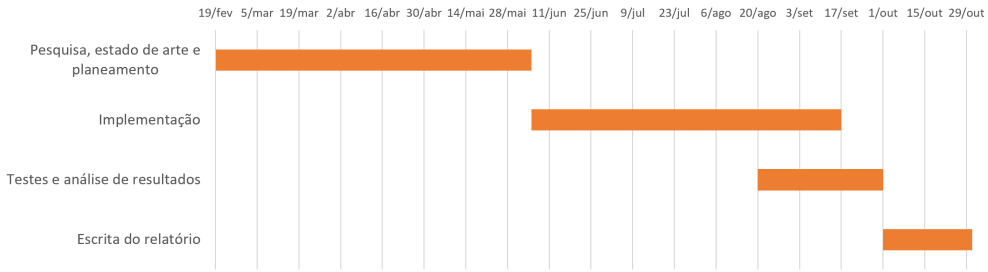


Figura 1.1: Calendarização

1.4 Estrutura do Relatório

Este relatório encontra-se dividido em cinco capítulos.

No capítulo 1 é feita uma introdução da dissertação desenvolvida, através da exposição do problema, do objetivo e da calendarização.

No capítulo 2 é abordado o estado de arte, descrevendo as técnicas mais utilizadas noutros estudos e que foram utilizadas neste trabalho.

No capítulo 3 são explicadas as funcionalidades implementadas na aplicação desenvolvida.

No capítulo 4 são mostrados os testes, a demonstração e a análise de resultados, utilizando um caso de simulação como exemplo.

No capítulo 5 são apresentadas as conclusões deste trabalho e são ainda perspetivados desenvolvimentos futuros.

Capítulo 2

Estado da arte

Neste capítulo o cenário do problema a tratar é apresentado bem como as técnicas mais utilizadas em artigos de cenários semelhantes. São ainda apresentadas as técnicas que são utilizadas neste trabalho e como conclusão as funcionalidades que são necessárias neste trabalho.

2.1 Mobilidade e Tráfego em TCE

Em TCE, as pessoas (utilizadores) movem-se dentro de um espaço limitado (recinto). A par com o percurso inerente de cada utilizador, está também o seu tráfego gerado através do seu *smartphone*. O comportamento de cada utilizador quanto a estes dois fatores é intrínseco e dependente do decorrer do evento. Como tal, de forma a evitar falhas é necessário ter um planeamento de rede que assegure que o pico de tráfego gerado em determinados instantes do evento não afete a rede de forma negativa, pois esses picos trarão falhas na rede. De forma a evitar que tal aconteça, é necessário estudar o comportamento dos utilizadores, bem como caracterizar a mobilidade e o tráfego de todo o universo de utilizadores. É ainda relevante agrupar os dados de mobilidade dos utilizadores de forma a obter várias áreas do recinto, para poder ser estudado o tráfego e o número de utilizadores obtidos em cada zona. As zonas serão também utilizadas para a mobilidade do utilizador, substituindo as coordenadas da localização do mesmo. É conveniente usar mecanismos que ajudem a estudar o comportamento dos utilizadores, tais como técnicas que permitam decidir qual o intervalo de tempo a considerar (em vez de considerar os dados referentes a toda a duração do evento evitando que hajam dados desatualizados a afetar o estudo do comportamento dos utilizadores e consequentemente a sua previsão) e ainda uma forma de organizar os dados de maneira que seja possível prever para que zonas um utilizador se poderá dirigir

consoante a sua zona no momento. No caso pretendido a rede é do tipo Wi-Fi e contempla uma técnica em que os APs podem-se mover consoante a necessidade de rede em cada área do recinto, através da utilização de UAVs [4]. Deste modo, é possível satisfazer a necessidade dos utilizadores em tempo real ajustando o posicionamento destes UAVs. Vários autores realizaram estudos de forma a melhorar o planeamento de rede.

Em [5] é feita uma análise às falhas da rede em eventos do tipo TCE, em que o foco dos autores é mostrar que as mudanças na distribuição dos utilizadores, no comportamento de cada utilizador e no tipo de aplicações utilizadas durante este tipo de eventos resultam numa degradação significativa do desempenho da rede. Este artigo é interessante para o trabalho a desenvolver, uma vez que comprova que existem falhas neste tipo de eventos. Porém, nenhuma solução é apresentada.

Em [6], os dados de várias torres celulares de 3G e LTE (Long Term Evolution) de zonas metropolitanas são extraídos e analisados. De forma a organizar os dados, é utilizada uma técnica de *Clustering*, denominada de *Hierarchical Clustering*, que consiste em ir agrupando os dados aos pares, sendo esses pares mais uma vez agrupados, e assim sucessivamente, até se chegar ao número de *clusters* pretendido [7]. Esta técnica é utilizada de forma a que se obtenham áreas em que os utilizadores possuam um comportamento característico na geração de tráfego. O grande objetivo desse trabalho foi criar um modelo que combina informações de tempo, localização e frequência para analisar os padrões de tráfego de milhares de torres celulares.

No caso do artigo [8], o objetivo dos autores é compreender a dinâmica do tráfego da Internet em grandes redes celulares, o que, segundo os autores, é útil para o projeto de rede, resolução de problemas, avaliação do desempenho e otimização da rede. Os dados utilizados para este estudo foram recolhidos de uma operadora de telecomunicações. Os dados correspondem ao tráfego de dispositivos móveis durante uma semana. O método *K Means Clustering* (explicado em 2.2.1) é utilizado de forma a categorizar os vários tipos de dispositivos utilizados. É ainda criado um modelo de previsão de tráfico com base em cadeias de Markov. Este modelo é designado de Modelo de Markov, que é um modelo que possui uma matriz de transição de estados em que esta matriz possui um conjunto de propriedades tais como: número de linhas igual ao número de colunas, a soma de cada linha é sempre igual a 1. Este modelo possui a propriedade de que estando num estado, apenas é possível transitar para um estado contíguo ou manter-se no mesmo.

Os artigos [6] e [8] aproximam-se da solução pretendida, uma vez que nos dois casos há uma caracterização do tráfego e no segundo caso há um modelo de previsão de tráfego. Porém nestes casos os APs são fixos, o que, em caso de eventos do tipo TCE, é desvantajoso face à utilização de UAVs. Isto deve-se ao

facto de que podem existir obstáculos no recinto do evento e a interferência com o campo de visão da multidão. Estes estudos também se distanciam do pretendido uma vez que tratam de casos de redes celulares, sendo que se pretendem redes Wi-Fi.

Existem ainda vários outros estudos acerca deste tema, sendo que o *Clustering* é uma técnica frequentemente utilizada [9][10][11]. A técnica *K Means* é utilizada em [11].

2.2 Clustering

O *Clustering* [12] é uma técnica que tem como função agrupar vários dados segundo o seu grau de semelhança, sendo que cada agrupamento de dados é denominado de *cluster*. Existem vários algoritmos que utilizam esta técnica, sendo que, por norma, cada ponto apenas pode pertencer a um e um só *cluster*. Um dos algoritmos que utiliza esta técnica é o *K Means Clustering*.

2.2.1 K Means Clustering

Este algoritmo [12] possui duas etapas, representadas na figura 2.1. A primeira etapa é a de atribuição ao *cluster*, em que todos os pontos de dados são atribuídos ao *cluster* mais próximo. Esta atribuição é feita tendo em conta a distância euclidiana do ponto ao *cluster*. O número de *clusters* tem de ser indicado pelo utilizador e a posição inicial dos mesmos pode ser aleatória ou também indicada. Na segunda etapa, é calculado o ponto equivalente à distância média de todos os pontos pertencente a esse grupo, designado de centroide. A duas etapas são iterativas, ou seja, é atribuído um valor de cada vez ao *cluster* e o centroide é reajustado para o ponto médio dos pontos pertencentes ao grupo. O número de iterações pode ser indicado pelo utilizador ou deixar a cargo do algoritmo, que irá realizar o número de iterações necessário para que o reajustamento da posição do centroide seja mínimo.

O *K Means Clustering* foi o algoritmo escolhido devido às suas vantagens, uma vez que é relativamente fácil de implementar e é um método rápido ao ser executado. Porém possui a desvantagem de que, inicialmente, é impossível garantir qual é o número ideal de *clusters* para os dados que estejam a ser analisados pelo algoritmo. Apesar de ser possível utilizar o algoritmo utilizando um número qualquer de *clusters*, deve-se utilizar um número que balanceie a complexidade do problema (quanto maior o número de *clusters*, maior será a complexidade) com a precisão da atribuição dos dados aos *clusters*. De forma a obter este valor, são utilizados métodos complementares de forma a garantir que o número de *clusters* é o ideal. Esses métodos são o *Elbow Method* e o *Average Silhouette Method*,entre

outros. Estes métodos devem ser utilizados conjuntamente de forma a garantir que foi escolhido o número de *clusters* certo.

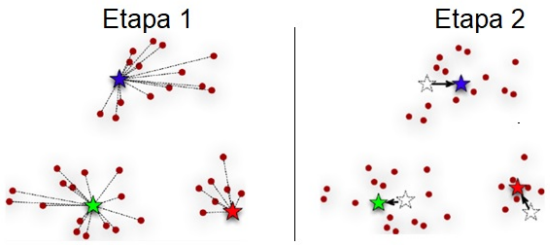


Figura 2.1: Etapas do *K Means Clustering*

2.2.2 Elbow Method

O *Elbow Method* [11] é um método que tem como função encontrar o número ideal de *clusters* a utilizar no *K Means Clustering* ou noutro método de *Clustering*, sendo, por esta razão, um método complementar. Tendo de partida um intervalo de número de clusters, o algoritmo relaciona o WSS (Within Sum of Squares), que é um indicador do número de iterações necessárias e do erro associado à etapa de atribuição do *K means*, com o respetivo número de *clusters*. De modo a encontrar o número ideal, deve-se analisar o gráfico, que tem a forma de um braço, sendo que o número pretendido estará no cotovelo do gráfico, ou *elbow*. Este algoritmo possui as seguintes etapas:

- Colocar o método em funcionamento, variando o valor de *clusters* entre um limite mínimo e máximo, por exemplo entre 1 a 10;
- Para cada valor é calculado o WSS;
- O gráfico que relaciona os dados é gerado, como mostrado na figura 2.2;
- O *elbow* significa o número otimo de *clusters*.

