

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Voyage to MARS: A Mobility Aware Recommender System

Ricardo Leal



Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Teresa Galvão

Co-Supervisor: Pedro Maurício Costa

July 15, 2015

Voyage to MARS: A Mobility Aware Recommender System

Ricardo Leal

Mestrado Integrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

Chair: Prof. Rosaldo Rossetti

External Examiner: Prof. Jorge Cardoso

Supervisor: Prof. Teresa Galvão

July 15, 2015

Abstract

Mobility is a concept that has been with humanity since its inception. Crucial to modern life, most people use mobility on a daily basis. It is a necessity. However, the over reliance on private transportation, cars in particular, over public transportation is having severe consequences on the environment. Carbon emissions, pollution and climate change are problems most cities are facing right now. This also causes a serious social and economic impact. It is of the utmost importance that we, as a society, work together to deal with this problem. The best way to do this is by promoting alternate modes of transportation like walking, cycling and, when those are not feasible, public transport.

Previous work has shown that an effective way of promotion public transportation would be for public transport operators to cater to their passengers through the use of partnerships with local commerce and services using a system that would reward users for their use of public transportation. The more they use it, the more benefits they will reap.

The work done in this thesis uses the concept of a recommender system to improve on that. A recommender system is a software tool or technique that provides suggestions to a user based on their characteristics, explicit or implicit interests, similar users preferences or items' attributes.

The system conceptualized to manage this partnerships and harbor the reward system to foster public transportation is called Voyager. It is presented to passengers in the form of a mobile application for smartphones that will offer its users discounts, products or simply suggestions. Voyager will manage the interactions between public transport operators, providers and public transportation users. There will be various different types of offers available. "Reward offers" will be awarded to the user proportionally with their use of public transportation. Promotional offers will try to influence users to frequent new places as an effort to promote dead zones in the city. Other types of offers will have different goals like entertaining the user while he is waiting for the bus with, for example, an offer for a coffee in a coffee shop nearby.

The implementation of Voyager would bring advantages to all major stakeholders. Public transportation users would be given offers and enjoy an overall better service. Public transport operators would have a new way of encouraging passengers to use even more public transportation and have a way to unite all the different partnerships and campaigns in one place using an organized and easier to manage system. Providers would have access to the huge client-base of public transportation systems gaining a competitive advantage over similar businesses and a faster way to create campaigns either by offering discounts and products or promoting events, target the campaigns to a specific audience and track and evaluate the success of said campaigns.

The biggest challenge of Voyager would be to attach the offers to the correct users: the ones who have interest in them. For the creation of a superior system, it is absolutely necessary the integration of a recommender system in Voyager. The recommender system developed for this project is called MARS: a mobility aware recommender system. MARS is able to recommend items to users based on their interests and their mobility profile. The tests conducted showed MARS does its job successfully and with a high degree of accuracy.

Resumo

Mobilidade é um conceito que pertence à humanidade desde sempre. É crucial para a vida moderna uma vez que a maioria das pessoas se desloca diariamente. É uma necessidade. Contudo, a dependência excessiva no transporte privado, automóveis em particular, em detrimento dos transportes públicos tem causado consequências severas para o meio ambiente. Emissões de carbono, poluição e alteração do clima são problemas que muitas cidades estão atualmente a lidar. Os efeitos não são apenas ambientais, mas também económicos e sociais. É de extrema importância que nós, como uma sociedade, trabalhemos juntos para enfrentar este problema. A melhor maneira de o fazer é promover modos de transporte alternativos como andar a pé, de bicicleta e, quando estes não são viáveis, transportes públicos.

Há projetos prévios que mostram que existe uma maneira eficaz de promover transportes públicos seria através do uso de parcerias com o comércio e serviços locais tendo por base um sistema que recompensa os utilizadores conforme o seu uso de transportes públicos. Quanto mais utilizarem, mais benefícios terão.

O trabalho desenvolvido nesta tese usa o conceito de um sistema de recomendação neste contexto. Um sistema de recomendação é uma ferramenta de *software* ou técnica que oferece sugestões a um utilizador baseado nas suas características, interesses (explícitos ou implícitos), preferências de utilizadores similares ou atributos de itens.

O sistema conceptualizado para gerir as parcerias e recompensas para fomentar a utilização de transportes públicos foi apelidado de Voyager e será apresentado aos passageiros sob a forma de uma aplicação móvel para *smartphone* que vai oferecer descontos, produtos ou simples sugestões. O Voyager vai gerir as interações entre os operadores de transportes públicos, os provedores das ofertas e os utilizadores de transportes públicos. Ofertas promotoras servirão para influenciar os utilizadores a frequentar novos lugares como forma de promover zonas mortas na cidade. Outros tipos de ofertas terão diferentes objetivos como entreter o utilizador enquanto ele espera pelo autocarro com uma oferta de café num sítio nas redondezas.

A implementação do Voyager traria vantagens a todos os intervenientes. Os utilizadores de transportes públicos receberiam ofertas e usufruiriam de um serviço melhor. Os operadores teriam uma nova forma de incentivar os passageiros a usar mais transportes públicos e teriam uma ferramenta para juntar num só lugar todas as parcerias e campanhas num sistema organizado e fácil de gerir. Os provedores teriam acesso à enorme base de clientes que utiliza transportes públicos ganhando uma vantagem competitiva perante outros negócios da mesma área.

O maior desafio do Voyager seria atribuir ofertas aos utilizadores corretos: aqueles que estariam interessados. Para a criação de um sistema de sucesso é absolutamente necessário a integração de um sistema de recomendação no Voyager. O sistema de recomendação desenvolvido no âmbito deste projeto é o MARS: A Mobility Aware Recommender System. O MARS consegue recomendar itens a utilizadores baseado nos seus interesses e perfil de mobilidade. Os testes efetuados mostram que o MARS executa essa tarefa com sucesso e com grande precisão.

Acknowledgements

First of all, I want to thank my supervisor, Prof. Teresa Galvão, for all the support and wise words.

I also want to thank my co-supervisor, Pedro Maurício Costa, for all the help and feedback throughout the project and especially for his unparalleled skill in naming new systems (MARS and Voyager were both names suggested by him).

Also to all the Seamless Mobility team. It was a pleasure to work alongside you, even if on different projects. Our discussions about the concepts inside this dissertations were of great value to me.

To my family and friends.

Ricardo Leal

*“Showing the right content to the right people is an ongoing process.
We’ll be working on that forever.”*

Mark Zuckerberg

Contents

1	Introduction	1
2	Related Works	3
2.1	Mobility and Public Transport	3
2.1.1	Public Transportation in Porto	4
2.2	"The Mobiganha Project"	7
2.2.1	Mobipag	7
2.2.2	Mobiganha	8
2.3	Recommender Systems	10
2.3.1	The Long Tail	10
2.3.2	Applications of Recommender Systems	11
2.3.3	Types of Recommender Systems	12
2.3.4	Location-Aware Recommender Systems	15
3	Voyager	17
3.1	Brief Vision of the System	17
3.2	Offers	18
3.2.1	Offer Description	18
3.2.2	Reward System	18
3.2.3	Offer Characteristics	20
3.2.4	Offer Claim Models	21
3.2.5	Campaign evaluation	26
3.3	Requirements	26
3.3.1	Functional Requirements	26
3.3.2	Non-Functional Requirements	27
3.4	Advantages for the main stakeholders	27
3.5	Integration with existing systems	28
3.6	Recommendations	28
4	MARS: A Mobility Aware Recommender System	29
4.1	Data	29
4.1.1	Mobility Data	29
4.1.2	Profile Data	30
4.1.3	Offer Data	30
4.2	Methodology	30
4.2.1	Baseline	30
4.2.2	Location	32
4.2.3	Location and Frequency	32

CONTENTS

4.2.4	Clustering	33
4.3	Implementation	34
4.3.1	Baseline	34
4.3.2	Location	35
4.3.3	Location and Frequency	35
4.3.4	Clustering	37
4.4	Results	40
5	Conclusions and Future Work	45
5.1	Future Work	45
5.1.1	Voyager	45
5.1.2	User's Interests Data	46
5.1.3	A Hybrid System	46
5.1.4	Walking Distance	46
5.1.5	Clustering	47
5.1.6	Time	47
	References	49

List of Figures

2.1	Evolution of validations in Porto Metropolitan Area	4
2.2	Division of Porto Metropolitan Area in zones for journeys starting in zone S8 . . .	5
2.3	Andante Monthly Subscription (on the left) and Andante Azul (on the right). . . .	6
2.4	The validation of a ticket.	7
2.5	Simplified business model for Mobiganha [San14]	9
2.6	The long tail	11
3.1	Model 1 - Normal	21
3.2	Model 2 - Discard	22
3.3	Model 3 - Auction	23
3.4	Model 4 - First Come First Served	24
3.5	Model 5 - Accept / Reject	25
4.1	Algorithm ratings comparison	41
4.2	Recommendations for a single user using the four iterations of the MARS algorithm: baseline (top left), location-based (top right), location- and frequency based (bottom left) and clustering (bottom right).	43
5.1	The straight line distance and the walking distance	47

LIST OF FIGURES

List of Tables

2.1	Types of Andante Cards	6
2.2	A utility matrix representing ratings of movies.	12
2.3	User and Item Profile Vectors Example	13
4.1	Offer Profile Example	31
4.2	Output Ratings Matrix	34
4.3	Algorithms Parameters and Overall Results	40

LIST OF TABLES

Abbreviations

RS	Recommender System
MARS	Mobility Aware Recommender System
LARS	Location Aware Recommender System
WWW	<i>World Wide Web</i>

Chapter 1

Introduction

Mobility is one of the most crucial aspects of the modern world. While our mobility aspirations keep getting higher with the passage of time, there are problems we have not yet dared to face. There is a rising concern for the way our mobility is affecting the world around us and our quality of life. Mobility is something essential for us as humans but it cannot come at too high a cost for the future. Congestion, pollution, climate change are only a few of the ways our current mobility habits are affecting our life. They are having a highly negative impact on an economic, social and environmental level. The best way to turn this around is to adapt our mobility habits and promote a more sustainable mobility. This can be achieved with a shift from private to public transportation.

Following a recent study [[San14](#)] that states partnerships with local businesses are the best way for public transport operators to encourage and increase the use of public transportation through a system that offers products, discounts and suggestions for the passengers, the work of this thesis focuses on the process that will grant this offers to public transportation users. This system will be beneficial not only to the public transport operator, with a boost to the use of public transportation, and to the user, who will be awarded offers for his effort, but also to the local business that will have a new way of attracting clients.

This thesis has the goal to explore in what way this kind of tool can be used in the context of urban mobility in [Chapter 3](#) and how can offers be targeted to public transportation users using their mobility profiles in [Chapter 4](#).

Introduction

Chapter 2

Related Works

In this chapter, all the work that preceded this dissertation and necessary to its understanding will be carefully laid out. It is divided in three seemingly unrelated but very much connected parts. "Mobility and Public Transport" relates to the context. "The Mobiganha Project" is about a precursor project whose concept Voyager is based on. "Recommender Systems" is about the more technical component needed to understand MARS, the mobility-aware recommender system at the heart of Voyager.

2.1 Mobility and Public Transport

Mobility is and has always been part of what makes us human. We move virtually every day. For work. For fun. For everything. Transport is fundamental to our society and to our economy. It enables economic growth and job creation. Mobility is vital to our prosperity and to guarantee the quality of life of citizens as they enjoy their freedom to travel. [E.C11] Mobility is not just a privilege. It is a necessity.

All over Europe, our cities are facing mostly the same problems: congestion, road safety, security, pollution, climate change due to carbon dioxide emissions, etc. Increasing traffic in urban areas leads to permanent congestion. Carbon emissions are affecting the global climate with irreversible long term consequences. This has negative economic, social and environmental impacts. The annual costs are estimated at almost 100 billion Euro [Com07]. In the light of the new challenges we face, mobility must be sustainable. Transport is the one sector where such a reduction in energy use and emissions is proving to be extraordinarily difficult to achieve.

In the current social and economical display, one of the best ways to face this challenges is through the use of modes of transportation that place less of an impact on the environment. [Ban11] This means to encourage practices like walking, cycling and a greater use of public transport.

Related Works

The consensus is that there needs to be a shift from private transportation to public transportation. This is so difficult to achieve because of the inability to strike the right balance between the comfort of private transportation and the sustainability provided by public transportation. [OHM⁺12] We cannot place too much priority on comfort but placing too much importance on sustainability restricts people's freedom of movement and makes a city a less convenient and attractive place to live. The solution is a high quality collective transport that is both smooth and safe. [Com07]

The mechanisms linking mobility and wellbeing are culturally, materially and politically specific, but in contexts where good public transport is available as a right and bus travel not stigmatised, it is experienced as a major contributor to wellbeing, rather than a transport choice of last resort. Bus services in particular remain a stigmatised form of mobility in many settings: used only where there are no other transport options. [GJR14] If we want to improve the sustainability of cities, this needs to change.

The aforementioned shift from private transportation to public transportation necessary for all the reasons stated above is not happening and the opposite is actually taking effect in areas like the Porto Metropolitan Area.