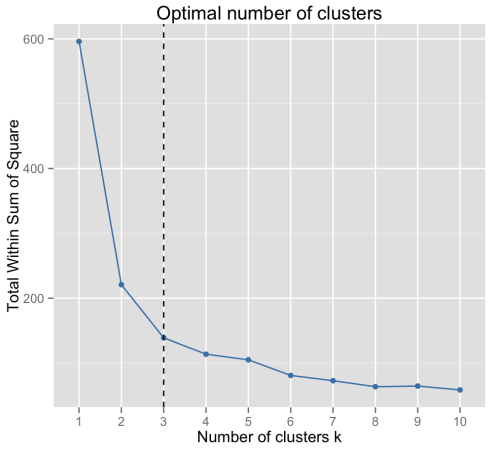


Figura 2.2: Representação gráfica do *Elbow Method*

Este método é de simples interpretação pois apenas é necessário encontrar o *elbow* do gráfico gerado, contudo, existem situações em que este processo não é suficiente para solucionar o problema, pois em alguns casos o *elbow* do gráfico não é percetível ou alonga-se para mais do que um valor, como se pode ver na figura 2.3.

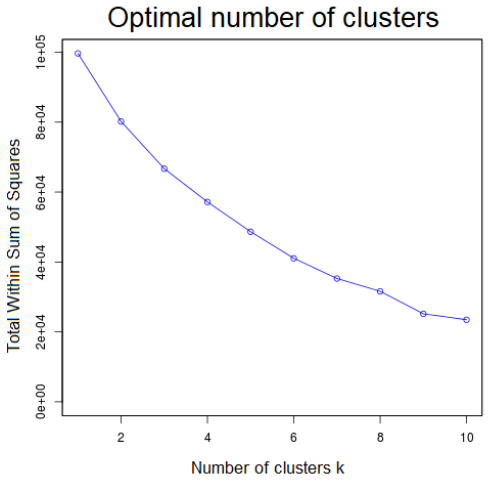


Figura 2.3: Numero ótimo de *clusters* impercetível no *Elbow Method*

2.2.3 Average Silhouette Method

O método *Average Silhouette* [13] tem a mesma função que o *Elbow Method*, sendo também um método apenas complementar. Neste caso, o algoritmo faz uma

avaliação do quão semelhantes são os dados em cada agrupamento de dados. Os resultados deste método variam entre -1 e 1. Os valores próximos de 0 indicam *clusters* sobrepostos. Valores negativos geralmente indicam que uma amostra foi atribuída ao *cluster* errado, devido a existir um *cluster* diferente em que os dados têm mais semelhanças com a amostra. Valores próximos de 1 revelam valores concisos. Relativamente à equação 2.1, para cada ponto p, é encontrada a distância média entre p e todos os outros pontos no mesmo agrupamento (esta medida corresponde a A). Em seguida, é encontrada a distância média entre p e todos os pontos no *cluster* mais próximo (corresponde a B). O coeficiente de silhueta para p é definido como a diferença entre B e A dividido pelo maior dos dois (MAX(A, B)).

$$s(p) = \frac{B(p) - A(p)}{MAX(A(p), B(p))} \tag{2.1}$$

Depois de obter o valor da equação para cada ponto, é feita a média dos valores de forma a obter o valor para cada *cluster*. A representação gráfica deste método encontra-se na figura 2.4.

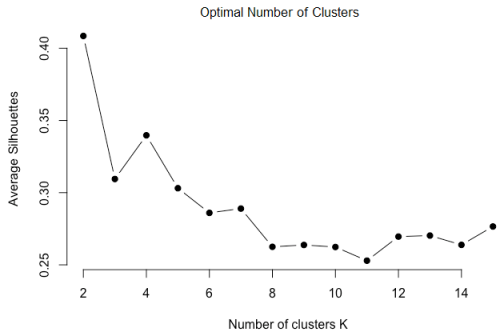


Figura 2.4: Representaçao gráfica do *Average Silhouette Method*

Este algoritmo nem sempre é o ideal para conseguir o valor exato de agrupamentos a utilizar, pois é possível existirem vários valores semelhantes o que torna a escolha difícil. Como tal, deve-se utilizar este algoritmo conjuntamente com o *Elbow Method*, de forma a reduzir a redundância existente tanto neste método como no *Elbow*.

2.3 Janela Deslizante

A janela deslizante é uma técnica que permite considerar apenas uma parte de um vetor em vez de considerar todas as posições do mesmo. Para isso, é definido um tamanho de janela qualquer, e após cada iteração esta janela é movida

para a direita. Isto significa que a primeira iteração terá de ter em conta o primeiro valor do vetor e a última terá em conta o último valor do vetor. Esta técnica é semelhante à utilizada no protocolo TCP (*Transmission Control Protocol*).

Na figura 2.5 encontra-se um exemplo de um vetor com valores inteiros em que está a ser utilizada uma janela deslizante de tamanho três. Neste caso existem 8 iterações no total.

Figura 2.5: Janela Deslizante

2.4 Matriz de transição

Uma matriz de transição [14] é uma matriz que contém as probabilidades de, existindo um espaço em que certo dado se encontra num estado qualquer, transitar para outro estado ou manter-se no mesmo. Isto significa que as linhas correspondem ao estado de origem e as colunas ao estado de destino. Esta matriz tem como propriedades ser sempre quadrada pois possui número de linhas e colunas iguais ao número de estados existentes e ter soma igual a 1 em todas as linhas. Um exemplo de uma matriz de transição encontra-se na figura 2.6. Esta matriz será utilizada para estudar a movimentação dos utilizadores entre zonas, sendo que a cada estado corresponderá um *cluster*.

$$P = (p_{ij}) = \begin{bmatrix} p_{00} & p_{01} & \cdots & p_{0N} \\ p_{10} & p_{11} & \cdots & p_{1N} \\ \cdots & \cdots & \cdots & \cdots \\ p_{N0} & p_{N1} & \cdots & p_{NN} \end{bmatrix}$$

Figura 2.6: Matriz de transição

2.5 Python

Python é uma linguagem de programação *open source* interpretada de alto nível, criada por Guido van Rossum e lançado pela primeira vez em 1991. Esta linguagem possui gestão automática de memória e suporta múltiplos paradigmas de programação, incluindo orientação a objetos, imperativos, funcionais e processuais, e possui uma biblioteca ampla e abrangente.

Tem como vantagens ter um código de fácil leitura, uma vasta comunidade de utilizadores e uma biblioteca diversificada.

O Python é usado em várias áreas:

- Desenvolvimento Web;
- Computação científica e numérica;
- Educação;
- *Graphic User Interfaces*;
- Desenvolvimento de software;
- Negócios e comércio.

2.6 Conclusão

Neste capítulo é feita uma contextualização do cenário que se pretende estudar, sendo feita uma análise a artigos de estudo a contextos semelhantes.

São ainda introduzidas os conceitos que são utilizados nesta Tese, tais como o *clustering*, através do algoritmo *K Means* e das suas técnicas complementares, a janela deslizante e as matrizes de transição.

É feita uma breve descrição à linguagem utilizada para a aplicação a desenvolver.

Este capítulo é muito útil para a familiarização das ferramentas mais utilizadas no contexto pretendido. Do ponto de vista da solução do problema, esta passa por criar uma aplicação em *Python*, utilizando as ferramentas populares, que satisfaça os seguintes requisitos:

- Dados de entrada: Deverão ser lidos dados de entrada que incluirão os locais em que os utilizadores se encontram ao longo do tempo e o tráfego médio gerado, bem como os limites do recinto, o tamanho da janela deslizante a utilizar para efeitos de previsão, o número de grupos pretendidos e o número do utilizador do qual se pretende visualizar o respetivo gráfico;
- Divisão dos dados em grupos: De maneira a facilitar a interpretação, a organização e a previsão dos dados, estes devem ser incluídos em grupos. Neste caso será usado o algoritmo *K Means*;
- Matriz de transição: devem ser criadas matrizes de previsão no global e por utilizador, serão para o efeito utilizadas matrizes de transição de estados;
- Previsão de dados: a aplicação deve conseguir prever dados futuros;
- Comparação da previsão: Devem ser geradas formas de comparar os dados de previsão com os dados que realmente foram introduzidos, em forma de erro associado;
- Representação gráfica: De forma a poder comparar os valores reais e de previsão, serão gerados gráficos em que os valores reais e os valores de previsão poderão ser comparados.

Capítulo 3

Implementação

Neste capítulo é descrito o modelo do sistema. É também explicado como foram usadas as principais técnicas e o algoritmo de previsão.

3.1 Modelo do sistema

O sistema criado é uma ferramenta de análise e previsão de mobilidade dos utilizadores e de tráfego em cada zona do recinto e possui 3 aplicações: 1 principal e 2 complementares. Tal é necessário devido ao *K Means* necessitar de valores quanto ao número de *clusters* a utilizar. Para além da aplicação principal que contém todos os requisitos necessários para este trabalho, foram criadas duas aplicações complementares: uma para o *Elbow Method* e outra para o *Average Silhouettes Method*. Isto deve-se ao facto de que estes métodos complementares podem ter valores redundantes e para isso é necessário, previamente ao início da aplicação principal, a interpretação do utilizador para escolher o número de *clusters* ótimo, recorrendo sempre à interpretação dos dados destes métodos complementares. Para isso, é necessária a compreensão destes métodos (explicados nas subsecções 2.2.2 e 2.2.3) de forma a decidir se algum destes métodos falhou. Apesar de estes valores serem importantes para a aplicação principal, é possível escolher qualquer valor para o número de *clusters*. A aplicação irá funcionar independentemente do número de *clusters* ser ótimo, porém não utilizar o valor ótimo terá impacto nos resultados. Todos os valores utilizados nestas aplicações são índices, tornando-as aplicações genéricas e adaptáveis a qualquer contexto.

As aplicações complementares possuem apenas duas etapas, conforme se pode confirmar pela figura 3.1:

- Leitura de dados: As aplicações necessitam dos limites inferior e superior

do recinto (dado lido através da linha de comandos) e das posições dos utilizadores nos diversos instantes em que for feita a leitura de dados (dado lido através de um ficheiro de texto). Isto deve-se ao facto de que podem existir valores de utilizadores fora deste recinto ligados à Internet, que estejam, por exemplo, na fila para a entrada. Estas aplicações apenas tratam os valores de utilizadores dentro do recinto, sendo posteriormente gerado um novo *cluster* caso existam utilizadores fora do recinto, apenas considerando esses utilizadores;

- Avaliação de dados: Os dados são avaliados conforme a aplicação escolhida, sendo o resultado o número ótimo de *clusters* do ponto de vista do método.

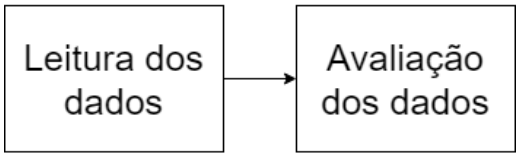


Figura 3.1: Etapas das aplicações complementares

Quanto à aplicação principal, as suas etapas encontram-se na figura 3.2.

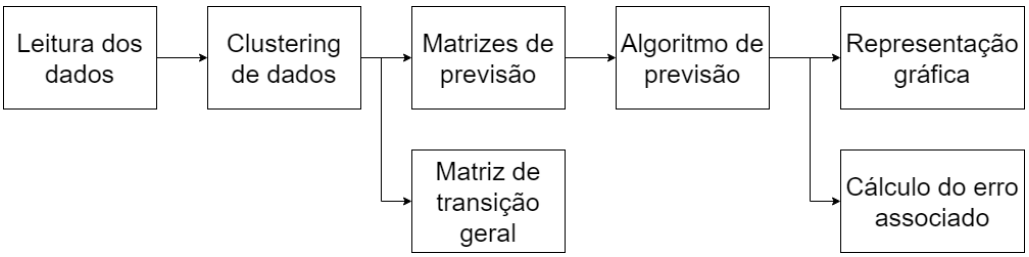


Figura 3.2: Etapas da aplicação principal

A leitura de dados inclui os seguintes parâmetros: posições dos utilizadores e tráfego médio gerado por cada utilizador, limites do recinto, tamanho da janela deslizante, número de grupos pretendidos e número dos utilizadores dos quais se pretende visualizar o respetivo gráfico. As posições dos utilizadores podem ser contínuas ou discretas, porém em todos os exemplos estas foram sempre consideradas como discretas (isto é, existe sempre uma distância D entre cada posição). Os dados das posições dos utilizadores são relativos aos vários instantes de tempo, tendo entre si intervalos regulares. O primeiro instante corresponde ao instante inicial, 0, e a partir daí correspondem a T, 2T, até ao instante final, nT, sendo que n corresponde ao total de instantes do contexto. No caso dos limites do recinto,

o sistema considerará recintos quadrados, com limite mínimo e máximo, sendo que as distâncias reais podem ser calculadas através da fórmula 3.1 , em que D corresponde às distâncias consideradas pelo sistema, i ao índice e d às distâncias reais (como por exemplo as coordenadas (1,1) e (1,2) do sistema, com 2 m de índice, possuem 1 de distância entre si em relação ao sistema e 2 m de distância real. São utilizadas distâncias euclidianas.

$$d = i * D \tag{3.1}$$

Será feito o agrupamento de dados em *clusters* (*clustering*), através da técnica *K Means*, sendo que o número de grupos pretendidos será o correspondente aos resultados das técnicas complementares.

Serão criadas várias matrizes: uma que inclui todos os utilizadores e todos os instantes (matriz de transição geral), outra que inclui todos os utilizadores mas que considera apenas um intervalo de instantes de tempo, sendo que este intervalo corresponde ao tamanho da janela deslizante (matriz de previsão geral) e por fim, uma matriz por cada utilizador considerando apenas um intervalo de instantes de tempo (matriz de previsão por utilizador).

De forma a poder comparar qualitativamente os valores reais dos valores de previsão, estes serão feitos através do cálculo do erro associado.

A representação gráfica serve para comparar os dados reais dos dados previstos, tanto de utilizadores como dos vários *clusters*. Os gráficos por utilizador são referentes à sua mobilidade, enquanto que os gráficos por *cluster* são relativos ao número de utilizadores (correspondente ao somatório de utilizadores num dado instante) e ao tráfego gerado nessa zona (correspondente ao somatório de tráfego gerado por cada utilizador). No caso do tráfego gerado por *cluster*, o tráfego gerado por cada utilizador será considerado constante (tráfego médio) e analisado como sendo o somatório do tráfego gerado pelos utilizadores que estão naquele *cluster*. Apenas os dados da mobilidade dos utilizadores é prevista, sendo que esses dados são utilizados para os cálculos de cada *cluster*.

3.2 Dados de entrada

Os dados de entrada são inseridos através de argumentos na linha de comandos e de um ficheiro de texto. Através da linha de comandos são inseridos os seguintes argumentos:

- Número de *clusters* (*n_clusters*): número de *clusters* que se pretende que o algoritmo obtenha;

- Utilizadores (*users_plot*): Apesar do código executado gerar valores para todos os utilizadores, é possível selecionar quais se pretende visualizar o respetivo gráfico de mobilidade;
- Limites (*limits*): Este argumento representa os limites inferior e superior. São considerados apenas recintos quadrados. Caso existam valores de posições fora do recinto, é criado um *cluster* extra apenas para estes casos;
- Janela deslizante (*window*): tamanho da janela deslizante.

Um exemplo da sintaxe destes dados de entrada é o seguinte:

-n_clusters 5 -users_plot 0 3 19 -limits 0 9 -window 20

Através do ficheiro de texto são lidas as posições dos utilizadores num vetor por utilizador e o tráfego médio gerado durante todo o evento por cada utilizador. Através deste ficheiro são também lidos o número de utilizadores e o número de dados obtidos. Os dados de um utilizador encontram-se em apenas uma linha e cada linha representa o número do utilizador (e.g. a linha 1 corresponde ao utilizador 1). Os dados das posições e os dados do tráfego são separados com o símbolo ;. No caso da mobilidade, cada par de coordenadas (x,y) (estes valores são índices relativos à posição de cada utilizador, dependendo do valor real do tamanho do recinto) corresponde ao instante de tempo correspondente, ou seja, o primeiro valor corresponde ao instante inicial, o segundo valor corresponde ao instante 1, e assim sucessivamente. Um exemplo de uma linha deste ficheiro é:

8,16 8,17 2,17 18,9 14,7 2,8 3,1 25,3 25,3 25,3;5

3.3 Clustering de dados

O algoritmo *K Means* é responsável por agrupar os dados referentes às posições dos utilizadores em *clusters*, e funciona conforme descrito em 2.2.1. Nesta fase, são apenas considerados os dados que estão dentro dos limites inseridos nos dados de entrada. Os dados de entrada são comparados com os limites introduzidos previamente e, caso hajam valores fora desses limites, é criado mais um *cluster*, recorrendo mais uma vez ao *K Means* de forma a que este gere apenas um *cluster* que agrupa apenas os dados que não pertençam aos limites introduzidos. Os dados que pertencem ao recinto, ou seja, que estão dentro dos limites previamente inseridos, são agrupados num número de *clusters* correspondente ao valor inserido inicialmente. É importante também salientar que neste caso os dados que geram os *clusters* são utilizados independentemente do tempo em que se encontram, gerando desta forma áreas fixas para que seja mais fácil gerar o algoritmo de previsão. É então gerado um gráfico que mostra apenas os dados dentro dos limites, como é exemplificado na figura 3.3.

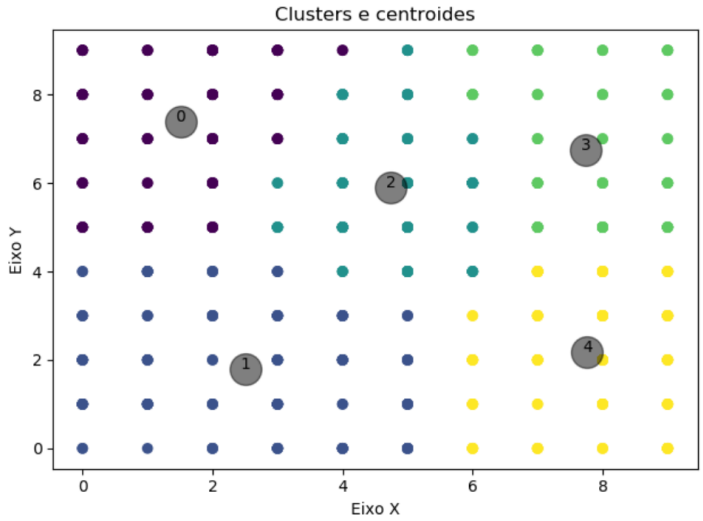


Figura 3.3: Gráfico de clusters e centroides

As posições de cada utilizador são traduzidas pelo respetivo *cluster* em que as posições se inserem. São tambem mostrados os centroides de cada *clusters* bem como os *clusters* em que cada utilizador esteve (no caso ilustrativo existe o *cluster* extra referente a utilizadores fora do recinto, o *cluster* 5), como por exemplo:

Centroides dos clusters:

- *Cluster* 0: [1.51 7.39];
- *Cluster* 1: [2.51 1.8];
- *Cluster* 2: [4.74 5.9];
- *Cluster* 3: [7.74 6.75];
- *Cluster* 4: [7.76 2.17];
- *Cluster* 5: [5.14 4.15].

Clusters em que cada um dos utilizadores esteve, por ordem cronológica (dados reais):

- Utilizador 0: [4, 4, 4, 2, 4, 2, 2, 2, 3, 3, 5, 3, 2, 0, 2, 2, 3, 4, 3, 5, 5, 4, 4, 1, 1, 4, 1, 2, 1, 2, 2, 1, 1, 5, 5, 1, 1, 1, 4, 4]
- Utilizador 19: [4, 2, 0, 2, 2, 2, 3, 2, 2, 0, 0, 0, 0, 5, 0, 2, 3, 2, 2, 3, 5, 5, 3, 3, 3, 3, 2, 3, 3, 3, 3, 5, 5, 5, 5, 4, 4, 4, 4, 5]

3.4 Matriz de transição geral

A matriz de transição geral é calculada tendo em conta os dados de todos os utilizadores, sendo esta quadrada de tamanho variável consoante o número de *clusters*. Cada linha desta matriz tem somatório igual a 1. Engloba todas as posições de todos os utilizadores em todos os instantes. Esta matriz é calculada com base nos *clusters* de cada utilizador. Cada linha representa o *cluster* em que o utilizador se encontra e cada coluna representa o *cluster* para o qual o utilizador se irá deslocar, ou seja, cada valor da matriz M(i,j) representa a probabilidade de transição do *cluster* i para o *cluster* j. Por exemplo, existindo 3 *clusters* (de 0 a 2) e um utilizador que se desloca do *cluster* 0 ao cluster 2, a matriz será atualizada na linha 0 e na coluna 2. No fim do processo, cada valor é dividido pelo somatório de cada linha da matriz, originando assim probabilidades de transição. Um exemplo de uma matriz (em que o cluster é designado por c) é o seguinte:

$$\begin{array}{c} \\ c_0 \\ c_1 \\ c_2 \end{array} \begin{array}{c} \begin{array}{ccc} c_0 & c_1 & c_2 \end{array} \\ \begin{bmatrix} 0.25 & 0.25 & 0.5 \\ 0.45 & 0.25 & 0.3 \\ 0.5 & 0.45 & 0.05 \end{bmatrix} \end{array}$$

Utilizando a primeira linha da matriz como referência, primeiramente é constatado que, por exemplo, existiram dois utilizadores a deslocarem-se do *cluster* 0 para o *cluster* 2, um utilizador que se desloca do *cluster* 0 para o *cluster* 1 e outro utilizador que está no *cluster* 0 e que se mantém no mesmo. A primeira linha da matriz é representada por: [1 1 2]. De seguida, a matriz é dividida pelo somatório da linha (que é igual a 4). O resultado dessa linha é: [0.25 0.25 0.5]. Na figura 3.4 encontra-se um fluxograma representativo do cálculo desta matriz. Esta matriz não é utilizada para efeitos de previsão, sendo utilizada apenas para ser possível estudar o comportamento geral dos utilizadores ao longo de todos os instantes e ser possível comparar com os valores de previsão obtidos.