2.1.1 Public Transportation in Porto

Porto Metropolitan Area is a Portuguese metropolitan area based in the city of Porto but encompassing 16 more counties. This is the example of an area that despite the efforts of the competent authorities and even in face of a economic crisis that still affects the country, registers a decrease in the use of public transportation as can be noticed in Fig. 2.1.

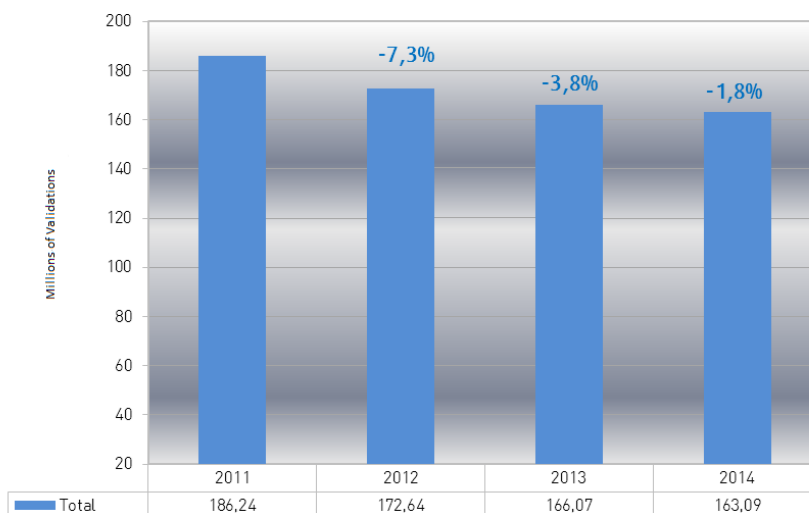


Figure 2.1: Evolution of validations in Porto Metropolitan Area

Related Works

To understand the work exposed of this dissertation, it's necessary to explain how public transportation in Porto Metropolitan Area works (for simplicity's sake, it will from now on be referred solely as Porto).

2.1.1.1 The Andante

The main ticketing system in Porto is called Andante. For traveling in this area, it is necessary to know how the system works.

This public transportation network is constituted by both public and private operators where most people use the Andante. Known for its complexity, the Andante system divides the Porto Metropolitan Area into zones as seen in Fig. 2.2.

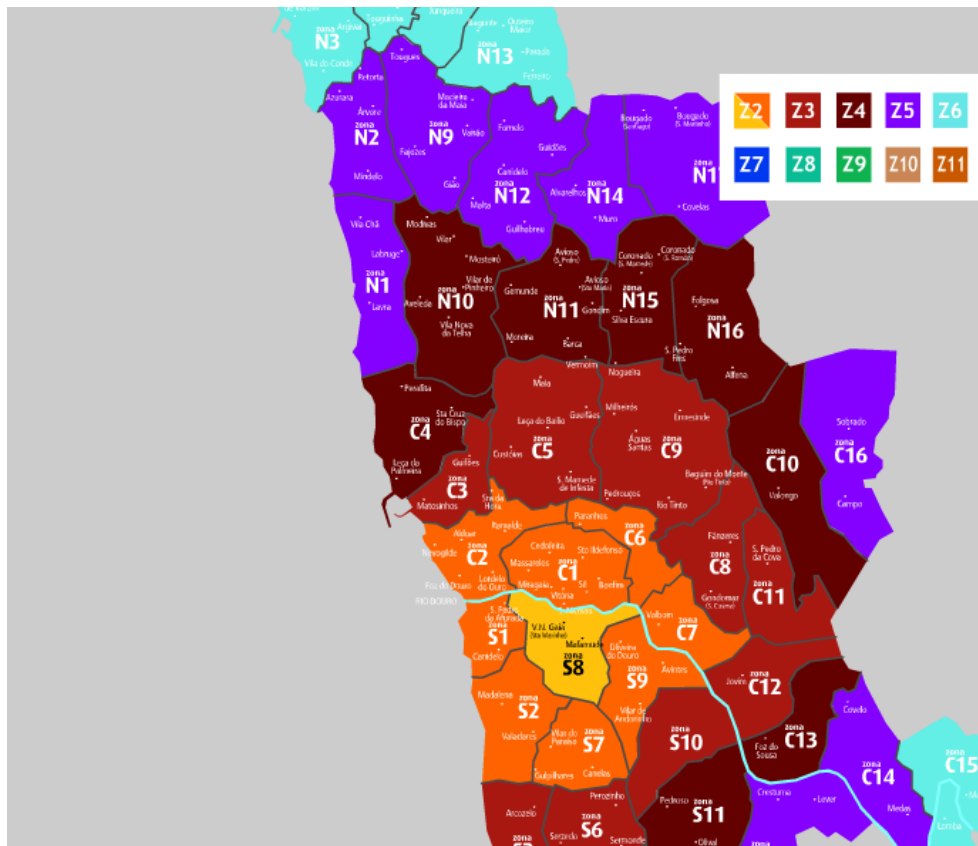


Figure 2.2: Division of Porto Metropolitan Area in zones for journeys starting in zone S8

The price of the ticket is not determined by the number of vehicles but by the number of zones the passenger wants to travel to relative to the check-in zone. The tickets are named "zX" with X being the number of zones (from 2 to 12). Thus, if a passenger wanted to travel one or two zones, he should choose a z2 ticket. If he needed to travel between 12 zones, he should chose a z12 ticket.

The physical form of the tickets is the Andante card: a contactless cards with RFID capabilities. It is possible to charge the card with several tickets as long as they are of the same type. For

Related Works

example, I can only have z2 tickets in the same card. It's not possible to have a z2 and a z3 ticket on the same card. There are four types of Andante cards:

Table 2.1: Types of Andante Cards

Type of Andante Card	Description
Andante Azul (Blue)	A normal rechargeable card for occasional users. The passenger can travel in the chosen zones between one hour (for z2) and three hours (for z12). The card can be seen on the right in Fig. 2.3.
Andante 24	A card for 24h. The price escalates according with the number of zones.
Andante Tour	Aimed at tourists, the Tour card allows the passenger to use all the network during one or three days.
Monthly subscription	A card for people who use public transportation intensively. The most economic option for a daily commute. The card can be seen on the left in Fig. 2.3.



Figure 2.3: Andante Monthly Subscription (on the left) and Andante Azul (on the right).

The system is based in an open architecture (without gates). The passenger has to pass the card in one of the available machines at the beginning of each trip, or when changing lines or vehicles, as illustrated on Fig. 2.4. This act is known as validation. The passenger doesn't need to validate the ticket when checking out after the trip.



Figure 2.4: The validation of a ticket.

2.2 "The Mobiganha Project"

"Mobiganha" is the project whose concept the system detailed in this dissertation is based on. It is based on a reward system that benefits users the more they use public transportation in Porto Metropolitan Area.

2.2.1 Mobipag

The ticketing system is the main root of expenditures of public transportation companies. They represent a big part of this companies' budgets. The Mobipag project was intended to cut costs for the operators with an application that would perform the same functions as the Andante card for a fraction of the cost.

With the Mobipag application, the user can buy and validate tickets without using the card. It would be unreasonable to expect the application to replace the Andante card since there are still many public transportation users that would not adopt of this kind of technology, especially senior citizens. Also, while the smartphone is becoming increasingly diversified, there are still many people who do not own one and it is a sizable investment. However, many users would prefer a service like this and the more passengers that used the application, the more public transport operators would save in ticketing costs.

The application was developed for Android devices with a 2.2 version or newer and implements all of the functionalities of the Andante system. It also adds a consulting feature that gives the user the option to overview his transactions as well as oversee the history of ticket usage. The application works through the Internet so the user needs to be connected either through Wi-Fi or Mobile Data. [FDFa14]

2.2.2 Mobiganha

To bring more users to the idea of Mobipag and to encourage them to adhere to this new payment method, a new solution was thought based on a "voucher market". In this context, a voucher is a redeemable certificate worth a certain monetary value which may be spent only for specific reasons or on a specific type of goods. A voucher market is a business that gives or sells vouchers to users for profit. The most popular example of a business like this would be Groupon. This solution would give users vouchers according with their use of public transportation. The application was integrated in Mobipag and called "Mobiganha".

A study was made to discover what would incite someone to use public transportation more often and, to people that don't use public transportation, what would instigate them to start doing so. The study was made in the form of a Focus Group. One of the main reasons cited was that, for families, private transportation was the cheaper option, so solutions that somehow reduced user costs were welcome. One suggestion was for employers to give incentives to workers that use public transportation, but it was unclear what companies would tangibly benefit from doing that. The solution that gathered universal support was rewarding users for using public transportation through the available mobile payments application. It was seen as the best way to keep the existing client base loyal and to gather new clients.

These vouchers would be obtained through partnerships with local business owners (eg. restaurants) and public entities (eg. museums). Public transportation operators actually already have an extensive network of partnerships with these entities, but promotional campaigns all happen in a mainly disorganized fashion with each new campaign needing considerable manpower and a marketing push as well as all kind of bureaucracies. These entities would have a new way to promote their venues using the enormous client base of public transportation companies.

In short, what would be the advantages of Mobiganha for all the stakeholders?

For public transportation users:

- The ability to resort to an exclusive voucher market and receive vouchers that could help reduce costs
- The opportunity to discover new and interesting places that would use Mobiganha to gain a foothold in the market

For local businesses and public entities:

- Access to the huge client-base of public transportation services
- The opportunity to gain a competitive edge against similar businesses
- Bypassing the usual red tape of these partnerships and high maintenance marketing campaigns and gaining a quick access to the customer: create a voucher and it's done!

For public transportation operators:

- Mobiganha would attract users to Mobipag resulting in savings from the reduced number of physical tickets
- Uniting all the diverse partnerships and campaigns in one place through an organized, easier to manage system

In Fig. 2.5 it is possible to see the Mobiganha business model and its stakeholders. Transport operators would have less ticketing expenses, local businesses and public entities would attract new costumers and public transportation users would have a big market with discounts ready to be used, incentivizing them to keep using public transportation. To cover the maintenance costs of a system like this, both the transport operators and the entities would have to pay a small fee. Also, entities could highlight their promotions through the payment of additional credits.

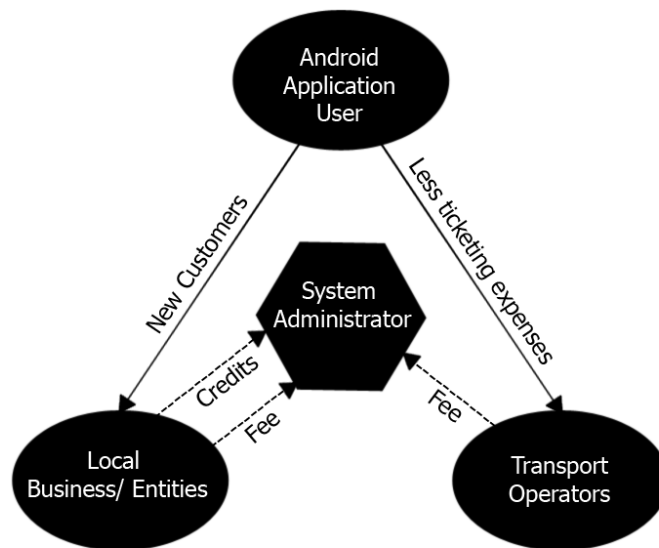


Figure 2.5: Simplified business model for Mobiganha [San14]

The system uses the mile system used in most airline operators. The user would gain points when buying tickets and spend points when buying vouchers. The system would consist of an Android application for the public transportation users and a web application so the entities can input their promotions. A prototype of both was already developed by the Mobipag team. [San14]

2.3 Recommender Systems

In everyday life, we often rely on recommendations from other people in making routine, daily decisions: which books to read, movies to watch, restaurants to go to. These recommendations can come from friends, magazines or simply by word of mouth. [RV97]

A recommender system is a software tool or technique that provides suggestions to a user based on their explicit or implicit preferences, other users preferences, user's characteristics or items' attributes. "Item" is the commonplace term used to denote what the system recommends to users. [TM11] Recommender systems are best known for their use in e-commerce [SKR01] stores like Amazon.com, but they are present in several other business models. The necessity and popularity of recommender systems can be explained by something called *The Long Tail* phenomenon.

2.3.1 The Long Tail

Brick and mortar stores have only a limited amount of shelf-space. They are bound by physical delivery systems, limited storage and distribution, thus they can only show their customer a small fraction of all the items that exist. On the other hand, online stores can make everything available for the customer. For example, a typical large brick and mortar bookstore has approximately between 40 and 100 thousand books while a large online retailer has approximately 3 million. [BHS06] At this moment in time, Amazon.com has more than 30 million books in their catalog.

Recommendation in physical stores is fairly simple because it's not possible to tailor the store to each individual customer. Retailers make orders based on popularity and highlight the items they think are the most popular, the ones they think will sell more.