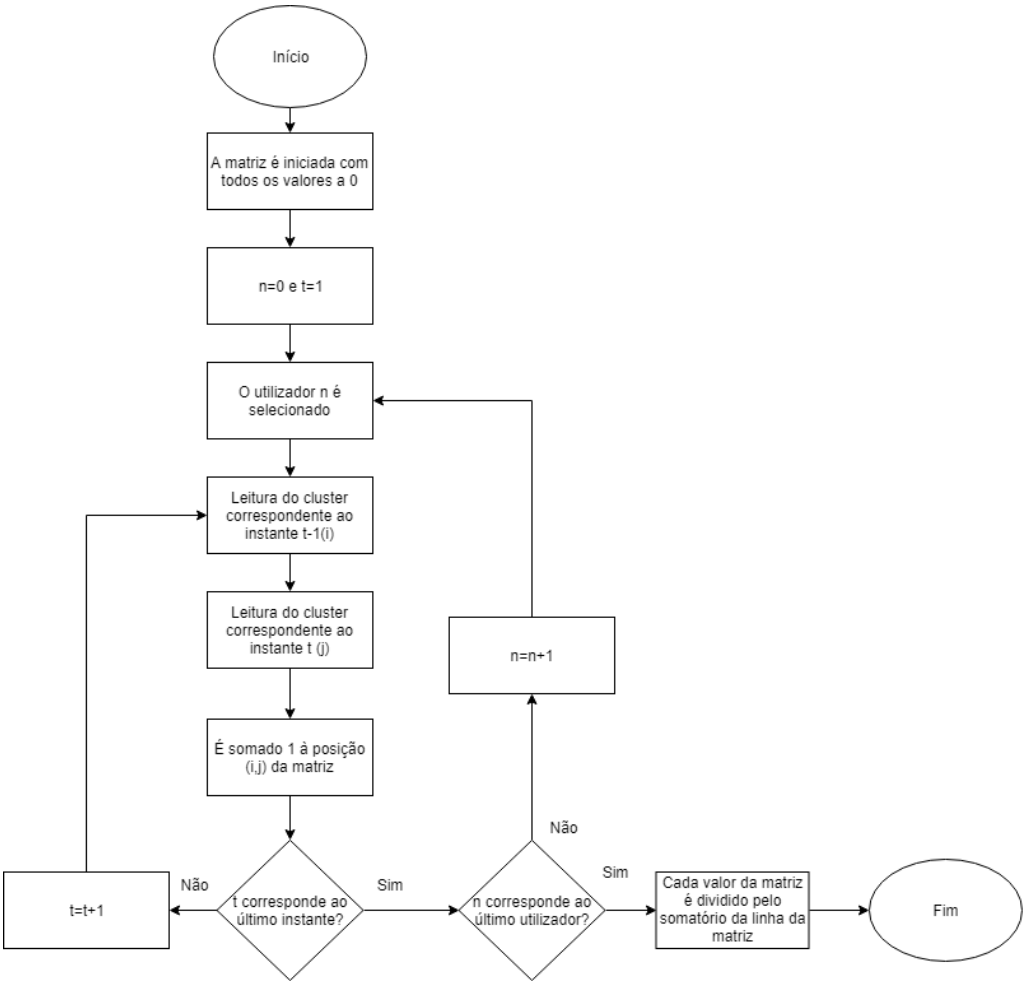


Figura 3.4: Fluxograma da matriz de transição geral

3.5 Matrizes de previsão

As matrizes utilizadas para efeitos de previsão são geradas de forma semelhante à da matriz de transição geral, a diferença reside no facto de que estas matrizes consideram apenas os dados do tamanho da janela deslizante em vez de considerarem todos os instantes. Estas matrizes são constantemente atualizadas com o decorrer do evento. Por exemplo, se o tamanho da janela deslizante for de 3, esta irá utilizar os valores dos instantes 0, 1 e 2 para prever o *cluster* em que o utilizador estará no instante 3. Para prever o valor do instante 4, esta matriz é atualizada utilizando os valores dos instantes 1, 2 e 3. O valor correspondente a 3 é referente à zona em que o utilizador realmente esteve, não considerando valores de previsão. Isto porque caso o algoritmo de previsão falhe num instante, esse valor iria afetar não só a qualidade do mesmo nesse instante, mas também iria

gerar problemas nos valores seguintes. Existem dois tipos de matrizes de previsão: uma geral e outra por utilizador. A diferença é que a matriz geral utiliza os valores de todos os utilizadores para a janela deslizante pretendida, enquanto que a matriz por utilizador utiliza os valores de apenas um utilizador para a janela pretendida. O fluxograma do caso da matriz geral de previsão encontra-se na figura 3.5.

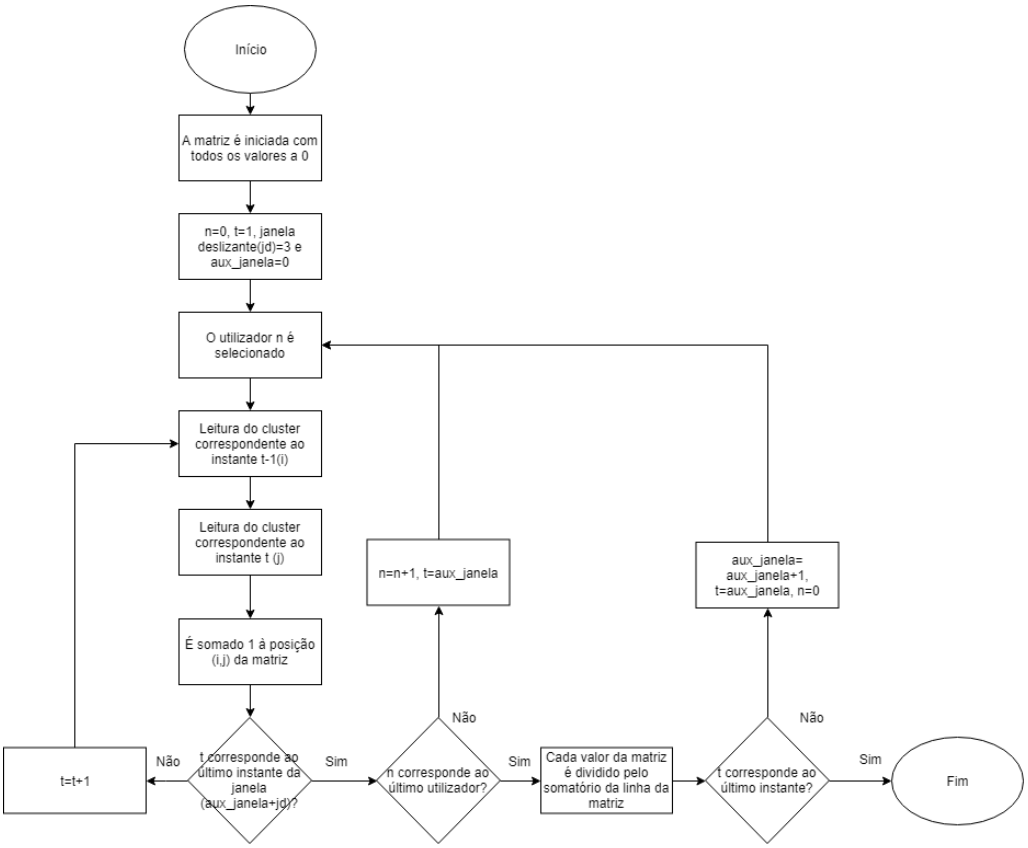


Figura 3.5: Fluxograma da matriz de previsão geral

3.6 Algoritmo de previsão

O algoritmo de previsão recorre na maior parte das vezes à matriz de previsão de cada utilizador, reservando a matriz de previsão geral para casos excecionais. Estes casos correspondem a situações em que a matriz do utilizador possui uma linha da matriz com todos os valores a zero (por exemplo, se um utilizador se encontrar no *cluster* 0 e a primeira linha da matriz ter um somatório igual a 0) uma vez que até àquele instante ainda nunca se deslocou do *cluster* em que se encontra no instante correspondente. Nesse caso a matriz de previsão geral é utilizada, recorrendo ao comportamento de todos os utilizadores. Caso as duas

matrizes possuam essa linha a zeros, é gerado um número inteiro aleatório de 0 até ao número de *clusters*. O método começa a prever no instante correspondente ao tamanho da janela deslizante, servindo-se, portanto, das primeiras matrizes de transição geradas. Por exemplo, se o tamanho da janela for de 3, o primeiro valor que o algoritmo irá prever será o correspondente ao instante 3, utilizando as matrizes de previsão que contêm os valores dos instantes 0,1 e 2. No instante seguinte, irá prever o instante 4 através das matrizes de previsão que contêm os valores dos instantes 1,2 e 3. A cada instante que é necessário prever, este algoritmo utiliza as matrizes de previsão correspondentes, mantendo-se assim a par das atualizações destas matrizes.

O algoritmo recorre à última posição prevista para ter referência da zona em que o utilizador se encontra, à exceção de quando este faz pela primeira vez a previsão, neste caso recorrendo a um valor real. A linha da matriz referente ao número do *cluster* em que o utilizador se encontra é utilizada para saber as probabilidades de transição desse *cluster*. É então gerado um numero aleatório de 0 a 1 e comparado com os valores da linha em causa e decidido o *cluster* para o qual o utilizador previsivelmente irá. Esta decisão é feita tomando as probabilidades de cada linha como intervalos. Utilizando o exemplo de o utilizador se encontrar no *cluster* 0 e essa linha da matriz conter os valores [0.25 0.25 0.5] e tendo um número gerado igual a 0.49, este valor será comparado com os intervalos [0;0,25] (intervalo referente ao utilizador se dirigir para o cluster 0),]0,25;0,5] (intervalo referente ao utilizador se dirigir para o cluster 1), e]0,5;1] (intervalo referente ao utilizador se dirigir para o cluster 2), sendo que neste caso, seguindo a lógica do algoritmo, o utilizador irá deslocar-se para o *cluster* 1.

É importante realçar que cada instante corresponde a cada leitura de dados, que pode ter qualquer intervalo de tempo consoante o contexto. Os *clusters* previstos em que cada um dos utilizadores esteve, por ordem cronológica (dados de previsão) são:

- Utilizador 0: [4, 4, 4, 2, 4, 2, 2, 2, 3, 3, 5, 3, 2, 0, 2, 2, 3, 4, 3, 5, 3, 5, 5, 3, 3, 5, 5, 4, 3, 5, 4, 3, 5, 4, 4, 3, 4, 4, 1, 2]
- Utilizador 19: [4, 2, 0, 2, 2, 2, 3, 2, 2, 0, 0, 0, 0, 5, 0, 2, 3, 2, 2, 3, 2, 2, 3, 5, 5, 5, 0, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 2]

É então gerado o gráfico dos dados reais (correspondente à representação gráfica dos *clusters* de cada utilizador em 3.3) e dos dados de previsão da mobilidade do utilizador, como mostrado na figura 3.6. A linha vertical corresponde ao último valor real tido em conta no caso da previsão, sendo que até esse valor as duas representações são idênticas.