The gap between the online and the physical worlds is called *the long tail* phenomenon and is illustrated in Fig. 2.6 on page 11. The vertical axis represents popularity and the horizontal axis the products ordered by popularity. The short head, the most popular products, are the ones that are available in brick and mortar stores. The other products are the less popular ones. However, when put together they represent a big chunk of the cake: the long tail. Online stores can provide the entire range of products: the long tail as well as the popular products.

This huge diversity of products forces online stores to make personalized recommendations to each individual user. They can't present all available options to the user and they can't expect users to know about every product they might like. Showing only the most popular options would deprive them from selling most of their items. The long tail phenomenon shows how recommender systems are essential to online institution's business model.

A common story to illustrate how the long tail phenomenon with a good recommender system can completely change the narrative of a product is the story of a book called "Touching the Void". This mountain climbing book was released in 1988 and, despite some critical acclaim, was never a commercial success. One decade later, when "Touching the Void" was nearly out of print, another mountain climbing book called "Into Thin Air" was released to great sales. The Amazon.com algorithm noticed some users who liked both books and started recommending one to the readers

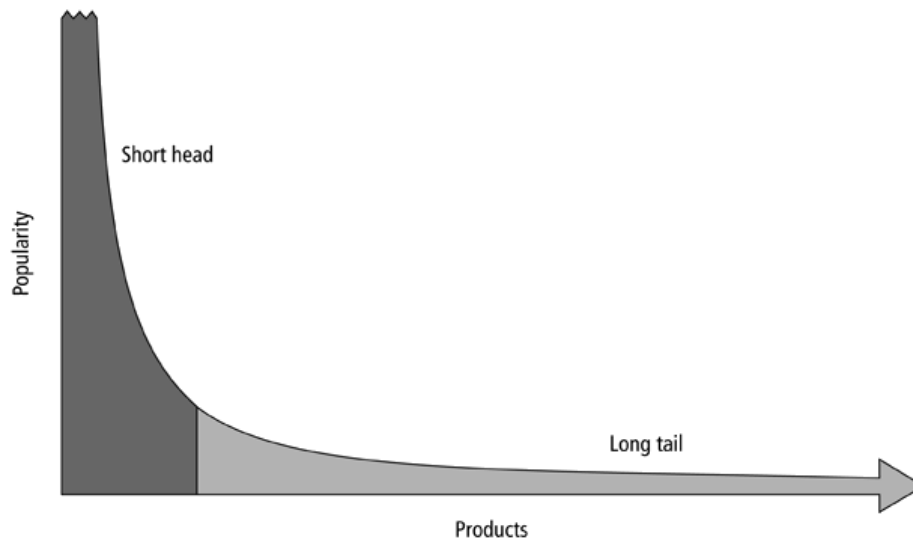


Figure 2.6: The long tail

of the other. Suddenly, "Touching the Void" started selling again. Eventually, "Touching the Void" sales even surpassed "Into Thin Air". [And07]

2.3.2 Applications of Recommender Systems

There are many applications of recommender systems, but right now the most important are:

Product Recommendations: The most common use of recommender system is in online retailers. Online stores like the aforementioned Amazon.com make product recommendations based on the products the user has previously bought and products purchased by similar users.

Movie and Music Recommendations: The upswing in popularity of services like Netflix for movies and Spotify for music will only increased the need for recommender systems since these services like a perfect fit for recommendations. The recommendations are made based on the movies the user liked, normally with a rating provided by the user. The need for good recommendations is such that Netflix offered one million dollars to the first algorithm that beat its recommender system by 10%. [BKV10]

Articles Recommendations: News services have also been trying to recommend news articles to their readers based on the articles they've read in the past.

Social Recommendations: Social networking sites like Facebook and LinkedIn make recommendations of friends and groups through the analysis of the user's social circles.

There are many more applications of recommender systems and not all of them are online. Recommender systems are also being applied in the physical world as will be depicted in section 2.3.4.

2.3.3 Types of Recommender Systems

The most important concept necessary to understand recommender system is the utility matrix. This matrix represents the preference that each user has for each item. This values can be an integer, in this case an integer between 1 and 5, the rating the user gave said item, or they can be a binary number, 1 if the user used the item and 0 if he did not.

	HG1	HG2	HG3	SLP	TA1	TA2
Arya	4			5	1	
Brienne	5	5	4			
Cersei				2	4	5
Daenerys		3				3

Table 2.2: A utility matrix representing ratings of movies.

Usually this kind of matrix is very sparse. Most of values are unknown because the user only rated / used a very small subset of the items available. What a recommender system does is trying to predict the blank spaces of an utility matrix: what is the level of interest the user has for an item he has not rated or used? The recommender system does not need to get the value for every cell in the utility matrix. The relevant data here are the highest values of each row. But there is no need to know all of them: a large subset of the highest values is enough.

As stated before, this type of matrix is generally very sparse, but of course the denser the matrix, the better the quality of the recommendations. So one of the bigger challenges of someone trying to engineer a recommender system is how to populate the matrix. There are two general approaches to acquiring this data:

- Ask users to rate items. Most movie recommendation systems like Netflix and sites like YouTube and Amazon do this. This is not the most effective way because people are not always willing to rate items. The simple fact that the rating came from a person willing to do it, may render it biased. Also, if the rating is visible, people sometimes opt to overcompensate.
- Make inferences from users' behavior. If a user buys a book, watches a movie or listens to a song, we infer that the user is interested in this item. This method will only result in binary ratings: 1 implies interest and 0 is blank. The resultant utility matrix would not have blank spaces, only 1s and 0s.

There are two broad types of recommender systems:

Related Works

- Content-based systems make recommendations based on the attributes of the items. The items recommended to a user are those whose properties are similar to items the user has manifested a preference.
- Collaborative filtering systems make recommendations based on the preferences of similar users.

2.3.3.1 Content-based systems

In a content-based system, recommendations are based on the attributes of the items. The first step is to construct an item profile. For some types of items properties are easily accessible. For example: for a movie it is easy to get certain information like actors, director and genre. The same can be said for a book: the author and genre are quickly found. However, for a certain type of items these traits are harder to define. Documents and web pages don't have any immediate characteristics. Images have the same problem and one of the best workarounds is the use of tags for classifying images. This method proved effective enough that Netflix started to recruit subjects to tag their movies.

After defining the characteristics of the item, a user profile is needed. The user profile summarizes the preferences of the user: how does he feel about each and every characteristic of the item. To do this, we need to resort to the utility matrix. The utility matrix seen in Table 2.2 represents ratings of movies on a 1-5 scale, with 5 being the highest rating. Blank spaces represent the situation where the user has not rated the movie. The movies' abbreviations are: HG - The Hunger Games, SLP - Silver Linings Playbook, TA - The Avengers. Assuming the item profile is constituted by a set of actors, with the user's row in the utility matrix it's possible to know the level of interest a user has of an actor. For example, Jennifer Lawrence is in three movies watched by Bran. In the user profile for Bran, the component for Jennifer Lawrence will have a value of 4.3, the average of all the movies Bran saw that Jennifer Lawrence starred in.

Both the item and user profile are represented as vectors. The similarity between these vectors will determine the interest the user has in the item. For the calculation of the similarity to be possible these vectors have to be identical. Continuing the thought process of the example, the vectors would have all the actors of all the movies in the system, so the vector would be really sparse. The calculation of the similarity between the user's profile and the item's profile would be made with a similarity function like Jaccard similarity or Cosine similarity.

Table 2.3: User and Item Profile Vectors Example

Profile	Cat AA	Cat AB	Cat AC	...	Cat ZZ
User Profile	1	1	0	...	0
Item Profile	0	1	1	...	1

2.3.3.2 Collaborative filtering systems

In a collaborative filtering system, recommendations are based on the preferences of similar users. In this type of system we don't need to create an item or user profiles. The item profile is represented by a column in the utility matrix and the user profile by a row. To find similar users, we measure the similarity between the vector of the user in question and all the other users with a similarity function. The most used or better rated items by the similar users are recommended.

Another version of this type of system uses other type of similarity. Humans are complicated and often paradoxical creatures, much more so than items. For example: it's very unlikely for a movie to be a cowboy movie and an horror movie at the same time, while it's very possible for an individual to like both cowboy and horror movies. This means it is easier to detect similar items because they are of the same genre than it is to detect similar users because they both like one genre but their opinions differ in others.

Item-based collaborative filtering takes this into account and instead of finding similar users, it finds similar items. If the system only needs to calculate for a single user (a single row in the matrix) and not the whole matrix, this method can be a bit heavier because it needs to calculate the similar items for all the items in the row while the former method only needed to calculate the similar users once.

To clarify the difference between the two processes:

- User-based collaborative-filtering finds the users most similar to the user in question. The user interest in the item would be dictated by the number of similar users who used, liked or rated the item. After calculating this for all the items, the ones with a better rating of predicted interest would be recommended.
- Item-based collaborative-filtering finds the most similar items of an item. The user interest in the item would be determined by the number of similar items the user used, liked or rated. After calculating this for all the items, the ones with a better rating of predicted interest would be recommended.

2.3.3.3 Hybrid systems

Neither content-based or collaborative filtering system are perfect. To better understand both their advantages and disadvantages and when to use one or the other, an analysis of their strengths and weaknesses is needed.

For collaborative-filtering systems:

Pro Works for any kind of item. There is no need to do a feature selection.

Con Cold start problem. Since this method is based on the preferences of other users, if the system does not have enough users, it will be unable to find a match.

Related Works

- Con** Sparsity problem. Sometimes the user/item matrix is so sparse that it is hard to find users who have used or rated the same items.
- Con** First rater problem. Cannot recommend an item that has not been previously rated: new items will not be recommended.
- Con** New user problem. Cannot recommend to someone that has not used or rated an item.
- Con** Popularity bias. Cannot recommend to someone with unique taste (i.e. without similar users). Favors popular items.

For content-based systems:

- Pro** No cold start, sparsity or first rater problems. It does not need data on other users.
- Pro** Able to recommend to users with unique tastes.
- Con** For some types of items like images or documents it is very hard to find appropriate features to create an item profile.
- Con** New user problem. Cannot recommend to someone if there is no information on their interests.
- Con** Serendipity problem. Known as the "well of similarity", content-based systems are prone to recommend items too similar to the ones the user has already used or bought. [IDL⁺08] This introduces the term of serendipity to recommender systems: serendipity is when a user receives a recommendation for unsearched or surprising but still useful items. [GDBJ10]

The best way to bypass some of this disadvantages is to use a hybrid approach. There are several ways of doing this.

One way to use a hybrid approach is to simply implement both recommender systems and then combine the results. In this case, the final ranked list of items would be comprised of the best ranked items from the content-based system and the top items from the collaborative filtering system. While this method gives us a very good set of varied recommendations, it is by far the most costly performance-wise.

Another option is to make a real hybrid recommendation system mixing methods from both approaches. For example: a recommender system that uses primarily a collaborative filtering approach, when trying to recommend new items to the user (ie. items no user has actually used or rated) would opt for a content-based method and use the item's characteristics as a way of finding users interested in said item.

2.3.4 Location-Aware Recommender Systems

The massification of smartphones has brought many possibilities to recommender systems. There is reason to believe the future can and should use location as a way to better estimate users' interest

Related Works

in an item. As a result, recommendation services become more personalized and context-sensitive, while limiting the effects of information overload [Ric10]. In addition, the main benefit of implementing a recommendation service in a mobile environment is the ability to leverage context-sensitive information and deliver recommendations in a wide range of scenarios [BLPR12].

LARS A location aware recommender system that uses location-based ratings to make recommendations. LARS can produce three types of location-based ratings:

- Spacial ratings for non-spatial items. For example, a user at home rating a book.
- Non-spacial ratings for spatial items. For example, a user with unknown location rating a restaurant.
- Non-spacial ratings for spatial items. For example, a user at work rating a restaurant visited at lunch.

It was observed that users from a region prefer items that are patently different than items preferred by users from other regions, even if said regions are adjacent. This was named *preference locality*.

It was also observed that, when recommended items are spatial, users tend to travel a limited distance when visiting these venues. This was called *travel locality*. Suggests that spatial items closer in travel distance to a user should be given precedence. [LSEM12]

CityVoyager A real-world recommender system designed for mobile devices tries to apply a recommender system, widely used in online shopping, to real-world shopping. It uses the user's most frequented places based on location data obtained through GPS and recommends new stores using an item-based collaborative filtering algorithm. [TS06]

I'm feeling LoCo A context-aware recommender system that by automatically discovering the user's mode of transportation and mood can better predict the distance the user is willing to travel to get to the destination and making a recommendation accordingly. [SBEH12]

Chapter 3

Voyager

This chapter on a new system based on the concept behind Mobiganha and coupling it with recommender systems. The Mobiganha project showed that a rewarding passengers for the use of public transportation was the best way to encourage them to use it more often. Also, in section [2.3](#), we establish the importance of recommender systems and the huge difference they can make in many businesses by simply giving people what they want.

3.1 Brief Vision of the System

The envisioned system gives public transportation users the rewards they want. It is first and foremost a service that gives passengers offers of things they might be interested in. These offers can be vouchers for various products, but they can also be free trials or simply suggestions for certain events. But this service will not only focus on the offer system but on the integration of this system with the public transportation network, like using offers as incentives to influence users to avoid peak hours or keep them busy until the next bus arrives.