No caso mostrado na figura, existem 40 instantes (do 0, estado inicial, ao 39, estado final) sendo que o instante 20 é o primeiro a ser previsto.

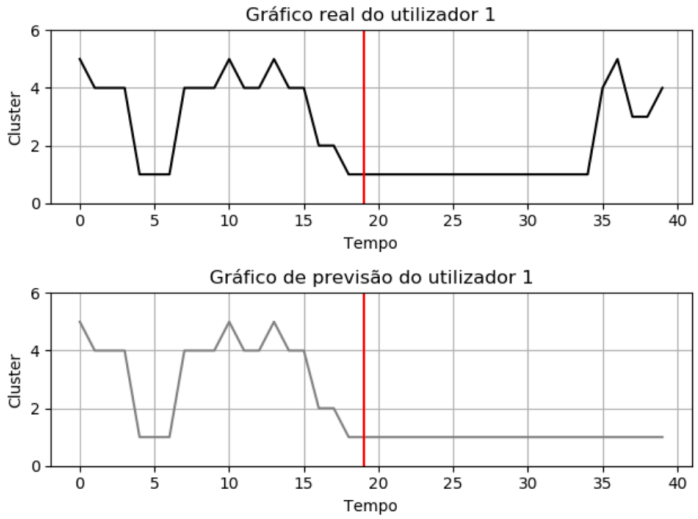


Figura 3.6: Gráfico de mobilidade do utilizador

Através destes dados, são também gerados gráficos em relação aos *clusters* existentes: gráfico em relação ao número de utilizadores e ao tráfego gerado pelos utilizadores num determinado instante. Estes gráficos são obtidos com base nos valores previstos da mobilidade. Em relação ao gráfico do número de utilizadores por zona, é feito o somatório dos utilizadores que se encontram em cada instante para cada *cluster*, através dos dados reais e dos dados de previsão, dando assim uma perspetiva ao nível de cada zona. É ainda calculado o tráfego gerado em cada instante, considerando os valores do tráfego médio dos utilizadores que se encontram em cada *cluster*. Por exemplo, se apenas o utilizador 0 e 19 se encontrarem no *cluster* 0 num dado instante, e tiverem valores de tráfego de, por exemplo, 1 e 2 Mbit/s, então nesse instante existem dois utilizadores nessa zona e o tráfego é de 3 Mbits/s. Estes gráficos encontram-se ilustrados na figura 3.7.

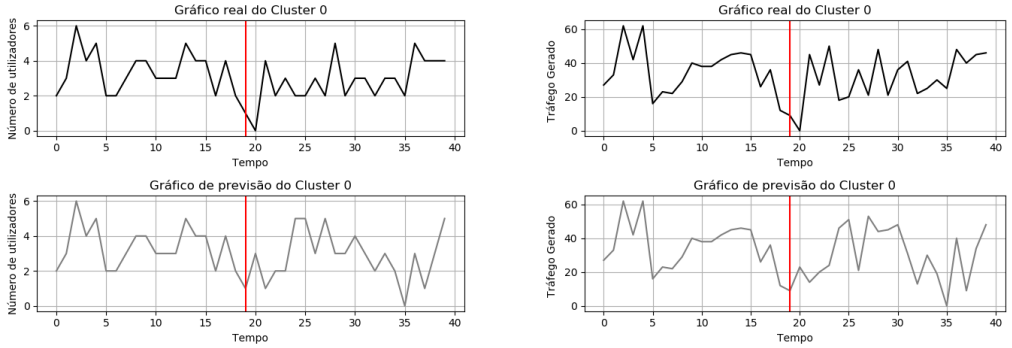


Figura 3.7: Gráfico do número de utilizadores e do tráfego gerado num *cluster*

3.7 Erro de previsão

O erro de previsão é obtido através da seguinte fórmula:

$$E(t) = \frac{|C_r(t) - C_p(t)|}{|C_r(t) - C_d(t)|} \tag{3.2}$$

Os dados representam:

- C_{r}(t): Coordenadas do centroide do cluster real para onde o utilizador se moveu no instante t;
- C_{p}(t): representa as coordenadas do centroide do *cluster* previsto;
- C_{d}(t): representa as coordenadas do centroide do *cluster* mais distante que o utilizador se pode deslocar naquele instante, a partir do *cluster* real onde este se encontra. Este valor é verificado através das matrizes de previsão;
- t: Instante de tempo da previsão;
- E(t): Erro de previsão no instante t.

Esta fórmula é utilizada para cada instante, sendo depois calculada a média do erro de todos os instantes para cada utilizador. É também calculado o erro global do algoritmo, que é calculado através da média do erro de todos os utilizadores.

3.8 Conclusão

Neste capítulo é apresentado o modelo do sistema, através da apresentação e da contextualização de todas as etapas necessárias. É apresentada a forma como os dados de entrada devem ser inseridos no sistema, através de uma descrição detalhada de cada argumento necessário. São ainda apresentadas as várias matrizes utilizadas e é explicada a forma como estas são obtidas. O algoritmo de previsão é descrito e é mostrado o *output* gerado, tanto a nível de gráficos por utilizador como a nível de gráficos por *cluster*. Por fim, a forma como o erro é calculado no sistema é elucidada.

Capítulo 4

Testes, demonstração e análise de resultados

Neste capítulo são feitos testes com as aplicações desenvolvidas, através de um caso de simulação criado. Os resultados são ainda demonstrados e analisados.

4.1 Caso de simulação

O caso de simulação consiste num exemplo de um festival de música, sendo que os dados são obtidos de 30 em 30 minutos e que o festival possui 4,5 horas, o que significa que existem 10 valores a ser lidos por utilizador. O universo de utilizadores é de 10. O índice do tráfego gerado é de 1 Mbit/s. O recinto é quadrado e possui 100 m de lado, sendo dividido em 20 posições tanto em comprimento como em largura (da posição (0,0) à posição (19,19)), ou seja, o índice é de 5 m, e tem as seguintes propriedades:

- Palco: retangular de comprimento de 50 m e de largura de 10 m. É delimitado no eixo dos X pelas posições 5 a 15 e no eixo dos Y pela posições 17 a 19;
- Barracas de comida: retangular de comprimento de 10 m e de largura de 25 m. É delimitado no eixo dos X pelas posições 0 a 2 e no eixo dos Y pela posições 5 a 10;
- Barracas de bebida: retangular de comprimento de 10 m e de largura de 20 m. É delimitado no eixo dos X pelas posições 17 a 19 e no eixo dos Y pela posições 6 a 10;

- WC (*Water Closet*): retangular de comprimento de 50 m e de largura de 5 m. É delimitado no eixo dos X pelas posições 0 a 10 e no eixo dos Y pela posições 0 a 1;
- Entrada: é feita através da posição 19 no eixo X e 5 no eixo Y.

A figura 4.1 representa o recinto que é utilizado para esta abordagem. É importante realçar que a figura não se encontra à escala.

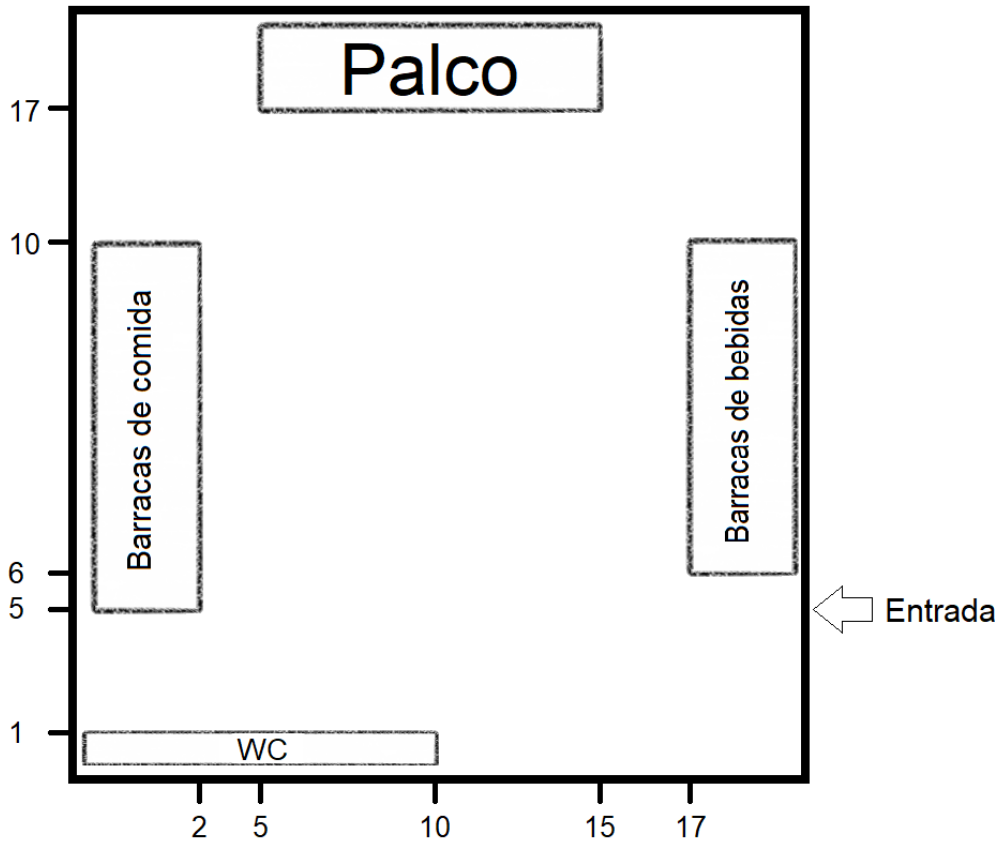


Figura 4.1: Recinto do festival

4.2 Dados introduzidos

Os dados introduzidos, assunto clarificado em 3.2, obtidos através de um ficheiro de texto, são:

8,16 8,17 2,17 18,9 14,7 2,8 3,1 25,3 25,3 25,3;5 (correspondente aos dados do utilizador 0)

12,16 9,17 18,8 14,8 2,9 6,0 25,3 25,3 25,3 25,3;20

3,8 2,7 18,7 5,15 17,14 5,1 18,7 14,17 12,7 25,3;50

18,8 10,17 10,17 8,5 1,6 18,9 9,16 11,9 19,0 25,3;15

4,1 2,11 18,6 6,17 14,17 13,15 8,9 10,0 16,10 18,3;25

21,3 10,1 3,6 7,10 9,11 9,12 9,1 8,7 18,9 25,3;45

21,2 18,7 14,8 10,9 2,7 1,9 12,6 12,17 13,17 10,3;10 (correspondente ao dados do utilizador 6)

22,4 18,8 0,0 5,5 10,10 7,17 16,16 18,5 15,0 18,10;30 (correspondente aos dados do utilizador 7)

23,3 13,0 4,4 2,10 3,8 5,8 10,7 18,6 15,14 2,0;40

24,3 21,2 18,9 15,14 10,9 3,7 1,0 2,0 19,0 16,1;5

O limite do recinto inferior é 0 e o superior é 19. Estes dados são testados primeiramente pelo método *elbow* e *average silhouettes*. Os resultados encontram-se nas figuras 4.2 e 4.3.