The system would reach the users in the form of a mobile application. This is the obvious choice and the clear best fit. If we want to reach public transportation users, always on the go, we have to get them where they are. This is the only way to make sure passengers have access to the system at all times, if either they are at home, at work or on the road.

Porto Metropolitan Area was chosen as the pilot area for testing for several reasons: it's the environment I have the best firsthand knowledge of, but especially the supplied data of validations from a whole year from this area created a fantastic opportunity for a more specialized system. Having said that, Voyager can easily be consciously implemented to other cities in the world.

There are three main stakeholders in the system:

Public transport operators The public transport operators find partners to provide offers to their users and manage the system.

Providers The local businesses or public entities that have partnerships with the public transport operators and provide the offers.

Public transportation users A public transportation user who has the application on his smart-phone and is interested in receiving offers.

3.2 Offers

3.2.1 Offer Description

The whole system depends on the offers made available to the user so it is essential to know what kind of offers we are talking about.

Giveaways Small value offers given for free by the provider. The idea behind these giveaways is that the business owner would be willing to offer something to a user to gain awareness, get him to come back to his establishment or even buy other items while he's there. **For example:** A coffee shop would offer a Portuguese custard tart. Some users would probably ask for a coffee to accompany it with.

Vouchers Discounts that can be employed in an infinite variety of items: from consumables to products but also events. **For example:** a meal discount in a restaurant, a coupon for the supermarket, a special promotion for a ticket to a concert, a discount in a CD in a music store.

Event suggestions Porto's cultural scene has been steadily growing in the last few years with a vast and varied selection of free events. From this perspective, the system is a service that would give users recommendations of events they are interested in. **For example:** outdoor cinema sessions, markets and fairs, concerts.

Premium Large value offers to be given to users who regularly use public transportation. **For example:** tickets for big music festivals with transport included.

3.2.2 Reward System

All offers would be given to a user according to his tastes and behavior but these offers can be classified by the way they are presented to the user.

Jumpstart Offers During the initial period after downloading the application, the user will receive some offers to get acclimated with the system.

Reward Offers Since the big goal of this system is to get users to use more public transportation, they will receive offers proportional with their use of public transportation. The intricacies of the system will be hidden from the users, but they will be warned the more they use public transportation, the more offers they will get. It would be great to give at least one

offer per day to keep the users engaged but that would always depend on the number of offers available.

Bonus Offers Goal-oriented offers that will only be obtained if the user reaches certain goals. In contrast with Reward Offers, he'll be fully able to track his progress. This is a way to keep the user interested in the application. He knows he'll get something, so he'll keep it in mind and won't discard it. For example: If the user reaches X amount of trips, he'll be able to choose one of three options.

Instant Offers Time-sensitive offers with a specific objective in mind.

- **Entertain the user while he is waiting for the bus.** For example: the user is waiting for a bus that only arrives in 30 minutes. In that situation, the application would offer a coffee in a nearby coffee shop to help the user pass the time.
- **Engage the user while he is in the bus** For example: If the user has a long journey ahead, the application would offer him something in the neighborhood of his destination: something to look forward to.
- **Change the user's route to avoid rush hour** If the user's usual bus is full, the application would give him an offer to delay him and help him avoid peak hour or offer him something in another spot while also recommending an alternate path.

Promotional Offers The idea behind these offer is to get the user out of his comfort zone, show him different and interesting places in which he doesn't usually hang out and also promote dead zones in the city.

The reward system for **Reward Offers** and **Bonus Offers** needs to be carefully analyzed. A points system in which the points are directly correlated with currency is not the way to go here. The goal of this system is not to improve the profitability of public transport but to create city-wide sustainability in transportation. If someone who uses the Monthly Subscription model, the public transport operator won't gain any more money if that passenger increases his use of public transports, but it is still important that he does so. The goal is to get everyone to use more public transportation.

If the reward system won't be based on money and ticket buying, we need to find other metrics to measure the use of public transportation. One of them is validations. To legally use a public transport vehicle, the user needs to validate his ticket. But that can't be our only measure. Let's picture the following:

- A user with a 30 minute commute who uses two public transportation vehicles and gets two validations
- A user who used only one vehicle, got one validation but traveled a much bigger distance for a longer amount of time

Should the former user have a benefit twice as big as the latter? Probably not. To counterbalance, Voyager would take advantage of the zone division in the Andante system and validations with tickets with more zone crossings would weigh more.

3.2.3 Offer Characteristics

Offers will have certain characteristics in the system.

Number of offers Each offer is part of a campaign with a certain amount of offers. This number is of course dependent on the amount of items the provider is willing to give. Unlimited can be an option if it is a discount for everybody that uses the application or an event suggestion.

Time constrains Some offers can only be redeemed in a specific period of time. For example, a coupon for a dinner in a restaurant has a specific time frame to be used. Some stores can also be interested in giving offer only during the slow hours of the day.

Target Audience Providers will also be able to set a target audience for an offer defined by parameters like gender, age or relationship status. This is incredibly important to refine the recommendations. For example: a woman's clothing store would only be of interest to women. Married woman won't be looking for wedding dresses. Etc.

Giftable Offers Users can give offers to other users as a gift.

Sharable Offers These are offers that can be shared with friends: the user will keep the offer but will be able to give a copy to a friend. They don't need to use the offer at the same time. Once the share is made, they are two individual offers.

3.2.4 Offer Claim Models

Now that it's established how the user will receive the offers, it's important to define how the user will claim them. The offers are limited for the purposes of the following diagrams. There are five ways the user can claim an offer:

Model 1 - Normal The first model, outlined in Fig. 3.1, is the standard. Offers are allocated to the most compatible users. The offer belongs to the user for him to do what he pleases.

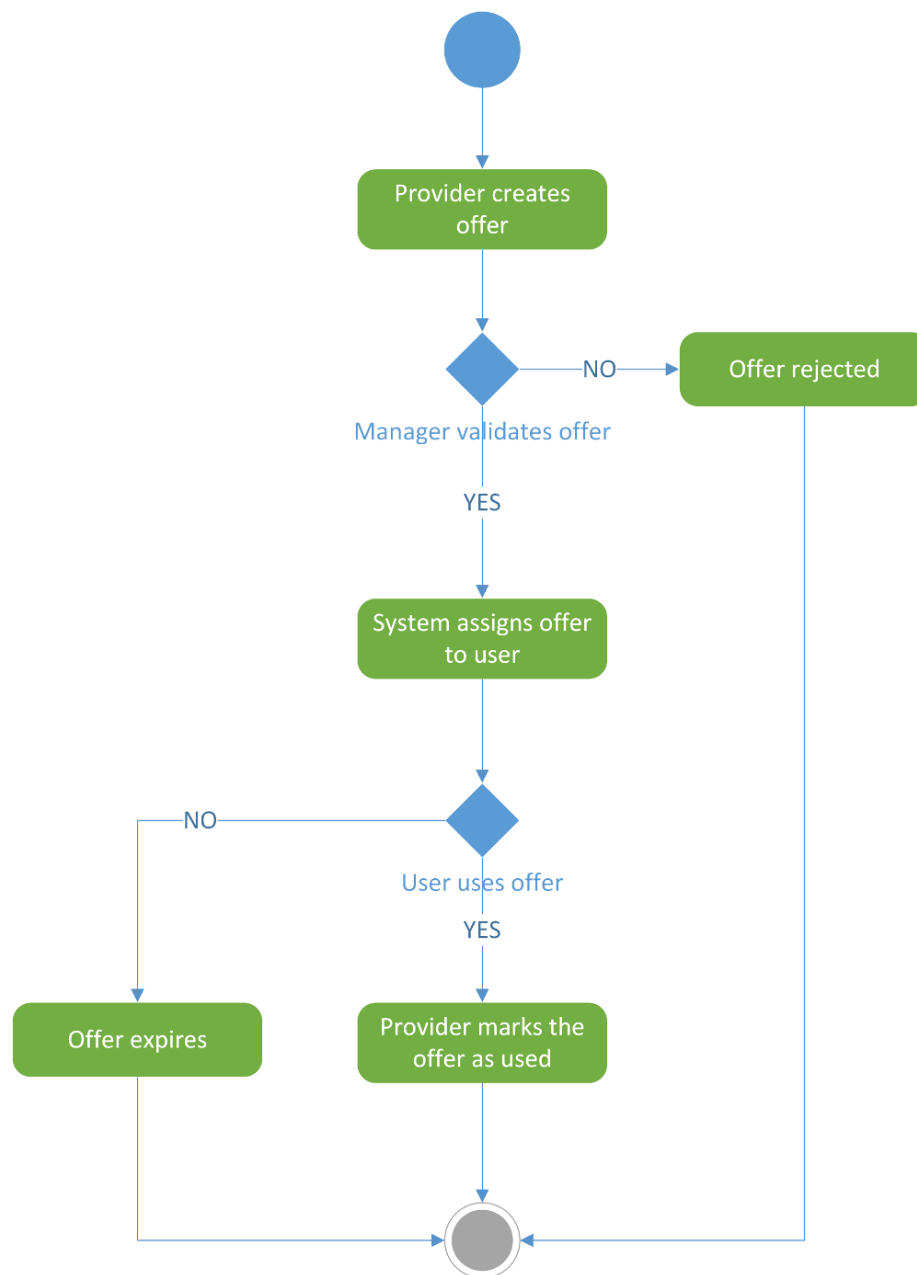


Figure 3.1: Model 1 - Normal

Voyager

Model 2 - Discard The second model is outlined in Fig. 3.2. Offers are allocated to the most compatible users. The offer belongs to the user for him to do what he pleases. It's similar to Model 1 but the user will have the option to discard an offer he doesn't want and that offer would be reallocated to another user.

The advantage of this model relative to the first one is that it reduces the change of an offer being unused and therefore going to waste when it could've been used by another user.

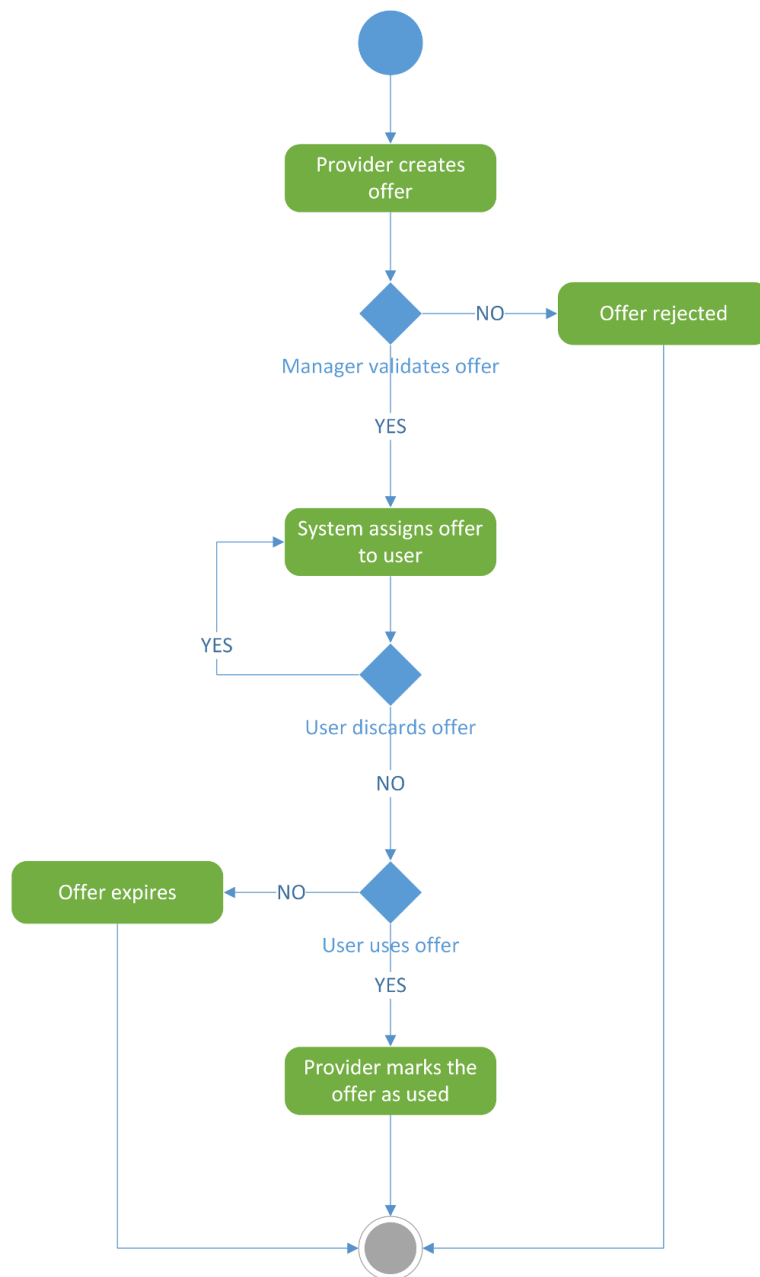


Figure 3.2: Model 2 - Discard

Model 3 - Auction The third model, outlined in Fig. 3.3, follows the paradigm of an auction without the need to bid more money than other users. Offers are allocated to all compatible users. The users need to claim the offer before others to be able to use it. For example, if there are 30 offers available and 50 compatible users, the offers would be available to the first 30 users to claim them.

The advantage of this model is that the probability of the offers being used increases because the user needs to make an effort to claim the offer. On the other hand, many users think of auctions as frustrating because of the possibility of not getting items they are interested in.

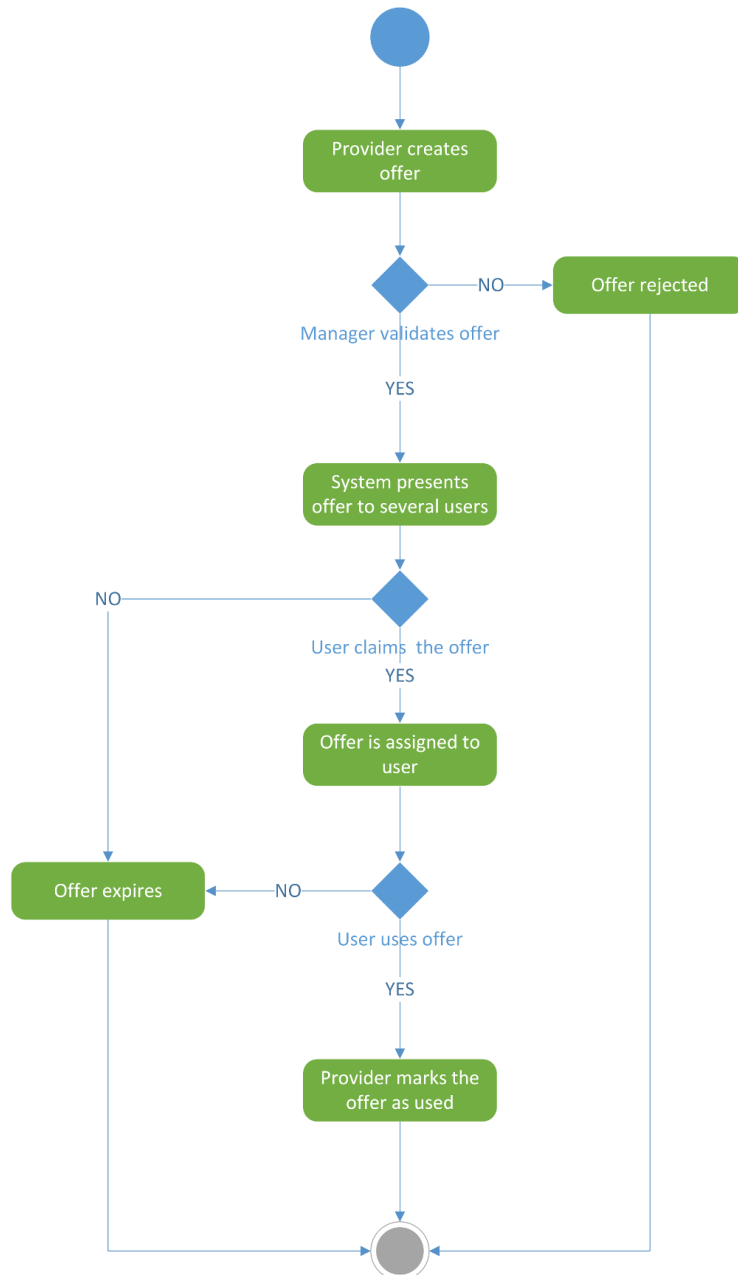


Figure 3.3: Model 3 - Auction

Model 4 - First Come First Served The fourth model is outlined in Fig. 3.4. In this model, offers are allocated to all compatible users. However, the user can only claim an offer at the moment of use. The offers go to the first users to take advantage of the offer.

Similarly to the auction model, this model, while effective in expediting the offers, can cause frustration to the user if when he arrives the offer is no longer available. This problem could be mitigated by warning the users when the offers is sold out.

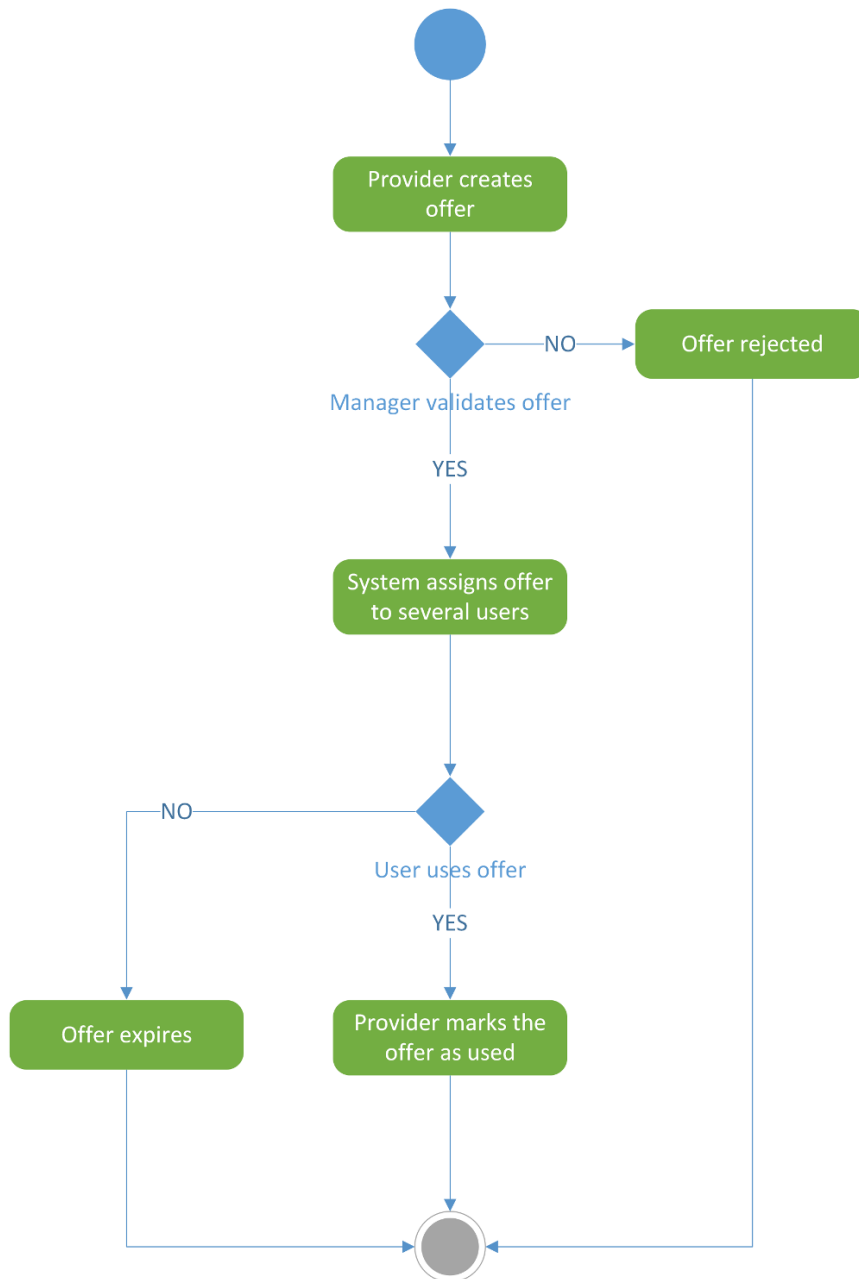


Figure 3.4: Model 4 - First Come First Served

Model 5 - Accept / Reject The fifth model is outlined in Fig. 3.5. In this model, offers are allocated to the most compatible users. When they receive an offer, then can accept or decline. If they accept, the offer belongs to them, but this decision is reversible. If they decline, the offer will be reallocated to another user and therefore is not reversible. If they don't make a choice in a predefined amount of time, they will lose the offer which would then be reallocated to another user.

The functioning of this method is identical to the Discard Model, but by asking people if they want an offer or not, we're more likely for them to say no if they don't like it. Many user just won't bother to discard an offer even if they don't like it.

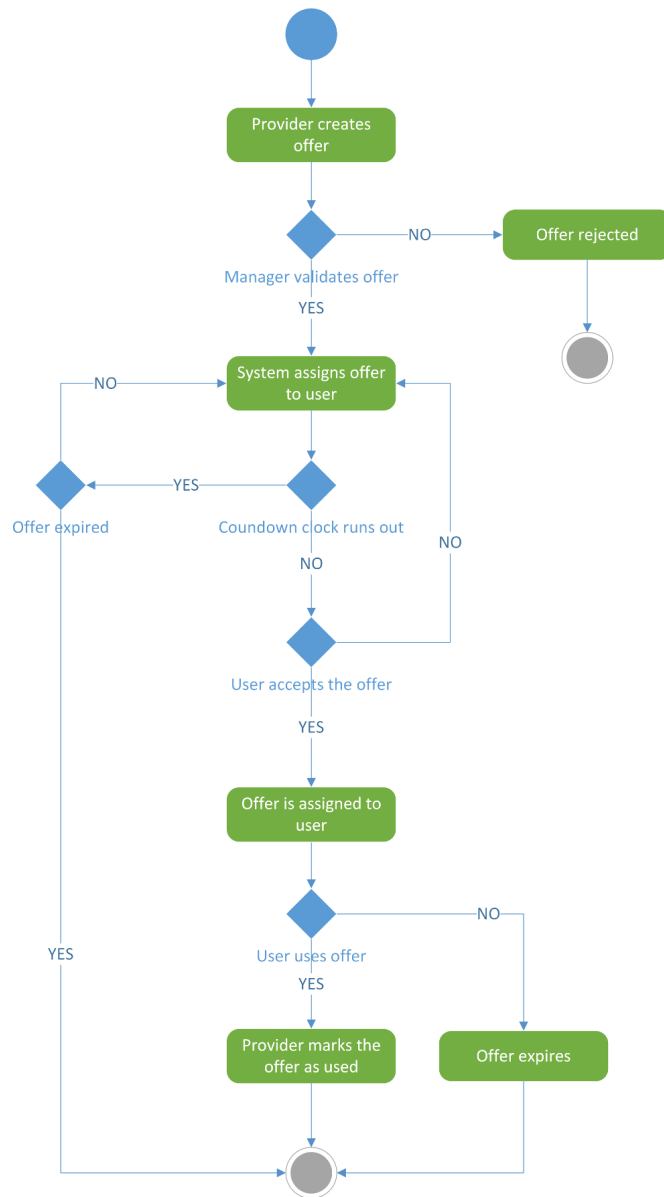


Figure 3.5: Model 5 - Accept / Reject

3.2.5 Campaign evaluation

One of the main complications the current partnerships with public transport operators is that there is no reliable and efficient way to measure the success of these campaigns. Voyager would offer a tool for providers to track the effect of these campaigns.

3.3 Requirements

3.3.1 Functional Requirements

The functional requirements of the Android application directed at public transportation users are:

- The system shall assign offer to the users. This is the main requirement of the application.
- The user shall be able to consult all offers available to him in any given moment.
- The user shall be able to see the details of any offer available to him in any given moment.
- The user shall be able to give his offer to a friend if that offer is giftable.
- The user shall be able to share an offer with a friend if that offer is shareable.
- The user shall be able to rate an offer.

The functional requirements of the Web application directed at the providers are:

- The provider shall be able to input an offer into the system.
- The offer provider shall be able to signal the offer as used.
- The provider shall be able to consult all offers he provided.
- The provider shall have access to statistics related to the campaigns he launches.

The functional requirements of the Web application directed at the system administrator are:

- The system administrator has to validate an offer for it to be available for recommendations.
- The system administrator shall also be able to input an offer into the system.
- The manager should be able to define parameters relative to the offers and public transportation. For example: which vehicles are full and need to be relieved from traffic.

3.3.2 Non-Functional Requirements

The non-functional requirements are divided into three main categories.

Performance Performance is crucial to this application. For instant offers, the system needs to calculate them in real time according to the user's location and destination. Other types of offers will have more processing time but the goal is to do an efficient use of the available resources.

Privacy By design, the system deals with a lot of personal data: social network profiles and user mobility data. This is an absolute necessity to protect the user and make sure his privacy is as close-guarded as possible.

Usability The interaction between the user and the application will in time-sensitive time periods. For example: Waiting for a bus. So the process from receiving an offer and using it has to be as simple as validating a ticket.

3.4 Advantages for the main stakeholders

For public transportation operators:

- A new and interesting way of encouraging passengers to use more public transportation
- Uniting all the diverse partnerships and campaigns in one place through an organized, easier to manage system
- The possibility to influence users and control the public transportation network in ways that improve sustainability

For providers:

- Access to the huge client-base of public transportation services
- The opportunity to gain a competitive edge against similar businesses
- Bypassing the usual red tape of these partnerships and high maintenance marketing campaigns and gaining a quick access to the customer: create a campaign and it's done!
- Option to easily target the campaigns to a specific audience
- Ability to track and evaluate the success of said campaigns

For public transportation users:

- Access to a new market of exclusive offers
- Offers tailored specifically to the user, avoiding SPAM and making sure the user only receives offers that have interests to him
- Discover new and interesting places

3.5 Integration with existing systems

Voyager could very well exist as a standalone system, but the integration with a mobile payments application could bring additional benefits to both the systems and the users similarly to the symbiotic relationship Mobiganha has with Mobipag. The two services aim to improve the public transport user's experience and users only interested in one of the applications could more easily be converted.