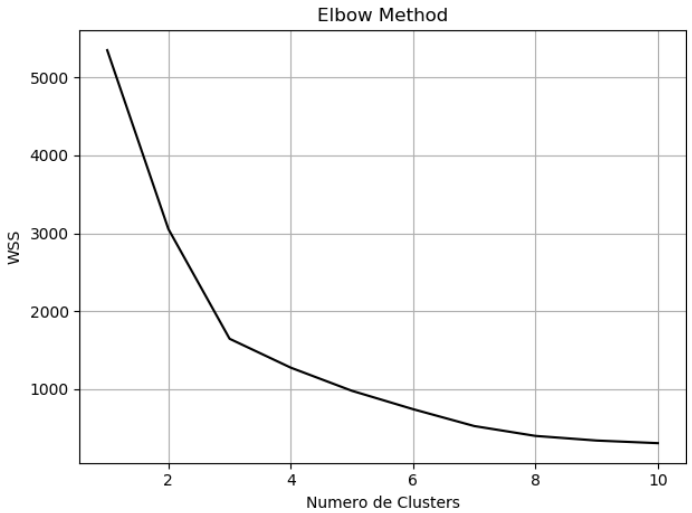


Figura 4.2: *Elbow Method*

Analisando os 2 métodos conjuntamente (explicado em 2.2.2 e 2.2.3) , é possível verificar que o método *Elbow* falha ao ler os valores, e para o método *Average Silhouette* o número ideal de *clusters* é igual a oito. A razão pela qual o método *Elbow* falhou é que após o valor do *elbow* há uma grande variabilidade dos valores, devendo estes valores estabilizarem após o número ótimo. Serão então considerados oito *clusters*. Os dados de entrada pela linha de comandos são:

-n_clusters 8 -users_plot 0 6 7 -limits 0 19 -window 6

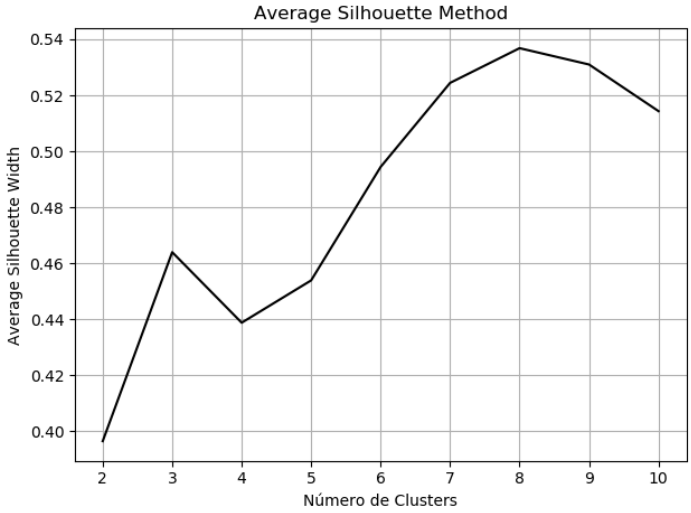


Figura 4.3: *Average Silhouette Method*

4.3 Execução do K Means

O algoritmo *K Means* gera o gráfico de *clusters* e centroides que se encontra na figura 4.4.

Os centroides dos *clusters* têm as seguintes coordenadas:

- *Cluster* 0: [2.67 7.53]
- *Cluster* 1: [2.88 0.38]
- *Cluster* 2; [7.4 16.6]
- *Cluster* 3: [9.54 8.4]
- *Cluster* 4: [10.4 1.]
- *Cluster* 5: [14.1 15.7]
- *Cluster* 6: [17.25 0.25]
- *Cluster* 7: [17.26 7.68]
- *Cluster* 8: [23.88 2.94] (*cluster* extra)

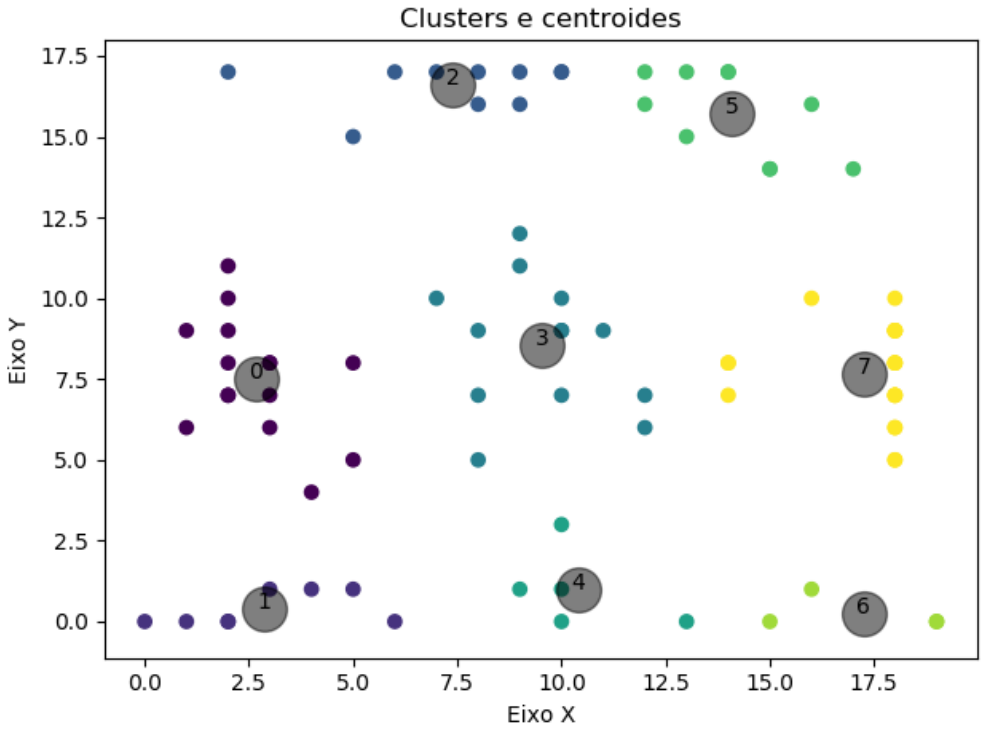


Figura 4.4: *K Means*

4.4 Matriz geral de transição

A matriz geral, explicitada em 3.4, é a seguinte:

	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8
c_0	0.33	0.2	0.	0.27	0.	0.	0.	0.2	0.
c_1	0.29	0.14	0.	0.	0.	0.	0.14	0.14	0.29
c_2	0.	0.	0.3	0.2	0.	0.3	0.	0.2	0.
c_3	0.23	0.	0.077	0.15	0.15	0.077	0.077	0.15	0.077
c_4	0.5	0.	0.	0.25	0.	0.	0.	0.25	0.
c_5	0.	0.2	0.1	0.3	0.1	0.2	0.	0.1	0.
c_6	0.	0.	0.	0.	0.	0.	0.33	0.33	0.33
c_7	0.12	0.059	0.24	0.059	0.	0.18	0.059	0.24	0.056
c_8	0.	0.	0.	0.	0.18	0.	0.	0.27	0.55

4.5 Gráficos por utilizador

Os gráficos dos utilizadores traduzem os valores reais (explicados na secção 3.3) e os valores de previsão obtidos (explicados na secção 3.6). Como exemplo são considerados os utilizadores 0, 6 e 7. Os valores reais destes utilizadores, por ordem cronológica, são:

- Utilizador 0: [2, 2, 2, 7, 7, 0, 1, 8, 8, 8];
- Utilizador 6: [8, 7, 7, 3, 0, 0, 3, 5, 5, 4];
- Utilizador 7: [8, 7, 1, 0, 3, 2, 5, 7, 6, 7].

Os valores de previsão (sendo que o primeiro valor previsto corresponde ao instante 6) destes utilizadores são:

- Utilizador 0: [2, 2, 2, 7, 7, 0, 0, 1, 8, 8];
- Utilizador 6: [8, 7, 7, 3, 0, 0, 0, 3, 5, 5];
- Utilizador 7: [8, 7, 1, 0, 3, 2, 5, 5, 7, 6].

Os gráficos destes utilizadores encontram-se nas figuras 4.5, 4.6 e 4.7. Através destes gráficos consegue-se perceber bastantes semelhanças, porém os valores possuem um desfasamento de 1 ou de 2 instantes de tempo. Utilizando como exemplo o utilizador 1 e os instantes 6 e 7, este desfasamento deve-se ao facto de que o algoritmo só considera a mudança de *cluster* do 0 para o 1 no instante 7, pois quando este está a prever a posição do utilizador no instante 6 este considera os instantes de 0 a 5.

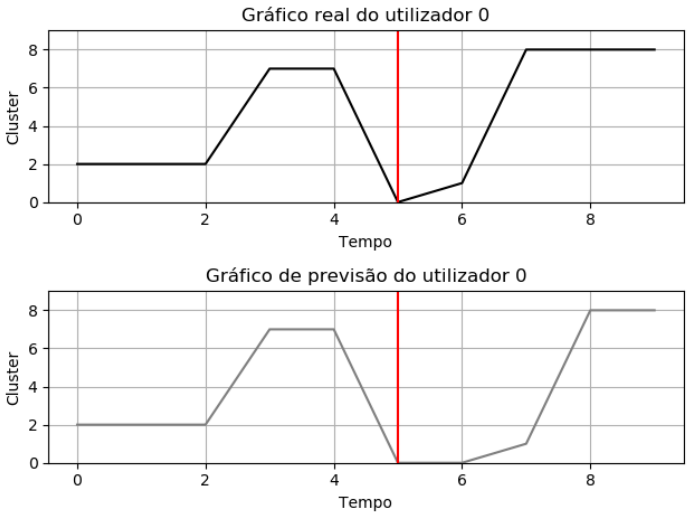


Figura 4.5: Gráfico Utilizador 0

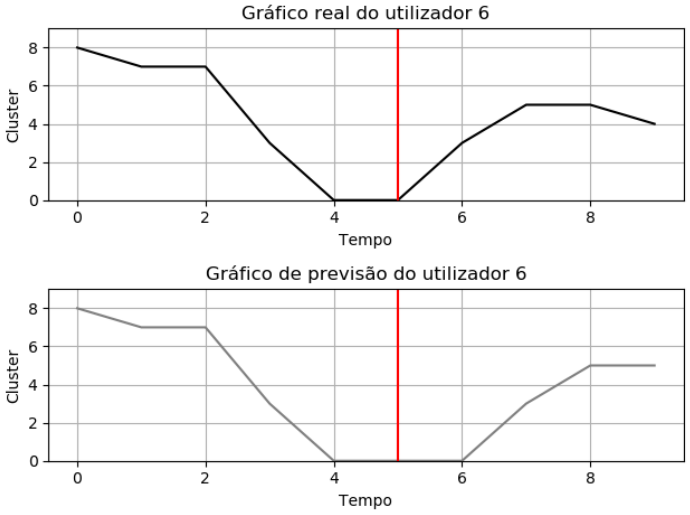


Figura 4.6: Gráfico Utilizador 6

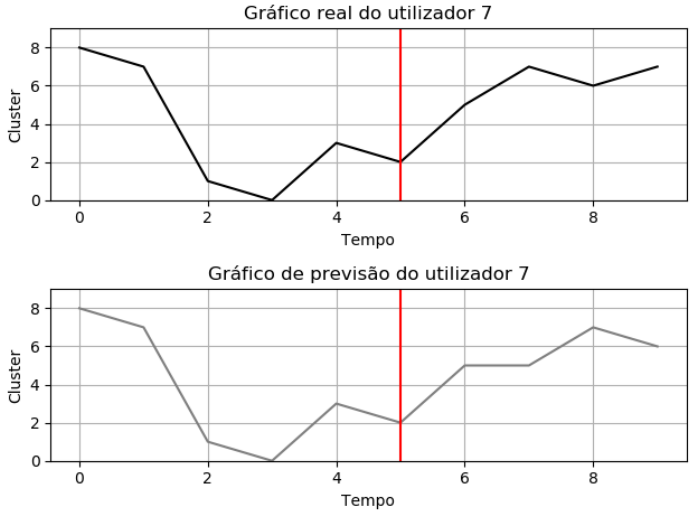


Figura 4.7: Gráfico Utilizador 7

4.6 Gráficos por cluster

Os gráficos por *cluster* subdividem-se em dois tipos, de acordo com o que foi descrito na secção 3.6. Os exemplos destes gráficos serão os dos *clusters* 2, 5, 6 e 8.

4.6.1 Por número de utilizadores

Os gráficos por número de utilizadores encontram-se nas figuras 4.8, 4.9, 4.10 e 4.11. Neste caso são sentidas pequenas discrepâncias, sendo que no caso do *cluster* 2 o gráficos são iguais ao longo de todos os instantes.

4.6.2 Por tráfego gerado

Os gráficos por número de utilizadores encontram-se nas figuras 4.12, 4.13, 4.14 e 4.15. Os dados obtidos possuem praticamente as mesmas discrepâncias sentidas no caso da subsecção anterior, sendo que o gráfico do *cluster* 2 é também semelhante.