3.6 Recommendations

The most distinctive and crucial part of this system is the assignment of offers to users. Which users should get which offers? That ruling is made by the recommender system. There are some peculiarities in Voyager that have to be addressed when building a recommender system that combines users' interests and mobility.

- The sparsity problem. Offers are limited. When a provider launches a campaign, normally there won't be a huge amount of offers. These are local business and, save for free event suggestions or discounts on a great range of materials (for example: a 20% discount on all electronics in a store), the number of offers is going to be small especially compared to the number of users. This is a different scenario from something like Netflix or Spotify: there is not limit on the amount of times you can stream a movie or a song. This means it will be very hard to find users who use the same offer and practically impossible to find similar users when taking several offers into account.
- The new item problem. Offers are new. When a provider launches a campaign, those offers will be a different item than all the other offers. No user will have used because it was just launched, yet it is expected of the system to start making recommendations for those offers right away.
- The location problem. Offers are generally not in the same place. Even if two users have similar tastes, they will not necessarily use or need the same offers just because they live in different locations. This aggravates the sparsity problem and makes it harder for the system to use other users preferences.
- Item profile. When a provider launches a campaign, he will fill out a form with several details about the offers' characteristics. This precious information will be useful when creating an user profile.
- Since Voyager is a system targeted at public transport users, in which they will use the system on the go, location and, especially, mobility needs to be a factor in the recommendation process.

This and other matters will be addressed in the next chapter that details Voyager's recommender system: MARS - Mobility Aware Recommender System.

Chapter 4

MARS: A Mobility Aware Recommender System

MARS is the recommender system engine used in Voyager. This mobility aware recommender system was built with a system like Voyager in mind, taking into account its necessities and constraints. This chapter details the methodology employed in the creation of the MARS algorithm, the implementation of said methodology, the data used for testing and showcasing the algorithm and the results obtained in testing.

An overview of the overall methodology applied to predict passenger interest in an offer is described. In this methodology four main steps were defined: (i) collection of data from smart cards from the public transport network in Porto, Portugal; (ii) generation of offers and user interests data for testing; (iii) establishing of a baseline algorithm; (iv) adaptation of the baseline algorithm to make predictions based on location and frequency of use.

4.1 Data

The data used is a combination of real mobility data and generated data, with the goal of providing a comprehensive dataset for investigating mobility recommendation systems.

4.1.1 Mobility Data

The public transport network of the Metropolitan Area of Porto covers an area of 1,575 km² and serves 1.75 million people. The users' mobility data was supplied by the transport provider, TIP. A total of 136.32 million journeys were recorded in the year of 2013. The system uses a smartcard ticket called Andante and is based on an open-gate validation system, where passenger check-in is required at the beginning of a journey and subsequent transfers. In this work, the users' mobility data was available from the year 2013. Although the full year was available, a subset was selected for the purposes of testing the algorithm. Thus, October 2013 was chosen for being a good sample

month with only one public holiday on the weekend and not many disruptive patterns (e.g. school holidays, events). We opted to further restrict the data geographically for an area of approximately 694 thousand square meters characterized for a great onflow of mobility. Narrowing down the set of testing data allows us to visually inspect the behavior of the algorithm adaptations at this stage, providing us with a platform to expand later. A random set of 127 users was selected, with check-ins in the target area, resulting in 2463 passenger check-ins.

4.1.2 Profile Data

While mobility data was ready for use, offer and user profiles were unavailable. As a result, 7 generic categories of interest were generated, to allow for a wide range of recommendations in controlled testing. The user interests were generated for the 7 generic categories of interests. Since a user can have several interests and the value for each of them is binary (i.e. the user is either interested or uninterested in that category), we opted for a set of all possible combinations for a total of 127 user interests profiles ($127 = 2^7 - 1$), the one case subtracted was the one where the user had no interests). Each of these combinations was then paired with the mobility passenger data previously selected, resulting in a total of 127 profiles provided with both mobility and interest.

4.1.3 Offer Data

Finally, the same number of offers was generated following the same methodology: all combinations of 7 generic categories minus the offer with no categories. Each of these offers was randomly assigned a geographic coordinate in the area defined earlier. Thus, each of this offers represents a location with a set of categories, for which different passengers will have different recommendation ratings.

4.2 Methodology

The goal is to provide a ranked list of offers to recommend to a user. In the context of mobility recommendations as opposed to online retailing it is reasonable to expect users to predominantly use offers on areas they frequent. The recommendations are going to be made based on two aspects: i) the user interest on the offer itself and ii) the location of the offer relative to the user's mobility profile.

4.2.1 Baseline

For the baseline algorithm, the recommendation was done solely based on the user's interest in the offer in question leaving the offer's location and the user's mobility profile aside. The intention behind this decision is to have a baseline to compare the algorithm when mobility is integrated.

To calculate user's interest in an offer, a content-based approach was chosen. In section [2.3.3.3](#), I summarize the advantages and disadvantages of each approach. The reasons behind this choice are the following:

MARS: A Mobility Aware Recommender System

- When providers create a campaign, they will have to input a required level of information about the offers. This means offers will have a clearly defined profile and will be well categorized. Using solely a collaborative filtering system would mean this reliable information would go to waste.
- No cold start problem. Even with few users, the system will be able to make recommendations (as long as they have a user profile with his category preferences).
- No sparsity problem. As seen in section 3.6, Voyager will have a very sparse users / offers used matrix. It will be very hard to find similar users: an essential step in collaborative filtering.
- No new item problem. New offers will always be arriving and rapidly expiring. Using collaborative filtering and not being able to recommend new items is a critical flaw.
- An hybrid system would also be a great choice, but looking at the resources and time available for this dissertation, it made more sense to opt for something simpler. The focus of this dissertation is more on integrating mobility in a recommender system than on building the optimal recommender system.
- By its very nature, collaborative filtering would include location in its recommendations because users would use offers locally advantageous to them and similar locations would increase similarity between users. Since the baseline algorithm would be a way to test a new mobility algorithm, a baseline deprived of location would be the best option. This problem would also likely affect a hybrid approach.

Following a content-based approach, an offer profile was created with a vector of feature-value pairs as seen in Table 4.1, with each pair representing a category: the feature consists of the category and the value is a binary that tells us if the offer can be classified as being part of that category (1 if it is in that category and 0 if it is not). An offer can have several categories. A user profile is also needed. The structure is exactly the same but instead of the categorization of an item, the vector represents the interests of an user with the binary telling us if he is interested in that category of items or not.

Table 4.1: Offer Profile Example

Profile	Cat A	Cat B	Cat C	...	Cat G
Offer N	0	1	1	...	0

Using both passenger and offer profiles as vectors, the distance between them indicates the interest a passenger has in a certain offer. To calculate this value we used the cosine similarity method in 4.1 due to its effectiveness and popularity [LSEM12], adapted to a binary based approach.

$$Similarity_{Rating} = \frac{A \cap B}{\sqrt{A} \times \sqrt{B}} \quad (4.1)$$

The cosine similarity adapted to a binary dataset measure divides the number of categories intersected in the two vectors by the product of the square roots of the total number of features or interests. The resulting value is between 0 and 1. The higher the cosine similarity, the higher the interest of the user in the offer is.

4.2.2 Location

The first adaptation to the baseline algorithm was focused on introducing location as a variable and apply it to the resulting recommendations. Let's call this new variable the location rating.

To get the location rating between a passenger and an offer, the distance between the offer's location and the closest stop used by the passenger is used. The distance between the two was calculated based on the coordinates provided. In addition, a radius was established, of what constitutes a close offer or how close an offer needs to be to become more attractive to the passenger. The location rating is calculated using equation 4.2. It results in a value between 0 and 1 (if the value is negative, the location rating is 0), the closer the stop the closer the value is to 1.

$$Location_{Rating} = 1 - \frac{distance}{radius} \quad (4.2)$$

To integrate this value with the recommender system, we used a weighted system in which the sum of the weights has to be 1. Using this method, offers outside of the radius would have their interest rating diminished, while items inside the radius would have an advantage according to how close they are from the closest stop. However, if an offer has a cosine similarity of 0, the final rating will be 0 independently of the location. If the user has no interest in an offer, the location becomes irrelevant.

While this approach allows recommendations that are closer to stops used by the passenger, there is a clear limitation: the closest stop is not necessarily the most relevant one for the passenger. As an example, an item could be less interesting near a stop used only once, while a stop used daily would result in more interesting recommendations for passengers. The next algorithm will also have the frequency of use of a stop into account.

4.2.3 Location and Frequency

Rather than just focusing only on location, the frequency of the location will also be taken into account. As a result, instead of calculating the location rating for the closest stop, the most relevant stop needs to be found, i.e. a stop that is both close and is frequently used. The ideal stop would be the most used stop (the one with most check-ins), which provide the basis for assessing frequency. For each passenger, the frequency rating of any stop will be the quotient of the total check-ins for that stop and the total check-in of the most used stop as seen in equation 4.3.

$$FrequencyRating = \frac{currentStop_{freq}}{mostUsedStop_{freq}} \quad (4.3)$$

We call the combination of the location rating and the frequency rating, mobility rating. This mobility rating, like the location rating, will be 0 if the stop is outside the defined radius.

The final equation seen in 4.4 takes into account the user interests through the cosine similarity but also the location of his stops and the frequency in which he checks in on them. Each of this variables will have a certain weight that can be calibrated.

$$R = w_{Sim} \times R_{Sim} + \underbrace{w_{Loc} \times R_{Loc} + w_{Freq} \times R_{Freq}}_{mobility\ rating} \quad (4.4)$$

4.2.4 Clustering

Even though the algorithm now takes location and frequency into account, it still has a small pitfall. For example, take a user with two offers with the same similarity rating but located in two different areas. In the first area the user has one stop with 20 check-ins. In the second area the user has two stops with 15 check-ins each. The offer in the first area would have a higher mobility rating even though the second area is more frequented by the user because currently the algorithm only takes into account the frequency of one stop. This normally happens when users have various possibilities to get to a destination and stops are located close together permitting the user to embark on the first vehicle to depart.

To address this weakness in the algorithm, a clustering algorithm was needed to bundle a group of stops and make the more frequented areas stronger. The criteria used to aggregate various stops was the following:

- All stops have to be in the same radius from each other. This radius is the same used in the calculation of the location rating.
- All stops need to have at least 90% offers located inside their radius in common.

When a cluster of stops is formed, something called a Super Stop is created. The Super Stop's frequency is the sum of the frequency of all the stops assembled for its formation and its latitude and longitude are the average of the coordinate values of all the stops that form the cluster.

4.3 Implementation

The goal of the MARS algorithm is to provide a ranked list of offers to recommend to a user. However, to facilitate testing the implementation will expand its scope to a whole matrix of users and offers as seen in Table 4.2. Each cell in the matrix, a value between 0 and 1, corresponding to the MARS rating represents the interest each user has in an offer.

Table 4.2: Output Ratings Matrix

Mars Output		Offers			
		Offer A	Offer B	...	Offer DW
Users	User 1	Rating 1-A	1-B	...	1-DW
	User 2	2-A	2-B	...	2-DW

	User 127	127-A	127-B	...	127-DW

The user object has an array of the stops frequented by the user and each stop has the coordinates and the number of times the user checked-in. The offer object contains the coordinates of that offer's location. Both the user and offer profiles have a structurally identical profile 4.1. with the user preferences and offer categories.

4.3.1 Baseline

```

1 // Inputs: users, items
2 // Output: matrix
3
4 for each user in users
5   for each item in items
6
7     ranking = calculateCosineSimilarity(user, item)
8     matrix.push(ranking)
9
10  end for
11 end for

```

Listing 4.1: Baseline Algorithm

4.3.2 Location

```

1 // Inputs: users, items
2 // Output: matrix
3
4 for each user in users
5   for each item in items
6
7     similarity = calculateCosineSimilarity(user,item)
8     rating = 0
9
10    if(similarity != 0)
11
12      bestLocationRating = 0
13
14      for each stop in user.stops
15
16        distance = calculateDistance(item,stop)
17        locationRanking = 1 - distance / radius
18
19        if (bestLocationRating < locationRating)
20          bestLocationRating = locationRating;
21        end if
22
23      end for
24
25      rating = interestWeight * similarity + locationWeight *
26              bestLocationRating
27
28    end if
29
30    matrix.push(rating)
31 end for

```

Listing 4.2: Location Algorithm

4.3.3 Location and Frequency

For a user in the users array: we find the most used stop and for each item in the items array we calculate the similarity between the user and the item. If we find some similarity, we proceed to calculate the mobility rating (a combination of the location and frequency ratings) for each of the user's stops. The stop with the best mobility rating will be the stop used to calculate the final rating in association with the similarity rating that represents the user's interests. That user is then inserted into the matrix. The same process is applied on all the other users.

MARS: A Mobility Aware Recommender System

```
1 // Inputs: users, items
2 // Output: matrix
3
4 for each user in users
5
6   for each stop in user.stops
7     if (stop.trips > mostUsedStop.trips)
8       mostUsedStop = stop
9     end if
10  end for
11
12  for each item in items
13    similarity = calculateCosineSimilarity(user,item)
14    rating = 0
15
16    if(similarity != 0)
17
18      for each stop in user.stops
19
20        distance = calculateDistance(item,stop)
21        locationRating = 1 - distance / radius
22        frequencyRating = stop.trips / mostUsedStop.trips
23
24        mobilityRating = locationWeight * locationRating + frequencyWeight *
25          frequencyRating
26
27        if (locationRating < 0)
28          mobilityRating = 0
29        end if
30
31        if (bestMobilityRating < mobilityRating)
32          bestMobilityRating = mobilityRating
33        end if
34      end for
35
36      rating = interestWeight * similarity + mobilityWeight * bestMobilityRating
37    end if
38
39    matrix.push(rating)
40  end for
41 end for
```

Listing 4.3: Location and Frequency Algorithm

4.3.4 Clustering

This implementation aims to achieve the clustering of all nearby stops using the criteria set in methodology. To cluster two stops, they would have to be in the same radius of each other (this radius is the same used for the location rating) and have at least 90% offers located inside their radius in common: let's call this being compatible. When adding a third or more stops, all the stops need to satisfy this criteria between each other. This compatibility is symmetric (for example, if offer A is compatible with offer B, offer B is compatible with offer A) but not transitional (for example, if offer A is compatible with offer B and B is compatible with C, this does not mean offer A is compatible with C). So a list of all the groups of compatible stops is needed, but not the subgroups that form them (for example, if a A, B and C are compatible, the sub-group A and B is not needed, only the group A, B and C).

This is a known graph problem called Listing All Maximal Cliques. A clique is subset of vertices of an undirected graph, such that every two distinct vertices in the graph are adjacent. In this rationale, the vertices would be stops and the edges would be the compatibility between the stops. A maximal clique is a clique that cannot get any bigger, in this case, it means a cluster that cannot add any other stop. To fully realize the clustering process, a list of all the maximal cliques is needed. The clique problem is NP-complete.

For a better understanding of the solution, the presentation of the algorithm will be split in parts.

In the first part, all the stops are compared with each other to check if they are compatible using a compatibility function that applies the criteria exposed in the methodologies in Section 4.2.4 and clusters of two stops are created and inserted in the processing queue. To optimize performance, the symmetry of the compatibility is exploited. Since the comparison is between the same vector, more than half of the compatibility matrix is discarded. That is, if A and B are compared, there is not need to compare B and A because the outcome will be the same.

```

1 offersInCommon = 0.9 // percentage of offers in common between two stops
2
3 for each stop i in user.stops
4   for each stop i+1 in user.stops
5
6     stop1 = user.stops[i]
7     stop2 = user.stops[i+1]
8
9     bool compatibility = compatibleclustering(stop1, stop2, radius, offersInCommon)
10
11    if(compatibility is true)
12      insert stop1 in cluster
13      insert stop2 in cluster
14      insert cluster in processingqueue
15    end if
16
```

```

17   end for
18 end for

```

Listing 4.4: Clustering Algorithm: Part I

With all the pairs of compatible stops in processing queue, the algorithm goes through the user stops again and checks if a stop is compatible with the cluster in the front of the processing queue that for now only contains pairs. For a stop to be compatible with a cluster, it needs to be compatible with all the stops in said cluster without already belonging to it. The flag used in code is called "overallcompatibility".

```

1 while(processingqueue != empty)
2
3   cluster = processingqueue.front
4
5   for each stop i in user.stops
6
7     stop1 = i
8     overallcompatibility = false
9
10    for each stop j in cluster
11
12      stop2 = j
13      compatibility = false
14
15      if(stop1 != stop2)
16        compatibility = compatibleclustering(stop1, stop2, radius, offersInCommon)
17      end if
18
19      if(compatibility is false)
20        overallcompatibility = false
21      end if
22
23    end for
24
25    ...
26
27    // the for and while continue in the next code snippet

```

Listing 4.5: Clustering Algorithm: Part II

If the stop is compatible with the cluster in the front of the processing queue that stop, a new expanded cluster is created containing the cluster and said stop and a flag is triggered confirming this not a maximal clique. If the newly expanded cluster is not already in the processing queue, he will be added.

If the cluster is not compatible with any of the user's stops, it means it is a maximal clique, there is no cluster that can contain him, so he will be added to clusters.

MARS: A Mobility Aware Recommender System

```
1
2  if(overallcompatibility is true)
3
4    maximalclique = false;
5
6    expandedcluster = cluster + stop1;
7
8    cluster_already_in_queue = false
9
10   for each c cluster in processingqueue
11     if(expandedcluster == c)
12       cluster_already_in_queue = true
13     end for
14
15     if(cluster_already_in_queue is false)
16       insert expandedcluster in processingqueue
17     end if
18   end if
19 end for
20
21 if(maximalclique is true)
22   cluster_already_added = false
23
24   for each c cluster in clusters
25     if(cluster == c)
26       cluster_already_added = true
27     end if
28   end for
29
30   if(cluster_already_added is false)
31     insert cluster in clusters
32   end if
33 end if
34
35 processingqueue.pop
36
37 end while
```

Listing 4.6: Clustering Algorithm: Part III

When the list of clusters listing all maximal cliques is complete, we use the clusters to create the Super Stops. The Super Stops are stops with a latitude and longitude calculated with the average of all the stops in their respective clusters. The frequency is the sum of the frequencies of all the stops in the cluster.

4.4 Results

The results obtained from the execution of the four proposed algorithms are discussed in this section. Firstly, the algorithms were set according to Table 4.3.

Table 4.3: Algorithms Parameters and Overall Results

	Radius	Similarity Weight	Location Weight	Frequency Weight	Avg. Rank (SD)
Baseline	-	100%	-	-	0.55 (0.05)
Location	200 m	50%	50%	-	0.36 (0.05)
Mobility	200 m	50%	25%	25%	0.39 (0.04)
Clustering	200 m	50%	25%	25%	0.37 (0.04)

The total number of recommendations per passenger is, on average, 111.79 (SD = 15) for the four versions of the recommender system. This makes sense because the location rating was not meant to be an excluding factor but a way to promote the offers closest to the stops. If an offer is outside the radius, its rating will be cut in half (at least if they both weight the same – as they did in our tests). The same could be said for the location and frequency based algorithm.

The average of rating in 4.3 is also consistent with the data. If the users' data and the offers' data used all possible combinations, the value of the rating, which in this case corresponds with the cosine similarity, would be 0.5. Since we opted to exclude the offer with no categories and the user with no interests because it doesn't make sense to recommend to a user with no interests or to have an offer that's not categorized, that value is a little bit higher. For the other algorithms, the rating decreased significantly. This is easily explained by the fact that it's harder to hit the spot in two variables than in only one. The difference between the location algorithm and the location frequency algorithm is too small to take any conclusions. To get a better take on the difference between these two algorithms, we need to do another type of analysis.

Fig. 4.1 presents a comparison between the frequencies of the ratings by algorithm. As noted earlier the number of items with 0% remains constant. However, the ratings are more evenly distributed in the Baseline algorithms than the others, as expected. The introduction of Location resulted in a strong penalization of the overall ratings, with most of the items grouped between the 30% and 50%. Finally, the Mobility algorithm, which takes into account both Location and Frequency, had the effect of promoting some of the ratings up.

The Mobility and Clustering algorithms differ slightly despite having the same weight distribution because the Super Stops have a higher frequency than the normal stops, so the frequency of the most used stop is going to be higher. This means that for most of the offers, the frequency rating is lower affecting the final rating.

However, the main goal of the algorithms is not to penalize or promote the rating, but rather to order the list of items. The rating in and of itself is a tool in the process of ranking the items. The rating for a user's interest in an item in an algorithm should not be compared to the rating given by another algorithm. What should be done instead is comparing the ranking of the item in an ordered list (in our implementation, correspondent with a row in the matrix).

MARS: A Mobility Aware Recommender System

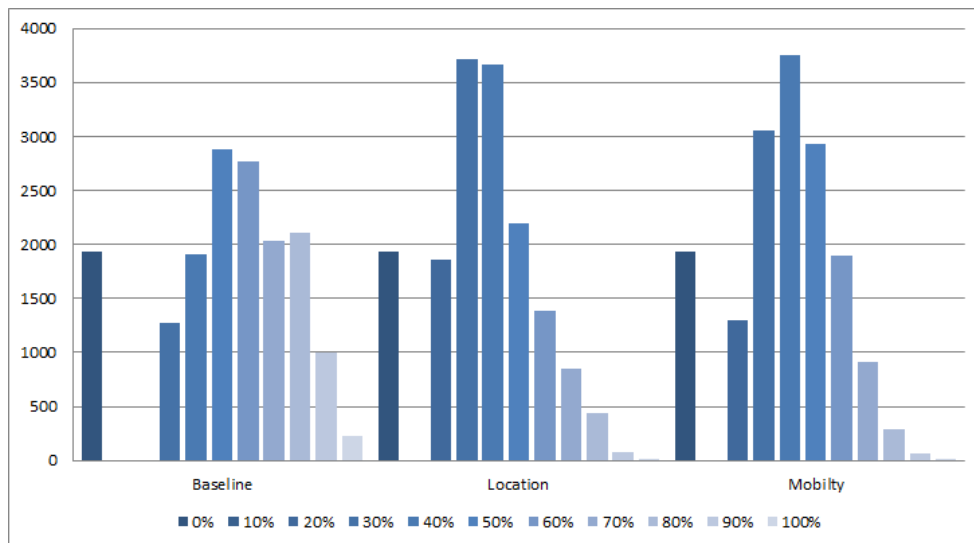


Figure 4.1: Algorithm ratings comparison

The performance of the algorithms is better perceived when taking an individual user, thus an illustrative example of a user is given. Fig. 4.2 shows four maps of the user interests in our previously mentioned sample area. The flags are stops and the dots are offers with the radius of the circle as well as the gradient representing the interest of the user in said item, bigger and darker being the more desirable items. However, the top items are shown in a different color, from green (highest) to yellow (lowest).

In the top map on the left, we can see the items are seemingly evenly arranged in the area. Let's compare this map with the one on the top right corresponding to the location algorithm. The chosen radius of the stops was traced with a circumference.

In the map of the location-based algorithm, all the items outside of the radius of the stops took a significant hit in relevance while the items inside the radius to the stops maintained or improved their size. It is also apparent a difference between the items on the fringe of the radius and the ones really close to the stops. This means the algorithm does what it set out to do, makes location a factor and highlights the items closer to the user's stops.

As previously discussed, frequency should also be taken into account. There should be and there is a difference between a stop a user frequents daily and a stop that is rarely used. Because of that difference, the next algorithm takes frequency into account. The bottom map on the left represents the frequency-based algorithm. The stops with a higher number of check-ins are displayed with a larger flag.

In the map of the location and frequency based algorithm, the items around the larger flag, the one that represents the stop with the most check-ins, gained more relevance compared to the ones around less traffic heavy stops. This means the goal of making the stops with more check-ins more important was successfully achieved. Note that these items are progressively closer to the stops most used by the user, adapting the recommendations to the mobility pattern.

MARS: A Mobility Aware Recommender System

Finally, the bottom map on the right represents the clustering algorithm. In this case there were two areas: one area with a stop with high frequency and another are with three stops with less frequency. However, in this algorithm, those three stops bundle together to form a Super Stop and this way can overpower the first area bringing the top offers to their radius. Again, the algorithm behaves as expected making the are most frequented by the user gain more relevance even though the data was scattered overcoming the biggest weakness of the location and frequency algorithm.