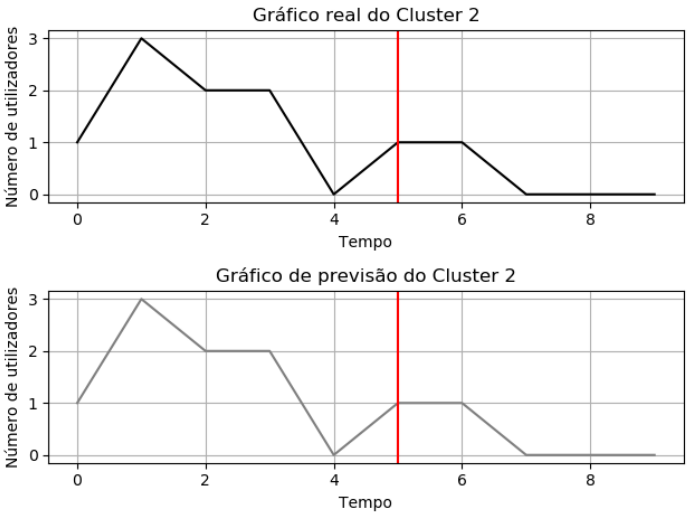


Figura 4.8: Gráfico Nº Utilizadores Cluster 2

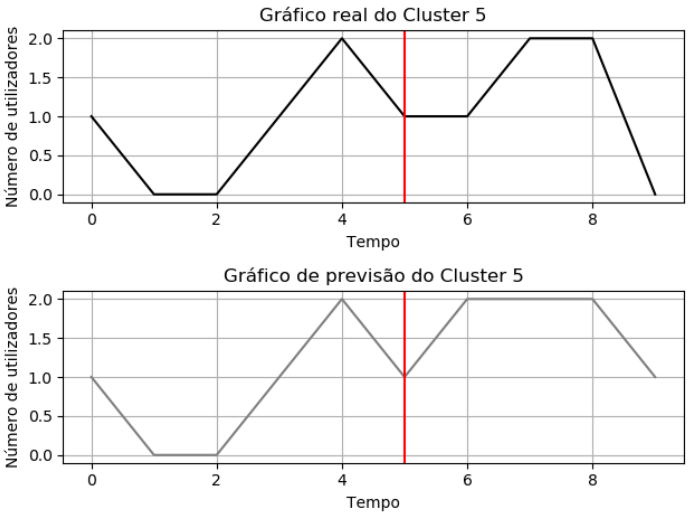


Figura 4.9: Gráfico Nº Utilizadores Cluster 5

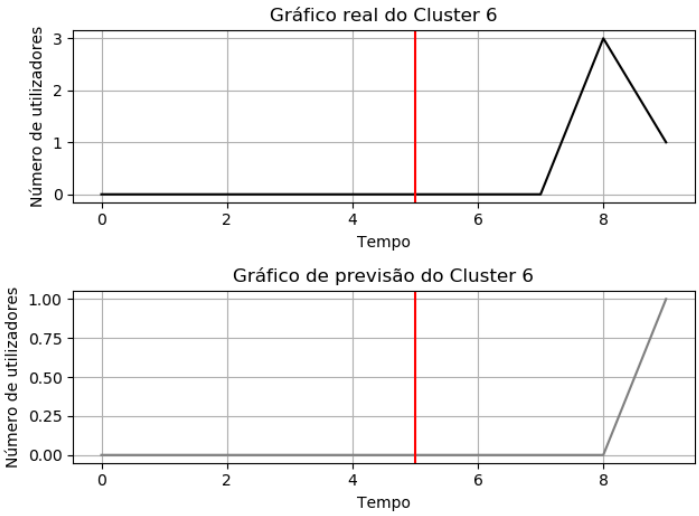


Figura 4.10: Gráfico Nº Utilizadores Cluster 6

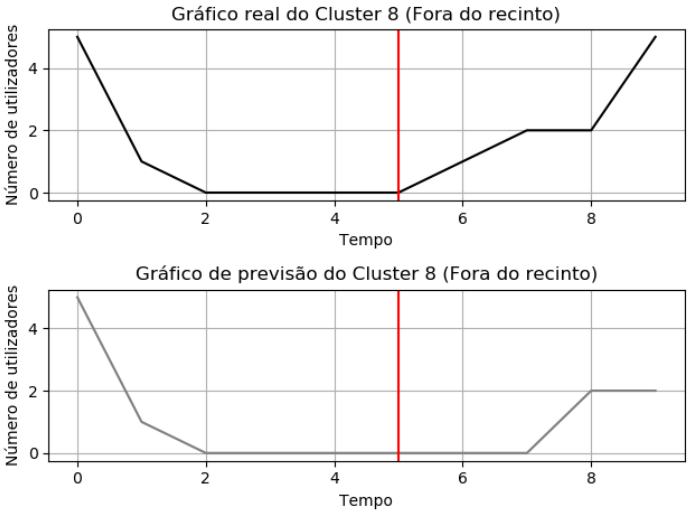


Figura 4.11: Gráfico Nº Utilizadores Cluster 8

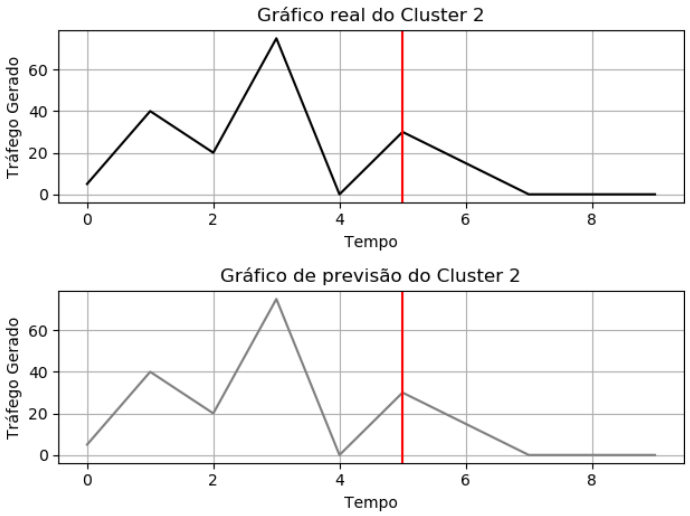


Figura 4.12: Gráfico Tráfego gerado Cluster 2

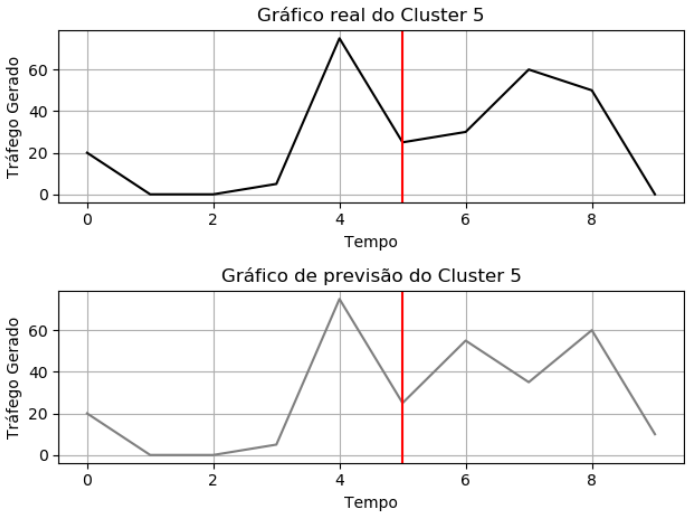


Figura 4.13: Gráfico Tráfego gerado Cluster 5

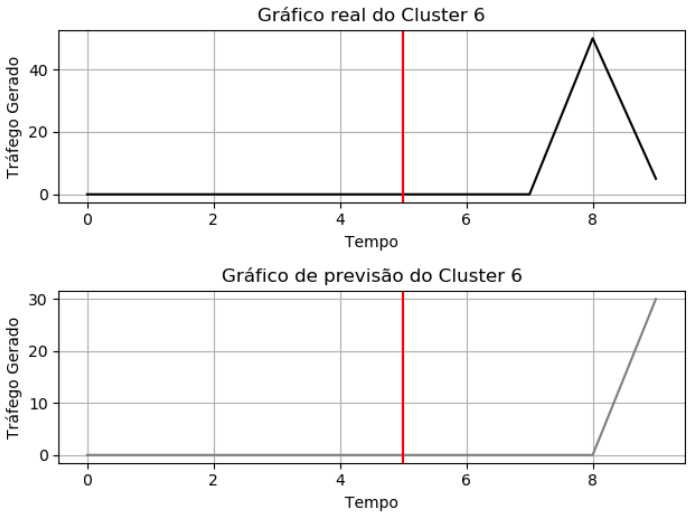


Figura 4.14: Gráfico Tráfego gerado Cluster 6

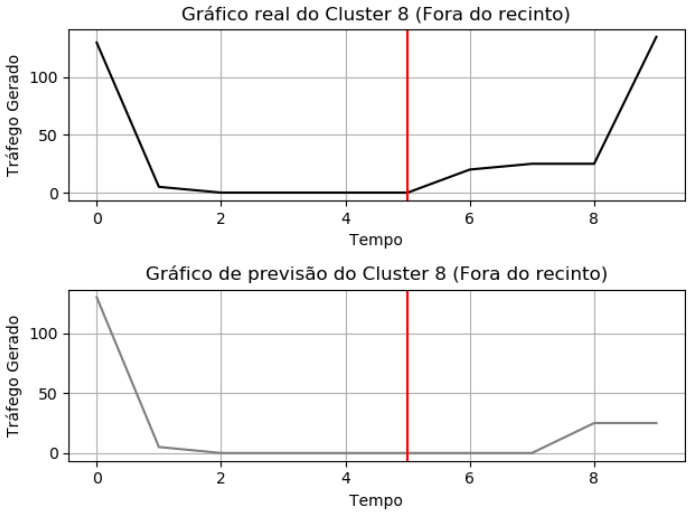


Figura 4.15: Gráfico Tráfego gerado Cluster 8

4.7 Erro associado

O erro global associado , para um tamanho da janela de previsão de 6,é de 54.53%, sendo que o erro de previsão para cada utilizador é:

- Erro global do utilizador 0 : 47.31%
- Erro global do utilizador 1 : 78.7%
- Erro global do utilizador 2 : 80.97%
- Erro global do utilizador 3 : 14.95%
- Erro global do utilizador 4 : 52.33%
- Erro global do utilizador 5 : 75.8%
- Erro global do utilizador 6 : 63.94%
- Erro global do utilizador 7 : 24.86%
- Erro global do utilizador 8 : 64.90%
- Erro global do utilizador 9 : 41.52%

4.8 Comparação de erro associado

De forma a descobrir como é o comportamento do algoritmo da previsão com uma janela deslizante diferente, o mesmo algoritmo é corrido com todos os dados iguais à exceção do tamanho da janela deslizante. O tamanho passará a ser de 5 e de 8. No primeiro caso, obtém-se um erro de 42.21%, em que o erro de previsão para cada utilizador é:

- Erro global do utilizador 0 : 39.64%
- Erro global do utilizador 1 : 19.64%
- Erro global do utilizador 2 : 57.01%
- Erro global do utilizador 3 : 46.8%
- Erro global do utilizador 4 : 48.98%
- Erro global do utilizador 5 : 58.03%
- Erro global do utilizador 6 : 49.36%
- Erro global do utilizador 7 : 40.8%

- Erro global do utilizador 8 : 55.11%
- Erro global do utilizador 9 : 6.77%

No segundo caso, obtém-se um erro de 37.66%, em que o erro de previsão para cada utilizador é:

- Erro global do utilizador 0 : 0.0%
- Erro global do utilizador 1 : 0.0%
- Erro global do utilizador 2 : 76.43%
- Erro global do utilizador 3 : 29.89%
- Erro global do utilizador 4 : 45.23%
- Erro global do utilizador 5 : 61.05%
- Erro global do utilizador 6 : 51.19%
- Erro global do utilizador 7 : 45.23%
- Erro global do utilizador 8 : 67.58%
- Erro global do utilizador 9 : 0.0%

Verifica-se que para uma janela de tamanho 5, o erro associado é menor que para uma janela de valor 6. Estes valores demonstram que, apesar de um tamanho da janela deslizante maior considerar maior número de dados, alguns desses dados podem já não estar atualizados de acordo com o comportamento dos utilizadores.

4.9 Conclusão

Todos os testes realizados através de este caso de simulação tiveram resultados positivos, validando assim todas as funcionalidades da aplicação desenvolvida. Através da interpretação dos dados, pode-se também concluir que os gráficos reais e os de previsão são bastante semelhantes.

O erro de previsão pode ser consequência de factos intrínsecos ao algoritmo e do tamanho da janela deslizante não ser o ideal. O tamanho da janela deslizante é um fator que altera o erro associado, mas não de forma linear (como foi demonstrado na secção anterior), ou seja, não é possível indicar qual o número ótimo do tamanho da janela deslizante sem se fazer um estudo para cada valor do tamanho da janela. Pode ainda estar relacionado com a forma como este é

calculado, sendo atribuído um erro muito elevado a cada instante em que o algoritmo falha, uma vez que para os gráficos mostrados neste capítulo o erro obtido parece não coincidir com a semelhança dos valores obtidos através da previsão.

Capítulo 5

Conclusões

Neste capítulo as conclusões do trabalho desenvolvido são expostas, fazendo uma análise quanto ao balanço de todo o trabalho e os possíveis desenvolvimentos futuros do mesmo.

5.1 Balanço

Esta tese envolveu várias etapas e envolve assuntos muito específicos, tendo, por isso, um tempo de pesquisa extenso, mas que foram úteis para a evolução da aplicação desenvolvida. No geral, este trabalho permitiu fazer um estudo do tráfego e da mobilidade em TCEs, sendo que o objetivo pretendido, o desenvolvimento de uma ferramenta que caracteriza e modeliza conjuntamente a mobilidade e o tráfego gerado pelos utilizadores, foi concluído com sucesso.

Quanto à ferramenta, que consiste numa aplicação, esta apresenta todas as funcionalidades definidas inicialmente. É também bastante simples de utilizar, pois os dados obtidos são facilmente interpretados, tendo também sido esta uma preocupação ao longo da elaboração da aplicação.

No geral, a aplicação funciona e cumpre todos os requisitos definidos. No entanto, os erros de previsão obtidos são um pouco elevados, por isso, a aplicação necessita de desenvolvimentos futuros de forma a reduzir este erro.

5.2 Desenvolvimentos Futuros

Como desenvolvimentos futuros pode-se realçar a alteração do algoritmo de *Clustering* pois o que foi utilizado não cobre todos os casos possíveis de dados. Para isto é necessário criar um outro algoritmo que avalie qual a melhor técnica a utilizar em cada caso.

É ainda necessário rever a fórmula utilizada para o erro, pois os valores que gerados por esta fórmula são muito elevados quando comparados aos gráficos obtidos.

Levanta-se também a questão do tamanho ideal da janela deslizante. Através dos dados do capítulo anterior, percebeu-se que o erro associado à previsão se altera consoante o tamanho da janela, mas não de forma linear. De maneira a melhorar este facto poderia ser criado um algoritmo que tenha funcionamento semelhante ao método *Elbow* e ao *Average Silhouette*, mas neste caso que avalie qual o tamanho ideal da janela deslizante de forma a obter o menor erro associado à previsão.

Bibliografia

[1] Natalie Duffield. How dedicated Wi-Fi is transforming the music festival experience | ITProPortal. Disponível em https://www.itproportal.com/2015/08/16/how-dedicated-wi-fi-transforming-music-festival-experience/. [citado na p. 1]

[2] Ricardo Bramão. Annual Report 2017 | 272 festivais de música - novo record |Associação Portuguesa de Festivais de Música, 2018. [citado na p. 2]

[3] Altice. MEO Sudoeste – Música e tecnologia juntas na edição de 2018 | Altice Portugal. Disponível em https://www.telecom.pt/pt-pt/media/comunicados/Paginas/2018/agosto/MEO-Sudoeste-—Música-e-tecnologia-juntas-na-edição-de-2018.aspx. [citado na p. 2]

[4] Eduardo Nuno Almeida, Rui Campos, and Manuel Ricardo. Traffic-aware multi-tier flying network: Network planning for throughput improvement. In *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6. IEEE, apr 2018. [citado na p. 6]

[5] M Zubair Shafiq, Ji Lusheng, Alex X Liu, Jeffrey Pang, Shobha Venkataraman, and Jia Wang. *A First Look at Cellular Network Performance during Crowded Events*. ACM, 2011. [citado na p. 6]

[6] Huandong Wang, Fengli Xu, Yong Li, Pengyu Zhang, and Depeng Jin. Understanding Mobile Traffic Patterns of Large Scale Cellular Towers in Urban Environment. In *Proceedings of the 2015 ACM Conference on Internet Measurement Conference - IMC '15*, pages 225–238, New York, New York, USA, 2015. ACM Press. [citado na p. 6]

[7] Ying Zhao and George Karypis. Evaluation of Hierarchical Clustering Algorithms for Document Datasets. Technical report, University of Minnesota, 2002. [citado na p. 6]

[8] M. Zubair Shafiq, Lusheng Ji, Alex X. Liu, and Jia Wang. Characterizing and modeling internet traffic dynamics of cellular devices. *ACM SIGMETRICS Performance Evaluation Review*, 39(1):265, jun 2011. [citado na p. 6]

[9] K. Tutschku and P. Tran-Gia. Spatial traffic estimation and characterization for mobile communication network design. *IEEE Journal on Selected Areas in Communications*, 16(5):804–811, jun 1998. [citado na p. 7]

[10] Amitabha Ghosh, Rittwik Jana, V. Ramaswami, Jim Rowland, and N. K. Shankaranarayanan. Modeling and characterization of large-scale Wi-Fi traffic in public hot-spots. In *2011 Proceedings IEEE INFOCOM*, pages 2921–2929. IEEE, apr 2011. [citado na p. 7]

[11] Trupti M Kodinariya and Prashant R Makwana. Review on determining number of Cluster in K-Means Clustering. *International Journal of Advance Research in Computer Science and Management Studies*, 1(6), 2013. [citado na p. 7, 8]

[12] Jiawei Han, Micheline Kamber, and Jian Pei. Cluster Analysis. In *Data Mining*, pages 443–495. Elsevier, 2012. [citado na p. 7]

[13] Peter J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 1987. [citado na p. 9]

[14] Marcelo Menezes Reis. Processos Estocásticos. Disponível em http://www.inf.ufsc.br/ marcelo.menezes.reis/Processos03.pdf. [citado na p. 11]