MARS: A Mobility Aware Recommender System

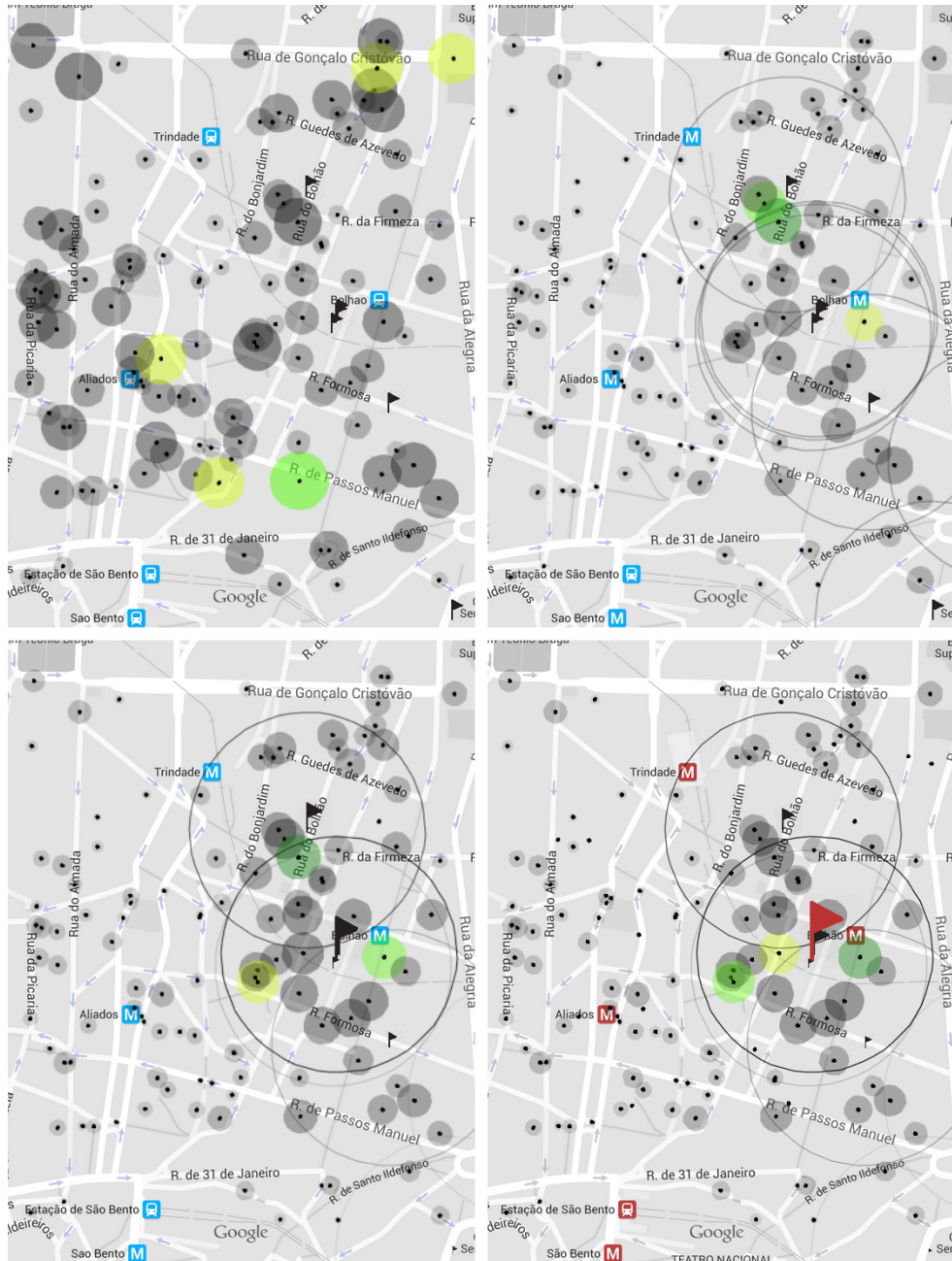


Figure 4.2: Recommendations for a single user using the four iterations of the MARS algorithm: baseline (top left), location-based (top right), location- and frequency based (bottom left) and clustering (bottom right).

MARS: A Mobility Aware Recommender System

Chapter 5

Conclusions and Future Work

In this thesis a literature review was done of the status of public transportation and recommender systems in Chapter 2 and went further with this concept than what was already established by Mobiganha.

The exploration of this concepts led to the specification of Voyager, a system with different types of offers that can be used by the system administrator with different goals from promoting some parts of the city to rewarding the most loyal users.

The investigation of current recommender systems showed that, despite location being a factor, a system that completely integrated mobility in its recommendations wasn't found. So MARS: A Mobility Aware Recommender System was created to bridge that gap and make recommendations using mobility data from public transportation users to assign the offers substantiated in Voyager.

5.1 Future Work

While I think Voyager could work really well as a basis for an implementation on a system of its kind and though MARS has some real value in making recommendations, the work in mobility-based recommender systems is far from done. These are some ideas on how to improve on the work developed in this thesis.

5.1.1 Voyager

Conceptually, one of Voyager's most ambitious features was the possibility of presenting instant offers to influence users. For example, if the user's usual bus is full, Voyager would present him with an offer in another area and an alternate route. If implemented, it would be interesting to analyze the ramifications of this. Would the user really change his usual route to receive an offer? What distance would the user be willing to travel to get an offer and would he be willing to use another vehicle to get there? How would this change and others like it affect the users, the public transport operator and the sustainability of the transportation network itself? Another Voyager

feature, providing promotional offers to users to promote a part of the city or an event. Could this be an effective way to make dead zones more appealing? These are all very complex and intriguing questions.

5.1.2 User's Interests Data

In MARS, the validation data in the Andante cars was used to create the user's mobility profile and the offer's characteristics would have to be inserted by the provider. However, the user's interest in the offer's categories was represented with generated data. In a real world implementation, the user's interest in a certain category would be calculated using the offers a user claimed, shared and redeemed or by asking him to fill out a form.

However, there are cleverer ways of doing this, especially in the age of social networks. If access to a social network like Facebook, Twitter or Foursquare was give, it would be possible to know even much more about the user right out of the gate, avoiding the new user problem that happens when the user doesn't want to fill out a form and hasn't used any offer yet.

5.1.3 A Hybrid System

In MARS, the prediction of a user's interest in an offer is calculated using a content-based method for reasons explained in section 4.2.1. However, as stated on section 2.3.3.3, there are better ways of estimating the interest a user has in an offer. I'm referring specifically to a hybrid recommender system that combines the content-based and the collaborative filtering approaches.

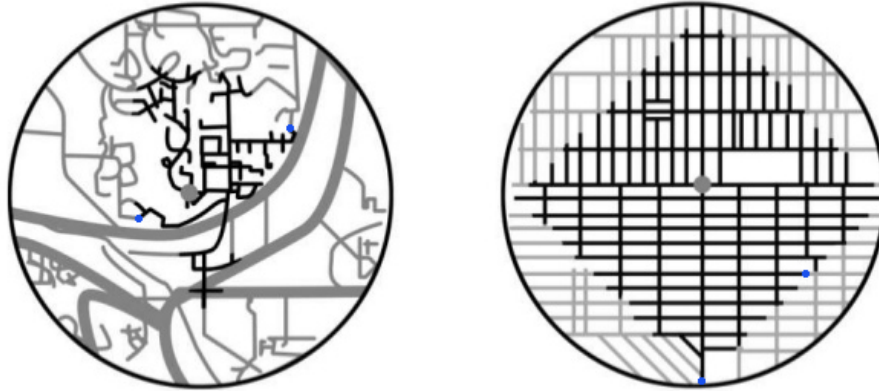
Using the information of the offers a user claimed, shared and redeemed to find similar users and see what they like, the recommendations could be more accurate. However, the combination of collaborative filtering techniques would have to be done in a thoughtful way because a regular approach to collaborative filtering would intrinsically have location as a variable. This would skew the algorithm in favor of closer offers before the inclusion of the location ranking, something that may or may not be desired. To avoid this, we could assume that when a user redeems an offer, for example, he isn't interested in the offer in and of itself but the whole category. For this to work, the offers would have to be very well categorized. For example: a bar can't be simply categorized as a bar, but as a salsa dancing mexican bar.

5.1.4 Walking Distance

All calculations in the MARS algorithm for all distances, both the distance between an offer and a stop in the ranking and the distance between two stops in the clustering, were done using a straight line distance (also known as flying or air distance). While the algorithm still works well using these values, it would certainly be more accurate if the calculation was done using the walking distance.

Figure 5.1 shows the layouts of two maps, one in a suburban community (Bellevue, Washington, USA) and other in a compact urban community (Phinney Ridge in Seattle, USA). The circle represents the straight line distance of approximately 1.6 kilometers from the dot on the center.

The other paths represent the farthest you can reach by walking the same 1.6 kilometers using the shortest route.



Walk-shed diagrams created by *Urban Design 4 Health*.

Figure 5.1: The straight line distance and the walking distance

Comparing the two distances in some points on the end of the path, it's possible to see there's a slight but not irrelevant difference between the straight line distance and the real walking distance in some of those points. The points signaled in blue show that discrepancy really well.

When implementing the system, it would also be interesting to study the distance public transportation users are willing to walk to redeem offers.

5.1.5 Clustering

The clustering in MARS presented a problem called the Listing All Maximal Cliques. That problem was successfully solved but maybe not in the most elegant way. It is a NP-complete problem so the solution is never going to be optimal. However, there is already a proven algorithm for this problem: Bron-Kerbosch algorithm. The consensus is that that is the best option for the problem.

The results of the clustering algorithm would not be different with this change but if MARS were someday to be implemented in a real word situation it could bring a not insignificant improvement to its performance.

Also, the approach we took to do the clustering using the features of MARS proved fortunate, yet it would be interesting to try different clustering techniques that group GPS coordinates and compare the results.

5.1.6 Time

In this thesis, mobility is referenced mainly as a combination of location and frequency, but our concept of mobility can be extended to include new variables if appropriate. One huge variable that could be taken into account is time. The offers of interest to a user could vary with either with

Conclusions and Future Work

the time of the day or the seasons of the year. Though that would unleash a whole new level of complexity, it would be very interesting to see the effects it could have on a recommender system.

References

- [And07] Chris Anderson. The Long Tail: Why the Future of Business Is Selling Less of More by Chris Anderson. *Journal of Product Innovation Management*, 24(3):1–30, 2007.
- [Ban11] David Banister. Cities, mobility and climate change. *Journal of Transport Geography*, 19(6):1538–1546, 2011.
- [BHS06] Erik Brynjolfsson, Yu Jeffrey Hu, and Michael D Smith. From Niches to Riches : Anatomy of the Long Tail From Niches to Riches : Anatomy of the Long Tail. *MIT Sloan Management Review*, 47:67, 2006.
- [BKV10] Robert M. Bell, Yehuda Koren, and Chris Volinsky. All together now: A perspective on the NETFLIX PRIZE. *Chance*, 23(1):24–24, 2010.
- [BLPR12] Linas Baltrunas, Bernd Ludwig, Stefan Peer, and Francesco Ricci. Context relevance assessment and exploitation in mobile recommender systems. In *Personal and Ubiquitous Computing*, volume 16, pages 507–526, 2012.
- [Com07] Commission of the European Communities. Green Paper - Towards a new culture for urban mobility. Technical report, European Commission, 2007.
- [E.C11] E.C. WHITE PAPER: Roadmap to a Single European Transport Area. *European Commission*, 2011.
- [FDFa14] Marta Campos Ferreira, Teresa Galvão Dias, and João Falcão. Design and Evaluation of a Mobile Payment System for Public Transport : the MobiPag STCP Prototype. (c):71–77, 2014.
- [GDBJ10] Mouzhi Ge, C Delgado-Battenfeld, and D Jannach. Beyond accuracy: evaluating recommender systems by coverage and serendipity. ... *on Recommender systems*, pages 257–260, 2010.
- [GJR14] Judith Green, Alasdair Jones, and Helen Roberts. More than A to B: the role of free bus travel for the mobility and wellbeing of older citizens in London. *Ageing and society*, 34(3):472–494, 2014.
- [IDL⁺08] Leo Iaquina, Marco De Gemmis, Pasquale Lops, Giovanni Semeraro, Michele Filannino, and Piero Molino. Introducing serendipity in a content-based recommender system. In *Proceedings - 8th International Conference on Hybrid Intelligent Systems, HIS 2008*, pages 168–173, 2008.
- [LSEM12] Justin J. Levandoski, Mohamed Sarwat, Ahmed Eldawy, and Mohamed F. Mokbel. LARS: A location-aware recommender system. In *Proceedings - International Conference on Data Engineering*, pages 450–461, 2012.

REFERENCES

- [OHM⁺12] Tatsuo Okuda, Shigeki Hirasawa, Nobuhiko Matsukuma, Takashi Fukumoto, and Akitoshi Shimura. Smart mobility for smart cities, 2012.
- [Ric10] Francesco Ricci. Mobile recommender systems. *Information Technology & Tourism*, 12(3):205–231, 2010.
- [RV97] Paul Resnick and Hal R Varian. Recommender systems. *Communications of the ACM*, 40(3):56–58, 1997.
- [San14] Vítor Santos. *Mobiganha - Innovative Services for Public Transportation Based in Mobile Payments*. FEUP, Porto, 2014.
- [SBEH12] Norma Saiph Savage, Maciej Baranski, Norma Elva Chavez, and Tobias Höllerer. I’m feeling LoCo: A Location Based Context Aware Recommendation System. *Advances in Location-Based Services SE - 3*, pages 37–54, 2012.
- [SKR01] J Ben Schafer, Joseph Konstan, and John Riedl. E-commerce recommendation applications. *Applications of Data Mining to Electronic . . .*, pages 115–153, 2001.
- [TM11] Nava Tintarev and Judith Masthoff. *Recommender Systems Handbook*, volume 54. Springer US, 2011.
- [TS06] Yuichiro Takeuchi and Masanori Sugimoto. CityVoyager : An Outdoor Recommendation. *Ubiquitous Intelligence and Computing*, 4159(Figure 1):625–636, 2006.